DISCRIMINATIVE INSTANCE WEIGHTING FOR DOMAIN ADAPTATION IN STATISTICAL MACHINE TRANSLATION

Authors: George Foster, Cyril Goutte, and Roland Kuhn (NRC Canada)
Presenter: Avneesh Saluja
The Claim

- Domain adaptation in SMT, and in NLP in general, a popular topic
- By incorporating several ideas:
  - Instance-weighting approach, at the level of phrase pairs
  - Overlapping features, designed to elicit “general language” and “similarity” characteristics
  - ML, instead of ME, training/learning criterion

the authors come up with an (improved?) domain adaptation scheme for MT
Why Domain Adaptation?

- Workshops, theses, papers, etc.
  - The brittleness of our models…
- In action: LMs for MT: Original vs. Translated Texts
- Theoretical background:
  - A theory of learning from different domains (Ben-David et al., Machine Learning, 2010)
  - Domain Adaptation of NLP Systems (J. Blitzer’s Thesis, 2008)
  - Domain Adaptation in Regression (Cortes & Mohri, ALT 2011)
- In MT: the pipeline approach prevents end-to-end adaptation scheme
- Assumption: all OOD data is homogeneous
Baseline Setups: Simplest Methods

- Throw everything into a big bucket:

- Let MERT handle it:

\[
\begin{pmatrix}
\pi_1 \\
\pi_2 \\
\vdots \\
\pi_m
\end{pmatrix}
\]
Baseline Setups: Linear Combination

- Linear models and MERT for adaptation problematic:
  - MERT assumes a flat loglinear model
- Optimize corpus log-likelihood instead of minimizing error

\[ \hat{\alpha} = \arg \max_\alpha \sum_{w,h} \hat{p}(w, h) \log \sum_i \alpha_i p_i(w|h) \]

\[ \hat{\alpha} = \arg \max_\alpha \sum_{s,t} \hat{p}(s, t) \log \sum_i \alpha_i p_i(s|t) \]
Baseline Setups: Linear Combination

- Linear models and MERT for adaptation problematic:
  - MERT assumes a flat loglinear model
- Optimize corpus log-likelihood instead of minimizing error

LM Weights: \( \hat{\alpha} = \arg \max_\alpha \sum_{w,h} \tilde{p}(w, h) \log \sum_i \alpha_i p_i(w|h) \)

TM Weights: \( p(s|t) = \frac{c_I(s, t) + \beta p_0(s|t)}{c_I(t) + \beta} \)
Baseline Setups: IR style

- Select “similar” sentence pairs from OOD that match sentences from ID
- Trained LM with in-domain data, evaluated on target side of OOD data
  - Select lowest perplexity sentences
  - Number of sentences to select tuned (optimize dev-set BLEU)
Instance Weighting: Model & Training

- Instance = Phrase Pair
- Potentially overlapping features defined for phrase pairs
- LM adaptation as in baseline
- TM adaptation: \( p(s|t) = \alpha_t p_I(s|t) + (1 - \alpha_t) p_o(s|t) \)

\[
c_o(s, t) \left[ 1 + \exp \left( -\sum_i \lambda_i f_i(s, t) \right) \right]^{-1} \cdot \frac{c_\lambda(s, t) + \gamma u(s|t)}{\sum_{s'} c_\lambda(s', t) + \gamma}
\]

- Jointly optimize feature and mixture weights via L-BFGS

\( (\hat{\alpha}, \hat{\lambda}) = \arg \max_{\alpha, \lambda} \sum_{s, t} \tilde{p}(s, t) \log p(s|t; \alpha, \lambda) \)

\( \gamma = 0 : \text{iw all} \)
\( \gamma \neq 0 : \text{iw all map} \)
Interpretation of the Model

• Why does downweighting original joint OOD counts work?
• Ideally, we want to maximize (log) likelihood w.r.t. (i.e., weighted by) “true” joint distribution of in-domain data:

\[ \hat{\theta} = \arg \max_{\theta} \sum_{s,t} p_{\hat{I}}(s, t) \log p_{\theta}(s|t) \]

Over all OOD phrase pairs

\[ \approx \arg \max_{\theta} \sum_{s,t} \frac{p_{\hat{I}}(s, t)}{p_{\hat{o}}(s, t)} c_{o}(s, t) \log p_{\theta}(s|t) \implies \]

\[ p_{\hat{\theta}}(s|t) = \frac{p_{\hat{I}}(s, t)}{p_{\hat{o}}(s, t)} \frac{c_{o}(s, t)}{\sum_{s'} p_{\hat{I}}(s', t) c_{o}(s', t)} \]

compare with

\[ \frac{c_{o}(s, t) w_{\lambda}(s, t) + \gamma u(s|t)}{\sum s' c_{o}(s', t) w_{\lambda}(s', t) + \gamma} \]

Ranges between 0 and 1

Does it make sense to “upweight”?
Features Used

**General Language**
- Phrase pair length
- Frequency of pair
- Rare source/target phrase frequencies (2x)
- IBM1 (OOD) ppl (2x)
- Mean & Min “document” or block frequencies (4x)
- Burstiness features (4x)

**Similarity**
- ID LM ppl over 1 & 2-grams (4x)
- OOV counts w.r.t. ID LM (2x)
- ID IBM1 model (2x)

**SVM Feature:**
- SVM classifier to classify ID and OOD phrase pairs
- Classifier result used as additional feature
Corpora & Setup

- **English <-> French**
  - ID: EMEA Medical corpus
  - OOD: Europarl
  - Dev/test: from EMEA corpus

- **Chinese -> English**
  - ID: NIST09 news-related corpora
  - OOD: Rest of NIST09
  - Dev: NIST05 evaluation + random training set sentences
  - Test: NIST06 & NIST08

- Standard phrase-based setup; 4-gram LM
- HMM + IBM2 WA union

<table>
<thead>
<tr>
<th>corpus</th>
<th>sentence pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europarl</td>
<td>1,328,360</td>
</tr>
<tr>
<td>EMEA train</td>
<td>11,770</td>
</tr>
<tr>
<td>EMEA dev</td>
<td>1,533</td>
</tr>
<tr>
<td>EMEA test</td>
<td>1,522</td>
</tr>
<tr>
<td>NIST OUT</td>
<td>6,677,729</td>
</tr>
<tr>
<td>NIST IN train</td>
<td>2,103,827</td>
</tr>
<tr>
<td>NIST IN dev</td>
<td>1,894</td>
</tr>
<tr>
<td>NIST06 test</td>
<td>1,664</td>
</tr>
<tr>
<td>NIST08 test</td>
<td>1,357</td>
</tr>
</tbody>
</table>

Table 1: Corpora
**Results – EMEA/EP**

### Table 2: Results, for EMEA/EP translation into English

<table>
<thead>
<tr>
<th>Method</th>
<th>EMEA/EP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fren</td>
</tr>
<tr>
<td>in</td>
<td>32.77</td>
</tr>
<tr>
<td>out</td>
<td>20.42</td>
</tr>
<tr>
<td>baseline</td>
<td>33.61</td>
</tr>
<tr>
<td>loglin</td>
<td>35.94</td>
</tr>
<tr>
<td>ir</td>
<td>33.75</td>
</tr>
<tr>
<td>lin lm</td>
<td>35.61</td>
</tr>
<tr>
<td>lin tm</td>
<td>35.32</td>
</tr>
<tr>
<td>map tm</td>
<td>35.15</td>
</tr>
<tr>
<td>lm+lin tm</td>
<td>36.42</td>
</tr>
<tr>
<td>lm+map tm</td>
<td>36.28</td>
</tr>
<tr>
<td>iw all</td>
<td>36.55</td>
</tr>
<tr>
<td>iw all map</td>
<td><strong>37.01</strong></td>
</tr>
<tr>
<td>iw all flat</td>
<td>36.50</td>
</tr>
<tr>
<td>iw gen map</td>
<td>36.98</td>
</tr>
<tr>
<td>iw sim map</td>
<td>36.82</td>
</tr>
<tr>
<td>iw svm map</td>
<td>36.79</td>
</tr>
</tbody>
</table>
Results - NIST

Table 2: Results, for EMEA/EP translation into English

<table>
<thead>
<tr>
<th>Method</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nst06</td>
</tr>
<tr>
<td>in</td>
<td>27.65</td>
</tr>
<tr>
<td>out</td>
<td>19.85</td>
</tr>
<tr>
<td>baseline</td>
<td>26.93</td>
</tr>
<tr>
<td>loglin</td>
<td>28.09</td>
</tr>
<tr>
<td>ir</td>
<td>——</td>
</tr>
<tr>
<td>lin lm</td>
<td>28.02</td>
</tr>
<tr>
<td>lin tm</td>
<td>27.16</td>
</tr>
<tr>
<td>map tm</td>
<td>27.20</td>
</tr>
<tr>
<td>lm+lin tm</td>
<td>27.83</td>
</tr>
<tr>
<td>lm+map tm</td>
<td>28.05</td>
</tr>
<tr>
<td>iw all</td>
<td>28.74</td>
</tr>
<tr>
<td>iw all map</td>
<td>30.04</td>
</tr>
<tr>
<td>iw all flat</td>
<td>28.31</td>
</tr>
<tr>
<td>iw gen map</td>
<td>29.81</td>
</tr>
<tr>
<td>iw sim map</td>
<td>29.66</td>
</tr>
<tr>
<td>iw svm map</td>
<td>——</td>
</tr>
</tbody>
</table>
Related Work

• Linear combination framework: Foster & Kuhn (ACL WMT, 2007)
  • Mixture weights are a function of several distance metrics
  • Downhill simplex to maximize BLEU on development set

• Motivation for instance weighting in NLP: Jiang & Zhai (ACL 2007)
  • Maximize expected log likelihood w.r.t. ID development set
  • This work applies the general concepts to MT

• Instance weighting through feature-based discriminative model: Matsoukas et al. (EMNLP 2009)
  • Sentence-level features, instead of phrase pair-level
  • Perceptron, instead of logistic regression
  • Optimize expected TER (over N-best) instead of log-likelihood
  • L-BFGS also

• General language & similarity features: Daumé (ACL 2007)
Conclusion

- Linear combination + instance weighting method for SMT domain adaptation
- Two-stage weighting:
  - Combine multinomial models: linearly
  - OOD phrase pair count weights: feature-based discriminative model
- Joint training of both sets of weights
- EMEA/EP (vs. strongest baseline):
  - Fr->En: +0.60 BLEU
  - En->Fr: +0.41 BLEU
- NIST (vs. strongest baseline)
  - NIST06: +0.99 BLEU
  - NIST08: +1.65 BLEU
Discussion

• Missing details:
  • Prior weight $\gamma$
  • No IR/SVM evaluation on NIST?
  • Example sentence showing improvement
  • Explicit comparison with sentence-level feature approach

• Analysis on how approach performs as a function of dataset size

• Is uniform prior the best choice?
• Is it necessary to have a two-stage model?
• A better way to incorporate Gigaword corpora?

Thank you!