Workpackage 4: Adaptive Translation Models
2nd Review Meeting

G. Sanchis-Trilles, D. Ortiz-Martínez, J. González-Rubio, F. Casacuberta

November 25, 2013
Index

1. Introduction

2. Task 1: Online Learning for Interactive Translation Prediction

3. Task 2: Active Learning for Interactive Translation Prediction

4. Task 3: Domain and User Adaptation
Introduction

- Main objective of this WP: adaptation in ITP
  1. Develop algorithms for efficient model adaptation
  2. Develop techniques for efficient sentence selection
  3. Explore approaches for domain and user adaptation

- Tasks:
  T4.1: On-line learning for ITP (months 1-24)
  T4.2: Active learning for ITP (months 13-24)
  T4.3: Domain and user adaptation (months 13-30)
Task 1: Online Learning in ITP

• First year: preliminary work with some restrictions (corpora or application)

• This year: extension to CasMaCat corpora and ITP experiments

• Three research directions:
  – Extended online learning experimentation
  – Distributed implementation of phrase-based model estimation
  – Online learning of log-linear weights
Online Learning Experiments

• First year experiments: online learning feasible in real time scenarios

• However, some crucial aspects were not studied:
  – Performance of incremental EM algorithm
  – Impact of update frequency in system performance
  – Batch versus online learning performance

• Integration within the CasMaCat Workbench

• Experiments conducted with official CasMaCat corpora

• Results reported in KSMR: $|\text{key-strokes}| + |\text{mouse-actions}| / |\text{reference characters}|$
Online Learning Experiments: EM Convergence

- Word alignments crucial in phrase-based models
- Online estimation requires the incremental EM
- Each training sample is discarded after being processed
Online Learning Experiments: Update Frequency

- The first 5,000 sentences of the Europarl training corpus were interactively translated.
- Different update frequencies were tested (every 10, 100 or 1,000 sentences).
- Results clearly show that performance is better as we increase the update frequency.
Online Learning Experiments: Batch vs Online

- Impact of frequent updates suggests to replace batch learning by online learning
Online Learning Experiments: Conclusions

• Incremental EM algorithm is competitive with batch EM

• Update frequency has a strong impact in system performance

• Ideally, models should be updated in a sentence-wise manner

• Batch learning is not appropriate for its use in online learning scenarios

• Online learning obtains the same or even better quality results than batch learning
Distributed Implementation of PB Model Estimation

• Initially implemented online learning techniques cannot be executed on large corpora

• First releases of Thot: no online estimation of word alignment models

• New functionality is being added to Thot, including:
  – Estimation of HMM-based alignment models using incremental EM
  – Map-Reduce implementation of PB estimation software

• The new software has been successfully applied in different tasks:
  – Experiments carried out during the second field trial
  – Extended online learning experiments presented above

• A new version of the Thot toolkit has been released
The Thot toolkit

• Open source toolkit for SMT (LGPL License)

• Hosted on github: https://github.com/daormar/thot

• Main features:
  – Fully-automatic SMT and ITP capabilities
  – Incremental learning
  – Scalable training

• Still under development but applied during second field trial
Online learning of log-linear weights

- During first year, promising results achieved in SMT

- Results did not carry over to ITP
  - Wordgraphs as compact representation of ITP search space pose problems:
  - Metric to be optimized does not depend on a single hypothesis
  - No single feature set for each weight set

- Approach adopted: weight sampling for building different quality wordgraphs
  - Gaussian sampling
  - Simplex sampling

- Experimentation with official CasMaCat corpora
Online learning of log-linear weights: results

Table 1: Results in KSMR of the different online learning strategies studied

<table>
<thead>
<tr>
<th>Method</th>
<th>weights</th>
<th>KSMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>—</td>
<td>40.6</td>
</tr>
<tr>
<td>original</td>
<td>—</td>
<td>42.8</td>
</tr>
<tr>
<td>gaussian</td>
<td>201</td>
<td>40.9</td>
</tr>
<tr>
<td>simplex</td>
<td>70</td>
<td>40.4</td>
</tr>
</tbody>
</table>

- Algorithms applied very positive in SMT, not so in ITP
- Mixed results point towards the need of further research
Task 2: Active Learning for ITP

- In ITP, the user is required to supervise all translations

- A more efficient approach can be implemented by:
  - Selectively supervising only a subset of the translations
  - Using an incremental SMT model to incorporate such subset
  - Maximizes the utility of each user interaction

- Related to the Active Interaction protocol in WP2, WP3
Implementation: Selective Supervision

- We decide which translations to supervise based on a translation-utility function.

- The user supervises a given percentage of the higher-scoring translations.

- Compute such score as:
  - **Uncertainty (U)**: Uncertainty of the SMT model on the translation.
  - **Information Density (ID)**: Weights U by representativeness of future translations.
  - **Coverage Augmentation (CA)**: Nr. of unknown n-grams in future translations.
  - As a baseline we consider a **Random (R)** utility function.
Results

![Graph showing BLEU and KSMR percentages with different curves for R, ID, CA, and AI categories.](image-url)
Conclusions

- Active learning has the potential to further improve the efficiency of ITP
- Active learning halves the user effort required to obtain quality translations
- Best performing strategy: coverage augmentation (though small differences)
Task 3: Domain and User Adaptation

- Adaptation critical when train and test domains mismatch

- Log-linear weights typically estimated on dev set
  - Relies on having sufficient matching (w.r.t. test) data
  - Else, estimates might be inaccurate

- Work conducted so far focused on Bayesian adaptation of log-linear weights
  - Formal framework for adaptation
  - Adaptation data is considered when computing system hypothesis
  - Gaussian prior over model parameters accounts for reliable predictions
Bayesian predictive adaptation

- Consider the integral over all the parametric space

\[ p(y|x; T, A) = \int p(y, \lambda|x; T, A)d\lambda \]

\[ \propto Z' \int p(A|\lambda; T)p(\lambda|T)p(y|x, \lambda) d\lambda \]

- Integral unfeasible ⇒ sampling with Markov chain Monte-Carlo

- Once an adequate sample \( S(\lambda_T) \) is obtained

\[ p(y|x; T, A) \approx Z' \sum_{\lambda \in S(\lambda_T)} p(y|x, \lambda) \]
Experiments with BPA: setup

- Conducted with standard Moses ($|\lambda| = 14$)

- Baseline: Phrase-pairs from Hansards, $\lambda$ estimated on Hansards dev.

- MCMC: $\mathcal{A}$ randomly extracted from new-domain training corpora, $\lambda_T$ as previous

- MERT: $\lambda$ re-estimated on $\mathcal{A}$ with MERT

- MIRA: $\lambda$ re-estimated on $\mathcal{A}$ with MIRA

- PRO: $\lambda$ re-estimated on $\mathcal{A}$ with pairwise optimization

- Such strategies not really fair: more costly, several translation steps are performed
Experiments with BPA: TER

![Graphs showing TER and CFI for different models]

- TER and CFI graphs for EMEA
- Models: baseline, MERT, MIRA, PRO, mcmc
- Data points for different model performances
- Graphs comparing models over the range of |A| values
Conclusions

• Formal and experimental results regarding log-linear weights
• Re-estimation unstable with low amounts of data, BPA stable
• Future work: log-linear feature adaptation
• Future work: other adaptation approaches:
  – Sentence selection
  – Language model adaptation
  – Translation model adaptation
Related publications

