Learning to Translate: A Query-Specific Combination Approach for Cross-Lingual Information Retrieval (CLIR)

Ferhan Ture and Elizabeth Boschee
{fture, eboschee}@bbn.com

Overview

Cross-Lingual Information Retrieval (CLIR)
Retrieve documents in a collection that are relevant to a query, where the query and documents are in different languages.
- Typically, the query is translated into the document language, or vice versa.
- Previous work has shown that:
  1. Combining different translation methods is beneficial.
  2. Effectiveness of each method differs greatly across queries and tasks.
   ➔ Our approach: A novel classification-based approach for learning how much weight to put on each translation method, for each query.

Query Translation (QT)

Methods
1-best: Output of Machine Translation (MT) system
N-best: Top N translations converted into a word by word probability distribution.
Word-based: Translation probabilities induced from word alignments from a parallel corpus.

Each method has strengths and weaknesses ➔ We take advantage of this fact to improve overall CLIR effectiveness.

Learning to Translate
A binary classification problem is designed for each translation method m:
Each query q is converted into a training instance using one of the two labeling methods:
Labeling
- By-measure: Label if m performs at least 90% as well as the best method.
- By-rank: Sort queries based on m’s effectiveness; top half gets Label=1.

We discard queries for which there is negligible difference between the effectiveness of the best and worst translation method.

Features
- Surface
  - Length
  - Stop words
  - Type of question
- Parse
  - Named entity
  - Syntactic constituents (e.g., is there an adverb in the query?)
- Translation
  - # unaligned words
  - # multi-aligned words
  - # self-aligned words
  - Entropy of translation probability distribution
- Index
  - Term frequency (tf)
  - Document frequency (df)
  - Probability assigned to OOV words

We train MaxEnt, SVM and decision tree classifiers using various feature subsets. The best classifier is selected on tuning data, under one of the three scenarios:

Tuning
- Fully-open
  - Train/tune in-domain
  - “leave one out”
- Half-blind
  - Train out-domain
- Fully-blind
  - Tune in-domain
  - Tune out-domain

➔ After training, classifier C_m is used for determining the weight of m in retrieval:
weight(m, q) ~ confidence of C_m that Label=1 for q

Evaluation

Baseline CLIR

<table>
<thead>
<tr>
<th>Task</th>
<th>1-best</th>
<th>10-best</th>
<th>Word</th>
<th>Uniform</th>
<th>Task-specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOLT_m</td>
<td>0.296</td>
<td>0.311</td>
<td>0.318</td>
<td>0.324</td>
<td>0.329</td>
</tr>
<tr>
<td>BOLT_l</td>
<td>0.370</td>
<td>0.406</td>
<td>0.407</td>
<td>0.431</td>
<td>0.431</td>
</tr>
<tr>
<td>TREC_m</td>
<td>0.292</td>
<td>0.298</td>
<td>0.301</td>
<td>0.314</td>
<td>0.318</td>
</tr>
<tr>
<td>NTCIR_l</td>
<td>0.146</td>
<td>0.152</td>
<td>0.141</td>
<td>0.162</td>
<td>0.162</td>
</tr>
</tbody>
</table>

Superscripts ** indicate statistically significant improvements (p < 0.05) over 1-best, 10-best, word-based approaches.

Our Approach

Our Approach: Effect of Labeling

<table>
<thead>
<tr>
<th>Task</th>
<th>By-Measure</th>
<th>By-Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOLT_m</td>
<td>0.342</td>
<td>0.330</td>
</tr>
<tr>
<td>BOLT_l</td>
<td>0.438</td>
<td>0.426</td>
</tr>
<tr>
<td>TREC_m</td>
<td>0.305</td>
<td>0.316</td>
</tr>
<tr>
<td>NTCIR_l</td>
<td>0.163</td>
<td>0.162</td>
</tr>
</tbody>
</table>

Mean Average Precision (MAP): measures how well the retrieval system ranked relevant documents w.r.t non-relevant ones.

Notes
- To compare the two labeling approaches, we trained a classifier with each and scored the retrieved documents (MAP):
  - For BOLT, by-measure is stat. sig. better.
  - For TREC, difference is due to two outlier queries (not stat. sig.). ➔ We decided to use by-measure labeling.

Percentage of queries for which the classifier was correct. Second value indicates “negligible queries”.

Our Approach: Effect of Tuning

Most Informative Features
- Translation-based and index-based features are selected in the best feature subset in almost all cases.
- Parse-based features most helpful in classifying NTCIR queries.

Conclusions
- Our results emphasize the benefit of query combination in CLIR, as it outperforms any single QT method on a variety of CLIR tasks.
- Finding a custom combination recipe for each query improves results even further.
- Degree of success depends highly on the availability of in-domain training/tuning data.

Future Work
- Our simple binary classifiers were successful on all four tasks, but we would like to experiment with a learning-to-rank framework for better optimization.
- Extending this approach to document translation is another interesting future direction.

Table: Query-Specific Combination

<table>
<thead>
<tr>
<th>Task</th>
<th>Query-Specific Combination</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>0.342**</td>
<td>0.330</td>
</tr>
<tr>
<td>Half</td>
<td>0.330</td>
<td>0.329</td>
</tr>
<tr>
<td>Blind</td>
<td>0.329</td>
<td>0.346</td>
</tr>
</tbody>
</table>

Classifiers trained on BOLT and TREC queries. Toned on held-out portion of NTCIR, tested on remaining.

Notes
- Our best MAP for each task is highlighted in boldface.
- “Oracle” is a hypothetical system that selects the best method for each query.
- Superscripts ** indicate stat. sig. improvements over uniform and task-specific approaches.