Finding parallel texts on the web using cross-language information retrieval

Achim Ruopp and Fei Xia
University of Washington
Jan 11, 2008
Motivation

• Parallel data are valuable resources for NLP:
  – Machine translation
  – Monolingual NLP via information projection:
  – Cross-lingual IR
  – ...

• Problem: Parallel corpora exist only for a limited set of language pairs.

• Task: Finding parallel texts on the Web
Example

Previous work:
• Ma and Liberman (1999)
• Chen and Nie (2000)
• Resnik and Smith (2003)
Main steps in identifying parallel text on the Web (Resnik and Smith, 2003)

1. Locating pages that might have parallel translations

2. Generating candidate page pairs that might be translations

3. Filtering out of non-translation candidate pairs
Our approach

1. Locating pages that might have parallel translations:
   - Sampling by sending queries

2. Generating candidate page pairs that might be translations:
   - Comparing different matching methods

3. Filtering out of non-translation candidate pairs:
   - Combining structural and content-based filtering
System Overview
Outline

• Motivation

• System description
  (1a) Sampling the source language L1
  (1b) Checking pages in the target language L2
  (2) Matching pages in L1 and L2
  (3) Filtering page pairs

• Experiments

• Conclusion and future work
(1a) Sampling L1

• The procedure:
  – Send a query to a search engine
  – Limit the search to pages in L1
  – The search engine returns up to 100 pages per query.

– An optional parameter: `inurl:<lang_id>`
  • Ex: `inurl:<en>` (for English)
The choice of source queries

• Obtain a vocabulary for L1
  – Collecting counts from corpora

• Random select mid-frequency words from the vocabulary
  – Avoid high-frequency and rare words

• Expand a query with semantically related words
Query expansion

• A common IR technique

• Based on page summaries returned by the one-word sampling query

• Summary terms ranked by frequency

• Leads to semantically related terms because of relevancy ranking of search engine results
  Shannon → information, Claude

• The original query expanded with one or more expansion terms
(1b) Checking pages in L2

• Translating the source query
  – Translate a query with a translation lexicon:
    inconvenience security travelers →
    unannehmlichkeit sicherheit”

  – Keeping m-best translations of a n-term query
    leads to $m^n$ target queries

  – An optional parameter: inurl:<lang_ID>
    • Ex:  inurl:<de> (for German)
Target language expansion

• An alternative to translating a complete n-term source query
  – First, translate the original one-term sample
  – Second, expand on target language side

• Resulting in m target queries instead of $m^n$

• Query expansion on the source side or on target side?
The “site:” parameter

• It allows limiting the query to a set of pre-defined sites.

• The search engine allows up to 30 sites.

• We use the top-30 sites returned for the source query.
(2) Matching pages in L1 and L2

• Using a fixed language list

• Levenshtein distance (a.k.a. Edit distance)

• URL part substitution
Using a language list

- Considered a match if two URLs differ only in language IDs

<table>
<thead>
<tr>
<th>en</th>
<th>de</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-us</td>
<td>de-de</td>
</tr>
<tr>
<td>en</td>
<td>ge</td>
</tr>
<tr>
<td>enu</td>
<td>deu</td>
</tr>
<tr>
<td>enu</td>
<td>ger</td>
</tr>
<tr>
<td>english</td>
<td>german</td>
</tr>
<tr>
<td>englisch</td>
<td>deutsch</td>
</tr>
</tbody>
</table>

http://ec.europa.eu/education/policies/rec_qual/recognition/diploma_de.html
Edit distance

• Calculating the edit distance between two URLs

• The pair is considered a match if the edit distance is less than n.
URL part substitution

• Sampling L1
  ➔ source URLs

• Replacing L1 names/ids in each source URL with L2 names/ids
  ➔ target URLs

• Checking whether the target URLs exist
(3) Filtering page pairs

• Structural filtering (Resnik and Smith, 2003)

• Content translation metric (Ma and Liberman, 1999)

• Linear combination
Structural filtering
(Resnik & Smith 2003)

• The HTML structure in each page is linearized

• The resulting sequences are aligned to determine the structural differences
  – Difference percentage (dp)
  – Length correlation of aligned non-markup chunks (r)
    • Pearson correlation coefficient over all aligned chunks in a page pair
    • Length of content in characters
Content translation metric

• Based on Ma and Liberman (1999)

\[ c(p_1, p_2) = \frac{\text{Num Of Translation Token Pairs}}{\text{Num Of Tokens in } p_1} \]

• Calculated on first 500 content words on page
  – Using a translation lexicon trained on Europarl corpora
Combining two kinds of metrics

- Structural metrics: dp and r

- Content-based metric: c

- Linear combination:

\[
 t_{dprc}(p_1, p_2) = \frac{a_{dp} \times (1 - dp(p_1, p_2)) + a_r \times r(p_1, p_2) + a_c \times c(p_1, p_2)}{3}
\]
Outline

• Motivation

• System description
  (1a) Sampling the source language L1
  (1b) Checking pages in the target language L2
  (2) Matching pages in L1 and L2
  (3) Filtering page pairs

• Experiments

• Conclusion and future work
Different settings for experiments

(1a) Sampling the source language L1
   – Source expansion
   – The “inurl:” parameter

(1b) Checking pages in the target language L2
   – Target expansion
   – The “inurl:” and “site:” parameter

(2) Matching pages in L1 and L2
   – Using a fixed list
   – Edit distance
   – URL part substitution
## Source expansion

<table>
<thead>
<tr>
<th>Expt ID</th>
<th>Expansion type</th>
<th>Query length</th>
<th>inurl: Param</th>
<th>Number of page pairs (before filtering)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>List</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>2</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Source</td>
<td>3</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>None</td>
<td>1</td>
<td>en/de</td>
<td>58</td>
</tr>
<tr>
<td>5</td>
<td>Source</td>
<td>2</td>
<td>en/de</td>
<td>72</td>
</tr>
<tr>
<td>6</td>
<td>Source</td>
<td>3</td>
<td>en/de</td>
<td>100</td>
</tr>
</tbody>
</table>
## Source expansion (cont)

<table>
<thead>
<tr>
<th>Expt ID</th>
<th>Expansion type</th>
<th>Query length</th>
<th>inurl: Param</th>
<th>Number of page pairs (after filtering)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>List</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Source</td>
<td>3</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>None</td>
<td>1</td>
<td>en/de</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>Source</td>
<td>2</td>
<td>en/de</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Source</td>
<td>3</td>
<td>en/de</td>
<td>25</td>
</tr>
</tbody>
</table>
## Target expansion

<table>
<thead>
<tr>
<th>Expt ID</th>
<th>Expansion type</th>
<th>Query length</th>
<th>inurl: Param</th>
<th>Number of page pairs (before filtering)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>List</td>
</tr>
<tr>
<td>7</td>
<td>None</td>
<td>1</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Target</td>
<td>2</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>Target</td>
<td>3</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>None</td>
<td>1</td>
<td>en/de</td>
<td>56</td>
</tr>
<tr>
<td>11</td>
<td>Target</td>
<td>2</td>
<td>en/de</td>
<td>107</td>
</tr>
<tr>
<td>12</td>
<td>Target</td>
<td>3</td>
<td>en/de</td>
<td>45</td>
</tr>
</tbody>
</table>
## Target expansion (cont)

<table>
<thead>
<tr>
<th>Expt ID</th>
<th>Expansion type</th>
<th>Query length</th>
<th>inurl: Param</th>
<th>Number of page pairs (after filtering)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>List</td>
</tr>
<tr>
<td>7</td>
<td>None</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Target</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Target</td>
<td>3</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>None</td>
<td>1</td>
<td>en/de</td>
<td>24</td>
</tr>
<tr>
<td>11</td>
<td>Target</td>
<td>2</td>
<td>en/de</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>Target</td>
<td>3</td>
<td>en/de</td>
<td>12</td>
</tr>
</tbody>
</table>
## Using site: parameter

<table>
<thead>
<tr>
<th>Expt ID</th>
<th>Expansion type</th>
<th>Query length</th>
<th>Site: Param</th>
<th>Number of page pairs (before filtering)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>List</td>
</tr>
<tr>
<td>1</td>
<td>None</td>
<td>1</td>
<td>No</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Source</td>
<td>2</td>
<td>No</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Source</td>
<td>3</td>
<td>No</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>None</td>
<td>1</td>
<td>Yes</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Source</td>
<td>2</td>
<td>Yes</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>Source</td>
<td>3</td>
<td>Yes</td>
<td>46</td>
</tr>
</tbody>
</table>
Observations: Sampling and Checking

• Query expansion increases the number of page pairs
  – Source and target query expansion lead to similar results
  – Difference between n=2 and n=3 is not significant

• Using `site:` and `inurl:` search parameters increases the number of discovered page pairs
  – But the parameters might miss candidate pairs that don’t follow the patterns
Observations: Matching methods

• URL part substitution >> Edit distance > Fixed language list

• Matching methods that use target queries are heavily impacted by relevancy rankings and/or translation quality.

• Edit distance matching method
  – Allows learning of URL patterns used for parallel pages
Results on filtering

• Combined filter
  – Evaluated in comparison to human judge
  – Precision: 88.9%
    • Encouraging on noisy test set
  – Recall of 36.4%
    • Low recall can be partially compensated for by submitting more queries
Conclusions

• It is possible to gather parallel pages by sending queries to the search engines.

• Query expansion and using `site:` and `inurl:` both help.

• The relevancy ranking and the limit on the number of pages are two major problems.
Future work

• To improve the precision and recall of the filtering step.

• To address the relevancy ranking and the page limit problem

• To study whether some queries are more productive than others

• To test the usefulness of the collected page pairs on applications such as MT.