Workpackage 3
Interactive Editing

Jesús González-Rubio, Hieu Hoang, Philipp Koehn, Herve Saint-Amand

November 27, 2012
<table>
<thead>
<tr>
<th>No.</th>
<th>Task</th>
<th>month</th>
<th>PM</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Sentence-level Estimation of Post-Editing Effort</td>
<td>1–24</td>
<td>7</td>
<td>on track</td>
</tr>
<tr>
<td>3.2</td>
<td>Word-level Confidence Measures</td>
<td>1–24</td>
<td>6</td>
<td>on track</td>
</tr>
<tr>
<td>3.3</td>
<td>Rules from Translation Memory</td>
<td>1–12</td>
<td>4</td>
<td>DONE</td>
</tr>
<tr>
<td>3.4</td>
<td>Visualisation of Word Alignment</td>
<td>1–12</td>
<td>4</td>
<td>DONE</td>
</tr>
<tr>
<td>3.5</td>
<td>Display Multiple Translation Options</td>
<td>1–36</td>
<td>5</td>
<td>planned</td>
</tr>
<tr>
<td>3.6</td>
<td>Authoring Assistance</td>
<td>19–36</td>
<td>5</td>
<td>planned</td>
</tr>
<tr>
<td>3.7</td>
<td>Automatic Reviewing</td>
<td>25–36</td>
<td>10</td>
<td>planned</td>
</tr>
</tbody>
</table>
Overview

- Task 3.1: Sentence-level estimation of post-editing effort (UPVLC)

- Task 3.2: Word-level confidence measures (UPVLC)

- Task 3.3: Rules from Translation Memory (UEDIN)

- Task 3.4: Visualization of word alignments (UEDIN)
Task 3.1
Sentence-Level Confidence Measures
Current Work

- Aid post-editing machine translation

- If machine translation output is too bad → do not show it

- Main approach: supervised learning problem with many features
  - research focused on dimensionality reduction methods
  - future work: train on CASMACAT field trial data

- Two publications
  - “PRHLT Submission to the WMT12 Quality Estimation Task”, Jesús González-Rubio and Alberto Sanchis and Francisco Casacuberta, WMT 2012
  - “Black Box Features for the WMT 2012 Quality Estimation Shared Task”, Christian Buck, WMT 2012
Task 3.2

Word-Level Confidence Measures
Approach

• Goal: Estimate the correctness of machine translated words

• Prior work on word-level confidence measures within interactive MT [González-Rubio et al., 2010]

• Implemented: confidence measure based on IBM Model 1
  – advantage: very fast to compute
  – other approaches: posterior methods, additional features

• Given source sentence $\mathbf{x} = x_0 \ldots x_j \ldots x_J$, confidence $\text{conf}(y_i)$ of words in translation $\mathbf{y} = y_1 \ldots y_i \ldots y_I$ computed from Model 1 probabilities $p(y_i|x_j)$

\[
\text{conf}(y_i) = \max_{0 \leq j \leq J} p(y_i|x_j),
\]

• Positive empirical results in simulated setting
Integration into Workbench

- Highlight words with low confidence

- Integration in UPVLC prototype

- Currently working on integration into CASMACAT Prototype
Planned Work

- Sequence models such as conditional random fields
- More features
  - linguistic information
  - lexical choice models
  - structural properties of the search graph
- Work on sub-word level (inflections)
- Exploitation of data generated by CASMACAT field trial
Task 3.3

Rules from Translation Memory
Main Idea

• Input

The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory

The second paragraph of Article 5 is deleted.

⇒ Part of the translation from TM fuzzy match

Part of the translation with SMT

The second paragraph of Article 21 is deleted.
Example

- Input sentence:

  The second paragraph of Article 21 is deleted.
Example

• Input sentence:

The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory:

The second paragraph of Article 5 is deleted.

= 

À l’article 5, le texte du deuxième alinéa est supprimé.
Example

• Input sentence:

   The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory:

   The second paragraph of Article 5 is deleted.

   =

   À l’article 5, le texte du deuxième alinéa est supprimé.

• Detect mismatch (string edit distance)
Example

• Input sentence:

   The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory:

   The second paragraph of Article 5 is deleted.
   \[
   \rightarrow \text{À l’article 5, le texte du deuxième alinéa est supprimé.}
   \]

• Detect mismatch (string edit distance)

• Align mismatch (using word alignment from GIZA++)
Example

- Input sentence:

  The second paragraph of Article 21 is deleted.

- Fuzzy match in translation memory:

  The second paragraph of Article 5 is deleted.

  =

  À l’article 5, le texte du deuxième alinéa est supprimé.

Output word(s) taken from the target TM
Example

• Input sentence:

  The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory:

  The second paragraph of Article 5 is deleted.

  =

  À l’article 5, le texte du deuxiéme alinéa est supprimé.

Output word(s) taken from the target TM

Input word(s) that still need to be translated by SMT
Example

• Input sentence:

   The second paragraph of Article 21 is deleted.

• Fuzzy match in translation memory:

   The second paragraph of Article 5 is deleted.

   \[=\]

   À l’article 5, le texte du deuxième alinéa est supprimé.

• Very Large Hierarchical Rule

   À l’article X, le texte du deuxième alinéa est supprimé.
   → The second paragraph of Article X is deleted.
Steps

- Fuzzy matching
  - based on string edit distance on words
    \[
    \text{FMS} = 1 - \frac{\text{edit-distance}(\text{source, tm-source})}{\max(|\text{source}|, |\text{tm-source}|)}
    \]
  - string edit distance on letters as tie breaker
  - details see [Koehn and Senellart, AMTA 2010]

- Word alignment of TM source and target
  - standard method

- Construction of very large rule
  - linking mismatch(input, TM source) to TM target
  - can be tricky
Multiple Choices

- Multiple fuzzy matches in TM with same score
  ⇒ consider all with optimal match score

- Same TM source with multiple translations
  ⇒ consider all with conditional probability score

- Fuzzy match choices integrated into decoder search
Quality Evaluation

Encoding TM fuzzy match as very large hierarchical grammar rules (VLR) outperforms baseline methods (Acquis task English-French)
Integration into Moses

- Proof-of-concept implementation before project [Koehn and Senellart, 2010]

- Integration into Moses
  - reimplementation of rule generation in C++
  - translation memory another rule table format
  - extremely easy to configure: just provide word-aligned parallel corpus

- Integration into CASMACAT Workbench
  - pass meta information about fuzzy match rule to GUI
  - highlight mismatched part with special color
  - not yet completed
  - will be tested in next field trial
Task 3.4

Visualization of Word Alignments
Idea

- Highlight source word(s) when cursor is positioned on target word
- Live updating of alignments when user changes translation
Computation of Alignment

- Training of alignment models
  - choice: HMM alignment model, since powerful and fast
  - train model with GIZA++ toolkit on parallel corpus

- Transmission of partial model to GUI
  - only word translation probabilities of source words in input needed
  - partial model compact enough to cause little overhead

- Computation of alignments
  - implementation of HMM inference and symmetrization in Javascript
  - dynamic programming allows computation of exact solution
  - requires tokenization of source (given) and target (various options)
  - special consideration for unknown words

- Demo
questions?