FBK @ IWSLT 2007

N. Bertoldi, M. Cettolo, R. Cattoni, M. Federico
FBK - Fondazione B. Kessler, Trento, Italy

Trento, 15 October 2007
Overview

- system architecture
- confusion network
- punctuation insertion
- improvement of lexicon
- use of multiple lexicons and language models
- system evaluation

Acknowledgments

- Hermes people: Marcello, Mauro, Roldano
The FBK SLT System

- input from speech (word-graph or 1-best) or text

- pre and post processing (optional)
  - use of the SRILM toolkit
  - **CN extraction**: lattice-tool
  - **punctuation insertion**: hidden-ngram
  - case restoring: disambig

- **Moses** is a text/CN decoder

- rescoring of $N$-best translations (optional)
Step 1: take the ASR word lattice

- arcs are labeled with words and acoustic and LM scores
- arcs have start and end timestamps
- any path is a transcription hypothesis
Step 2: approximate the word lattice into a *Confusion Network*

- a CN is a linear word graph
- arcs are labeled with *words* or with the *empty word* (ε-word)
- arcs are weighted with word *posterior probabilities*
- paths are a *superset* of those in the word lattice
- paths can have different lengths
- algorithm proposed by [Mangu, 2000]
  - exploit start and end timestamps of the lattice arcs
  - collapse/cluster close words
  - lattice-tool
**Step 3:** represent the CN as a *table*

<table>
<thead>
<tr>
<th>i.9</th>
<th>cannot.8</th>
<th>ϵ.7</th>
<th>say.6</th>
<th>ϵ.7</th>
<th>anything.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>hi.1</td>
<td>can.1</td>
<td>not.3</td>
<td>said.2</td>
<td>any.3</td>
<td>thing.1</td>
</tr>
<tr>
<td></td>
<td>ϵ.1</td>
<td></td>
<td>says.1</td>
<td></td>
<td>things.1</td>
</tr>
</tbody>
</table>
Step 3: represent the CN as a *table*

<table>
<thead>
<tr>
<th>i.9</th>
<th>cannot.8</th>
<th>$\epsilon$.7</th>
<th>say.6</th>
<th>$\epsilon$.7</th>
<th>anything.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>hi.1</td>
<td>can.1</td>
<td>not.3</td>
<td>said.2</td>
<td>any.3</td>
<td>thing.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\epsilon$.1</td>
<td>says.1</td>
<td>things.1</td>
<td></td>
</tr>
</tbody>
</table>

Notes

- text is a trivial CN
- CN can be used for representing ambiguity of the input
  - transcription alternatives
  - punctuation
  - upper/lower case
The Problem

- *punctuation* improves *readability* and *comprehension* of texts
- *punctuation marks* are important clues for the translation process
- most ASR systems generate output *without* punctuation
The Problem

- *punctuation* improves *readability* and *comprehension* of texts
- *punctuation marks* are important clues for the translation process
- most ASR systems generate output *without* punctuation

Our approach [Cattoni, Interspeech 2007]

- insert punctuation as a *pre-processing* step
- exploit *multiple* hypotheses of punctuation
- use *punctuated models* (i.e. trained on texts with punctuation)
- let the decoder choose the best punctuation (and translation)
**Step 1**: take the input *not-punctuated CN*
Step 2: extract the not-punctuated *consensus decoding*

i cannot say anything at this point are there any comments
**Step 3:** compute the *N-best* hypotheses of punctuation (with hidden-ngram)

<table>
<thead>
<tr>
<th>NBEST</th>
<th>Score</th>
<th>Sentence</th>
<th>Punctuation</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBEST_0</td>
<td>-15.270</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_1</td>
<td>-15.317</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_2</td>
<td>-16.275</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_3</td>
<td>-16.322</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>?</td>
</tr>
<tr>
<td>NBEST_4</td>
<td>-17.829</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_5</td>
<td>-18.284</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>?</td>
</tr>
<tr>
<td>NBEST_6</td>
<td>-18.331</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_7</td>
<td>-18.473</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
<tr>
<td>NBEST_8</td>
<td>-18.521</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>?</td>
</tr>
<tr>
<td>NBEST_9</td>
<td>-18.834</td>
<td>i cannot say anything</td>
<td>at this point</td>
<td>.</td>
</tr>
</tbody>
</table>
Step 4: compute the *punctuating CN* with *posterior probs* of multiple marks

<table>
<thead>
<tr>
<th>$i_1$</th>
<th>cannot$_1$</th>
<th>say$_1$</th>
<th>anything$_1$</th>
<th>$\epsilon$.9</th>
<th>at$_1$</th>
<th>this$_1$</th>
<th>point$_1$</th>
<th>$.7$</th>
<th>are$_1$</th>
<th>there$_1$</th>
<th>any$_1$</th>
<th>comments$_1$</th>
<th>$\epsilon$.6</th>
</tr>
</thead>
</table>
**Step 5:** *merge* the input CN and the punctuating CN
Step 6: get the final punctuated CN
Step 6: get the final punctuated CN

Notes
- this approach works with any speech input (1-best and CN) without punctuation and with partially punctuated input
Step 6: get the final punctuated CN

Notes

- this approach works with any speech input (1-best and CN) without punctuation and with partially punctuated input
- one system (with punctuated models) translates any input (text and speech)
Which is the better approach to add punctuation marks?
Which is the better approach to add punctuation marks?

• in the *source* as a *pre-processing* step
Which is the better approach to add punctuation marks?

- in the *source* as a *pre-processing* step
- in the *target* as a *post-processing* step
  - translate with not-punctuated models
  - add punctuation to the best translation (with hidden-ngram)
Which is the better approach to add punctuation marks?

- in the *source* as a *pre-processing* step
- in the *target* as a *post-processing* step
  - translate with not-punctuated models
  - add punctuation to the best translation (with hidden-ngram)

- evaluation
  - task: eval set 2006, TC-STAR English-to-Spanish
  - training data: FTE transcriptions of EPPS (36Mw English, 38Mw Spanish)
  - verbatim input (w/o punctuation), case-insensitive

<table>
<thead>
<tr>
<th>approach</th>
<th>BLEU</th>
<th>NIST</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>42.23</td>
<td>9.72</td>
<td>46.12</td>
<td>34.38</td>
</tr>
<tr>
<td>source</td>
<td>44.92</td>
<td>9.84</td>
<td>42.84</td>
<td>31.77</td>
</tr>
</tbody>
</table>
Do multiple punctuation hypotheses help to improve translation quality?
Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation)
  - case-insensitive

<table>
<thead>
<tr>
<th>input type</th>
<th># punctuation hyps</th>
<th>BLEU</th>
<th>NIST</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrb</td>
<td>1</td>
<td>44.92</td>
<td>9.84</td>
<td>42.84</td>
<td>31.77</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>45.33</td>
<td>9.83</td>
<td>42.58</td>
<td>31.59</td>
</tr>
</tbody>
</table>
Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation), 1-best
  - case-insensitive

<table>
<thead>
<tr>
<th>input</th>
<th>type</th>
<th># punctuation hyps</th>
<th>BLEU</th>
<th>NIST</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrb</td>
<td>1</td>
<td></td>
<td>44.92</td>
<td>9.84</td>
<td>42.84</td>
<td>31.77</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td></td>
<td>45.33</td>
<td>9.83</td>
<td>42.58</td>
<td>31.59</td>
</tr>
<tr>
<td>asr</td>
<td>1-best</td>
<td>1</td>
<td>35.62</td>
<td>8.37</td>
<td>57.15</td>
<td>44.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>36.01</td>
<td>8.41</td>
<td>56.78</td>
<td>44.39</td>
</tr>
</tbody>
</table>
Do multiple punctuation hypotheses help to improve translation quality?

- evaluation
  - verbatim (w/o punctuation), 1-best, and CN
  - case-insensitive

<table>
<thead>
<tr>
<th>input</th>
<th>type</th>
<th># punctuation hyps</th>
<th>BLEU</th>
<th>NIST</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrb</td>
<td>1</td>
<td>44.92</td>
<td>9.84</td>
<td>42.84</td>
<td>31.77</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>45.33</td>
<td>9.83</td>
<td>42.58</td>
<td>31.59</td>
<td></td>
</tr>
<tr>
<td>asr</td>
<td>1-best</td>
<td>35.62</td>
<td>8.37</td>
<td>57.15</td>
<td>44.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>36.01</td>
<td>8.41</td>
<td>56.78</td>
<td>44.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CN</td>
<td>36.22</td>
<td>8.46</td>
<td>56.39</td>
<td>44.37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>36.45</td>
<td>8.49</td>
<td>56.17</td>
<td>44.19</td>
<td></td>
</tr>
</tbody>
</table>
Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (GIZA++)
- combine both word-alignments in one symmetric way:
  - grow-diag-final, union, and intersection
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies
Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (GIZA++)
- combine both word-alignments in one symmetric way:
  - grow-diag-final, union, and intersection
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies

Ideas for improving the lexicon:

- use case-insensitive corpus for word-alignment, but case-sensitive extraction
Create a phrase-pair lexicon

- take a case-sensitive parallel corpus
- word-align the corpus in direct and inverse directions (GIZA++)
- combine both word-alignments in one symmetric way:
  - grow-diag-final, union, and intersection
- extract phrase pairs from a symmetrized word-alignment
- add single word translation from direct alignment
- score phrase pairs according to word and phrase frequencies

Ideas for improving the lexicon:

- use *case-insensitive* corpus for word-alignment, but case-sensitive extraction
- extract phrase pairs separately from more symmetrized word-alignments, concatenate them and compute their scores
How much improvement do we get?
How much improvement do we get?

- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets (’03-’05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

<table>
<thead>
<tr>
<th>symmetrization</th>
<th>text for word-alignment</th>
<th># phrase pairs</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>grow-diag-final</td>
<td>case-sensitive</td>
<td>496K</td>
<td>20.50</td>
<td>5.57</td>
</tr>
</tbody>
</table>
How much improvement do we get?

- evaluation
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets (’03-’05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

<table>
<thead>
<tr>
<th>symmetrization</th>
<th>text for word-alignment</th>
<th># phrase pairs</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>grow-diag-final</td>
<td>case-sensitive</td>
<td>496K</td>
<td>20.50</td>
<td>5.57</td>
</tr>
<tr>
<td></td>
<td>case-insensitive</td>
<td>507K</td>
<td>21.86</td>
<td>5.59</td>
</tr>
</tbody>
</table>
How much improvement do we get?

- **evaluation**
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets (’03-’05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

<table>
<thead>
<tr>
<th>symmetrization</th>
<th>text for word-alignment</th>
<th># phrase pairs</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>grow-diag-final</td>
<td>case-sensitive</td>
<td>496K</td>
<td>20.50</td>
<td>5.57</td>
</tr>
<tr>
<td></td>
<td>case-insensitive</td>
<td>507K</td>
<td>21.86</td>
<td>5.59</td>
</tr>
<tr>
<td>+union</td>
<td></td>
<td>507K</td>
<td>22.35</td>
<td>6.20</td>
</tr>
</tbody>
</table>
Improving Lexicon

How much improvement do we get?

- **evaluation**
  - task: IWSLT Chinese-to-English, 2006 eval set
  - training data: BTEC and dev sets (’03-’05)
  - weight optimization on 2006 dev set
  - verbatim input, case-sensitive

<table>
<thead>
<tr>
<th>symmetrization</th>
<th>text for</th>
<th># phrase pairs</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>grow-diag-final case-sensitive</td>
<td></td>
<td>496K</td>
<td>20.50</td>
<td>5.57</td>
</tr>
<tr>
<td></td>
<td>case-insensitive</td>
<td>507K</td>
<td>21.86</td>
<td>5.59</td>
</tr>
<tr>
<td>+union</td>
<td>”</td>
<td>507K</td>
<td>22.35</td>
<td>6.20</td>
</tr>
<tr>
<td>+intersection</td>
<td>”</td>
<td>5.2M</td>
<td>22.71</td>
<td>6.31</td>
</tr>
</tbody>
</table>
• *multiple training corpora*
  – non-homogeneous data (size, domain)
  – small corpus for domain adaptation
• **multiple training corpora**
  – non-homogeneous data (size, domain)
  – small corpus for domain adaptation

• **one TM and one LM**
  – concatenation of all corpora
  – corpus characteristics are (too?) smoothed
• **multiple training corpora**
  - non-homogeneous data (size, domain)
  - small corpus for domain adaptation

• **one TM and one LM**
  - concatenation of all corpora
  - corpus characteristics are smoothed

• **multiple TMs and multiple LMs**
  - **advantages**
    * more specialized models, more flexibility
    * easy combination/selection of models
    * effective (for TMs)
  - **drawbacks**
    * complexity of the model
How much improvement do we get?
How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    * baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06):
      3.8M phrase pairs, 362K 4-grams
    * EU Proceedings (39M phrase pairs, 16M 4-grams)
    * Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

<table>
<thead>
<tr>
<th>TM₁,LM₁</th>
<th>TM₂,LM₂</th>
<th>LM₃</th>
<th>OOV</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>-</td>
<td>-</td>
<td>1.68</td>
<td>28.70</td>
<td>5.76</td>
</tr>
</tbody>
</table>
How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    * baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06):
      3.8M phrase pairs, 362K 4-grams
    * EU Proceedings (39M phrase pairs, 16M 4-grams)
    * Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

<table>
<thead>
<tr>
<th>TM_1,LM_1</th>
<th>TM_2,LM_2</th>
<th>LM_3</th>
<th>OOV</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>-</td>
<td>-</td>
<td>1.68</td>
<td>28.70</td>
<td>5.76</td>
</tr>
<tr>
<td>&quot;</td>
<td>-</td>
<td>web</td>
<td>&quot;</td>
<td>29.66</td>
<td>5.83</td>
</tr>
</tbody>
</table>
How much improvement do we get?

- evaluation
  - task: IWSLT Italian-to-English, second half of 2007 dev set
  - training data:
    * baseline: BTEC, Named Entities, MultiWordNet and dev sets ('03-'06): 3.8M phrase pairs, 362K 4-grams
    * EU Proceedings (39M phrase pairs, 16M 4-grams)
    * Google Web 1T (336M 5-grams)
  - weight optimization on the first half of 2007 devset
  - verbatim input repunctuated with CN, case-insensitive

<table>
<thead>
<tr>
<th>TM₁,LM₁</th>
<th>TM₂,LM₂</th>
<th>LM₃</th>
<th>OOV</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>-</td>
<td>-</td>
<td>1.68</td>
<td>28.70</td>
<td>5.76</td>
</tr>
<tr>
<td>&quot;</td>
<td>-</td>
<td>web</td>
<td>&quot;</td>
<td>29.66</td>
<td>5.83</td>
</tr>
<tr>
<td>&quot;</td>
<td>EP</td>
<td>&quot;</td>
<td>0.28</td>
<td>30.79</td>
<td>5.92</td>
</tr>
</tbody>
</table>
1-best vs. Confusion Networks
1-best vs. Confusion Networks

<table>
<thead>
<tr>
<th>task</th>
<th>input</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, ASR</td>
<td>1bst</td>
<td>41.51</td>
</tr>
<tr>
<td></td>
<td>cn</td>
<td><strong>42.29</strong></td>
</tr>
</tbody>
</table>

* primary run

- CN outperforms 1-best
1-best vs. Confusion Networks

<table>
<thead>
<tr>
<th>task</th>
<th>input</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, ASR</td>
<td>1bst</td>
<td>41.51</td>
</tr>
<tr>
<td></td>
<td>cn</td>
<td>42.29*</td>
</tr>
<tr>
<td>JE, ASR</td>
<td>1bst</td>
<td>39.46*</td>
</tr>
<tr>
<td></td>
<td>cn</td>
<td>39.69</td>
</tr>
</tbody>
</table>

* primary run

- CN outperforms 1-best
- no inspection on CN for JE
Multiple TMs and LMs
## Multiple TMs and LMs

<table>
<thead>
<tr>
<th>task</th>
<th>TMs</th>
<th>LMs</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, clean</td>
<td>baseline</td>
<td>baseline</td>
<td>43.41</td>
</tr>
<tr>
<td></td>
<td>+EP</td>
<td>+EP+web</td>
<td>44.32*</td>
</tr>
</tbody>
</table>

* primary run
## Multiple TMs and LMs

<table>
<thead>
<tr>
<th>task</th>
<th>TMs</th>
<th>LMs</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, clean</td>
<td>baseline</td>
<td>baseline</td>
<td>43.41</td>
</tr>
<tr>
<td></td>
<td>+EP</td>
<td>+EP+web</td>
<td>44.32*</td>
</tr>
<tr>
<td>IE, ASR, CN</td>
<td>baseline</td>
<td>baseline</td>
<td>40.74</td>
</tr>
<tr>
<td></td>
<td>+EP</td>
<td>+EP+web</td>
<td>41.51*</td>
</tr>
</tbody>
</table>

* primary run
Multiple TMs and LMs

<table>
<thead>
<tr>
<th>task</th>
<th>TMs</th>
<th>LMs</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, clean</td>
<td>baseline</td>
<td>baseline</td>
<td>43.41</td>
</tr>
<tr>
<td></td>
<td>+EP</td>
<td>+EP+web</td>
<td>44.32*</td>
</tr>
<tr>
<td>IE, ASR, CN</td>
<td>baseline</td>
<td>baseline</td>
<td>40.74</td>
</tr>
<tr>
<td></td>
<td>+EP</td>
<td>+EP+web</td>
<td>41.51*</td>
</tr>
<tr>
<td>CE, clean</td>
<td>baseline</td>
<td>baseline</td>
<td>35.08</td>
</tr>
<tr>
<td></td>
<td>”</td>
<td>+web</td>
<td>33.94</td>
</tr>
<tr>
<td></td>
<td>”</td>
<td>+LDC</td>
<td>34.72*</td>
</tr>
</tbody>
</table>

* primary run

- additional TMs improves performance (+0.77 BLEU)
- Google Web LM severely affects performance on CE (-1.14 BLEU)
Future work

- punctuation insertion in other languages (Chinese, Japanese)
- use of *caseing* CN to for case restoring
Future work

• punctuation insertion in other languages (Chinese, Japanese)
• use of *caseing* CN to for case restoring
• automatic way of selecting corpora
Future work

- punctuation insertion in other languages (Chinese, Japanese)
- use of *caseing* CN to for case restoring

- automatic way of selecting corpora

- further inspection on the use of Google Web corpus
Thank you!
Chinese-to English

- word-alignment on ci texts, grow-diag-final + union + inter
- case sensitive models
- distortion models: distance-based and orientation-bidirectional-fe
- (stack size, translation option limit, reordering limit)= (2000, 50, 7)
- BTEC and dev sets ('03-'07) (TM₁: 5.9M phrase pairs, LM₁: 39K 6-grams)
  LDC: (TM₂: 27M phrase pairs)
  Google Web (LM₂: 336M 5-grams)
- 5 official runs
Japanese-to English

- word-alignment on ci texts, grow-diag-final + union + inter
- case sensitive models
- distortion models: distance-based and orientation-bidirectional-fe
- (stack size, translation option limit, reordering limit)=(2000,50,7)
- BTEC and dev sets ('03-'07) (TM$_1$: 9.1M phrase pairs, LM$_1$: 39K 6-grams)
  Reuters: (TM$_2$, 176K phrase pairs)
- 6 official runs
**Italian-to English**

- word-alignment on ci texts, grow-diag-final + union
- case insensitive TMs and LMs and case restoring
- distortion models: distance-based
- (stack size, translation option limit, reordering limit)=(200,20,6)
- BTEC NE, MWN, dev sets ('03-'07) (TM$_1$: 3.8M phrase pairs, LM$_1$: 362K 4-grams)
  EU Proceedings: (TM$_2$: 39M phrase pairs, LM$_2$: 16M 4-grams)
  Google Web (LM$_3$: 336M 5-grams)
- rescoring with 5K-best translations
- case-restoring with a 4-gram LM
- 12 official runs
• **Toolkit for SMT:**
  – translation of both text and CN inputs
  – incremental pre-fetching of translation options
  – handling multiple lexicons and LMs
  – handling of huge LMs and LexMs (up to Giga words)
  – on-demand and on-disk access to LMs and LexMs
  – factored translation model (surface forms, lemma, POS, word classes, ...)

• **Multi-stack DP-based decoder:**
  – theories stored according to the coverage size
  – synchronous on the coverage size

• **Beam search:**
  – deletion of less promising partial translations:
  – histogram and threshold pruning

• **Distortion limit:** reduction of possible alignments

• **Lexicon pruning:** limit the amount of translation options per span
• **log-linear statistical model**

• features of the *first* pass
  – (multiple) language models
  – direct and inverted word- and phrase-based (multiple) lexicons
  – word and phrase penalties
  – reordering model: distance-based and lexicalized (CE, JE)

• (additional) features of the *second* pass (IE)
  – direct and inverse IBM Model 1 lexicon scores
  – weighted sum of $n$-grams relative frequencies ($n = 1, \ldots, 4$) in $N$-best list
  – the reciprocal of the rank
  – counts of hypothesis duplicates
  – $n$-gram posterior probabilities in $N$-best list [Zens, 2006]
  – sentence length posterior probabilities [Zens, 2006]