HG8003 Technologically Speaking: The intersection of language and technology.

Statistical and Example-based Machine Translation

Francis Bond
Division of Linguistics and Multilingual Studies
http://www3.ntu.edu.sg/home/fcbond/
bond@ieee.org

Lecture 10
Location: LT8

HG8003 (2014)
## Schedule

<table>
<thead>
<tr>
<th>Lec.</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01-16</td>
<td>Introduction, Organization: Overview of NLP; Main Issues</td>
</tr>
<tr>
<td>2</td>
<td>01-23</td>
<td>Representing Language</td>
</tr>
<tr>
<td>3</td>
<td>02-06</td>
<td>Representing Meaning</td>
</tr>
<tr>
<td>4</td>
<td>02-13</td>
<td>Words, Lexicons and Ontologies</td>
</tr>
<tr>
<td>5</td>
<td>02-20</td>
<td>Text Mining and Knowledge Acquisition</td>
</tr>
<tr>
<td>6</td>
<td>02-27</td>
<td>Structured Text and the Semantic Web</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quiz</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recess</td>
</tr>
<tr>
<td>7</td>
<td>03-13</td>
<td>Citation, Reputation and PageRank</td>
</tr>
<tr>
<td>8</td>
<td>03-20</td>
<td>Introduction to MT, Empirical NLP</td>
</tr>
<tr>
<td>9</td>
<td>03-27</td>
<td>Analysis, Tagging, Parsing and Generation</td>
</tr>
<tr>
<td>10</td>
<td>Video</td>
<td>Statistical and Example-based MT</td>
</tr>
<tr>
<td>11</td>
<td>04-03</td>
<td>Transfer and Word Sense Disambiguation</td>
</tr>
<tr>
<td>12</td>
<td>04-10</td>
<td>Review and Conclusions</td>
</tr>
<tr>
<td>Exam</td>
<td>05-06</td>
<td>17:00</td>
</tr>
</tbody>
</table>

➢ Video week 10
Overview

➢ Review of Analysis, Tagging, Parsing and Generation

➢ Example-based Machine Translation

➢ Translation Memories

➢ Statistical Machine Translation
  ➢ Guest lecture by Philipp Koehn, University of Edinburgh
  ➢ http://videolectures.net/aerfaiss08_koehn_pbfs/
  Up to slide 77; note two videos
  ➢ Slides on Edventure: aerfaiss08_koehn_pbfs.pdf
Morphological Analysis and Tagging
Review of Morphological Analysis and Tagging

- Morphological analysis is the analysis of units within the word
  - Segmentation: splitting text into words
  - Lemmatization: finding the base form
  - Tokenization: splitting text into tokens (for further processing)

- Part of Speech tagging assigns POS tags to words or tokens
  - Often combined with morphological analysis
Segmentation

➢ Separate a stream into units
  ➢ non-spaced languages (Chinese, Thai, . . . )
  ➢ speech input

➢ Need both good lexicons and unknown word handling

➢ Typically learn rules from a tagged corpus
  ➢ treat rare words as unknown words

➢ Can pass ambiguity to the next stage
Lemmatization

➢ Lemmatization is the process of finding the stem or canonical form

➢ You must store all irregular forms

➢ You need rules for the rest (inflectional morphology)

➢ Rare words tend to be regular

   ➢ For languages without much morphology, you can expand everything offline

➢ Most rules depend on the part-of-speech

   ➢ So lemmatization is done with (or after) part-of-speech tagging
Tokenization

- Splitting **words** into **tokens** — the units needed for further parsing
  - Separating punctuation
  - Adding BOS/EOS (Beginning/End of sentence) markers
  - Splitting into stem+morph: *went* → *go*+ed
  - Normalization
    * *data base*
    * *data-base*
    * *database*
  - Possibly also chunking
    * *in order to* → *in_order_to*

- This process is very task dependent
Parts of Speech (POS)

➢ Four main open-class categories

   **Noun** heads a noun phrase, refers to things
   **Verb** heads a verb phrase, refers to actions
   **Adjective** modifies Nouns, refers to states or properties
   **Adverb** modifies Verbs, refers to manner or degree
   **Pronoun** *I, he, it*; **Auxiliary Verb** *be, have*

➢ Closed class categories vary more

   **Preposition** *in, of*: links noun to verb (postposition)
   **Conjunction** *and, because*: links like things
   **Determiner** *the, this, a*: delimits noun’s reference
   **Interjection** *Wow, um:*
   **Number** *three, 125*: counts things
   **Classifier** 頭 “animal”: classifies things
Part of Speech Tagging

➢ Exploit knowledge about distribution
  ➢ Create tagged corpora

➢ With them, it suddenly looks easier
  ➢ Just choose the most frequent tag for known words (I pronoun, saw verb, a article, . . .)
  ➢ Make all unknown words proper nouns
  ➢ This gives a baseline of 90% (for English)

➢ The upper bound is 97-99% (human agreement)
  ➢ The last few percent are very hard
Two opposite needs:

- Disambiguate early
  → Improve speed and efficiency
- Disambiguate late
  → Can resolve ambiguities with more information

Several Strategies:

- Prune: Discard low-ranking alternatives
- Use under-specification (keep ambiguity efficiently)
- Pack information in a lattice (keep ambiguity efficiently)

Combine tasks instead of pipe-lining
Parsing and Generation
Review of Parsing and Generation

➢ Parsing
   ➢ Words to representation

➢ Generation
   ➢ Representation to words

➢ Two main syntactic representations:
   ➢ Dependencies (word-to-word)
   ➢ Phrase Structure Trees (with phrasal nodes)
Algorithmic Complexity and Big-O Notation

—we measure the efficiency of an algorithm using big-O notation

\[ f(n) = O(g(x)) \text{ iff } f(n) < M|g(x)| \text{ for all } x > x_0 \]

—we describe the behaviour of a function for big numbers.

—we want the function \( g(x) \) to grow as slowly as possible

<table>
<thead>
<tr>
<th>Function</th>
<th>Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O(0) )</td>
<td>Constant</td>
<td>hash lookup</td>
</tr>
<tr>
<td>( O(n) )</td>
<td>Linear</td>
<td>parsing ( n ) sentences</td>
</tr>
<tr>
<td>( O(n^2) )</td>
<td>Quadratic</td>
<td>dependency parsing ( n ) word sentence</td>
</tr>
<tr>
<td>( O(n^3) )</td>
<td>Cubic</td>
<td>LR parsing ( n ) word sentence</td>
</tr>
<tr>
<td>( O(n^c) )</td>
<td>Polynomial of degree ( c )</td>
<td>HPSG parsing ( n ) word sentence (( c = 5 ))</td>
</tr>
<tr>
<td>( O(c^n) )</td>
<td>Exponential</td>
<td>ambiguity in ( n ) word sentence</td>
</tr>
<tr>
<td>( O(n!) )</td>
<td>Factorial ( c )</td>
<td></td>
</tr>
</tbody>
</table>
Comparing Growth of Functions

Comparison of different orders of complexity.

- $O(N)$
- $O(N \log N)$
- $O(N^2)$
- $O(N!)$
- $O(k)$
- $O(\log N)$
Efficiency is important

➤ Need to avoid exponential processing

➤ Least complex is best
  ➤ constant < linear < polynomial < exponential

➤ May sacrifice some accuracy for speed
  ➤ Discard low ranked paths (known as pruning)
Dependency and PSG

 Dependencies Grammars $O(n^2)$

Phrase Structure Grammars $O(n^3)$
Generation: Process

➢ Generation is the process of producing language

➢ In the general case, this involves:

➢ **Goal**
  ➢ Text Planner $\rightarrow$ *Text Plan* \hspace{1cm} (**What-to-say**)  
  ➢ Sentence Planner $\rightarrow$ *Sentence Plan* \hspace{1cm} (**How-to-say**)  
  ➢ Linguistic Realizer $\rightarrow$ *Surface Text*

➢ At the lowest level, it involves:

➢ Taking a semantic realization and producing a string
  ➢ Normally multiple strings are possible
    ∗ Over-generate and rank with a language model
Machine Translation
Approaches to Machine Translation

**Rule-based MT**  Find the meaning, translate it, generate it

**Example-based MT**  See how this has been translated before, translate it the same way

**Statistical MT**  Imagine that the text has been encoded in some way, try to decode it
“Man does not translate a simple sentence by doing deep linguistic analysis, rather, man does translation, first, by properly decomposing an input sentence into certain fragmental phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference.”

Makoto Nagao (1984)
EBMT Method

➤ When translating, reuse existing knowledge:

0. Compile and align a database of examples
1. Match input to a database of translation examples
2. Identify corresponding translation fragments
3. Recombine fragments into target text

➤ Example:

➤ Input: *He buys a book on international politics*
➤ Data:
   * He buys a notebook – *Kare wa noto o kau*
   * I read a book on international politics – *Watashi wa kokusai seiji nitsuite kakareta hon o yomu*
➤ Output: *Kare wa kokusai seiji nitsuite kakareta hon o kau*
EBMT ‘Pyramid’

Matching (Analysis) → Interlingua → Recombination (Generation)

Exact Match (Direct Translation) → Alignment (Transfer) → Source Language

Example-based Translation: Advantages/Disadvantages

➢ Advantages
  ➢ Correspondences can be found from raw data
  ➢ Examples give well structured output if the match is big enough

➢ Disadvantages
  ➢ Lack of well aligned bitexts
  ➢ Generated text tends to be incohesive
Open Questions

 Representation of examples:

 ➢ Which representation should be used for examples?
   * $n$-grams
     people saw her; saw her duck; her duck .
   * Dependencies
   * Phrase-structure trees

 ➢ How much information do we need encoded in examples? (lemmas, morphological, syntactic, semantic, ...)

Statistical and Example-based Machine Translation 24
How big should the translation unit be:

- Large → better translation / less likely to be used
- Small → better cover / leads to literal translations
- Mixed: combine different sizes (best of both worlds)

Score translation candidates:

- heuristics
  - size of units
  - semantic closeness to input
- probabilistic models (becomes SMT)
Early EBMT: ATR System (1991)


- Japanese-English translation: $N_1 \text{ no } N_2$ “$N_2$ of $N_1$” problem

- When EBMT is better suited than Rule-based MT
Translating “N₁ no N₂”

➢ の no “ADN” is an adnominal particle

➢ Variants: での deno, までの madeno, …

➢ “N₁ no N₂” → “N₂ of/for/in/φ/’s N₁”

youka no gogo The afternoon of the 8th
kaigi no mokuteki The objective of the conference
kaigi no sankaryou The application fee for the conference
isshukan no kyuka A week’s holiday
kyouto deno kaigi The conference in Kyoto
mittsu no hoteru Three hotels

➢ Many different combinations
Difficult linguistic phenomena

➢ It is difficult to hand-craft linguistic rules for “N₁ no N₂”
   ➢ Requires deep semantic analysis for each word
   ➢ So remember examples, and then match the closest
     input  *mikka no asa* “3rd of morning”
     match  *youka no gogo* → “the afternoon of the 8th”
     * matching was done using an ontology
     * 3rd ≈ 8th
     * morning ≈ afternoon
     Translation “the morning of the 3rd”
Example of a larger match

Head switching between the English verb *like* into French *plaire*.

➢ *Sam* likes the new laser printer.

➢ La nouvelle imprimante à laser plaît à *Sam*.

Remember the whole pattern:

➢ *X* likes *Y*

➢ *Y* plaît *X.*
When EBMT works better than Rule-based MT

➤ The translation rule is difficult to formulate

➤ General rule cannot accurately describe phenomena due to special cases (e.g. idioms)

➤ Translation cannot be made compositionally using only target words

➤ The sentence to be translated has a close match in the database.
EBMT Example System: Eureka

➢ A Hybrid Translation Method (Shirai, Bond and Takahashi, 1997)

➢ Dynamically filters example sentences
➢ Uses components of existing rule based systems

➢ General Approach

1. Select a set of candidate sentences similar to the input sentence
2. Select the most typical translation from the candidates’ translations
3. Use this translation and its source as templates to translate the input sentence

➢ The closest source language match may not be the best translation!
The Nikkei Average October contracts continued declining.

The Nikkei Average September contracts were lower.

The Nikkei over-the-counter average continued declining.

August contracts continued declining.
(2) Identify the Most Similar Pair

$S_I$ nikkei heikin 10 gatsu mono wa zokuraku .
$S_1$ nikkei heikin 9 gatsu mono wa zokuraku .
\hspace{1cm} (7/8 shared segments = 0.875 )
$S_2$ 8 gatsu mono wa zokuraku .
\hspace{1cm} (5/6 shared segments = 0.833)
$S_3$ nikkei-tentoo-heikin wa zokuraku .
\hspace{1cm} (3/4 shared segments = 0.75)

Actual metric in Eureka considers:

➤ the number of matching characters
➤ the number of non-matching characters
➤ the number of continuous matching characters
(3) Recombination

1. Find differences
2. Choose constituent
3. Replace differing constituent
4. Smooth the output
   ➤ Number agreement
   ➤ Filter adjuncts

$S_I$ nikkei heikin 10 gatsu mono wa zokuraku .
$S_t$ nikkei heikin 9 gatsu mono wa zokuraku .
$T_t$ The Nikkei Average September contracts were lower.
$T_I$ The Nikkei Average October contracts were lower.
Eureka is adaptive: also consider the translation

- When we select the translation fragment
  - Cluster candidates with similar translations
  - Choose the template from the largest cluster

  $T_1$  The Nikkei$_2$ Average$_1$ September$_1$ contracts$_2$ were lower$_1$

  $T_2$  August$_1$ contracts$_2$ continued$_2$ declining$_2$
  $T_3$  The Nikkei$_2$ over-the-counter$_1$ average$_1$ continued$_2$ declining$_2$

- The source and target template are: The sentence pair in the best cluster with the highest similarity to the input sentence: $T_2$

- Weed out atypical (strange) translations
Recombination

1. Find differences
2. Choose constituent
3. Replace differing constituent
4. Smooth the output
   ➢ Number agreement
   ➢ Filter adjuncts

\[ S_I \quad \text{nikkei heikin 10 gatsu mono wa zokuraku .} \]
\[ S_t \quad \text{8 gatsu mono wa zokuraku .} \]
\[ T_t \quad \text{August contracts continued declining.} \]
\[ T_I \quad \text{The Nikkei Average October contracts continued declining.} \]
General EBMT Issues:

Sentence or sub-sentence?

Sentence:
- Better quality translation
- Boundaries are easy to determine
- Harder to find a match

Sub-sentence:
- Studies suggest this is how humans translate

Boundary friction
- The handsome boy ate his breakfast ↔ Der schone Junge as seinen Fruhstuck
- I saw the handsome boy ↔ Ich sah den schonen Jungen
General EBMT Issues: Suitability of Examples

Some EBMT systems do not use raw corpus directly, but use manually-constructed examples or carefully-filtered set of real-world examples.

Real-world examples may contain:

- Examples that mutually reinforce each other (overgeneration)
- Examples that conflict
- Examples that mislead the distance metric
  - Watashi wa kompyuta o kyoyosuru ↔ I share the use of a computer
  - Watashi wa kuruma o tsukau ↔ I use a car
  - Watashi wa dentaku o shiyosuru ↔ * I share the use of a calculator

*Note: The examples are given in Japanese and their English translations are provided for clarity.
General EBMT Issues: Matching

➢ String matching / IR-style matching
   ➢ “This is shown as A in the diagram”
     ⇔ “This is shown as B in the diagram”
   ➢ “The large paper tray holds 400 sheets of paper”
     ⇔ “The small paper tray holds 300 sheets of paper”

➢ Matching by meaning:
   ➢ use thesaurus and distance based on semantic similarity

➢ Matching by structure:
   ➢ Tree edit distance, etc.
Once a set of examples close to the input are found, we need to carry out:

- **Alignment:**
  Identify which portion of the associated translation corresponds to input

- **Recombination:**
  Stitch together these portions to create smooth output
State of the Art

➢ EBMT does best with well aligned data in a narrow domain
  ➢ There are not so many domains with such data

➢ EBMT not used in commercial systems

➢ EBMT eclipsed by SMT in competitions

➢ Still a healthy research community

➢ EBMT and SMT converging
  ➢ EBMT adds probabilistic models
  ➢ SMT adds larger phrases
Translation Memories

Translation Memories are aids for human translators

- Store and index entire existing translations
- Before translating new text
  * Check to see if you have translated it before
  * If so, reuse the original translation

- Checks tend to be very strict ⇒ translation is reliable
  - Identical except for white-space differences

- Now extended to fuzzy matching and replacing
  - Equivalent to EBMT
  - More flexible, greater cover, less reliable
Translation Memories

➢ TM are popular with translators

➢ Well integrated with word processors

➢ The translator is in control

➢ Translation companies can pool memories, giving them an advantage

➢ Simple solutions sell well

➢ Good tools also integrate
  ➢ highlighted differences from closest match
  ➢ dictionary look up
  ➢ collocational look up (words in context)
Translation Memory: OmegaT

OmegaT is a free multiplatform Computer Aided Translation tool.

1) OmegaT is a free multiplatform Computer Aided Translation tool. OmegaT is, 自由に、かつ、無料で使える翻訳支援ツールです。<100% 1.6.2_total.tmx>

2) OmegaT is a Computer Assisted Translation tool. OmegaT は、翻訳支援ツールです。<66%>

3) OmegaT is a Computer Assisted Translation tool. OmegaT は、コンピュータを利用した翻訳ツールです。<66% 1.6.2_total.tmx>

'Computer Assisted Translation tool' = '翻訳支援ツール'

Statistical and Example-based Machine Translation
EBMT/TM Summary

➢ Example-based Machine Translation (EBMT)

0. Compile and align a database of examples
   1. Match input to a database of translation examples
   2. Identify corresponding translation fragments
   3. Recombine fragments into target text

➢ Translation Memories (TM)

➢ Reuse complete existing translations
➢ Keep the translator in control
➢ TM: Popular with human translators

➢ Recent TMs also allow some variables, mainly numbers and nouns
Statistical Machine Translation (SMT)

➢ Find the translation with the highest probability of being the best.
   ➢ Probability based on existing translations (bitext)

➢ Balance two things:
   ➢ Adequacy (how faithful the translation to the source)
   ➢ Fluency (how natural is the translation)

➢ These are modeled by:
   ➢ Translation Model: $P(T|S)$
     how likely is it that this translation matches the source
   ➢ Language Model: $P(T)$
     how likely is it that this translation is good English

➢ Overall: $\hat{T} = \arg\max_T P(S|T) = \arg\max_T P(T|S)(T)$
SMT: Basic Method

The basic method (Brown et al. 1990): Choose the English Translation with the highest probability of translating the Japanese text.

$$\hat{E} = \arg\max_E P(E|J)$$
I want strong coffee.
Strong coffee please.
I'd like to have some strong coffee.
...

\[
\hat{E} = \arg\max_{E} P(J|E)P(E)
\]
Word Alignment

show₁ me₂ the₃ one₄ in₅ the₆ window₇

ウィンドー₁  □    □    □    □    □    ■    ■
の₂      □    □    □    □    □    □    □
品物₃    □    □    □    ■    □    □    □
を₄      □    □    □    □    □    □    □
見せ₅    ■    ■    □    □    □    □    □
てください₆  ■    ■    □    □    □    □    □

➤ Not all words align
を₄ ACC, the₃, in₅

➤ Some alignments are unexpected
てください₁ te-kudasi “please” → me₂

➤ Tend to be diagonal
Another way of looking at it

\[ E = \text{NULL}_0 \quad \text{show}_1 \quad \text{me}_2 \quad \text{the}_3 \quad \text{one}_4 \quad \text{in}_5 \quad \text{the}_6 \quad \text{window}_7 \]

\[ J = \quad \text{ウィンドー}_1 \quad \text{の}_2 \quad \text{品物}_3 \quad \text{を}_4 \quad \text{見せ}_5 \quad \text{てください}_6 \]

\[ A = ( \quad 7 \quad 0 \quad 4 \quad 0 \quad 1 \quad 1 \quad ) \]

Note that the alignment is one way

- NULL only on source side (fertility)
- one to many mapping
Combining Alignments

Spanish to English

English to Spanish

Intersection
Combining Alignments

In SMT, these chunks are called phrases although they may not be linguistic units (e.g. Maria no ↔ Mary did not).

<table>
<thead>
<tr>
<th></th>
<th>Maria</th>
<th>no</th>
<th>dió</th>
<th>una</th>
<th>a</th>
<th>la</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Maria, Mary), (no, did not),
(slap, dió una bofetada), (verde, green),
(a la, the), (bruja, witch),
(Maria no, Mary did not),
(no dió una bofetada, did not slap),
(dió una bofetada a la, slap the),
(bruja verde, green witch),
(a la bruja verde, the green witch), …
Improving Alignments

➤ The most common alignment approaches use word cooccurrence over large bitexts

➤ How often do words $s_i$ and $t_j$ appear in the same sentence pair compared to appearing in different sentence pairs.

➤ Other possible features include

➤ same semantic class (from an ontology or cluster)
➤ orthographic similarity
➤ shared translations in a third language

➤ These are especially useful for matching rare words
Translation Model (IBM Model 4)

\[ P(J, A|E) \]
Fertility Model
\[ \prod n(\phi_i|E_i) \]
NULL Generation Model
\[ \left( m-\phi_0 \right) p_0^{m-2\phi_0} p_1^{\phi_0} \]
Lexicon Model
\[ \prod t(J_j|E_{A_j}) \]
Distortion Model
\[ \prod d_1(j - k|A(E_i)B(J_j)) \]
\[ \prod d_{1>}(j - j'|B(J_j)) \]

could you recommend another hotel
could could recommend another another another hotel
could could recommend NULL another another another hotel
ていただけ ます 紹介し を 他の ホテル か

 Millions of candidates are produced and ranked.
Euro Matrix

➤ Translation between EU languages (EU funded project)


➤ 18-40 million words, .6–1.3 million sentences

➤ Freely available text in all European Languages

Europarl: A Parallel Corpus for Statistical Machine Translation, Philipp Koehn, MT Summit 2005
# Euro Matrix Results

<table>
<thead>
<tr>
<th>Output Language</th>
<th>Danish</th>
<th>Dutch</th>
<th>German</th>
<th>Greek</th>
<th>English</th>
<th>Finnish</th>
<th>French</th>
<th>Italian</th>
<th>Portuguese</th>
<th>Spanish</th>
<th>Swedish</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>20.51</td>
<td>18.39</td>
<td>17.49</td>
<td>23.01</td>
<td>10.34</td>
<td>24.67</td>
<td>20.07</td>
<td>20.71</td>
<td>22.95</td>
<td>19.03</td>
<td>22.22</td>
</tr>
<tr>
<td>BLEU</td>
<td>22.35</td>
<td>23.40</td>
<td>20.75</td>
<td>25.36</td>
<td>11.88</td>
<td>27.75</td>
<td>21.36</td>
<td>23.28</td>
<td>25.49</td>
<td>20.51</td>
<td>24.30</td>
</tr>
<tr>
<td>BLEU</td>
<td>22.79</td>
<td>20.02</td>
<td>17.42</td>
<td>27.28</td>
<td>11.44</td>
<td>32.15</td>
<td>26.84</td>
<td>27.67</td>
<td>31.26</td>
<td>21.23</td>
<td>24.30</td>
</tr>
<tr>
<td>BLEU</td>
<td>23.73</td>
<td>21.13</td>
<td>18.54</td>
<td>26.13</td>
<td>30.00</td>
<td>12.63</td>
<td>32.48</td>
<td>35.37</td>
<td>38.47</td>
<td>22.68</td>
<td>34.02</td>
</tr>
<tr>
<td>BLEU</td>
<td>21.47</td>
<td>20.07</td>
<td>16.92</td>
<td>24.83</td>
<td>27.89</td>
<td>11.08</td>
<td>36.09</td>
<td>31.20</td>
<td>34.04</td>
<td>20.26</td>
<td>35.37</td>
</tr>
<tr>
<td>BLEU</td>
<td>23.27</td>
<td>20.23</td>
<td>18.27</td>
<td>26.46</td>
<td>30.11</td>
<td>11.99</td>
<td>39.04</td>
<td>32.07</td>
<td>37.95</td>
<td>21.96</td>
<td>38.47</td>
</tr>
<tr>
<td>BLEU</td>
<td>24.10</td>
<td>21.42</td>
<td>18.29</td>
<td>28.38</td>
<td>30.51</td>
<td>12.57</td>
<td>40.27</td>
<td>32.31</td>
<td>35.92</td>
<td>23.90</td>
<td>28.66</td>
</tr>
<tr>
<td>BLEU</td>
<td>30.35</td>
<td>21.94</td>
<td>18.97</td>
<td>22.86</td>
<td>30.20</td>
<td>15.37</td>
<td>29.77</td>
<td>23.94</td>
<td>25.95</td>
<td>28.66</td>
<td>24.30</td>
</tr>
</tbody>
</table>
Linguistic similarity affects the score:

- Highest: Spanish $\rightarrow$ French (BLEU = 40.27)
- Lowest: Italian $\rightarrow$ Finnish (BLEU = 11.08)

Creating all $n(n - 1)$ language pairs took a week

- It is easy to add new languages if you have a multi-lingual corpus

Translation done using the open source SMT System:

Moses <statmt.org>
More data improves BLEU: (Och, 2005)

- Doubling the translation model data gives a 2.5% boost.
- Doubling the language model data gives a 0.5% boost.
- For linear improvement in translation quality the data must increase exponentially
  * BLEU +10% needs $2^4 = 16$ times as much bilingual data
  * BLEU +20% needs $2^8 = 256$ times as much bilingual data
  * BLEU +30% needs $2^{12} = 4096$ times as much bilingual data

- Richer models improve quality ⇒ syntax based models

- Pruning bad models improves quality ⇒ multilingual models
Readings

➢ Good site for SMT: statmt.org

➢ Statistical Machine Translation: Jurafsky and Martin (2009), Chapter 25.3–8

➢ Now listen to Philip Koehn’s talk!

➢ Lecture by Philipp Koehn, University of Edinburgh
➢ http://videolectures.net/aerfaiss08_koehn_pbfs/

Up to slide 77; note two videos
➢ Slides on Edventure: aerfaiss08_koehn_pbfs.pdf