Carnegie Mellon University
Spring 2007

11-731 Machine Translation

Example-Based MT (EBMT)

Additional Instructor: Bob Frederking
Phone: 268-6656
Email: ref@cs.cmu.edu
Office Hours: By appointment
Introduction:

This is the second lecture on “empirical” techniques.

Example-Based MT is machine translation based on examples(!)

This is obviously a very loose definition, so we will describe:

- the motivations for this approach,
- the characteristics of EBMT systems,
- the taxonomy of EBMT systems,
- and two example systems in detail:
  - CMU’s Pangloss/DIPLOMAT EBMT
  - Trinity College’s ReVerb
Motivations for EBMT:

KBMT is hard: Knowledge engineering is labor-intensive and hard to get right, and we now have a lot of electronic data, with more becoming available all the time.

Incremental improvement should be possible: just add more examples to the system.

Hard-to-capture phenomena may be caught.

Translation efficiency: we don’t want to retranslate sentences we’ve already done. Much easier to just look up old translation (if it’s indexed).
Elusive phenomena: Word choice:

Database of examples for *the gold fields*:

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>the main fields</td>
<td>les principaux domaines</td>
</tr>
<tr>
<td>the following fields</td>
<td>les domaines suivants</td>
</tr>
<tr>
<td>these two fields</td>
<td>ces deux domaines</td>
</tr>
<tr>
<td>the specialized fields</td>
<td>les domaines specialises</td>
</tr>
<tr>
<td>the para-medical fields</td>
<td>acivites paramedicales</td>
</tr>
<tr>
<td>the magnetic fields</td>
<td>les champs magnetiques</td>
</tr>
<tr>
<td>the coal fields</td>
<td>les bassins-houiliers</td>
</tr>
<tr>
<td>the coal fields</td>
<td>les bassins</td>
</tr>
<tr>
<td>the corn fields</td>
<td>les champs de ble</td>
</tr>
</tbody>
</table>

*Gold* is closest to *coal*, so pick *bassins*.

Depends on having a good thesaurus/ontology.
Elusive phenomena: Complex noun phrases:

Direct translation of Japanese *no*:

afternoon of the 8th
application fee of the conference
conference of Tokyo
holiday of a week
*reservation of hotel
*hotels of three

Similar problem for Spanish or French *de*. 
Generic EBMT diagram
(from ReVerb):
Sub-sentential EBMT:

But since exact sentence matches only occur in special domains (manual revisions), we want to extend this to sub-sentence matches. This is where it gets interesting, since we need to:

- Find the most similar example (involves segmenting)
- Alter source side to match current input
- Alter target side the “same” way

But now computational costs start to go up; though development is still cheaper than KBMT.
Sub-sentence EBMT requirements:

Need to do “sub-sentential alignment” somehow.

Similarity requires a “distance metric” in the source language (at least). This can be closeness:

- of the lexical items in a hierarchy of terms/concepts from thesaurus/ontology,
- of the sequence of syntactic categories and function words,
- of the two syntactic structures,
- or combinations of these.

Example:
Remove [the bulb] and replace it [with a new one]

Might want to take into account frequency of chunk in source or target (or cooccurrence). This begins to get into statistics.

Also, need to recombine pieces: this leads to the “boundary friction” problem.
External Classification of EBMT:

EBMT is one of the “empirical approaches”: knowledge derived from data, rather than linguists’ heads. This is really a continuum, since:

- linguists today use “corpus-based” approaches to derive corpus-specific insights
- empirical systems require insights from empiricists’s heads to achieve reasonable performance.

After enough processing, the EBMT example-base could become a large number of specific transfer rules plus a mechanism for selecting the closest rule.

Distinction from other empirical approaches (such as “statistical MT”) is also not always totally clear; seems to be the use of a knowledge-base containing actual examples (but why isn’t a trigram table an example-base?)
Internal Classification of EBMT:

Examples of EBMT systems:

- Translation Memory (Commercial TM)
- Pangloss/DIPLOMAT EBMT (CMU)
- Cranias et al. (ILSP, Greece)
- ReVerb (Trinity Dublin)
- Iida (ATR-ITL)
- Sato & Nagao (JIST, Kyoto U.)

Range of types of EBMT:

- exact whole sentences
- surface substrings
- sequences of tags, simple transformations
- full dependency analysis, context-dependent tree operations
Pangloss/DIPLOMAT EBMT:

Very large sentence-aligned corpus: 726k sentence pairs! (Well, large for 1992.)

Due to large size, simple surface-level string matching effective (and necessary!)

As in other systems, two stage matching: SL’:SL, then SL:TL.

Also as in other systems, sub-sentential matching. Chunks from different corpus matches combined (with chunks from other engines) by chart manager, to be described in Multi-Engine MT lecture.

Two versions of EBMT system, Baroque and Modern.
PanEBMT: Corpus:

726k sentence pairs: mostly UN (500 Megabytes per language), plus 10k PAHO, 552 pairs DARPA-MT. (In early Spanish-to-English system.)

UN corpus contained many difficulties:

- inconsistent formatting
- sections alphabetized by country
- footnotes in random places
- heterogeneous mixture of documents

Threw out difficult material, and paragraphs with unequal numbers of sentences (5% of corpus).
PanEBMT 1: 
SL’ to SL match:

Sentence broken into segments at punctuation and words not in corpus. Words of sentence looked up in index, except for “stop list” of frequent words.

Find best set of chunks that cover input. Chunks are allowed to contain, but penalized for:

- Gaps up to five words
- Word order differences
- Mismatched words
- Morphological variations

Also penalized for different word ordering.

Ten best matches are returned for each chunk.
PanEBMT 1:
SL:TL matching:

Machine-readable dictionary (MRD) and morphological analyzers for both languages used to establish cross-language correspondences.

Compare SL and TL word stems to get longest TL substring, then find “best” substring of that.

Start with smallest possible translation (at least one unambiguous match), and then extend up to word that does not correspond to SL chunk.

Problems caused by:
• Idioms
• Repeated content words
• Word-sense confusion
PanEBMT 1:  
SL:TL scoring:

Score is based on minimizing the estimated number of keystrokes to correct translation. Derived by statistical training on a manually scored set of examples.

Eight variables trained:
1. Number of SL words with TL correspondance
2. Number of SL content words w/out TL correspondance
3. Number of TL content words w/out SL correspondance
4. Minimum of 2 and 3
5. Number of stop list words in both SL and TL
6. Number of SL content words w/out TL correspondance, but present in the TL sentence
7. Whether TL chunk is at beginning or end of sentence
8. Difference in number of stop list words between SL and TL
PanEBMT 2:

First version was partly in Lisp; never managed to index more than 25% of the corpus!

New version less “fancy”, but uses full corpus. Distinctly more effective: 70 to 85% coverage of inputs. Also much faster.

Stop list dropped!
PanEBMT 2:  
SL’:SL matching:

New version does not try to find optimal cover; produces all sub-sentential alignments it can find, lets MEMT sort it out later.

Chunks retrieved from corpus based on finding at least two contiguous words, and at least one of them unambiguously. For each chunk, last 5 occurrences returned.
PanEBMT 2:
SL:TL matching:

Configuration file enhancements:

• **Target-language root/synonym list:** morphological processing moved offline.

• **Word equivalence classes:** day and month names, countries, units, etc., that can be tokenized (variablized).

• **Optional words:** words that can be deleted or inserted.

Full sentence match used directly
PanEBMT 2:  
SL:TL scoring:

Otherwise, identify minimum and maximum segments, score every possible in-between chunk.

Scoring can be length-dependent. Weighted terms:
1. Number of SL words w/out TL correspondance
2. Number of TL words w/out SL correspondance
3. Minimum of 1 and 2
4. Number of SL content words w/out TL correspondance, but present in the TL sentence
5. Sentence boundaries shared between SL and TL
6. Elidable SL words
7. Insertable TL words
8. Different in length between SL and TL
**ReVerb EBMT:**

View EBMT as case-based reasoning (CBR), with sparse data requiring adaptation.

They claim that they abstract to a “dependency level” of representation, although the examples don’t seem to support this.

Best match of feature lists (tags), rather than dependency subtrees as in Sato’s system.

“Adaptability” scores is meant to penalize within-language or cross-language dependency.

Directly address “translation divergences”: links established based on lexical meaning, not POS. But they penalize matches for POS mismatches.

Corpus consists of bilingual English/German CorelDRAW manuals. (214 sentences?) Corpus was tagged for morphology, POS, and syntactic function, then hand-disambiguated. (No attachment implies no real dependency analysis).
Adaptability/Mappability for a chunk has 4 discrete values:

- Level 3: SLA:TLA mapping is one-to-one for all words
- Level 2: Syntactic Functions map, but not some POS tags
- Level 1: Functions differ, but lexical correspondence still holds
- Level 0: Cannot establish correspondence

Generalization ("Case templatization"): substitute POS tags for chunks that are mappable at a given level:

0: SUBJ FMAINV OBJ NFMAINV OBJ ADVL ADVL
1: SUBJ FMAINV OBJ NFMAINV OBJ ADVL ADVL
2: SUBJ allows OBJ NFMAINV OBJ ADVL on the kbd
3: Move allows you to move the window with the dir keys on the kbd

Extrapolation: 20,000 cases would give 20 level 3 templates.
ReVerb: Retrieval:

Cases indexed via SL words. Retrieval is finding a good SLA given the input SL’.

Retrieval is done in two stages:

- Phase 1: activate cases whose words exactly match input.
- Phase 2: extra activation for head, and for good mappability scores

(They confuse marker-passing and activation.)
ReVerb: Adaptation:

Runtime adaptation of TLA to account for SLA/SLA’ differences is based only on dependency, not linear order.

They replace the divergent piece, using a corresponding TL piece from the case base.

Divergent cases lead to errors, which are user corrected, and then stored as new cases.

Translation is quality scored, based on mappability of substitutions, and non-substituted pieces.

Use bilingual dictionary to augment case base.
ReVerb: Evaluation/Examples:

74% of the test cases found no exact matches, even at level 0 (fully variablismed). Quality increases, quantity decreases, with increasing mappability.

Phase 1: Doesn’t help on current small corpus.
Phase 2: User determines mappability threshold for recall.

Level 3:
SL’: Enter a prefix to be attached...
SL: Enter a suffix to be attached...
TL: Geben Sie hier ein Suffix das...
TL’: Geben Sie hier ein Praefix das...

Level 0:
SL’: Use the Zoom Tool to bring ...
SL: Use the Offset Command to specify...
TL: Mit der Option Abstand legen Sie...fest
TL’: Mit der Option Zoom des Hilf mittels Zoom...
Comparative Advantages/Disadvantages:

Note that ReVerb has much smaller DB of examples than PanEBMT, due to more required preprocessing: 214 sentence pairs vs. 726,000.

But in turn it potentially produces a higher % of good translations.
Comparing EBMT and SMT:

EBMT:

• Can work with very little data (one sentence pair!)
• Trains and decodes more quickly
• Generalizes differently; will always reproduce training examples
• Has historically trailed SMT systems a bit in DARPA competitions
• Less principled (at least in theory)