The CMU Example-Based Machine Translation System: A Case Study

Ralf Brown
21 March 2007
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - and their effects on development
- Current Efforts
- Where Next?
New Sentence (Source)
Yesterday, 200 delegates met with President Bush.

Matches to Source Found

Yesterday, 200 delegates met behind closed doors…

Difficulties with President Bush…

Alignment (Sub-sentential)

Yesterday, 200 delegates met behind closed doors…

Difficulties with President Bush over…

Translated Sentence (Target)
Gestern trafen sich 200 Abgeordnete mit Praesident Bush.
Outline

• Review: What is EBMT
• Origins of the CMU system
• Projects using EBMT
  – and their effects on development
• Current Efforts
• Where Next?
CMU-EBMT Origins

- The earliest implementation (1992) was in Lisp as part of the Pangloss system; various approaches to matching were tried.
- A second implementation in C was begun to replace the index-lookup code from the Lisp version for greater speed.
  - This version already performed contiguous-phrase matching, but conflated all function words in matching.
- The current C++ implementation was begun in 1995 as a complete replacement for the Lisp/C hybrid.
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - Pangloss
  - DIPLOMAT
  - TONGUES
  - TIDES / GALE
- Current Efforts
- Where Next?

- Started as a knowledge-based translation system
- But sponsors kept changing the criteria for input texts
- So had to add rapid-development capabilities, hence EBMT and MEMT
- All the Pangloss translation engines produced phrasal translations, so a language modeler was added
EBMT (1995)

- Used an inverted-file index
  - for every word type, lists all occurrences in the training text
- Found maximal phrases by scanning down occurrence lists for adjacent words in input
- Simple heuristic aligner to find corresponding translation
- Matches processed from most-recently added until sufficient (3-5) alignable matches found
  - score for each candidate translation is the alignment score
Inverted-File Index: Attributes

• Advantages:
  ➔ Fast incremental update - just add the new occurrence records
  ➔ Easy to find multiple variants in parallel
  ➔ Easy to find matches with gaps (just scan down occurrence lists for non-adjacent words)

• Disadvantage:
  ➔ Lookup time and memory requirements scale linearly with corpus size
Pangloss-Lite (1995)

- The Pangloss system was a full suite called the Translator's Workstation, including post-editing facilities, visualization, etc.
- TWS implemented in Lisp, loaded Prolog-based Panglyzer KBMT system, C-based EBMT/LM
- very slow start-up times (15 minutes!), so implemented a simple wrapper around EBMT, LM, and a dictionary
  - resulted in a translation system that started up in a few seconds
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - Pangloss
  - DIPLOMAT
  - TONGUES
  - TIDES / GALE
- Current Efforts
- Where Next?
DIPLOMAT (1996-2000)

- Speech-to-Speech Translation
  - Croatian/English
  - Haitian Creole/English
  - Korean/English
- Initial domain was refugee interviews
- Had to run on portable equipment
DIPLOMAT: Issues

- Assume completely computer-illiterate interviewee ("farmer in his fields")
  - asymmetric interface: keep necessary complexity on the interviewer's side
- Can't talk to someone until you put a headset on them?
  - found a noise-canceling microphone that looks much like a telephone handset
  - implemented pre-recorded instructions ("We will use this machine to communicate. Talk into the microphone....")
DIPLOMAT: Hardware

- Needed to run two speech recognizers, two speech synthesizers, two translators, and two user interfaces
- Final configuration was three networked laptops [133-166 Mhz Pentium, 48 MB RAM each]
  1) English ASR, English synth, and interviewer UI
  2) foreign ASR, foreign synth, and interviewee UI
  3) both translators
DIPLOMAT: Croatian

- Problem: almost no parallel text available
- Solution: hire Croatian speakers to translate text
  - generated ~1 million words of text over three years
DIPLOMAT: Haitian Creole

- Problem: almost no parallel text available
- Solution: hire Haitian speakers to translate text
- Problem 2: Standard orthography? WHAT standard orthography? 
  - different translators used different spellings
- Solution: cross-checks and semi-automatic substitutions
DIPLOMAT: Korean

- Parallel text was available
- Translators were available
- Problem: not representable in 8-bit characters
  - had to implement Unicode support
- Problem2: very different word order than English (S-O-V)
  - experimental source-word reordering: bracket subject and object noun phrases to identify verb complex, then move verb between subject and object
  - partially worked, but project ended before reordering was integrated
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - Pangloss
  - DIPLOMAT
  - TONGUES
  - TIDES / GALE
- Current Efforts
- Where Next?
TONGUES (2000-2001)

- Follow-on to DIPLOMAT sponsored by the US Army Chaplain School
- Croatian-English Speech-to-Speech
- target device must fit in a cargo pocket
  - selected Toshiba Libretto subnotebook modified with a touch-screen
- unlike DIPLOMAT, got an actual field test
TONGUES Field Test (Zagreb)
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - Pangloss
  - DIPLOMAT
  - TONGUES
  - TIDES / GALE
- Current Efforts
- Where Next?
TIDES (2002-2005) / GALE (2005-)

• Focus shifts towards huge training corpora
  – 200+ million words
  – (originally had 100k small data track, quickly dropped)
• Goal is maximum translation quality with little or no regard for resource requirements (including translation time – minutes per sentence is OK)
Outline

• Review: What is EBMT
• Origins of the CMU system
• Projects using EBMT
  – and their effects on development
• Current Efforts
• Where Next?
EBMT (2005-2006)

- Can take a lattice of hypotheses as input
- New index: Burrows-Wheeler transformed version of a suffix array
- Many more matches are processed, and translation score is now a combination of alignment score, translation probability, and contextual weighting
- Training examples can be generalized, but that capability sees little use on large corpora
Lattice Processing

- Sometimes, the input is ambiguous, and we want to preserve that ambiguity
  - multiple word segmentations for Asian languages
  - confusion networks or word lattices from speech recognizers
  - multiple morphological analyses
- Solution: instead of taking a simple string as input, use a lattice and match all possible paths through the lattice against the training corpus
Burrows-Wheeler-Style Index

- **Advantages:**
  - Compact: compressed version can take less than 18 bits per occurrence in a corpus of billions of words (uncompressed is 32 bits per)
  - Fast: \(O(\log N)\) lookups
  - Lookup results represent all occurrences of an n-gram in constant space
  - Can incrementally reconstruct source text from index

- **Disadvantages:**
  - No incremental updates
  - Lookups with a gap degenerate to linear time
The Burrows-Wheeler Transform

• A block-sorting transformation originally devised for data compression
  ➔ Now used in many of the best-performing compression programs, such as bzip2
• Groups together all occurrences of an n-gram in the text
• BWT is normally applied to bytes
  ➔ For EBMT, we use 32-bit word identifiers
• End-of-Text marker is normally sorted before any member of the alphabet
  ➔ For EBMT, sort it after all other word IDs
Benefits of BW-Transformed Text

- The resulting C and V vectors lend themselves to effective compression because C (representing the vocabulary) is entirely monotonically increasing, as is each segment of V (successor pointers) bounded by pointers from C
- The transform is entirely reversible
- All occurrences of a given n-gram are contiguous; all occurrences of an n+1-gram are contained within the range of the occurrences of the n-gram
  - Lookups consist of successive binary searches within the range of occurrences for the previous result
  - All occurrences can be represented by the start and end of the range
A Problem for EBMT

- The BW-transformed text shows relative positions, not absolute positions
  - To get absolute positions without reconstructing the entire text, additional information must be stored
- The EBMT corpus is a set of textual units, i.e. records, whose boundaries need not be spanned
  - Therefore, we don't care what follows an end-of-record marker
- Thus, a successor pointer that points outside the corpus can encode a record number by *how much* outside the corpus it points
EBMT Record Numbers

- Reserve the highest ¼ of the 32-bit numbers as end-of-record markers; this leaves 3 billion values for both word ID and word location and 1 billion values for record numbers.
- The record number becomes the EoR marker's value less the value of the first EoR value.
- When generating the V array, leave EoR markers alone; we can guarantee that they will be pointing outside the corpus.
Retrieving a Matched Instance

- Once the phrasal matches have been found, iterate on each occurrence and
  - Follow successor pointers in the index until EoR
  - Extract the record number from the EoR marker
  - Retrieve the corresponding example from the corpus
  - Perform word-level alignment, etc. to generate a translation hypothesis for the phrase
Compressing the Index

- Since the index consists of runs of increasing values, mostly with small differences, we can delta-encode the V array
  - Use one byte per entry and an overflow table
  - Values 1-191 represent the difference from the previous entry's value
  - Values 192-255 point into the overflow table, which contains a 32-bit absolute value
  - Retrieval overhead is limited by forcing at least one absolute value per 64-entry bucket
Compression Performance

- Uncompressed = 32 bits per entry
- Compressed = minimum 9, maximum 40 bits per entry
  - For larger corpora, typically averages 17-19 bits per entry
  - Translation times 10-20% higher when using compression, but index is 40-50% smaller
Motivation for Context

- Most EBMT systems treat the training corpus as a bag of examples
- Training corpora are typically sets of documents
- So are the inputs to be translated
- Within a single document, there tends to be a uniformity of usage
- Adjacent sentences will have similar referents for pronouns, etc.
- Therefore, use of context should improve translation quality
Approach

• Looking at two kinds of context:
  – intra-sentential: re-use of a single training example for various fragments of an input sentence
    • multiple fragments from a single training instance will give us increased confidence in the translation
  – inter-sentential: use of portions of the same or adjacent training instances that were used for the previous input sentence
    • takes advantage of document-level coherence
Intra-Sentential Example

Training Instances

John visited the bank yesterday morning to get some cash.

Bill strolls along the bank every time he comes to the river.

Test Input:

{John} visited {the bank} yesterday morning {to get some cash.}

John went to the bank to get some cash.

Bill strolls along {the bank} every time he comes to the river.

bonus for two other matches  default weight
Using Context

Training Documents

... John and Mary were walking in the park. "Let's go to the bank."
... John needed some cash. "I'll go to the bank in the morning."
... The flight instructor told John, "don't bank the plane too sharply."

Input being translated:

"I need some cash. Will you go to the bank?"

Without context:

"Let's {go to the bank}.
"I'll {go to the bank} in the morning."

equal weight

With context:

(no contextual match)
"Let's {go to the bank}.

(used "some cash" previously)
"I'll {go to the bank} in the morning."

default weight

increased weight
Intra-Sentential Context

- Matching retrieves multiple examples, and gives a quality score based on the weighted average of the retrieved examples
- Give more weight to training instances that have already been used in translating the current sentence
  - biases scores toward such instances, making them more likely to be selected by the decoder
- Used a greedy approach for ease and efficiency of implementation
Intra-Sentential Context (2)

- Maintain an array of counts, one per training instance
- After each match is processed, increment the count for the corresponding training instance
- Adjust weight of the match by current count
- Matches are processed in order by starting offset in the input sentence and reverse order by length
  - matches automatically get a bonus if they are a substring of some other match
  - disjoint matches only receive boost for matches located earlier in the input sentence
Inter-Sentential Context

- Instead of discarding the array of match counts on completing a sentence, make a copy.
- Look at match counts not just for active training instance, but also those immediately before and after.
- Score bonus is a weighted sum of the counts within five sentence pairs.
## Computing Context Bonuses

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Base Weights</th>
<th>Local</th>
<th>Inter.</th>
<th>Merge Eq.</th>
<th>Final Wt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wt=2</td>
<td>2*1.05</td>
<td>0</td>
<td>1</td>
<td>4.20</td>
<td>3.55 (9.0%)</td>
</tr>
<tr>
<td>Wt=1</td>
<td>1*1.11</td>
<td>1</td>
<td>0</td>
<td>2.22</td>
<td></td>
</tr>
<tr>
<td>Wt=3</td>
<td>1*1.27</td>
<td>1</td>
<td>2</td>
<td>5.08</td>
<td>30.43 (77.5%)</td>
</tr>
<tr>
<td></td>
<td>1*1.33</td>
<td>0</td>
<td>0</td>
<td>1.33</td>
<td></td>
</tr>
<tr>
<td>Wt=1</td>
<td>3*1.41</td>
<td>1</td>
<td>3</td>
<td>21.15</td>
<td></td>
</tr>
<tr>
<td>Wt=1</td>
<td>1*1.72</td>
<td>0</td>
<td>0</td>
<td>1.72</td>
<td>5.28 (13.4%)</td>
</tr>
<tr>
<td>Wt=1</td>
<td>1*1.78</td>
<td>1</td>
<td>0</td>
<td>3.56</td>
<td></td>
</tr>
</tbody>
</table>
Generalization in EBMT

- Basic matching in our EBMT system is between strings
  - he went to several seminars last month
  - she went to several seminars in June
- But those strings need not be surface forms
  - morphological base forms (roots/stems)
    - he [go] to several [seminar] last month
  - equivalence classes
    - he went to several <event-p> last <timespan>
  - templates
    - <PERSON> went to <NP> <TIMESPEC>
Clustering for Generalization

- Too much work to manually create equivalence classes
- Automated clustering methods to the rescue:
  - members of the equivalence class can be used interchangeably, thus appear in similar contexts
  - create a term vector from the words surrounding every instance of a word of interest, then cluster the vectors
Spectral Clustering (1)

- A clustering method that uses a nonlinear dimensionality reduction technique (eigenvalues) on distance matrices
- Can correctly separate non-convex clusters -- even when one completely surrounds another
- Methods exist to automatically determine the correct number of clusters
Spectral Clustering (2)

- Use cosine similarity as the distance metric
- Perform local distance scaling
  \[ d(a,b) > d(b,a) \] if \( a \) has many near neighbors and \( b \) has few nearby neighbors
- Extract first \( K \) eigenvectors, stack them to form a matrix, normalize the matrix \((Y)\), and then perform \( k \)-means clustering using each row of \( Y \) as a point in \( K \) dimensions
Spectral Clustering (3)

• First results for English/French:
  – +1.37% @ 10k, +29.08% @ 20k, +3.88% @ 30k

• More recently:
  – 100 clusters @ 20k  0.1442 → 0.1952 (+35.37%)
  – but attempts to refine the clustering didn't help
    • seeded: 0.1935
    • filtered: as low as 0.1886
Seeded Clustering

• What if we can add a bit of human smarts?
  – have the human provide a set of equivalence classes with a few examples for each class
  – automatically expand upon those seed classes
• Unfortunately, seeding did not help translation when using Spectral Clustering
Generalizing with Morphology

- Matching on base forms can allow more matches
  - but need ability to filter matches and re-inflect on target side
- Splitting off affixes can improve alignments
  - affixes often correspond to separate words or particles in the other, less-inflected language
  - preliminary result for Mapudungun-Spanish: 0.15304 $\rightarrow$ 0.16142 (+5.48%)
- Separating compound words may allow mix-and-match use or generalization of their parts
  - Herzkrankheit $\rightarrow$ Herz krankheit $\rightarrow$ \{organ\}krankheit
Generalizing with Morphology (2)

- Arabic has rich morphology
  - affixes corresponding to English articles, prepositions, etc.
  - inflectional morphology
  - early performance numbers:
    BLEU 0.20619 → 0.23099 (+11.46%)
  - more recently, with more training data and improved system building: 0.45 → 0.47
Outline

- Review: What is EBMT
- Origins of the CMU system
- Projects using EBMT
  - and their effects on development
- Current Efforts
- Where Next?
Future Work

• Respect document boundaries
• Make use of document-level similarity
  – boost weight of training instances in the training documents which are most similar to the input document
• Context-aware decoder
  – pass origin information to the decoder and have it bias its search toward selecting sets of arcs from single training documents
• Better/faster clustering
• Learn source-text reordering
• Use multiply-segmented input