Introduction to MT

CSE 415
Fei Xia
Linguistics Dept
02/24/06
Outline

• MT in a nutshell
• Major challenges
• Major approaches
• Introduction to word-based statistical MT
MT in a nutshell
What is the ultimate goal of translation?

• Translation: source language $\rightarrow$ target language (S$\rightarrow$T)

• Ultimate goal: find a “good” translation for text in S:
  – Accuracy: faithful to S, including meaning, connotation, style, …
  – Fluency: the translation is as natural as an utterance in T.
Translation is hard, even for human

• Novels

• Word play, jokes, puns, hidden message.

• Concept gaps: double jeopardy, go Greek, fen sui, ....

• Cultural factor:
  – A: Your daughter is very talented.
  – B: She is not that good ➔ Thank you.

• Other constraints: lyrics, dubbing, poem.
“Crazy English” by Richard Lederer

• “Compound” words: Let’s face it: English is a crazy language. There is no egg in eggplant or ham in hamburger, neither apple nor pine in pineapple.

• Verb+particle: When a house burns up, it burns down. You fill in a form by filling it out and an alarm clock goes off by going on.

• Predicate+argument: When the stars are out, they are visible, but when the lights are out, they are invisible. And why, when I wind up my watch, I start it, but when I wind up this essay, I end it?
A brief history of MT
(Based on work by John Hutchins)

• The pioneers (1947-1954): the first public MT demo was given in 1954 (by IBM and Georgetown University).

• The decade of optimism (1954-1966): ALPAC (Automatic Language Processing Advisory Committee) report in 1966: "there is no immediate or predictable prospect of useful machine translation."
A brief history of MT (cont)

- The aftermath of the ALPAC report (1966-1980): a virtual end to MT research
- The 1980s: Interlingua, example-based MT
- The 1990s: Statistical MT
- The 2000s: Hybrid MT
Where are we now?

- Huge potential/need due to the internet, globalization and international politics.

- Quick development time due to SMT, the availability of parallel data and computers.

- Translation is reasonable for language pairs with a large amount of resource.

- Start to include more “minor” languages.
What is MT good for?

- Rough translation: web data
- Computer-aided human translation
- Translation for limited domain
- Cross-lingual information retrieval

- Machine is better than human in:
  - Speed: much faster than humans
  - Memory: can easily memorize millions of word/phrase translations.
  - Manpower: machines are much cheaper than humans
  - Fast learner: it takes minutes or hours to build a new system. Erasable memory 😊
Evaluation of MT systems

• Unlike many NLP tasks (e.g., tagging, chunking, parsing, IE, pronoun resolution), there is no single gold standard for MT.

• Human evaluation: accuracy, fluency, …
  – Problem: expensive, slow, subjective, non-reusable.

• Automatic measures:
  – Edit distance
  – Word error rate (WER)
  – BLEU
  – …
Major challenges in MT
Major challenges

• Getting the right words:
  – Choosing the correct root form
  – Getting the correct inflected form
  – Inserting “spontaneous” words

• Putting the words in the correct order:
  – Word order: SVO vs. SOV, …
  – Translation divergence
Lexical choice

• Homonymy/Polysemy: bank, run

• Concept gap: no corresponding concepts in another language: go Greek, go Dutch, fen sui, lame duck, …

• Coding (Concept ➔ lexeme mapping) differences:
  – More distinction in one language: e.g., “cousin”
  – Different division of conceptual space:
Choosing the appropriate inflection

• Inflection: gender, number, case, tense, …

• Ex:
  – Number: Ch-Eng: all the concrete nouns:
    \texttt{ch\_book} \rightarrow \text{book, books}
  – Gender: Eng-Fr: all the adjectives
  – Case: Eng-Korean: all the arguments
  – Tense: Ch-Eng: all the verbs:
    \texttt{ch\_buy} \rightarrow \text{buy, bought, will buy}
Inserting spontaneous words

- **Determiners: Ch-Eng:**
  - ch_book ➔ a book, the book, the books, books

- **Prepositions: Ch-Eng**
  - ch_November ➔ … in November

- **Conjunction: Eng-Ch:**
  - Although S1, S2 ➔ ch_although S1, ch_but S2

- **Dropped argument: Ch-Eng:**
  - ch_buy le ma ? ➔ Has Subj bought Obj ?
Major challenges

• Getting the right words:
  – Choosing the correct root form
  – Getting the correct inflected form
  – Inserting “spontaneous” words

• Putting the words in the correct order:
  – Word order: SVO vs. SOV, …
  – Translation divergence
Word order

- SVO, SOV, VSO, ...
- VP + PP → PP VP
- VP + AdvP → AdvP + VP
- Adj + N → N + Adj
- NP + PP → PP NP
- NP + S → S NP
- P + NP → NP + P
Translation divergences (based on Bonnie Dorr’s work)

• Thematic divergence: I like Mary ➔
  S: Marta me gusta a mi (‘Mary pleases me’)

• Promotional divergence: John usually goes home ➔
  S: Juan suele ira casa (‘John tends to go home’)

• Demotional divergence: I like eating ➔ G: Ich esse gern
  (“I eat likingly”)  

• Structural divergence: John entered the house ➔
  S: Juan entro en la casa (‘John entered in the house’)

Translation divergences (cont)

• Conflational divergence: I stabbed John ➔
  S: Yo le di punaladas a Juan (‘I gave knife-wounds to John’)

• Categorial divergence: I am hungry ➔
  G: Ich habe Hunger (‘I have hunger’)

• Lexical divergence: John broke into the room ➔
  S: Juan forzo la entrada al cuarto (‘John forced the entry to the room’)

Ambiguity

• Ambiguity that needs to be “resolved”:
  – Ex1: wh-movement
    • Eng: *Why* do you think that he came yesterday?
    • Ch: you *why* think he yesterday come ASP?
    • Ch: you think he yesterday *why* come?
  – Ex2: PP-attachment: “he saw a man with a telescope”
  – Ex3: lexical choice: “a German teacher”
Ambiguity (cont)

• Ambiguity that can be “carried over”.
  – Ex1: “Mary and John bought a house last year.”

• Important factors:
  – Language pair
  – Type of ambiguity
Major approaches
What kinds of resources are available to MT?

- Translation lexicon:
  - Bilingual dictionary

- Templates, transfer rules:
  - Grammar books

- Parallel data, comparable data

- Thesaurus, WordNet, FrameNet, …

- NLP tools: tokenizer, morph analyzer, parser, …

➤ There are more resources for major languages than “minor” languages.
Major approaches

- Transfer-based
- Interlingua
- Example-based (EBMT)
- Statistical MT (SMT)
- Hybrid approach
<table>
<thead>
<tr>
<th>Word</th>
<th>Meaning (interlingua)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer-based Phrase-based SMT, EBMT</td>
<td>Word-based SMT, EBMT</td>
</tr>
</tbody>
</table>

The MT triangle

- **Analysis**: Transfer-based
- **Synthesis**: Phrase-based SMT, EBMT

```
[Analysis] -> [Meaning (interlingua)] -> [Synthesis]
```

- Word -> Meaning (interlingua) -> Word
Transfer-based MT

- Analysis, transfer, generation:
  1. Parse the source sentence
  2. Transform the parse tree with transfer rules
  3. Translate source words
  4. Get the target sentence from the tree

- Resources required:
  - Source parser
  - A translation lexicon
  - A set of transfer rules

- An example: Mary bought a book yesterday.
Transfer-based MT (cont)

- Parsing: linguistically motivated grammar or formal grammar?
- Transfer:
  - context-free rules? A path on a dependency tree?
  - Apply at most one rule at each level?
  - How are rules created?
- Translating words: word-to-word translation?
- Generation: using LM or other additional knowledge?
- How to create the needed resources automatically?
Interlingua

• For \( n \) languages, we need \( n(n-1) \) MT systems.
• Interlingua uses a language-independent representation.
• Conceptually, Interlingua is elegant: we only need \( n \) analyzers, and \( n \) generators.

• Resource needed:
  – A language-independent representation
  – Sophisticated analyzers
  – Sophisticated generators
Interlingua (cont)

• Questions:
  – Does language-independent meaning representation really exist? If so, what does it look like?
  – It requires deep analysis: how to get such an analyzer: e.g., semantic analysis
  – It requires non-trivial generation: How is that done?
  – It forces disambiguation at various levels: lexical, syntactic, semantic, discourse levels.
  – It cannot take advantage of similarities between a particular language pair.
Example-based MT

• Basic idea: translate a sentence by using the closest match in parallel data.
• First proposed by Nagao (1981).
• Ex:
  – Training data:
    • w1 w2 w3 w4 $\rightarrow$ v2 v3 v1 v4
    • W3’ $\rightarrow$ v3’
  – Test sent:
    • w1 w2 w3’ $\rightarrow$ v2 v3’ v1
EMBT (cont)

• Types of EBMT:
  – Lexical (shallow)
  – Morphological / POS analysis
  – Parse-tree based (deep)

• Types of data required by EBMT systems:
  – Parallel text
  – Bilingual dictionary
  – Thesaurus for computing semantic similarity
  – Syntactic parser, dependency parser, etc.
Statistical MT

• Sentence pairs: word mapping is one-to-one.
  – (1) S: a b c
    T: l m n
  – (2) S: c b
    T: n m

⇒ (a, l) and
    (b, m), (c, n), or
    (b, n), (c, m)
SMT (cont)

• Basic idea: learn all the parameters from parallel data.

• Major types:
  – Word-based
  – Phrase-based

• Strengths:
  – Easy to build, and it requires no human knowledge
  – Good performance when a large amount of training data is available.

• Weaknesses:
  – How to express linguistic generalization?
## Comparison of resource requirement

<table>
<thead>
<tr>
<th></th>
<th>Transfer-based</th>
<th>Interlingua</th>
<th>EBMT</th>
<th>SMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>dictionary</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Transfer rules</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parser</td>
<td>+</td>
<td>+</td>
<td>+ (?)</td>
<td></td>
</tr>
<tr>
<td>semantic analyzer</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>parallel data</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>others</td>
<td></td>
<td>Universal representation</td>
<td>thesaurus</td>
<td></td>
</tr>
</tbody>
</table>
Hybrid MT

• Basic idea: combine strengths of different approaches:
  – Transfer-based: generalization at syntactic level
  – Interlingua: conceptually elegant
  – EBMT: memorizing translation of n-grams; generalization at various level.
  – SMT: fully automatic; using LM; optimizing some objective functions.
Types of hybrid HT

• Borrowing concepts/methods:
  – EBMT from SMT: automatically learned translation lexicon
  – Transfer-based from SMT: automatically learned translation lexicon, transfer rules; using LM

• Using multiple MT systems in a pipeline:
  – Using transfer-based MT as a preprocessor of SMT

• Using multiple MT systems in parallel, then adding a re-ranker.
Summary

• Major challenges in MT
  – Choose the right words (root form, inflection, spontaneous words)
  – Put them in right positions (word order, unique constructions, divergences)
Summary (cont)

• Major approaches
  – Transfer-based MT
  – Interlingua
  – Example-based MT
  – Statistical MT
  – Hybrid MT
Additional slides
Introduction to
word-based SMT
Word-based SMT

• Classic paper: (Brown et al., 1993)
• Models 1-5
• Source-channel model

\[ T^* = \arg \max \limits_T P(T \mid S) \]
\[ = \arg \max \limits_T \frac{P(S \mid T)P(T)}{P(S)} \]
\[ = \arg \max \limits_T P(S \mid T)P(T) \]

\[ E^* = \arg \max \limits_E P(F \mid E)P(E) \]
Word alignment

- Ex:
  - F: $f_1$ $f_2$ $f_3$ $f_4$ $f_5$
  - E: $e_1$ $e_2$ $e_3$ $e_4$
Modeling $p(F \mid E)$ with alignment $a$

$$P(F \mid E) = \sum_{a} P(a, F \mid E)$$

$$= \sum_{a} P(a \mid E) * P(F \mid a, E)$$
IBM Model 1
Generative process

• To generate F from E:
  – Pick a length $m$ for F, with prob $P(m \mid l)$
  – Choose an alignment $a$, with prob $P(a \mid E, m)$
  – Generate Fr sent given the Eng sent and the alignment, with prob $P(F \mid E, a, m)$. 
Final formula for Model 1

\[
P(F \mid E) = \frac{P(m \mid l)}{(l + 1)^m} \prod_{j=1}^{m} \sum_{i=1}^{l} P(f_j \mid e_i)
\]

\(m\): Fr sentence length
\(l\): Eng sentence length
\(f_j\): the \(j^{th}\) Fr word
\(e_i\): the \(i^{th}\) Eng word

Two types of parameters:
• Length prob: \(P(m \mid l)\)
• Translation prob: \(P(f_j \mid e_i)\), or \(t(f_j \mid e_i)\),
Estimating $t(f|e)$: a naïve approach

• A naïve approach:
  – Count the times that $f$ appears in $F$ and $e$ appears in $E$.
  – Count the times that $e$ appears in $E$.
  – Divide the 1st number by the 2nd number.

• Problem:
  – It cannot distinguish true translations from pure coincidence.
  – Ex: $t(el \mid white) \approx t(blanco \mid white)$

• Solution: count the times that $f$ aligns to $e$. 
Estimating $t(f|e)$ in Model 1

- When each sent pair has a unique word alignment
- When each sent pair has several word alignments with prob
- When there are no word alignments
When there is a single word alignment

• We can simply count.

• Training data:

   Eng: \ b \ c \ b
   \   \   \   \\
Fr: \ x \ y \ y

• Prob:
  – $ct(x,b)=0$, $ct(y,b)=2$, $ct(x,c)=1$, $ct(y,c)=0$
  – $t(x|b)=0$, $t(y|b)=1.0$, $t(x|c)=1.0$, $t(y|c)=0$
When there are several word alignments

- If a sent pair has several word alignments, use fractional counts.

- Training data:

  \[ P(a|E,F)=0.3 \quad 0.2 \quad 0.4 \quad 0.1 \quad 1.0 \]
  \[ b \quad c \quad b \quad c \quad b \quad c \quad b \quad c \quad b \]
  \[ \big| \quad \big| \quad \big/ \quad \big/ \quad \big\times \quad \big| \quad \big| \]
  \[ x \quad y \quad x \quad y \quad x \quad y \quad x \quad y \quad y \]

- Prob:
  - \( Ct(x,b)=0.7, \ Ct(y,b)=1.5, \ Ct(x,c)=0.3, \ Ct(y,c)=0.5 \)
  - \( P(x|b)=7/22, \ P(y|b)=15/22, \ P(x|c)=3/8, \ P(y|c)=5/8 \)
Fractional counts

Let $C_t(f, e)$ be the fractional count of $(f, e)$ pair in the training data, given alignment prob $P$.

$$C_t(f, e) = \sum_{E,F} \sum_a \left( P(a \mid E,F) \* \sum_{j=1}^{\mid F \mid} \delta(f, f_j) \delta(e, e_{a_j}) \right)$$

Alignment prob

Actual count of times $e$ and $f$ are linked in $(E,F)$ by alignment $a$

$$t(f \mid e) = \frac{C_t(f, e)}{\sum_{x \in V_F} C_t(x, e)}$$
When there are no word alignments

- We could list all the alignments, and estimate $P(a \mid E, F)$.

\[
P(a \mid E, F) = \frac{P(a, F \mid E)}{\sum_a P(a, F \mid E)} = \frac{\prod_{j=1}^{m} t(f_j \mid e_{a_j})}{\sum_a \prod_{j=1}^{m} t(f_j \mid e_{a_j})}
\]
Formulae so far

$$P(a \mid E, F) = \frac{P(a, F \mid E)}{\sum_a P(a, F \mid E)} = \frac{\prod_{j=1}^m t(f_j \mid e_{a_j})}{\sum_a \prod_{j=1}^m t(f_j \mid e_{a_j})}$$

$$Ct(f, e) = \sum_{E,F} \sum_a (P(a \mid E, F) * \sum_{j=1}^{\mid F \mid} \delta(f, f_j) \delta(e, e_{a_j}))$$

$$t(f \mid e) = \frac{Ct(f, e)}{\sum_{x \in V_F} Ct(x, e)} \quad \Leftarrow \text{New estimate for } t(f \mid e)$$
The EM algorithm

1. Start with an initial estimate of $t(f | e)$: e.g., uniform distribution
2. Calculate $P(a | F, E)$
3. Calculate $C_t (f, e)$, Normalize to get $t(f|e)$
4. Repeat Steps 2-3 until the “improvement” is too small.
So far, we estimate $t(f \mid e)$ by enumerating all possible alignments

- This process is very expensive, as the number of all possible alignments is $(l+1)^m$.

$$Ct(f,e) = \sum_{E,F} \sum_{a} \left( P'(a \mid E,F) \times \sum_{j=1}^{\left| F \right|} \delta(f, f_j) \delta(e, e_{a_j}) \right)$$

**Prev iteration’s Estimate of Alignment prob**

**Actual count of times e and f are linked in (E,F) by alignment a**
No need to enumerate all word alignments

- Luckily, for Model 1, there is a way to calculate $Ct(f, e)$ efficiently.

$$ Ct(f, e) = \sum_{E,F} \frac{t'(f | e) * (\sum_{i=0}^{\mid E \mid} \delta(e, e_i)) * (\sum_{j=0}^{\mid F \mid} \delta(f, f_j))}{\sum_{i'=0}^{\mid E \mid} t'(f | e_{i'})} $$

$$ t(f | e) = \frac{Ct(f, e)}{\sum_{x \in V_F} Ct(x, e)} $$
The algorithm

1. Start with an initial estimate of $t(f \mid e)$: e.g., uniform distribution
2. Calculate $P(a \mid F, E)$
3. Calculate $C_t(f, e)$, Normalize to get $t(f \mid e)$
4. Repeat Steps 2-3 until the “improvement” is too small.
An example

• Training data:
  – Sent 1: Eng: “b c”, Fr: “x y”
  – Sent 2: Eng: “b”, Fr: “y”

• Let’s assume that each Eng word generates exactly one Fr word

• Initial values for $t(f|e)$:
  
  \[
  t(x|b)=t(y|b)=1/2, \quad t(x|c)=t(y|c)=1/2
  \]
After a few iterations

|       | $t(x|b)$ | $t(y|b)$ | $t(x|c)$ | $t(y|c)$ | a1  | a2  |
|-------|---------|---------|---------|---------|-----|-----|
| init  | $1/2$  | $1/2$  | $1/2$  | $1/2$  | $-  $| $-  $|
| $1^{st}$ iter | $1/4$  | $3/4$  | $1/2$  | $1/2$  | $1/2$| $1/2$|
| $2^{nd}$ iter | $1/8$  | $7/8$  | $3/4$  | $1/4$  | $1/4$| $3/4$|
Summary for word-based SMT

- Main concepts:
  - Source channel model
  - Word alignment

- Training: EM algorithm

- Advantages:
  - It requires only parallel data
  - Its extension (phrase-based SMT) produces the best results.