Why Machine Translation?

- translation is expensive
- growing on-line demand for translation (on-the-fly)
- globalization, growing export, new markets
- political issues (EU, support of minority languages ...)
- tourism, movies, news
- ... MT makes us laugh:
  - Input: Vem vann allsvenskan i fjol?
  - Google 2010: Who stole headlines last year?

MT combines various aspects of computational linguistics
(in a naturally occurring task)

Why is translation hard?

Languages are different on many levels:

- lexicon, syntax, semantics
  - source language ambiguity
  - cross-lingual divergences
  - target language variation
- pragmatics, style, culture, background

Computers have big problems with ambiguity.

Lexical ambiguities across languages

(from Jurafsky & Martin, 2008: Speech and Language Processing)
Types of Systematic Divergences (Habash, 2002)

1. categorical divergences:
   - tener celos (N) (lit. “to have jealousy”) ↔ to be jealous (A)
2. conflation: ir flotando (lit. “to go floating”) ↔ to float
3. structural divergence: entrar en N (lit. “to enter in N”) ↔ to enter N
4. head swapping:
   - entrar corriendo (lit. “to enter running”) ↔ to run in
5. thematic divergence:
   - me gustan uvas (lit. “to-me they-please grapes”) ↔ I like grapes

Variation in Target language

Redundancy of natural languages:

- translate: “Vid avslutad kurs ...”
  - On completion of the course ...
  - After completion of the course ...
  - Having completed the course ...
  - After finishing the course ...
  - Once the course has been completed ...
  - ...

Which one is best? How do we decide that?

What can we do? ... browsing quality

- Balance between quality and input restrictions

<table>
<thead>
<tr>
<th>Automatic translation depending on task</th>
<th>fully automatic</th>
<th>computer-aided (human) translation (CAT)</th>
<th>fully automatic (FAHOMT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>general purpose rough translation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>browsing quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>editing quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>publishing quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>limited domains</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on-line service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>localization, ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product manuals, limited domains</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Different views on MT

“Linguist’s view”: Three independent steps

1. source language analysis (understanding)
2. source to target language transfer (or interlingua)
3. target language generation (grammatical & fluent)

“Engineer’s view”: MT as decoding

“I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information coded in the text” (Warren Weavers (1947/49), Rockefeller Foundation)

“Every time I fire a linguist the performance goes up” (Fred Jelinek)

A brief history of MT (based on Hutchins)

1. Precursors and pioneers, 1933–1945
   - interest in cryptography, information theory, coding
   - Georgetown experiment (rus-eng) → big success
   - classical MT approaches (direct, transfer, interlingua)
   - ALPAC report 1966: No need for MT! It’s useless!
   - The spirit is willing but the flesh is weak
   - is translated into the Russian to the equivalent of
   - The vodka is good, but the steak is lousy.
   - Operational and commercial systems, 1976–1989
     - Systran, METAL
     - computer-aided translation (especially in Japan)
     - Revival of MT research
   - Boom of Data-driven MT, 1990 – now
     - SMT, EBMT, hybrid models
     - automatic evaluation (BLEU, ...)
     - “syntax-based” SMT (synchronous grammars ...)
     - Google Translate (now: support for 58 languages)

→ MT is “in” again!
Transfer-based MT

- many lexical transfer rules, fewer structural rules
- often feature-based representations
- often simple preference mechanisms: more specific first
  - on → på
  - come.vb → kom.vb
  - come on → kom igen

  - sit.vb on NP → sitta.vb på NP
  - sit.vb on the couch → sitta.vb i soffan
- often simple (morphological) generation

What are the problems?

- expensive (difficult) grammar engineering
- exponential ambiguity vs. early commitment
- variation & preference
- coverage & robustness

→ Good quality can be achieved but low coverage!
→ Increase coverage using data & (lexical) rule induction
→ Need example data!
Motivation
History of MT
Transfer-based MT
Example-based MT
Statistical MT

Time for Questions & Short Break

To sum up part I ....

- huge demand for machine translation (private & professional)
- challenging task for computational linguistics with a loooooong history
- classical rule-based approaches (direct, transfer, interlingua)

Part II

... even transfer-based MT needs example data

- example-based MT
- statistical MT
- statistical alignment models

Data-driven Machine Translation

Can we learn MT from example translations?

**Idea 1:** translation by analogy (EBMT)

- collect a large database of examples with translations
- when translating new sentences:
  - retrieve similar examples including partial matches
  - preference for longer matches
  - take partial matches and merge them together (recombination)
- advanced: generalize examples to templates

Need many examples! → Large parallel corpora!

Example-Based Machine Translation

The classical example (Sato & Nagao, 1990)

- **translate:** “He buys a book on international politics.”
- **examples:**
  1. He buys a notebook.
     Kare wa nito o kau.
   HE topic NOTEBOOK obj BUY.
  2. I read a book on international politics.
     Watashi wa kokusai seiji nitsuite kakareta hon o yomu.
     I topic INTERNATIONAL POLITICS ABOUT CONCERNED BOOK obj READ.
- **output:** “Kare wa kokusai seiji nitsuite kakareta hon o kau.”
Example-Based MT

Issues to be addressed:

▸ aquiriting & storing large example databases
▸ matching suitable example fragments
▸ alignment of fragments to target language
▸ recombination of translated fragments
▸ ranking of alternative solutions

Can we do this in a formal computational model?

Statistical Machine Translation

Can we learn MT models from example translations?

Noisy channel for MT: “What could have been the sentence that has generated the observed source language sentence?”

Probabilistic view on MT

\[ \hat{E} = \arg\max_E P(E|F) \]
\[ = \arg\max_E \frac{P(F|E)P(E)}{P(F)} \]
\[ = \arg\max_E P(F|E)P(E) \]

Translation model: \( P(F|E) \), estimated from (big) parallel corpora
Language model: \( P(E) \), estimated from (huge) monolingual target language corpora, takes care of fluency
Decoder: global search for \( \arg\max_E P(F|E)P(E) \) for a given sentence \( F \)
SMT training

Translation model \( P(F|E) \)
- decompose into word/phrase alignment models (Why?)
- estimate alignment model parameters

Language model \( P(E) \)
- stochastic N-gram models
- maximum-likelihood estimations, smoothing

Generative word alignment models

Words in observed source language \( (F) \) have been generated by words in target language \( (E) \).

\[
\begin{array}{cccc}
1 & 2 & 0 & 3 \\
huset & är & NULL & litet \\
\end{array}
\]

\[
\begin{array}{cccc}
\uparrow & \uparrow & \uparrow & \\
the & house & is & just small \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

\[ A : \{1 \rightarrow 1, 2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 0, 5 \rightarrow 3\} \]

Alignment-based translation model: \( P(F|E) = \sum_A P(F, A|E) \)

Note: Word alignment is often not monotonic!
(Try to align “klein ist das Haus” to “the house is small”.)

Learning alignment models

- need word alignment to estimate model parameters
- need model parameters to do alignment

→ learning with incomplete data: EM

1. initialize model (random, uniform, prior)
2. compute expected values for missing data (alignment) (apply current model to the data) → E-step
3. estimate new model using the completed data → M-step
4. goto 2 until model does not change (significantly)

→ iterative hill-climbing to find (local) maximum
→ initialization is important!
→ run sequence: IBM1 → IBM2 → HMM → IBM3 → IBM4 → IBM5
Learning alignment models (IBM 1)

\[ E = \arg\max_E P(F|E)P(E) \]

- now we have the translation model \( P(F|E) \)
- we still need the language model \( P(E) \)

→ Use standard N-gram language models

Higher IBM Models

Statistical Machine Translation: Language Modeling

What is the likelihood \( P(E) \) to observe sentence \( E \)?

\[ P_{LM}(\text{the house is small}) > P_{LM}(\text{small the is house}) \]

Estimate \( N \) – gram probabilities from corpora (MLE):

\begin{align*}
  P(E) &= P(e_1, e_2, e_3, \ldots, e_j) \\
  P(E) &= P(e_1) \cdot P(e_2|e_1) \cdot P(e_3|e_1, e_2) \cdot \ldots \cdot P(e_j|e_1, \ldots, e_{j-1}) \\
  P(E) &\approx p(e_1) \cdot P(e_2|e_1) \cdot \ldots \cdot P(e_j|e_{j-n}, \ldots, e_{j-1}) \\
  P(e_j|e_{j-n}, \ldots, e_{j-1}) &= \frac{\text{count}(e_{j-n}, \ldots, e_{j-1})}{\sum_k \text{count}(e_{j-n}, \ldots, e_{j-1}, e_k)}
\end{align*}
Statistical Machine Translation: Language Modeling

Problem with MLE: zero counts!

▶ some N-grams are never observed (→ \( \text{count}(e_1, e_2) = 0 \))
(Why is that a problem?)

→ Smoothing! (reserve probability mass for unseen events)

▶ add-one smoothing
▶ Good-Turing smoothing
▶ Kneser-Ney smoothing
▶ model interpolation

Summary of Word-Based SMT

Word-Based SMT and the Noisy Channel Model:

▶ statistical word alignment → \( P(F|E) \)
▶ language modeling → \( P(E) \)
▶ global decoding \( \text{argmax}_E P(F|E)P(E) \)

Word-by-word translation is too weak!

▶ contextual dependencies, local reordering
▶ non-compositional constructions
▶ n:m relations

→ look at larger chunks! → Phrase-Based SMT!

Time for Questions & Short Break

To sum up part II ....

▶ data-driven MT: use parallel corpora
▶ example-based MT: translate by analogy
▶ statistical MT: learn to “decode”
▶ train translation model from aligned parallel corpora
▶ train target language model from monolingual corpora
▶ translation = (probabilistic) search problem

Part III

Word-based SMT has its limits!

▶ phrase-based SMT
▶ decoding
▶ tree-based models
Phrase-based SMT

\[ E = \arg \max_E \left( \phi(f_i | e_i) \ast d(start_i, end_{i-1}) \ast P(E) \ast \omega^{\text{length}(E)} \right) \]

- **phrase translation probabilities** from phrase alignments: 
  \[ \phi(f_i | e_i) = \frac{\text{count}(f_i, e_i)}{\sum \text{count}(f_i, e_j)} \]

- **distance-based distortion**: 
  \[ d(start_i, end_{i-1}) = \alpha^{\text{length}(start_i - end_{i-1}) - 1} \]

- **OR lexicalized distortion** (learned from aligned parallel corpus)

- **word penalty** \( \omega \) to avoid preference for short candidates

Word Alignment Symmetrization

start with intersection, add adjacent links (from union) ...

Phrase extraction

Get ALL phrase pairs that are consistent with word alignments

Example phrase pairs:

- (Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green).
- (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
  (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
  (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch).
- (Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch),
  (no daba una bofetada a la bruja verde, did not slap the green witch),
  (Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)
Phrase extraction

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch)

Log-linear SMT Models

Instead of noisy-channel model \( \hat{E} = \arg\max_E P(F|E)P(E) \):

- model posterior directly: \( \hat{E} = \arg\max_E P(E|F) \)
- many feature functions \( h_m(E, F) \) may influence \( P(E|F) \)
  - phrase translation model \( E \rightarrow F \)
  - phrase translation model \( F \rightarrow E \)
  - lexical weights from underlying word alignment
  - a language model \( P(E) \)
  - lexicalized reordering model
  - length features (word/phrase costs/penalties)

\[ P(E|F) = \text{weighted combination of feature functions!} \]
Hypothesis expansion

... and continue adding more hypothesis
→ exponential explosion of search space! (→ needs pruning & limits)

Hypothesis Stacks

▶ here: based on number of foreign words translated
▶ expand all hypotheses from one stack during translation
▶ place expanded hypotheses into appropriate stacks
→ get n-best list of translations

Tree-based Models

Synchronous Context-Free Grammars (SCFGs):
▶ productions with two right-hand sides (source & target)
▶ specify relations between constituents (non-terminals)

NP → (i, watashi wai)
NP → (the box, hako woi)
V → (open, akemasu)

▶ attach probabilities to rules as in PCFGs
Tree-based Models

- Synchronous grammars generate pairs of trees.
- Only linked non-terminals can be expanded.

\[ \langle S^{10}, S^{10} \rangle \]
\[ \langle NP^{11}, VP^{12} \rangle, NP^{11}, VP^{12} \rangle \]
\[ \langle NP^{11}, V^{13} \rangle, NP^{14}, NP^{14}, V^{13} \rangle \]
\[ \langle i, V^{13} \rangle, NP^{14}, watashi, wa, NP^{14}, V^{13} \rangle \]
\[ \langle i, open, NP^{14}, watashi, wa, NP^{14}, akemasu \rangle \]
\[ \langle i, open the box, watashi, wa, hako, wo, akemasu \rangle \]

Decoding by parsing

- Can use well-known algorithms for statistical parsing.
- Chart parsing for synchronous CFGs (→ chart decoding).
- Stack/cube pruning to make decoding more efficient (skip unlikely productions early).
- Difficult: Integration of large-scale language models.

Learning Tree-based Models

- Probabilistic grammars → trained on treebanks.
- Probabilistic synchronous grammars → trained on aligned parallel treebanks.

But if we don’t have aligned parallel treebanks?

→ Induce synchronous grammar from plain parallel corpora!

- Heavily restrict grammar rules (number of non-terminals, reordering of linked non-terminals)
- EM-style learning, or ...
- Rule extraction guided by word alignment

“inversion transduction grammars”, “hierarchical phrase models”, ...

We might have parse trees on one side of the training data.

→ “tree-to-string” & “string-to-tree” models

- More context with increased number of non-terminals
- Focus on linguistically motivated structures (at least on one side)
- Same techniques as before (grammar learning, chart decoding)

(“Syntax-augmented Machine Translation”, ...)
**Tree-based Models**

Parsed data on both sides of the training corpus

→ “tree-to-tree” models

▶ rule extraction guided by word alignment, or ...
▶ explicit tree alignment (constituent alignment)
▶ problem: structural mismatches, noisy alignment
   → rule simplification (for example, binarization)
   → increase coverage & rule confidence

**Other ideas on how to use syntax in SMT**

▶ tree-based language models (target side)
▶ tree-based reordering
   ▶ post-processing (target side)
   ▶ pre-processing (source side)

Most research is now on tree-based/syntax-enhanced MT!

**Summary**

▶ large demand for machine translation
▶ challenging application for computational linguistics
▶ main focus today: data-driven techniques
   ▶ (phrase-based) statistical MT
   ▶ tree-based MT based on statistical parsing

What we didn’t talk about:

▶ How to get data (parallel & monolingual)
▶ How to evaluate automatic translation
▶ How to integrate MT in professional workflows
Some extra slides ...

Log-linear SMT Models

\[ P(E|F) = \text{weighted} (\lambda_m) \text{ combination of feature functions} \ (h_m) \]

\[ P(E|F) = \frac{1}{Z} \exp \sum_{m=1}^{M} \lambda_m h_m(E, F) \]

\[ \hat{E} = \arg\max_E P(E|F) = \arg\max_E (\log P(E|F)) \]

\[ = \arg\max_E \sum_{m=1}^{M} \lambda_m h_m(E, F) \]

How to learn weights \( \lambda_m \)?

- **Minimum error rate training** (MERT) on development set!
- Measure error in terms of BLEU scores \((n\text{-best list})\)
- Iterative adjustment of model parameters (slow but effective!)

Factored Translation Models

- represent (source and target) words by **factors**

\[ \text{Input} \quad \text{Output} \]

- translate lemma and POS separately
- generate surface word forms from translated factors

Training Factored models

- word alignments + symmetrization
- phrase extraction + scoring on factors

\[ \Rightarrow \text{naturally hat john} \quad \text{naturally john has} \]

\[ \Rightarrow \text{ADV V NNP} \quad \text{ADV NNP V} \]

\[ \rightarrow \text{create phrase tables for each translation factor} \]
Automatic Generalization: EBMT Templates

- replace named entities (PER, LOC, ORG), numbers, dates with place-holder variables:
  - John Hancock was in Philadelphia on July 4th.
  - PERSON was in CITY on DATE.
    PERSON war am DATE in CITY.

- generalization by syntactic category (e.g. NP)
  - rekodo no nagasa wa saidai 512 baito de aru.
    The maximum length of records is 512 bytes.
  - X[NP] no nagasa wa saidai Y[N] baito de aru.
    The maximum length of X[NP] is Y[N] bytes.

- generalization by semantic role/feature
  play X[NP/sport]  play X[NP/instrument]
  X[NP] o suru      X[NP] o hiku