Introduction to Statistical Machine Translation

Philipp Koehn
28 November 2008
Topics

- Introduction
- Word-based models and the EM algorithm
- Decoding
- Phrase-based models
- Open source: Moses
- Syntax-based statistical MT
- Factored models
- Large-Scale discriminative training
Machine translation

- Task: translate this into English

- One of the oldest problems in Artificial Intelligence

- AI-hard: reasoning and world knowledge required
The Rosetta stone

- Egyptian language was a mystery for centuries
- 1799 a stone with Egyptian text and its translation into Greek was found
  ⇒ Humans could learn how to translated Egyptian
Parallel data

- Lots of translated text available: 100s of million words of translated text for some language pairs
  - a book has a few 100,000s words
  - an educated person may read 10,000 words a day
  $\rightarrow$ 3.5 million words a year
  $\rightarrow$ 300 million a lifetime
  $\rightarrow$ soon computers will be able to see more translated text than humans read in a lifetime

$\Rightarrow$ Machine *can learn* how to translated foreign languages
Statistical machine translation

- Components: **Translation model, language model, decoder**
The machine translation pyramid

interlingua

foreign semantics

foreign syntax

foreign words

english semantics

english syntax

english words
Word-based models

- Translation process is *decomposed into smaller steps*, each is tied to words

- Original models for statistical machine translation [Brown et al., 1993]
Phrase-based models

- Foreign input is segmented in **phrases**
  - *any sequence of words*, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

[from Koehn et al., 2003, NAACL]
Syntax-based models

[Diagram showing tree structures for reordering, insertion, translation, and taking leaves with corresponding Japanese sentence: "Kare ha ongaku wo kiku no ga daisuki desu"]

[from Yamada and Knight, 2001]
Automatic evaluation

• Why automatic evaluation metrics?
  – Manual evaluation is too slow
  – Evaluation on large test sets reveals minor improvements
  – Automatic tuning to improve machine translation performance

• History
  – Word Error Rate
  – BLEU since 2002

• BLEU in short: Overlap with reference translations
Automatic evaluation

• Reference Translation
  – the gunman was shot to death by the police.

• System Translations
  – the gunman was police kill.
  – wounded police jaya of
  – the gunman was shot dead by the police.
  – the gunman arrested by police kill.
  – the gunmen were killed.
  – the gunman was shot to death by the police.
  – gunmen were killed by police ?SUB>0 ?SUB>0
  – al by the police.
  – the ringer is killed by the police.
  – police killed the gunman.

• Matches
  – green = 4 gram match (good!)
  – red = word not matched (bad!)
Automatic evaluation

- BLEU correlates with human judgement
  - multiple reference translations may be used
Correlation? [Callison-Burch et al., 2006]

- DARPA/NIST MT Eval 2005
  - Mostly statistical systems (all but one in graphs)
  - One submission **manual post-edit** of statistical system’s output
  - Good adequacy/fluency scores *not reflected* by BLEU
Correlation? [Callison-Burch et al., 2006]

- Comparison of
  - good statistical system: high BLEU, high adequacy/fluency
  - bad statistical sys. (trained on less data): low BLEU, low adequacy/fluency
  - Systran: lowest BLEU score, but high adequacy/fluency
Automatic evaluation: outlook

• Research questions
  – why does BLEU fail Systran and manual post-edits?
  – how can this overcome with novel evaluation metrics?

• Future of automatic methods
  – automatic metrics too useful to be abandoned
  – evidence still supports that during system development, a better BLEU indicates a better system
  – final assessment has to be human judgement
Competitions

- Progress driven by MT Competitions
  - NIST/DARPA: Yearly campaigns for Arabic-English, Chinese-English, newstexts, since 2001
  - IWSLT: Yearly competitions for Asian languages and Arabic into English, speech travel domain, since 2003
  - WPT/WMT: Yearly competitions for European languages, European Parliament proceedings, since 2005

- Increasing number of statistical MT groups participate
Euromatrix

• Proceedings of the European Parliament
  – translated into 11 official languages
  – entry of new members in May 2004: more to come...

• Europarl corpus
  – collected 20-30 million words per language
  → 110 language pairs

• 110 Translation systems
  – 3 weeks on 16-node cluster computer
  → 110 translation systems
# Quality of translation systems

- **Scores** for all 110 systems  

<table>
<thead>
<tr>
<th></th>
<th>da</th>
<th>de</th>
<th>el</th>
<th>en</th>
<th>es</th>
<th>fr</th>
<th>fi</th>
<th>it</th>
<th>nl</th>
<th>pt</th>
<th>sv</th>
</tr>
</thead>
<tbody>
<tr>
<td>da</td>
<td></td>
<td>18.4</td>
<td>21.1</td>
<td>28.5</td>
<td>26.4</td>
<td>28.7</td>
<td>14.2</td>
<td>22.2</td>
<td>21.4</td>
<td>24.3</td>
<td>28.3</td>
</tr>
<tr>
<td>de</td>
<td>22.3</td>
<td></td>
<td>20.7</td>
<td>25.3</td>
<td>25.4</td>
<td>27.7</td>
<td>11.8</td>
<td>21.3</td>
<td>23.4</td>
<td>23.2</td>
<td>20.5</td>
</tr>
<tr>
<td>el</td>
<td>22.7</td>
<td>17.4</td>
<td></td>
<td>27.2</td>
<td>31.2</td>
<td>32.1</td>
<td>11.4</td>
<td>26.8</td>
<td>20.0</td>
<td>27.6</td>
<td>21.2</td>
</tr>
<tr>
<td>en</td>
<td>25.2</td>
<td>17.6</td>
<td>23.2</td>
<td></td>
<td>30.1</td>
<td>31.1</td>
<td>13.0</td>
<td>25.3</td>
<td>21.0</td>
<td>27.1</td>
<td>24.8</td>
</tr>
<tr>
<td>es</td>
<td>24.1</td>
<td>18.2</td>
<td>28.3</td>
<td>30.5</td>
<td></td>
<td>40.2</td>
<td>12.5</td>
<td>32.3</td>
<td>21.4</td>
<td>35.9</td>
<td>23.9</td>
</tr>
<tr>
<td>fr</td>
<td>23.7</td>
<td>18.5</td>
<td>26.1</td>
<td>30.0</td>
<td>38.4</td>
<td></td>
<td>12.6</td>
<td>32.4</td>
<td>21.1</td>
<td>35.3</td>
<td>22.6</td>
</tr>
<tr>
<td>fi</td>
<td>20.0</td>
<td>14.5</td>
<td>18.2</td>
<td>21.8</td>
<td>21.1</td>
<td>22.4</td>
<td></td>
<td>18.3</td>
<td>17.0</td>
<td>19.1</td>
<td>18.8</td>
</tr>
<tr>
<td>it</td>
<td>21.4</td>
<td>16.9</td>
<td>24.8</td>
<td>27.8</td>
<td>34.0</td>
<td>36.0</td>
<td>11.0</td>
<td></td>
<td>20.0</td>
<td>31.2</td>
<td>20.2</td>
</tr>
<tr>
<td>nl</td>
<td>20.5</td>
<td>18.3</td>
<td>17.4</td>
<td>23.0</td>
<td>22.9</td>
<td>24.6</td>
<td>10.3</td>
<td>20.0</td>
<td></td>
<td>20.7</td>
<td>19.0</td>
</tr>
<tr>
<td>pt</td>
<td>23.2</td>
<td>18.2</td>
<td>26.4</td>
<td>30.1</td>
<td>37.9</td>
<td>39.0</td>
<td>11.9</td>
<td>32.0</td>
<td>20.2</td>
<td></td>
<td>21.9</td>
</tr>
<tr>
<td>sv</td>
<td>30.3</td>
<td>18.9</td>
<td>22.8</td>
<td>30.2</td>
<td>28.6</td>
<td>29.7</td>
<td>15.3</td>
<td>23.9</td>
<td>21.9</td>
<td>25.9</td>
<td></td>
</tr>
</tbody>
</table>

[from Koehn, 2005: Europarl]
What makes MT difficult?

• Some language pairs more difficult than others

• Birch et al [EMNLP 2008] showed 75% of the differences in BLEU scores due to
  – morphology on target side (vocabulary size)
  – historic distance of languages (cognate ratio)
  – degree of reordering requited

• Not a factor: morphology on source
  – note: Arabic–English fairly good, despite rich morphology in Arabic
Available data

• Available parallel text
  – **Europarl**: 40 million words in 11 languages http://www.statmt.org/europarl/
  – **Acquis Communitaire**: 8-50 million words in 20 EU languages
  – **Canadian Hansards**: 20 million words from Ulrich Germann, ISI
  – Chinese/Arabic to English: over 100 million words from LDC
  – lots more French/English, Spanish/French/English from LDC

• Available monolingual text (for language modeling)
  – 2.8 billion words of English from LDC
  – trillions of words on the web
More data, better translations

- **Log-scale improvements** on BLEU:
  Doubling the training data gives constant improvement \((+1 \text{% BLEU})\)
More LM data, better translations

- Also **log-scale improvements** on BLEU:
  - doubling the training data gives constant improvement \((+0.5 \%BLEU)\)
  - (last addition is 218 billion words out-of-domain web data)

[from Och, 2005: MT Eval presentation]
Word-based models and the EM algorithm
Lexical translation

- How to translate a word → look up in dictionary

  **Haus** — *house, building, home, household, shell.*

- **Multiple translations**
  - some more frequent than others
  - for instance: *house,* and *building* most common
  - special cases: *Haus* of a *snail* is its *shell*

- Note: During all the lectures, we will translate from a foreign language into English
Collect statistics

• Look at a parallel corpus (German text along with English translation)

<table>
<thead>
<tr>
<th>Translation of Haus</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>house</td>
<td>8,000</td>
</tr>
<tr>
<td>building</td>
<td>1,600</td>
</tr>
<tr>
<td>home</td>
<td>200</td>
</tr>
<tr>
<td>household</td>
<td>150</td>
</tr>
<tr>
<td>shell</td>
<td>50</td>
</tr>
</tbody>
</table>
Estimate translation probabilities

• Maximum likelihood estimation

\[
p_f(e) = \begin{cases} 
0.8 & \text{if } e = \text{house,} \\
0.16 & \text{if } e = \text{building,} \\
0.02 & \text{if } e = \text{home,} \\
0.015 & \text{if } e = \text{household,} \\
0.005 & \text{if } e = \text{shell.}
\end{cases}
\]
Alignment

• In a parallel text (or when we translate), we **align** words in one language with the words in the other

```
1 2 3 4
das Haus ist klein
the house is small
1 2 3 4
```

• Word **positions** are numbered 1–4
Alignment function

- Formalizing *alignment* with an **alignment function**

- Mapping an English target word at position $i$ to a German source word at position $j$ with a function $a : i \rightarrow j$

- Example

\[
a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}
\]
Reordering

- Words may be reordered during translation

\[
a : \{1 \to 3, 2 \to 4, 3 \to 2, 4 \to 1\}
\]
One-to-many translation

- A source word may translate into \textit{multiple} target words

```
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
das & Haus & ist & klitzeklein \\
the & house & is & very & small \\
1 & 2 & 3 & 4 & 5 \\
\end{array}
```

\[ a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\} \]
Dropping words

- Words may be **dropped** when translated
  - The German article *das* is dropped

```
1 2 3

das Haus ist klein
```

```
1 2 3 4

house is small
```

\[ a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\} \]
Inserting words

- Words may be *added* during translation
  - The English *just* does not have an equivalent in German
  - We still need to map it to something: special *NULL* token

\[ a : \{ 1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4 \} \]
IBM Model 1

- **Generative model**: break up translation process into smaller steps
  - **IBM Model 1** only uses *lexical translation*

- Translation probability
  - for a foreign sentence \( \mathbf{f} = (f_1, \ldots, f_{l_f}) \) of length \( l_f \)
  - to an English sentence \( \mathbf{e} = (e_1, \ldots, e_{l_e}) \) of length \( l_e \)
  - with an alignment of each English word \( e_j \) to a foreign word \( f_i \) according to the alignment function \( a : j \rightarrow i \)

  \[
P(\mathbf{e}, a | \mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})
  \]

  - parameter \( \epsilon \) is a *normalization constant*
### Example

**das** | **Haus** | **ist** | **klein**
---|---|---|---
|  
| \(e\) & \(t(e|f)\) | \(e\) & \(t(e|f)\) | \(e\) & \(t(e|f)\) | \(e\) & \(t(e|f)\)  
| the & 0.7 | house & 0.8 | is & 0.8 | small & 0.4  
| that & 0.15 | building & 0.16 | 's & 0.16 | little & 0.4  
| which & 0.075 | home & 0.02 | exists & 0.02 | short & 0.1  
| who & 0.05 | household & 0.015 | has & 0.015 | minor & 0.06  
| this & 0.025 | shell & 0.005 | are & 0.005 | petty & 0.04  

\[
p(e,a|f) = \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})
\]

\[
= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4
\]

\[
= 0.0028 \epsilon
\]
Learning lexical translation models

• We would like to *estimate* the lexical translation probabilities \( t(e|f) \) from a parallel corpus

• ... but we do not have the alignments

• **Chicken and egg problem**
  – if we had the *alignments*,
    → we could estimate the *parameters* of our generative model
  – if we had the *parameters*,
    → we could estimate the *alignments*
EM algorithm

- **Incomplete data**
  - if we had *complete data*, would could estimate *model*
  - if we had *model*, we could fill in the *gaps in the data*

- **Expectation Maximization (EM) in a nutshell**
  - initialize model parameters (e.g. uniform)
  - assign probabilities to the missing data
  - estimate model parameters from completed data
  - iterate
EM algorithm

... la maison ... la maison blue ... la fleur ...

... the house ... the blue house ... the flower ...

• Initial step: all alignments equally likely

• Model learns that, e.g., *la* is often aligned with *the*
EM algorithm

... la maison ... la maison blue ... la fleur ...

... the house ... the blue house ... the flower ...

• After one iteration

• Alignments, e.g., between *la* and *the* are more likely
EM algorithm

... la maison ... la maison bleu ... la fleur ...

... the house ... the blue house ... the flower ...

- After another iteration

- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely (*pigeon hole principle*)
EM algorithm

... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...

- Convergence

- Inherent hidden structure revealed by EM
EM algorithm

... la maison ... la maison bleu ... la fleur ...  

/  \  /  \  /  \  /  \  
... the house ... the blue house ... the flower ... 

\[ p(\text{la}|\text{the}) = 0.453 \]
\[ p(\text{le}|\text{the}) = 0.334 \]
\[ p(\text{maison}|\text{house}) = 0.876 \]
\[ p(\text{bleu}|\text{blue}) = 0.563 \]

- Parameter estimation from the aligned corpus
EM Algorithm consists of two steps

**Expectation-Step**: Apply model to the data
- parts of the model are hidden (here: alignments)
- using the model, assign probabilities to possible values

**Maximization-Step**: Estimate model from data
- take assign values as fact
- collect counts (weighted by probabilities)
- estimate model from counts

Iterate these steps until **convergence**
IBM Model 1 and EM

- We need to be able to compute:
  - Expectation-Step: probability of alignments
  - Maximization-Step: count collection
IBM Model 1 and EM

- **Probabilities**
  \[ p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05 \]
  \[ p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8 \]

- **Alignments**

  \[ p(\text{e, a}|\text{f}) = 0.56 \quad p(\text{e, a}|\text{f}) = 0.035 \quad p(\text{e, a}|\text{f}) = 0.08 \quad p(\text{e, a}|\text{f}) = 0.005 \]
  \[ p(\text{a}|\text{e, f}) = 0.824 \quad p(\text{a}|\text{e, f}) = 0.052 \quad p(\text{a}|\text{e, f}) = 0.118 \quad p(\text{a}|\text{e, f}) = 0.007 \]

- **Counts**
  \[ c(\text{the}|\text{la}) = 0.824 + 0.052 \]
  \[ c(\text{the}|\text{maison}) = 0.118 + 0.007 \]
  \[ c(\text{house}|\text{la}) = 0.052 + 0.007 \]
  \[ c(\text{house}|\text{maison}) = 0.824 + 0.118 \]
Higher IBM Models

<table>
<thead>
<tr>
<th>IBM Model 1</th>
<th>lexical translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Model 2</td>
<td>adds absolute <strong>reordering model</strong></td>
</tr>
<tr>
<td>IBM Model 3</td>
<td>adds <strong>fertility model</strong></td>
</tr>
<tr>
<td>IBM Model 4</td>
<td>relative reordering model</td>
</tr>
<tr>
<td>IBM Model 5</td>
<td>fixes <strong>deficiency</strong></td>
</tr>
</tbody>
</table>

- Only IBM Model 1 has **global maximum**
  - training of a higher IBM model builds on previous model

- Computationally biggest change in Model 3
  - trick to simplify estimation does not work anymore
  - **exhaustive** count collection becomes computationally too expensive
  - **sampling** over high probability alignments is used instead
IBM Model 4

Mary did not slap the green witch
Mary not slap slap slap the green witch
Mary not slap slap slap NULL the green witch
Maria no daba una botefada a la verde bruja
Maria no daba una bofetada a la bruja verde

n(3|slap)
p-null
t(la|the)
d(4|4)
Decoding
Statistical Machine Translation

- Components: Translation model, language model, decoder
Phrase-Based Translation

- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated

- Each phrase is translated into English

- Phrases are reordered
## Phrase Translation Table

- Phrase Translations for “den Vorschlag”:

| English              | $\phi(e|f)$ | English              | $\phi(e|f)$ |
|----------------------|------------|----------------------|------------|
| the proposal         | 0.6227     | the suggestions      | 0.0114     |
| ’s proposal          | 0.1068     | the proposed         | 0.0114     |
| a proposal           | 0.0341     | the motion           | 0.0091     |
| the idea             | 0.0250     | the idea of          | 0.0091     |
| this proposal        | 0.0227     | the proposal ,       | 0.0068     |
| proposal             | 0.0205     | its proposal         | 0.0068     |
| of the proposal      | 0.0159     | it                   | 0.0068     |
| the proposals        | 0.0159     | ...                  | ...        |
Decoding Process

- Build translation left to right
  - *select foreign* words to be translated
Decoding Process

- Build translation \textit{left to right}
  - select foreign words to be translated
  - find \textit{English} phrase translation
  - add \textit{English} phrase to end of partial translation
Decoding Process

- Build translation left to right
  - select foreign words to be translated
  - find English phrase translation
  - add English phrase to end of partial translation
  - *mark foreign* words as translated
Decoding Process

- *One to many* translation
Decoding Process

- Many to one translation
• *Many to one* translation
Decoding Process

- **Reordering**

Maria no dio una bofetada a la bruja verde

Mary did not slap the green

Philipp Koehn Statistical Machine Translation 28 November 2008
Decoding Process

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio una bofetada</th>
<th>a la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

| Mary | did not | slap | the | green | witch |

- Translation *finished*
Translation Options

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td>a slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Look up possible phrase translations
  - many different ways to segment words into phrases
  - many different ways to translate each phrase
Hypothesis Expansion

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td>a slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td>to</td>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>slap</td>
<td>to</td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Start with **empty hypothesis**
  - e: no English words
  - f: no foreign words covered
  - p: probability 1
Hypothesis Expansion

- Pick *translation option*
- Create *hypothesis*
  - e: add English phrase Mary
  - f: first foreign word covered
  - p: probability 0.534

**Philipp Koehn**

Statistical Machine Translation

28 November 2008
A Quick Word on Probabilities

• Not going into detail here, but...

• Translation Model
  – phrase translation probability $p(\text{Mary}|\text{Maria})$
  – reordering costs
  – phrase/word count costs
  – ...

• Language Model
  – uses trigrams:
  – $p(\text{Mary did not}) = 
    p(\text{Mary}|\text{START}) \times p(\text{did}|\text{Mary,START}) \times p(\text{not}|\text{Mary did})$
Hypothesis Expansion

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

Mary did not give a slap to the green witch.

- Add another hypothesis
Hypothesis Expansion

- Further *hypothesis expansion*
Hypothesis Expansion

• ... until all foreign words covered
  – find best hypothesis that covers all foreign words
  – backtrack to read off translation
Hypothesis Expansion

- Adding more hypothesis

⇒ *Explosion* of search space

Philipp Koehn

Statistical Machine Translation

28 November 2008
Explosion of Search Space

- Number of hypotheses is \textit{exponential} with respect to sentence length

\Rightarrow Decoding is NP-complete [Knight, 1999]

\Rightarrow Need to \textit{reduce search space}

- risk free: hypothesis \textit{recombination}
- risky: \textit{histogram/threshold pruning}
Hypothesis Recombination

- Different paths to the *same* partial translation
Hypothesis Recombination

- Different paths to the same partial translation

⇒ Combine paths
  - drop weaker path
  - keep pointer from weaker path (for lattice generation)
Hypothesis Recombination

- Recombined hypotheses do not have to *match completely*
- No matter what is added, weaker path can be dropped, if:
  - *last two English words* match (matters for language model)
  - *foreign word coverage* vectors match (effects future path)
Hypothesis Recombination

- Recombined hypotheses do not have to match completely
- No matter what is added, weaker path can be dropped, if:
  - last two English words match (matters for language model)
  - foreign word coverage vectors match (effects future path)

⇒ Combine paths
Pruning

- Hypothesis recombination is *not sufficient*

⇒ Heuristically *discard* weak hypotheses early

- Organize Hypothesis in **stacks**, e.g. by
  - *same* foreign words covered
  - *same number* of foreign words covered
  - *same number* of English words produced

- Compare hypotheses in stacks, discard bad ones
  - **histogram pruning**: keep top \( n \) hypotheses in each stack (e.g., \( n = 100 \))
  - **threshold pruning**: keep hypotheses that are at most \( \alpha \) times the cost of best hypothesis in stack (e.g., \( \alpha = 0.001 \))
Hypothesis Stacks

- Organization of hypothesis into stacks
  - here: based on *number of foreign words* translated
  - during translation all hypotheses from one stack are expanded
  - expanded Hypotheses are placed into stacks
Comparing Hypotheses

• Comparing hypotheses with *same number of foreign words* covered

Maria no dio una bofetada a la bruja verde

- e: Mary did not
- f: **-------
- p: 0.154

- e: the
- f: -----**--
- p: 0.354

better partial translation

covers easier part --> lower cost

• Hypothesis that covers *easy part* of sentence is preferred

⇒ Need to consider *future cost* of uncovered parts
Future Cost Estimation

- Estimate cost to translate remaining part of input

- Step 1: estimate future cost for each translation option
  - look up translation model cost
  - estimate language model cost (no prior context)
  - ignore reordering model cost
  → $\text{LM} \times \text{TM} = p(\text{to}) \times p(\text{the}|\text{to}) \times p(\text{to the}|\text{a la})$
Future Cost Estimation: Step 2

- Step 2: find *cheapest cost* among translation options

\[
\begin{array}{c}
\text{a la} \\
\downarrow \\
\text{to the} & \text{cost} = 0.0372 \\
\downarrow \\
\text{to} & \text{cost} = 0.0299 \\
\downarrow \\
\text{the} & \text{cost} = 0.0354
\end{array}
\]
Future Cost Estimation: Step 3

• Step 3: find *cheapest future cost path* for each span
  – can be done *efficiently* by dynamic programming
  – future cost for every span can be *pre-computed*
Future Cost Estimation: Application

- Use future cost estimates when *pruning* hypotheses

- For each *uncovered contiguous span*:
  - look up *future costs* for each maximal contiguous uncovered span
  - *add* to actually accumulated cost for translation option for pruning
A* search

- Pruning might drop hypothesis that lead to the best path (search error)

- **A* search**: safe pruning
  - future cost estimates have to be accurate or underestimates
  - **lower bound** for probability is established early by **depth first search**: compute cost for one complete translation
  - if cost-so-far and future cost are worse than **lower bound**, hypothesis can be safely discarded

- Not commonly done, since not aggressive enough
Limits on Reordering

- Reordering may be **limited**
  - **Monotone** Translation: No reordering at all
  - Only phrase movements of at most $n$ words

- Reordering limits *speed* up search (polynomial instead of exponential)

- Current reordering models are weak, so limits *improve* translation quality
• Search graph can be easily converted into a word lattice
  – can be further mined for n-best lists
  → enables reranking approaches
  → enables discriminative training
### Sample N-Best List

- **Simple N-best list:**

<table>
<thead>
<tr>
<th>Translation</th>
<th>Reordering LM</th>
<th>TM</th>
<th>Word Penalty</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>this is a small house</td>
<td>0</td>
<td>-27.0908</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>this is a little house</td>
<td>0</td>
<td>-28.1791</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>it is a small house</td>
<td>0</td>
<td>-27.108</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>it is a little house</td>
<td>0</td>
<td>-28.1963</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>this is an small house</td>
<td>0</td>
<td>-31.7294</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>it is an small house</td>
<td>0</td>
<td>-32.3094</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>this is an little house</td>
<td>0</td>
<td>-33.7639</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>this is a house small</td>
<td>-3</td>
<td>-31.4851</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>this is a house little</td>
<td>-3</td>
<td>-31.5689</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>it is an little house</td>
<td>0</td>
<td>-34.3439</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>it is a house small</td>
<td>-3</td>
<td>-31.5022</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>this is an house small</td>
<td>-3</td>
<td>-32.8999</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>it is a house little</td>
<td>-3</td>
<td>-31.586</td>
<td>-3.21888</td>
<td>-5</td>
</tr>
<tr>
<td>this is an house little</td>
<td>-3</td>
<td>-32.9837</td>
<td>-1.83258</td>
<td>-5</td>
</tr>
<tr>
<td>the house is a little</td>
<td>-7</td>
<td>-28.5107</td>
<td>-2.52573</td>
<td>-5</td>
</tr>
<tr>
<td>the is a small house</td>
<td>0</td>
<td>-35.6899</td>
<td>-2.52573</td>
<td>-5</td>
</tr>
<tr>
<td>is it a little house</td>
<td>-4</td>
<td>-30.3603</td>
<td>-3.91202</td>
<td>-5</td>
</tr>
<tr>
<td>the house is a small</td>
<td>-7</td>
<td>-28.7683</td>
<td>-2.52573</td>
<td>-5</td>
</tr>
<tr>
<td>it ’s a small house</td>
<td>0</td>
<td>-34.8557</td>
<td>-3.91202</td>
<td>-5</td>
</tr>
<tr>
<td>this house is a little</td>
<td>-7</td>
<td>-28.0443</td>
<td>-3.91202</td>
<td>-5</td>
</tr>
<tr>
<td>it ’s a little house</td>
<td>0</td>
<td>-35.1446</td>
<td>-3.91202</td>
<td>-5</td>
</tr>
<tr>
<td>this house is a small</td>
<td>-7</td>
<td>-28.3018</td>
<td>-3.91202</td>
<td>-5</td>
</tr>
</tbody>
</table>
Phrase-based models
Word alignment

- Notion of **word alignment** valuable

- Shared task at NAACL 2003 and ACL 2005 workshops
Word alignment with IBM models

• IBM Models create a *many-to-one* mapping
  
  – words are aligned using an *alignment function*
  
  – a function may return the same value for different input
    (one-to-many mapping)
  
  – a function can not return multiple values for one input
    (*no many-to-one* mapping)

• But we need *many-to-many* mappings
Symmetrizing word alignments

• Intersection of GIZA++ bidirectional alignments
Symmetrizing word alignments

- *Grow* additional alignment points [Och and Ney, CompLing2003]
Growing heuristic

GROW-DIAG-FINAL(e2f,f2e):
neighboring = ((-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1))
alignment = intersect(e2f,f2e);
GROW-DIAG(); FINAL(e2f); FINAL(f2e);

GROW-DIAG():
iterate until no new points added
for english word e = 0 ... en
  for foreign word f = 0 ... fn
    if ( e aligned with f )
      for each neighboring point ( e-new, f-new ):
        if ( ( e-new not aligned and f-new not aligned ) and
            ( e-new, f-new ) in union( e2f, f2e ) )
          add alignment point ( e-new, f-new )

FINAL(a):
for english word e-new = 0 ... en
  for foreign word f-new = 0 ... fn
    if ( ( e-new not aligned or f-new not aligned ) and
        ( e-new, f-new ) in alignment a )
      add alignment point ( e-new, f-new )
Phrase-based translation

- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered
Phrase-based translation model

- Major components of phrase-based model
  - **phrase translation model** $\phi(f|e)$
  - **reordering model** $\omega^{\text{length}(e)}$
  - **language model** $p_{LM}(e)$

- Bayes rule
  \[
  \arg \max_e p(e|f) = \arg \max_e p(f|e)p(e) \\
  = \arg \max_e \phi(f|e)p_{LM}(e)\omega^{\text{length}(e)}
  \]

- Sentence $f$ is decomposed into $I$ phrases $\vec{f}_1 = \bar{f}_1, ..., \bar{f}_I$

- Decomposition of $\phi(f|e)$
  \[
  \phi(\vec{f}_I|\vec{e}_I) = \prod_{i=1}^{I} \phi(f_i|e_i) d(a_i - b_{i-1})
  \]
Advantages of phrase-based translation

- *Many-to-many* translation can handle non-compositional phrases

- Use of *local context* in translation

- The more data, the *longer phrases* can be learned
Phrase translation table

- Phrase translations for *den Vorschlag*

| English                  | $\phi(e|f)$ | English                  | $\phi(e|f)$ |
|--------------------------|------------|--------------------------|------------|
| the proposal             | 0.6227     | the suggestions          | 0.0114     |
| ’s proposal              | 0.1068     | the proposed             | 0.0114     |
| a proposal               | 0.0341     | the motion               | 0.0091     |
| the idea                 | 0.0250     | the idea of              | 0.0091     |
| this proposal            | 0.0227     | the proposal ,           | 0.0068     |
| proposal                 | 0.0205     | its proposal             | 0.0068     |
| of the proposal          | 0.0159     | it                       | 0.0068     |
| the proposals            | 0.0159     | ...                      | ...        |
How to learn the phrase translation table?

- Start with the **word alignment**:

  ![](image)

- Collect all phrase pairs that are **consistent** with the word alignment.
• **Consistent with the word alignment**

  phrase alignment has to contain all alignment points for all covered words

  \[(\overline{e}, \overline{f}) \in BP \iff \forall e_i \in \overline{e} : (e_i, f_j) \in A \rightarrow f_j \in \overline{f} \]

  AND \[\forall f_j \in \overline{f} : (e_i, f_j) \in A \rightarrow e_i \in \overline{e}\]
Word alignment induced phrases

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

(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)
Word alignment induced phrases

(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)
Word alignment induced phrases

(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)
Word alignment induced phrases (5)

(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)
Probability distribution of phrase pairs

- We need a **probability distribution** $\phi(f|e)$ over the collected phrase pairs

⇒ Possible *choices*
  - *relative frequency* of collected phrases: $\phi(f|e) = \frac{\text{count}(f,e)}{\sum_f \text{count}(f,e)}$
  - or, conversely $\phi(e|f)$
  - use *lexical translation probabilities*
Reordering

- **Monotone** translation
  - do not allow any reordering
  $\rightarrow$ worse translations

- **Limiting** reordering (to movement over max. number of words) helps

- **Distance-based** reordering cost
  - moving a foreign phrase over $n$ words: cost $\omega^n$

- **Lexicalized** reordering model
Lexicalized reordering models

- Three orientation types: monotone, swap, discontinuous

- Probability $p(swap|e, f)$ depends on foreign (and English) phrase involved
Learning lexicalized reordering models

- Orientation type is *learned during phrase extractions*

- *Alignment point* to the *top left* (monotone) or *top right* (swap)?

- For more, see [Tillmann, 2003] or [Koehn et al., 2005]
Open Source Machine Translation
Research Process

- new ideas
- prototype
- experiments
- research paper
- dissemination
- rebuild prototype
- new ideas

SMT is increasingly a big systems field building prototypes requires huge efforts

Philipp Koehn

Statistical Machine Translation

28 November 2008
Research Process

SMT is increasingly a big systems field

building prototypes requires huge efforts

new ideas

prototype

experiments

research paper

dissemination

rebuild prototype

new ideas
Requirements for Building MT Systems

• **Data resources**
  – *parallel* corpora (translated texts)
  – *monolingual* corpora, especially for output language

• **Support tools**
  – basic *corpus preparation*: tokenization, sentence alignment
  – *linguistic* tools: tagger, parsers, morphology, semantic processing

• **MT tools**
  – word alignment, *training*
  – *decoding* (translation engine)
  – tuning (optimization)
  – re-ranking, incl. posterior methods
Who will do MT Research?

- If MT research requires the development of *many resources*
  - who will be able to do relevant research?
  - who will be able to deploy the technology?

- A *few* big labs?

- ... or a *broad network* of academic and commercial institutions?
MT is diverse

- Many different **stakeholders**
  - academic researchers
  - commercial developers
  - multi-lingual or trans-lingual content providers
  - end users of online translation services
  - human translation service providers

- Many different **language pairs**
  - few languages with rich resources: *English, Spanish, German, Chinese, ...*
  - many second tier languages: *Czech, Danish, Greek, ...*
  - many under-resourced languages: *Gaelic, Basque, ...*
Open Research

SMT is increasingly a big systems field

building prototypes requires huge efforts

sharing of resources reduces duplication of efforts

new ideas

prototype

experiments

research paper

dissemination

re-use prototype

new ideas
Making Open Research Work

• Non-restrictive licensing

• Active development
  – working high-quality prototype
  – ongoing development
  – open to contributions

• Support and dissemination
  – support by email, web sites, documentation
  – offering tutorials and courses
Moses: Open Source Toolkit

- **Open source** statistical machine translation system (developed from scratch 2006)
  - state-of-the-art *phrase-based* approach
  - novel methods: *factored translation models*, *confusion network decoding*
  - support for *very large models* through *memory-efficient* data structures

- Documentation, source code, binaries available at http://www.statmt.org/moses/

- Development also **supported by**
  - EC-funded *TC-STAR* project
  - *US* funding agencies DARPA, NSF
  - universities (Edinburgh, Maryland, MIT, ITC-irst, RWTH Aachen, ...)

Philipp Koehn  
Statistical Machine Translation  
28 November 2008
Call for Participation: 3rd MT Marathon

- Prague, Czech Republic, January 26-30

- Events
  - winter school (5-day course on MT)
  - research showcase
  - open source showcase: call for papers, due December 2nd
  - open source hands-on projects

- Sponsored by EuroMatrix project — free of charge
Syntax-based models
Advantages of Syntax-Based Translation

- *Reordering* for syntactic reasons
  - e.g., move German object to end of sentence

- Better explanation for *function words*
  - e.g., prepositions, determiners

- Conditioning to *syntactically related words*
  - translation of verb may depend on subject or object

- Use of *syntactic language models*
  - ensuring grammatical output
Syntactic Language Model

- **Good syntax tree** → good English
- Allows for **long distance constraints**

- Left translation preferred by syntactic LM
String to Tree Translation

• Use of English *syntax trees* [Yamada and Knight, 2001]
  – exploit *rich resources* on the English side
  – obtained with statistical parser [Collins, 1997]
  – *flattened tree* to allow more reorderings
  – works well with syntactic language model
Yamada and Knight [2001]

Kare ha ongaku wo kiku no ga daisuki desu

[from Yamada and Knight, 2001]
## Reordering Table

| Original Order   | Reordering       | \(p(\text{reorder} | \text{original})\) |
|------------------|------------------|---------------------|
| PRP VB1 VB2      | PRP VB1 VB2      | 0.074               |
| **PRP VB1 VB2**  | **PRP VB2 VB1**  | **0.723**           |
| PRP VB1 VB2      | VB1 PRP VB2      | 0.061               |
| PRP VB1 VB2      | VB1 VB2 PRP      | 0.037               |
| PRP VB1 VB2      | VB2 PRP VB1      | 0.083               |
| PRP VB1 VB2      | VB2 VB1 PRP      | 0.021               |
| VB TO            | VB TO            | 0.107               |
| **VB TO**        | **TO VB**        | **0.893**           |
| TO NN            | TO NN            | 0.251               |
| **TO NN**        | **NN TO**        | **0.749**           |
Decoding as Parsing

- **Chart Parsing**

```
  PRP
  he
```

- **Pick Japanese** *words*

- **Translate into** *tree stumps*
Decoding as Parsing

- Chart Parsing

```
  PRP  NN  TO
  he  music  to
kare  ha  ongaku  wo  kiku  no  ga  daisuki desu
```

- Pick Japanese words

- Translate into tree stumps
Decoding as Parsing

- Adding some *more entries*...
Decoding as Parsing

- Combine entries

kare ha ongaku wo kiku no ga daisuki desu
Decoding as Parsing

- PRP: he
- NN: music
- TO: to
- VB: listening

kare ha ongaku wo kiku no ga daisuki desu
Decoding as Parsing

kare ha ongaku wo kiku no ga daisuki desu
Decoding as Parsing

- *Finished* when all foreign words covered

Philipp Koehn  Statistical Machine Translation  28 November 2008
Yamada and Knight: Training

- Parsing of the English side
  - using Collins statistical parser

- EM training
  - translation model is used to map training sentence pairs
  - EM training finds low-perplexity model
  → unity of training and decoding as in IBM models
Is the Model Realistic?

• Do English trees *match* foreign strings?

• Crossings between French-English [Fox, 2002]
  – 0.29-6.27 per sentence, depending on how it is measured

• Can be reduced by
  – *flattening tree*, as done by [Yamada and Knight, 2001]
  – detecting *phrasal* translation
  – *special treatment* for small number of constructions

• Most coherence between *dependency structures*
Chiang: Hierarchical Phrase Model

- **Chiang** [ACL, 2005] (best paper award!)
  - context free bi-grammar
  - *one non-terminal* symbol
  - right hand side of rule may include non-terminals and terminals

- *Competitive* with phrase-based models in 2005 DARPA/NIST evaluation
Types of Rules

- **Word** translation
  - $X \rightarrow \text{maison} \parallel \text{house}$

- **Phrasal** translation
  - $X \rightarrow \text{daba una bofetada} \mid \text{slap}$

- **Mixed** non-terminal / terminal
  - $X \rightarrow X \text{ bleue} \parallel \text{blue } X$
  - $X \rightarrow \text{ne } X \text{ pas} \parallel \text{not } X$
  - $X \rightarrow X_1 X_2 \parallel X_2 \text{ of } X_1$

- **Technical rules**
  - $S \rightarrow S X \parallel S X$
  - $S \rightarrow X \parallel X$
Learning Hierarchical Rules

X → X verde || green X
Learning Hierarchical Rules

\[
X \rightarrow \text{a la } X \parallel \text{the } X
\]
Details of Chiang’s Model

- Too many rules
  - → filtering of rules necessary

- Efficient parse decoding possible
  - hypothesis stack for each span of foreign words
  - only one non-terminal → hypotheses comparable
  - length limit for spans that do not start at beginning
Clause Level Restructuring [Collins et al.]

- **Why clause structure?**
  - languages *differ vastly* in their clause structure
    (English: SVO, Arabic: VSO, German: fairly *free order*; a lot details differ: position of adverbs, sub clauses, etc.)
  - large-scale restructuring is a *problem* for phrase models

- **Restructuring**
  - *reordering* of constituents (main focus)
  - add/drop/change of *function words*

- Details see [Collins, Kucerova and Koehn, ACL 2005]
Clause Structure

- **Syntax tree** from German parser
  - statistical parser by Amit Dubay, trained on TIGER treebank
Reordering When Translating

- **Reordering** when translating into English
  - tree is *flattened*
  - clause level constituents line up
Clause level reordering is a well defined task

- label German constituents with their English order
- done this for 300 sentences, two annotators, high agreement

Philipp Koehn  Statistical Machine Translation  28 November 2008
Systematic Reordering German $\rightarrow$ English

- Many types of reorderings are **systematic**
  - move verb group together
  - subject - verb - object
  - move negation in front of verb

⇒ Write rules by hand
  - apply rules to test and training data
  - train standard phrase-based SMT system

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline system</td>
<td>25.2%</td>
</tr>
<tr>
<td>with manual rules</td>
<td>26.8%</td>
</tr>
</tbody>
</table>
Other Syntax-Based Approaches

- **ISI**: extending work of Yamada/Knight
  - more complex rules
  - performance approaching phrase-based

- **Prague**: Translation via *dependency structures*
  - parallel Czech–English dependency treebank
  - tecto-grammatical translation model [EACL 2003]

- **U.Alberta/Microsoft**: *treelet translation*
  - translating from English into foreign languages
  - using dependency parser in English
  - project *dependency tree* into foreign language for training
  - map parts of the dependency tree (“treelets”) into foreign languages
Other Syntax-Based Approaches

- Context feature model for rule selection and reordering
  - SVM for rule selection in hierarchical model [Chan et al., 2007]
  - maximum entropy model for reordering [Xiong et al., 2008; He et al., 2008]

- *Reranking* phrase-based SMT output with syntactic features
  - create n-best list with phrase-based system
  - POS tag and parse candidate translations
  - rerank with syntactic features
  - see [Koehn, 2003] and JHU Workshop [Och et al., 2003]

- JHU Summer workshop 2005
  - *Genpar*: tool for syntax-based SMT
Syntax: Does it help?

- **Getting there**
  - for some languages competitive with best phrase-based systems

- **Some evidence**
  - work on reordering German
  - ISI: better for Chinese–English
  - automatically trained tree transfer systems promising

- Challenges
  - if real syntax, we need *good parsers* — are they good enough?
  - syntactic annotations add a level of *complexity*
    → difficult to handle, slow to train and decode
  - few researchers good at statistical modeling and syntactic theories
Factored Translation Models
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Statistical machine translation today

- Best performing methods based on **phrases**
  - short sequences of words
  - no use of explicit syntactic information
  - no use of morphological information
  - currently best performing method

- Progress in **syntax-based** translation
  - tree transfer models using syntactic annotation
  - still shallow representation of words and non-terminals
  - active research, improving performance
One motivation: morphology

- Models treat *car* and *cars* as completely different words
  - training occurrences of *car* have no effect on learning translation of *cars*
  - if we only see *car*, we do not know how to translate *cars*
  - rich morphology (German, Arabic, Finnish, Czech, ...) → many word forms

- Better approach
  - analyze surface word forms into **lemma** and **morphology**, e.g.: *car + plural*
  - translate lemma and morphology separately
  - generate target surface form
Factored translation models

- **Factored representation** of words

  - **Input**: word, lemma, part-of-speech, morphology, word class
  - **Output**: word, lemma, part-of-speech, morphology, word class

- **Goals**
  - **Generalization**, e.g. by translating lemmas, not surface forms
  - **Richer model**, e.g. using syntax for reordering, language modeling)
Related work

- **Back off** to representations with richer statistics (lemma, etc.)
- Use of additional annotation in **pre-processing** (POS, syntax trees, etc.)
  [Collins et al., 2005, Crego et al, 2006]
- Use of additional annotation in **re-ranking** (morphological features, POS, syntax trees, etc.)
  [Och et al. 2004, Koehn and Knight, 2005]

→ we pursue an **integrated approach**

- Use of syntactic **tree structure**

→ may be **combined** with our approach
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Decomposing translation: example

- **Translate** lemma and syntactic information *separately*

\[
\begin{align*}
\text{lemma} & \Rightarrow \text{lemma} \\
\text{part-of-speech} & \Rightarrow \text{part-of-speech} \\
\text{morphology} & \Rightarrow \text{morphology}
\end{align*}
\]
Decomposing translation: example

- **Generate surface** form on target side

  ![Diagram showing the process of generating a surface form from a lemma, part-of-speech, and morphology.]

Phyllis Koehn  Statistical Machine Translation  28 November 2008
Translation process: example

Input: \((Autos, Auto, NNS)\)

1. Translation step: lemma \(\Rightarrow\) lemma
   \((?, car, ?), (?, auto, ?)\)

2. Generation step: lemma \(\Rightarrow\) part-of-speech
   \((?, car, NN), (?, car, NNS), (?, auto, NN), (?, auto, NNS)\)

3. Translation step: part-of-speech \(\Rightarrow\) part-of-speech
   \((?, car, NN), (?, car, NNS), (?, auto, NNP), (?, auto, NNS)\)

4. Generation step: lemma,part-of-speech \(\Rightarrow\) surface
   \((car, car, NN), (cars, car, NNS), (auto, auto, NN), (autos, auto, NNS)\)
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Model

- Extension of *phrase model*

- Mapping of foreign words into English words broken up into steps
  - **translation step**: maps foreign factors into English factors (on the phrasal level)
  - **generation step**: maps English factors into English factors (for each word)

- Each step is modeled by one or more *feature functions*
  - fits nicely into log-linear model
  - weight set by discriminative training method

- Order of mapping steps is chosen to optimize search
Phrase-based training

- Establish word alignment (GIZA++ and symmetrization)
Phrase-based training

• Extract phrase

⇒ natürlich hat john — naturally john has
Factored training

- Annotate training with factors, extract phrase

⇒ ADV V NNP — ADV NNP V
Training of generation steps

• Generation steps map target factors to target factors
  – typically trained on target side of parallel corpus
  – may be trained on additional monolingual data

• Example:  *The/DET man/NN sleeps/VBZ*
  – count collection
    - count(*the*,DET)++
    - count(*man*,NN)++
    - count(*sleeps*,VBZ)++
  – evidence for probability distributions (max. likelihood estimation)
    - p(DET|*the*), p(*the*|DET)
    - p(NN|*man*), p(*man*|NN)
    - p(VBZ|*sleeps*), p(*sleeps*|VBZ)
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Phrase-based translation

- Task: translate this sentence from German into English

   er geht ja nicht nach hause
Translation step 1

- Task: translate this sentence from German into English

- Pick phrase in input, translate
Translation step 2

- Task: translate this sentence from German into English

```
er geht ja nicht nach hause
```

- Pick phrase in input, translate
  - it is allowed to pick words *out of sequence* (*reordering*)
  - phrases may have multiple words: *many-to-many* translation
Translation step 3

- Task: translate this sentence from German into English

- Pick phrase in input, translate

\[ \text{er} \quad \text{geht} \quad \text{ja} \quad \text{nicht} \quad \text{nach} \quad \text{hause} \]

\[ \text{he} \quad \text{does not} \quad \text{go} \]
Translation step 4

- Task: translate this sentence from German into English

```
er geht ja nicht nach hause
```

- Pick phrase in input, translate

```
he does not go home
```
Translation options

<table>
<thead>
<tr>
<th>er</th>
<th>geht</th>
<th>ja</th>
<th>nicht</th>
<th>nach</th>
<th>hause</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>is</td>
<td>yes</td>
<td>not</td>
<td>after</td>
<td>house</td>
</tr>
<tr>
<td>it</td>
<td>are</td>
<td>is</td>
<td>do not</td>
<td>to</td>
<td>home</td>
</tr>
<tr>
<td>, it</td>
<td>goes</td>
<td>, of course</td>
<td>does not</td>
<td>according to</td>
<td>chamber</td>
</tr>
<tr>
<td>, he</td>
<td>go</td>
<td>is</td>
<td>is not</td>
<td>in</td>
<td>at home</td>
</tr>
<tr>
<td>it is</td>
<td>not</td>
<td>is</td>
<td>not after</td>
<td>to</td>
<td>home</td>
</tr>
<tr>
<td>he will be</td>
<td>is not</td>
<td>does not</td>
<td>not to</td>
<td>following</td>
<td>under house</td>
</tr>
<tr>
<td>it goes</td>
<td>does not</td>
<td>do not</td>
<td></td>
<td></td>
<td>return home</td>
</tr>
<tr>
<td>he goes</td>
<td>is after all</td>
<td>not</td>
<td></td>
<td></td>
<td>do not</td>
</tr>
<tr>
<td>is</td>
<td>are</td>
<td>does</td>
<td>not after</td>
<td>not to</td>
<td></td>
</tr>
<tr>
<td>are</td>
<td>is not</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not</td>
<td>is not</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>are not</td>
<td>is not a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Many translation options to choose from
• The machine translation decoder does not know the right answer

→ *Search problem* solved by heuristic beam search
Decoding process: precompute translation options

er  geht  ja  nicht  nach  hause
Decoding process: start with initial hypothesis

er geht ja nicht nach hause
Decoding process: hypothesis expansion

er geht ja nicht nach hause

are
Decoding process: hypothesis expansion

er  geht  ja  nicht  nach  hause

he
are
it
Decoding process: hypothesis expansion

er geht ja nicht nach hause
are it he goes does not go to home

Philipp Koehn Statistical Machine Translation 28 November 2008
Decoding process: find best path

er geht ja nicht nach hause
Factored model decoding

- Factored model decoding introduces *additional complexity*

- Hypothesis expansion not any more according to simple translation table, but by *executing a number of mapping steps*, e.g.:
  1. translating of *lemma* → *lemma*
  2. translating of *part-of-speech, morphology* → *part-of-speech, morphology*
  3. generation of *surface form*

- Example: *haus|NN|neutral|plural|nominative* → {
  *houses|house|NN|plural, homes|home|NN|plural, buildings|building|NN|plural, shells|shell|NN|plural*
}

- Each time, a hypothesis is expanded, these mapping steps have to applied
Efficient factored model decoding

- Key insight: executing of mapping steps can be *pre-computed* and stored as translation options
  - apply mapping steps to all input phrases
  - store results as *translation options*
→ decoding algorithm *unchanged*
Efficient factored model decoding

• Problem: *Explosion* of translation options
  - originally limited to 20 per input phrase
  - even with simple model, now 1000s of mapping expansions possible

• Solution: *Additional pruning* of translation options
  - *keep only the best* expanded translation options
  - current default 50 per input phrase
  - decoding only about 2-3 times slower than with surface model
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
- Outlook
Adding linguistic markup to output

- Generation of POS tags on the target side
- Use of high order language models over POS (7-gram, 9-gram)
- Motivation: syntactic tags should enforce syntactic sentence structure model not strong enough to support major restructuring
Some experiments

- English–German, Europarl, 30 million word, test2006

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>best published result</td>
<td>18.15</td>
</tr>
<tr>
<td>baseline (surface)</td>
<td>18.04</td>
</tr>
<tr>
<td>surface + POS</td>
<td>18.15</td>
</tr>
</tbody>
</table>

- German–English, News Commentary data (WMT 2007), 1 million word

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
</tr>
</tbody>
</table>

- Improvements under sparse data conditions
- Similar results with CCG supertags [Birch et al., 2007]
Sequence models over morphological tags

- Violation of noun phrase agreement in gender
  - *das schwarze* and *schwarze Himmel* are perfectly fine bigrams
  - but: *das schwarze Himmel* is not

- If relevant n-grams does not occur in the corpus, a lexical n-gram model would fail to detect this mistake

- Morphological sequence model: \( p(N\text{-male}|J\text{-male}) > p(N\text{-male}|J\text{-neutral}) \)
Local agreement (esp. within noun phrases)

- High order language models over POS and morphology

- Motivation
  - \textit{DET-sgl NOUN-sgl} good sequence
  - \textit{DET-sgl NOUN-plural} bad sequence
Agreement within noun phrases

• Experiment: 7-gram POS, morph LM in addition to 3-gram word LM
• Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Agreement errors in NP</th>
<th>devtest</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>15% in NP $\geq$ 3 words</td>
<td>18.22 BLEU</td>
<td>18.04 BLEU</td>
</tr>
<tr>
<td>factored model</td>
<td>4% in NP $\geq$ 3 words</td>
<td>18.25 BLEU</td>
<td>18.22 BLEU</td>
</tr>
</tbody>
</table>

• Example
  – baseline: ... zur zwischenstaatlichen methoden ...
  – factored model: ... zu zwischenstaatlichen methoden ...

• Example
  – baseline: ... das zweite wichtige änderung ...
  – factored model: ... die zweite wichtige änderung ...
Morphological generation model

- Our motivating example

- Translating lemma and morphological information more robust
Initial results

• Results on 1 million word News Commentary corpus (German–English)

<table>
<thead>
<tr>
<th>System</th>
<th>In-doman</th>
<th>Out-of-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
<td>15.01</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
<td>15.03</td>
</tr>
<tr>
<td>Morphgen model</td>
<td>14.38</td>
<td>11.65</td>
</tr>
</tbody>
</table>

• What went wrong?
  – why back-off to lemma, when we know how to translate surface forms?
  → loss of information
Solution: alternative decoding paths

- Allow both surface form translation and morphgen model
  - prefer surface model for known words
  - morphgen model acts as back-off
Results

- Model now beats the baseline:

<table>
<thead>
<tr>
<th>System</th>
<th>In-domain</th>
<th>Out-of-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
<td>15.01</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
<td>15.03</td>
</tr>
<tr>
<td>Morphgen model</td>
<td>14.38</td>
<td>11.65</td>
</tr>
<tr>
<td>Both model paths</td>
<td>19.47</td>
<td>15.23</td>
</tr>
</tbody>
</table>
Adding annotation to the source

• Source words may **lack sufficient information** to map phrases
  
  – English-German: what case for noun phrases?
  – Chinese-English: plural or singular
  – pronoun translation: what do they refer to?

• Idea: **add additional information** to the source that makes the required information available locally (where it is needed)

• see [Avramidis and Koehn, ACL 2008] for details
Case Information for English–Greek

- Detect in English, if noun phrase is subject/object (using parse tree)
- Map information into case morphology of Greek
- Use case morphology to generate correct word form
Obtaining Case Information

- Use syntactic parse of English input
  (method similar to semantic role labeling)
Results English-Greek

- Automatic BLEU scores

<table>
<thead>
<tr>
<th>System</th>
<th>devtest</th>
<th>test07</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>18.13</td>
<td>18.05</td>
</tr>
<tr>
<td>enriched</td>
<td>18.21</td>
<td>18.20</td>
</tr>
</tbody>
</table>

- Improvement in verb inflection

<table>
<thead>
<tr>
<th>System</th>
<th>Verb count</th>
<th>Errors</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>311</td>
<td>19.0%</td>
<td>7.4%</td>
</tr>
<tr>
<td>enriched</td>
<td>294</td>
<td>5.4%</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

- Improvement in noun phrase inflection

<table>
<thead>
<tr>
<th>System</th>
<th>NPs</th>
<th>Errors</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>247</td>
<td>8.1%</td>
<td>3.2%</td>
</tr>
<tr>
<td>enriched</td>
<td>239</td>
<td>5.0%</td>
<td>5.0%</td>
</tr>
</tbody>
</table>

- Also successfully applied to English-Czech
Discriminative Training
Overview

- Evolution from generative to discriminative models
  - IBM Models: purely generative
  - MERT: discriminative training of generative components
  - More features $\rightarrow$ better discriminative training needed

- Perceptron algorithm

- Problem: overfitting

- Problem: matching reference translation
The birth of SMT: generative models

- The definition of translation probability follows a **mathematical derivation**

\[
\arg\max_e p(e|f) = \arg\max_e p(f|e) \ p(e)
\]

- Occasionally, some **independence assumptions** are thrown in for instance IBM Model 1: word translations are independent of each other

\[
p(e|f, a) = \frac{1}{Z} \prod_i p(e_i|f_{a(i)})
\]

- Generative story leads to **straight-forward estimation**
  - maximum likelihood estimation of component probability distribution
  - **EM algorithm** for discovering hidden variables (alignment)
Log-linear models

• IBM Models provided mathematical justification for factoring components together

\[ p_{LM} \times p_{TM} \times p_D \]

• These may be weighted

\[ p^\lambda_{LM} \times p^\lambda_{TM} \times p^\lambda_D \]

• Many components \( p_i \) with weights \( \lambda_i \)

\[ \prod_i p_i^{\lambda_i} = \exp(\sum_i \lambda_i \log(p_i)) \]

\[ \log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i \log(p_i) \]
Knowledge sources

- Many different knowledge sources useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word count
  - phrase count
  - drop word feature
  - phrase pair frequency
  - additional language models
  - additional features
Set feature weights

- Contribution of components $p_i$ determined by weight $\lambda_i$

- Methods
  - *manual setting* of weights: try a few, take best
  - *automate* this process

- Learn weights
  - set aside a *development corpus*
  - set the weights, so that *optimal translation performance* on this development corpus is achieved
  - requires *automatic scoring* method (e.g., BLEU)
Discriminative training

- Model: generate n-best list
- Change feature weights
- Score translations
- Find feature weights that move up good translations

Philipp Koehn
Statistical Machine Translation
28 November 2008
Discriminative vs. generative models

• Generative models
  – translation process is broken down to steps
  – each step is modeled by a probability distribution
  – each probability distribution is estimated from the data by maximum likelihood

• Discriminative models
  – model consist of a number of features (e.g. the language model score)
  – each feature has a weight, measuring its value for judging a translation as correct
  – feature weights are optimized on development data, so that the system output matches correct translations as close as possible
Discriminative training

• Training set (development set)
  – different from original training set
  – small (maybe 1000 sentences)
  – must be different from test set

• Current model translates this development set
  – n-best list of translations (n=100, 10000)
  – translations in n-best list can be scored

• Feature weights are adjusted

• N-Best list generation and feature weight adjustment repeated for a number of iterations
Learning task

- Task: *find weights*, so that feature vector of the correct translations *ranked first*

<table>
<thead>
<tr>
<th>TRANSLATION</th>
<th>IM</th>
<th>TM</th>
<th>WP</th>
<th>SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mary not give slap witch green .</td>
<td>-17.2</td>
<td>-5.2</td>
<td>-7</td>
<td>1</td>
</tr>
<tr>
<td>2 Mary not slap the witch green .</td>
<td>-16.3</td>
<td>-5.7</td>
<td>-7</td>
<td>1</td>
</tr>
<tr>
<td>3 Mary not give slap of the green witch .</td>
<td>-18.1</td>
<td>-4.9</td>
<td>-9</td>
<td>1</td>
</tr>
<tr>
<td>4 Mary not give of green witch .</td>
<td>-16.5</td>
<td>-5.1</td>
<td>-8</td>
<td>1</td>
</tr>
<tr>
<td>5 Mary did not slap the witch green .</td>
<td>-20.1</td>
<td>-4.7</td>
<td>-8</td>
<td>1</td>
</tr>
<tr>
<td>6 Mary did not slap green witch .</td>
<td>-15.5</td>
<td>-3.2</td>
<td>-7</td>
<td>1</td>
</tr>
<tr>
<td>7 Mary not slap of the witch green .</td>
<td>-19.2</td>
<td>-5.3</td>
<td>-8</td>
<td>1</td>
</tr>
<tr>
<td>8 Mary did not give slap of witch green .</td>
<td>-23.2</td>
<td>-5.0</td>
<td>-9</td>
<td>1</td>
</tr>
<tr>
<td>9 Mary did not give slap of the green witch .</td>
<td>-21.8</td>
<td>-4.4</td>
<td>-10</td>
<td>1</td>
</tr>
<tr>
<td>10 Mary did slap the witch green .</td>
<td>-15.5</td>
<td>-6.9</td>
<td>-7</td>
<td>1</td>
</tr>
<tr>
<td>11 Mary did not slap the green witch .</td>
<td>-17.4</td>
<td>-5.3</td>
<td>-8</td>
<td>0</td>
</tr>
<tr>
<td>12 Mary did slap witch green .</td>
<td>-16.9</td>
<td>-6.9</td>
<td>-6</td>
<td>1</td>
</tr>
<tr>
<td>13 Mary did slap the green witch .</td>
<td>-14.3</td>
<td>-7.1</td>
<td>-7</td>
<td>1</td>
</tr>
<tr>
<td>14 Mary did not slap the of green witch .</td>
<td>-24.2</td>
<td>-5.3</td>
<td>-9</td>
<td>1</td>
</tr>
<tr>
<td>15 Mary did not give slap the witch green .</td>
<td>-25.2</td>
<td>-5.5</td>
<td>-9</td>
<td>1</td>
</tr>
</tbody>
</table>

**rank translation**

**feature vector**
Och’s minimum error rate training (MERT)

- **Line search** for best feature weights

```plaintext
given: sentences with n-best list of translations
iterate n times
    randomize starting feature weights
    iterate until convergences
    for each feature
        find best feature weight
        update if different from current
return best feature weights found in any iteration
```
Methods to adjust feature weights

- **Maximum entropy** [Och and Ney, ACL2002]
  - match *expectation* of feature values of model and data

- **Minimum error rate** training [Och, ACL2003]
  - try to *rank best translations first* in n-best list
  - can be adapted for various error metrics, even BLEU

- **Ordinal regression** [Shen et al., NAACL2004]
  - *separate* $k$ worst from the $k$ best translations
BLEU error surface

- Varying one parameter: a rugged line with many local optima
### Unstable outcomes: weights vary

<table>
<thead>
<tr>
<th>component</th>
<th>run 1</th>
<th>run 2</th>
<th>run 3</th>
<th>run 4</th>
<th>run 5</th>
<th>run 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance</td>
<td>0.059531</td>
<td>0.071025</td>
<td>0.069061</td>
<td>0.120828</td>
<td>0.120828</td>
<td>0.072891</td>
</tr>
<tr>
<td>lexdist 1</td>
<td>0.093565</td>
<td>0.044724</td>
<td>0.097312</td>
<td>0.108922</td>
<td>0.108922</td>
<td>0.062848</td>
</tr>
<tr>
<td>lexdist 2</td>
<td>0.021165</td>
<td>0.008882</td>
<td>0.008607</td>
<td>0.013950</td>
<td>0.013950</td>
<td>0.030890</td>
</tr>
<tr>
<td>lexdist 3</td>
<td>0.083298</td>
<td>0.049741</td>
<td>0.024822</td>
<td>-0.000598</td>
<td>-0.000598</td>
<td>0.023018</td>
</tr>
<tr>
<td>lexdist 4</td>
<td>0.051842</td>
<td>0.108107</td>
<td>0.090298</td>
<td>0.111243</td>
<td>0.111243</td>
<td>0.047508</td>
</tr>
<tr>
<td>lexdist 5</td>
<td>0.043290</td>
<td>0.047801</td>
<td>0.020211</td>
<td>0.028672</td>
<td>0.028672</td>
<td>0.050748</td>
</tr>
<tr>
<td>lexdist 6</td>
<td>0.083848</td>
<td>0.056161</td>
<td>0.103767</td>
<td>0.032869</td>
<td>0.032869</td>
<td>0.050240</td>
</tr>
<tr>
<td>lm 1</td>
<td>0.042750</td>
<td>0.056124</td>
<td>0.052090</td>
<td>0.049561</td>
<td>0.049561</td>
<td>0.059518</td>
</tr>
<tr>
<td>lm 2</td>
<td>0.019881</td>
<td>0.012075</td>
<td>0.022896</td>
<td>0.035769</td>
<td>0.035769</td>
<td>0.026414</td>
</tr>
<tr>
<td>lm 3</td>
<td>0.059497</td>
<td>0.054580</td>
<td>0.044363</td>
<td>0.048321</td>
<td>0.048321</td>
<td>0.056282</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.052111</td>
<td>0.045096</td>
<td>0.046655</td>
<td>0.054519</td>
<td>0.054519</td>
<td>0.046538</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.052888</td>
<td>0.036831</td>
<td>0.040820</td>
<td>0.058003</td>
<td>0.058003</td>
<td>0.066308</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.042151</td>
<td>0.066256</td>
<td>0.043265</td>
<td>0.047271</td>
<td>0.047271</td>
<td>0.052853</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.034067</td>
<td>0.031048</td>
<td>0.050794</td>
<td>0.037589</td>
<td>0.037589</td>
<td>0.031939</td>
</tr>
<tr>
<td>phrase-pen.</td>
<td>0.059151</td>
<td>0.062019</td>
<td>-0.037950</td>
<td>0.023414</td>
<td>0.023414</td>
<td>-0.069425</td>
</tr>
<tr>
<td>word-pen</td>
<td>-0.200963</td>
<td>-0.249531</td>
<td>-0.247089</td>
<td>-0.228469</td>
<td>-0.228469</td>
<td>-0.252579</td>
</tr>
</tbody>
</table>
Unstable outcomes: scores vary

- Even different scores with different runs (varying 0.40 on dev, 0.89 on test)

<table>
<thead>
<tr>
<th>run</th>
<th>iterations</th>
<th>dev score</th>
<th>test score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>50.16</td>
<td>51.99</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>50.26</td>
<td>51.78</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>50.13</td>
<td>51.59</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>50.10</td>
<td>51.20</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>50.16</td>
<td>51.43</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>50.02</td>
<td>51.66</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>50.25</td>
<td>51.10</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>50.21</td>
<td>51.32</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>50.42</td>
<td>51.79</td>
</tr>
</tbody>
</table>
More features: more components

• We would like to add more components to our model
  – multiple language models
  – domain adaptation features
  – various special handling features
  – using linguistic information

→ MERT becomes even less reliable
  – runs many more iterations
  – fails more frequently
More features: factored models

- Factored translation models break up phrase mapping into smaller steps
  - multiple translation tables
  - multiple generation tables
  - multiple language models and sequence models on factors

→ Many more features
Millions of features

- Why **mix** of discriminative training and generative models?

- Discriminative training of all components
  - phrase table [Liang et al., 2006]
  - language model [Roark et al, 2004]
  - additional features

- **Large-scale** discriminative training
  - millions of features
  - training of full training set, not just a small development corpus
Perceptron algorithm

- Translate each sentence
- If no match with reference translation: update features

set all lambda = 0
do until convergence
for all foreign sentences f
set e-best to best translation according to model
set e-ref to reference translation
if e-best != e-ref
for all features feature-i
lambda-i += feature-i(f,e-ref)
lambda-i -= feature-i(f,e-best)
Problem: overfitting

• Fundamental problem in machine learning
  – what works best for training data, may not work well in general
  – rare, unrepresentative features may get too much weight

• Especially severe problem in phrase-based models
  – long phrase pairs explain well individual sentences
  – ... but are less general, suspect to noise
  – EM training of phrase models [Marcu and Wong, 2002] has same problem
Solutions

• **Restrict to short phrases**, e.g., maximum 3 words (current approach)
  
  – limits the power of phrase-based models
  – … but not very much [Koehn et al, 2003]

• **Jackknife**
  
  – collect phrase pairs from one part of corpus
  – optimize their feature weights on another part

• IBM direct model: **only one-to-many** phrases [Ittycheriah and Salim Roukos, 2007]
Problem: reference translation

- Reference translation may be anywhere in this box

- If produceable by model \(\rightarrow\) we can compute feature scores
- If not \(\rightarrow\) we can not
Some solutions

- **Skip sentences**, for which reference can not be produced
  - invalidates large amounts of training data
  - biases model to shorter sentences

- Declare candidate translations closest to reference as **surrogate**
  - closeness measured for instance by smoothed BLEU score
  - may be not a very good translation: odd feature values, training is severely distorted
Better solution: early updating?

- At some point the reference translation **falls out** of the search space
  - for instance, due to *unknown words*:

  | Reference: | The group attended the meeting in Najaf ... |
  | System:    | The group meeting was attended in UNKNOWN ... |

  only update features involved in this part

- Early updating [Collins et al., 2005]:
  - stop search, when reference translation is not covered by model
  - only update **features involved in partial** reference / system output
Conclusions

• Currently have proof-of-concept implementation

• Future work: Overcome various technical challenges
  – reference translation may not be produceable
  – overfitting
  – mix of binary and real-valued features
  – scaling up

• More and more features are unavoidable, let’s deal with them