Moses
Machine Translation with Open Source Software

Philipp Koehn and Hieu Hoang
4 November 2010

Outline

09:30-10:00 Introduction
10:00-11:00 Hands-on Session — you will need a laptop
11:00-11:30 Break
11:30-12:30 Advanced Topics
Basic Idea

Training
- Training Data
- Linguistic Tools
- Parallel corpora
- Monolingual corpora
- Dictionaries

Statistical Machine Translation System

Using
- Source Text
- Statistical Machine Translation System
- Translation

Statistical Machine Translation History

around 1990
- Pioneering work at IBM, inspired by success in speech recognition

1990s
- Dominance of IBM's word-based models, support technologies

early 2000s
- Phrase-based models

late 2000s
- Tree-based models
Moses History

2002
Pharaoh decoder, precursor to Moses (phrase-based models)

2005
Moses started by Hieu Hoang and Philipp Koehn (factored models)

2006
JHU workshop extends Moses significantly

since late 2006
Funding by EU projects EuroMatrix, EuroMatrixPlus

2009
Tree-based models implemented in Moses

Moses in Academia

• Built by academics, for academics

• Reference implementation of state of the art
  – researchers develop new methods on top of Moses
  – developers re-implement published methods
  – used by other researchers as black box

• Baseline to beat
  – researchers compare their method against Moses
Developer Community

- Main development at University of Edinburgh, but also:
  - Fondazione Bruno Kessler (Italy)
  - Charles University (Czech Republic)
  - DFKI (Germany)
  - RWTH Aachen (Germany)
  - others...
- Code shared on Sourceforge
- Main forum: support and developer mailing lists
- Main event: Machine Translation Marathon (next: September 2011, Trento)
  - annual open source convention
  - presentation of new open source tools
  - hands-on work on new open source projects
  - summer school for statistical machine translation

Open Source Components

- Moses distribution uses external open source tools
  - word alignment: GIZA++, Berkeley aligner
  - language model: SRILM, IRSTLM, RANDLM
  - scoring: BLEU, TER, METEOR
- Other useful tools
  - sentence aligner
  - syntactic parsers
  - part-of-speech taggers
  - morphological analyzers
Other Open Source MT Systems

- **Joshua** — Johns Hopkins University
  http://joshua.sourceforge.net/

- **CDec** — University of Maryland
  http://cdec-decoder.org/

- **Jane** — RWTH Aachen
  http://www-i6.informatik.rwth-aachen.de/jane/

- Very similar technology
  - Joshua implemented in Java, others in C++
  - Joshua and Jane support only tree-based models
  - Phrasal supports only phrase-based models

- Open sourcing tools increasing trend in NLP research

---

Moses in Industry

- Distributed with LGPL — free to use

- Competitive with commercial SMT solutions
  (Language Weaver, Google, ...)

- But:
  - not easy to use
  - requires significant expertise for optimal performance
  - integration into existing workflow not straight-forward
Case Studies

European Commission —
uses Moses in-house to aid human translators

Autodesk —
showed productivity increases in translating manuals when post-editing output from a custom-build Moses system

Systran —
developed statistical post-editing using Moses

Asia Online —
offers translation technology and services based on Moses

Pangea —
language service provider builds Moses systems for its data

Phrase-Based Model

- Foreign input is segmented in phrases
- Each phrase is translated into English
- Phrases are reordered
Phrase Translation Options

- Many translation options to choose from

Phrase Translation Options

- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order
  ➔ Search problem solved by heuristic beam search
Decoding: Precompute Translation Options

consult phrase translation table for all input phrases

Decoding: Start with Initial Hypothesis

initial hypothesis: no input words covered, no output produced
Decoding: Hypothesis Expansion

er geht ja nicht nach house

pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion

he are it

create hypotheses for all other translation options
Decoding: Hypothesis Expansion

Decoding: Find Best Path

also create hypotheses from created partial hypothesis

backtrack from highest scoring complete hypothesis
Computational Complexity

- The suggested process creates exponential number of hypothesis
- Reduction of search space: pruning
  → Decoder may not find the model-best translation

---

Factored Representation

- Factored representation of words

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>word</td>
</tr>
<tr>
<td>lemma</td>
<td>lemma</td>
</tr>
<tr>
<td>part-of-speech</td>
<td>part-of-speech</td>
</tr>
<tr>
<td>morphology</td>
<td>morphology</td>
</tr>
<tr>
<td>word class</td>
<td>word class</td>
</tr>
</tbody>
</table>

- Goals
  - generalization, e.g. by translating lemmas, not surface forms
  - richer model, e.g. using syntax for reordering, language modeling)
Factored Model

Example:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>word</td>
</tr>
<tr>
<td>lemma</td>
<td>lemma</td>
</tr>
<tr>
<td>part-of-speech</td>
<td>part-of-speech</td>
</tr>
<tr>
<td></td>
<td>morphology</td>
</tr>
</tbody>
</table>

Decomposing the translation step
Translating lemma and morphological information more robust

Syntax Models

**String to String**
John misses Mary
⇒ Marie manque à Jean

**String to Tree**
John misses Mary
⇒ S
  NP
  Marie
  V
  manque
  VP
  à
  NP
  Jean

**Tree to String**
S
NP
V
NP
John
misses
Mary
⇒ Marie manque à Jean

**Tree to Tree**
S
NP
V
NP
John
misses
Mary
⇒ Marie manque à Jean
Syntax Decoding

1. PRO
   she

2.>NN
coffee

3.>NN
trinken

Sie
PER
will
VFIN

Trinke
NP

Trinke
NN

Kaffee
NN

Syntax Decoding

1. PRO
   she

2.>NN
coffee

3.>NN
drink

Sie
PER
will
VFIN

Trinke
NP

Trinke
NN

Kaffee
NN
Advanced Topics

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system
Hands-On Session

Advanced Topics
Advanced Features

- **Data and domain adaptation**
  - Speed vs. quality
  - Speed vs. memory use
  - Language models
  - Instructions to decoder
  - Input formats
  - Output formats
  - Minimum Bayes risk decoding
  - Translation models
  - Experiment management system

Data

- Parallel corpora → translation model
  - sentence-aligned translated texts
  - translation memories are parallel corpora
  - dictionaries are parallel corpora

- Monolingual corpora → language model
  - text in the target language
  - billions of words easy to handle
Domain Adaptation

- The more data, the better
- The more in-domain data, the better
  (even in-domain monolingual data very valuable)
- Multiple models
  - train a translation model for each domain corpus
  - train a language model for each domain corpus
  - use all, tune weights for each model
  - alternative: interpolate language model
- Always tune towards target domain

Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Experiment management system
Speed

- Easy speed-up: multi-threaded decoding

```
--threads NUM
```

- Requires boost library

- Does not currently work for:
  - syntax-based decoding
  - IRSTLM
  - randLM

Speed vs. Quality

- Decoder search creates very large number of partial translations ("hypotheses")
- Decoding time $\sim$ number of hypotheses created
- Translation quality $\sim$ number of hypothesis created
Hypothesis Stacks

- Phrase-based: One stack per number of input words covered
- Number of hypothesis created = sentence length \times stack size \times applicable translation options

Pruning Parameters

- Regular beam search
  - `--stack NUM` max. number of hypotheses contained in each stack
  - `--table-limit NUM` max. num. of translation options per input phrase
  - search time roughly linear with respect to each number

- Cube pruning
  (fixed number of hypotheses are added to each stack)
  - `--search-algorithm 1` turns on cube pruning
  - `--cube-pruning-pop-limit NUM` number of hypotheses added to each stack
  - search time roughly linear with respect to pop limit
  - note: stack size and translation table limit have little impact in speed
Syntax Hypothesis Stacks

- One stack per input word span
- Number of hypothesis created = sentence length$^2 \times$ number of hypotheses added to each stack

Trade-Off Speed vs Quality

optimum for practical use

optimum for winning competitions
Advanced Features

- Data and domain adaptation
- Speed vs. quality
- **Speed vs. memory use**
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system

Speed vs. Memory Use

![Diagram showing process size (RAM) with Language model, Translation model, and Working memory organized with more efficient representation of language model using cache and disk storage.]
**Speed vs. Memory Use**

Typical Europarl file sizes:
- Language model
  - 170 MB (trigram)
  - 412 MB (5-gram)
- Phrase table
  - 11GB
- Lexicalized reordering
  - 9.4GB
→ total = 20.8 GB

**Speed vs. Memory Use**

- Load into memory
  - fast decoding
  - large memory usage
  - large load time

- Load-on-demand
  - store indexed model on disk
  - binary format
  - minimal start-up time, memory usage
  - slower decoding
Speed vs. Memory Use

Phrase Table:

Phrase-based
export LC_ALL=C
cat pt.txt | sort | ./processPhraseTable -ttable 0 0 - \ -nscores 5 -out out.file

Hierarchical / Syntax
export LC_ALL=C
./CreateOnDiskPt 1 1 5 100 2 pt.txt out.folder

Lexical Reordering Table:

export LC_ALL=C
processLexicalTable -in r-t.txt -out out.file

Language Models (later)

---

Speed vs. Memory Use

Change ini file

Phrase-based
[ttable-file]
1 0 0 5 out.file

Hierarchical / Syntax
[ttable-file]
2 0 0 5 out.folder

Lexical Reordering Table
[distortion-file]
0-0 wbe-msd-bidirectional-fe-allff 6 out.file
Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- **Language models**
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system

Language Models

- Probability of the output
- Very important in MT, for all SMT models → improve fluency
- Huge amount of training data easy to obtain
  - monolingual
  - can scrape from websites etc.
- But:
  - training takes a long time
  - large memory requirement during decoding
  - large load time
- IRSTLM and RandLM especially designed to tackle large data issues
- Developed by FBK-irst, Trento, Italy
- Create a binary format which can be read from disk as needed
  - reduces memory but slower decoding
- Quantization of probabilities
  - reduces memory but lose accuracy
  - probability stored in 1 byte instead of 4 bytes

IRSTLM in Moses

- Compile the decoder with IRSTLM library
  ```bash
  ./configure --with-irstlm=[root dir of the IRSTLM toolkit]
  ```
- Change ini file to use IRSTLM implementation
  ```
  [lmmodel-file]
  1 0 3 file/path
  ```
IRSTLM: Training

- Specialized training for large corpora
  - parallelization
  - reduce memory usage

- Training:
  ```
  build-lm.sh -i "gunzip -c corpus.gz" -n 3
  -o train.irstlm.gz -k 10
  ```
  - `-n 3` = n-gram order
  - `-k 10` = split training procedure into 10 steps

IRSTLM: Binary Format

- Create binary format:
  ```
  compile-lm language-model.srilm language-model.blm
  ```

- Load-on-demand:
  ```
  rename file .mm
  ```
Randomized language model

- For huge corpora (e.g. 100 billion words)

- Lossy compression
  - Makes false positive mistakes
  - frequency of mistakes can be varied with a parameter

- Typically \( \frac{1}{10} \) size of SRI / IRST language model

- Maybe use as secondary LM to complement conventional LM
  - out-of-domain data scraped from the web
  - high-order n-gram, eg. 6-7 gram

RandLM: Use in Moses

- Compile the decoder with RandLM library
  
  ```
  ./configure --with-randlm=[root dir of the RandLM toolkit]
  ```

- Change ini file to use RandLM implementation

  ```
  [lm0_model-file]
  0 0 3 /path/to/file  # conventional lm
  5 0 7 /path/to/file  # rand lm
  ```
RandLM: Training

- Train from text corpus
  
  ```
  ./buildlm -struct BloomMap -falsepos 8 -values 8 -order 3
  -output-prefix model
  < corpus.txt
  ```

- Convert SRILM language model
  
  ```
  ./buildlm -struct BloomMap -falsepos 8 -values 8 -order 3
  -output-prefix model
  -input-path lm.srilm -input-type arpa
  ```

Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- **Instructions to decoder**
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system
Specifying Translations with XML

- Translation tables for numbers?

| f    | e      | p(f|e) |
|------|--------|-------|
| 2003 | 2003   | 0.7432|
| 2003 | 2000   | 0.0421|
| 2003 | year   | 0.0212|
| 2003 | the    | 0.0175|
| 2003 | ...    | ...   |

- Instruct the decoder with XML instruction

  \[
  \text{the revenue for } \texttt{<num translation="2003">} 2003 \texttt{</num> is higher than } ...
  \]

- Deal with different number formats

  \[
  \text{er erzielte } \texttt{<num translation="17.55">} 17.55 \texttt{</num> Punkte .}
  \]

Walls and Zones

- Specification of reordering constraints

- Zone
  
  sequence to be translated without reordering with outside material

- Wall
  
  hard reordering constraint, no words may be reordered across

- Local wall
  
  wall within a zone, not valid outside zone
Walls and Zones: Examples

- Requiring the translation of quoted material as a block
  
  He said `<zone>"yes"</zone>`.

- Hard reordering constraint
  
  Number 1: `<wall/> the beginning`.

- Local hard reordering constraint within zone
  
  A new plan `<zone>(<wall/> maybe not new <wall/>)</zone>` emerged.

- Nesting
  
  The `<zone>"new <zone>(old)"</zone>` proposal.

Preserving Markup

- How do you translate this:

  `<h1>My Home Page</h1>
  I really like to `<b>eat</b>` chicken!`

- Solution 1: XML translations, walls and zones

  `<x translation="<h1>"/> <wall/> My Home Page <wall/>
  <x translation="</h1>"/>

  I really like to `<zone>`<x translation="<b>"/> <wall/> eat <wall/>
  <x translation="</b>"/> </zone>` chicken!

  (note: special XML characters like `<` and `>` need to be escaped)
Preserving Markup

- Solution 2: Handle markup externally
  - track word positions and their markup
    
    
    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
    |---|---|---|---|---|---|---|
    |   |   |   | <b>eat</b> |   |   |   |

  - translate without markup
    
    I really like to eat chicken !

  - keep word alignment to source
    
    | 1 | 5 | 2 | 3-4 | 6 | 7 |
    |---|---|---|-----|---|---|
    | Ich | esse | wirklich | gerne | Hühnchen | ! |

  - re-insert markup
    
    Ich <b>esse</b> wirklich gerne Hühnchen!

Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- **Input formats**
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system
**Example: Misspelt Words**

- Misspelt sentence:
  
The room was *exellent but the hallway was *filty .

- Strategies for dealing with spelling errors:
  
  - Create correct sentence with correction
    × problem: if not corrected properly, adds more errors
  
  - Create many sentences with different corrections
    × problem: have to decode each sentence, slow

**Confusion Network**

The room was *exellent but the hallway was *filty .

**Input to decoder:**

Let the decoder decide
Example: Diacritics

- Correct sentence
  Trung Quốc cảnh báo Mỹ về luật tiền tệ

- Something a non-native person might type
  Trung Quoc canh bao My ve luat tien te

- Confusion network

![Confusion Network Diagram]

Confusion Network Specification

Argument on command line

```
./moses -inputtype 1
```

Input to moses

```
the 1.0
room 1.0
was 1.0
excel 0.33 excellent 0.33 excellence 0.33
but 1.0
the 1.0
hallway 1.0
was 1.0
guilty 0.5 filthy 0.5
```
Example: Chinese Word Segmentation

- Unsegmented sentence

  硬质合金号称"工业牙齿"

- Incorrect segmentation

  硬质合金号称"工业牙齿"

- Correct segmentation

  硬质合金号称"工业牙齿"

---

**Lattice**

**Input to decoder:**

Let the decoder decide.

---
Example: Compound Splitting

- Input sentence
  
  einen wettbewerbsbedingten preissturz

- Different compound splits

```
0  einer  [wettbewerb] wettbewerbs
    [bedingten]  [preis] sturz
```

- Let the decoder decide

Lattice Specification

Command line argument

```
./moses -inputtype 1
```

Input to Moses (PLF format - Python Lattice Format)

```
(
  ('einer', 1.0, 1),
  ('wettbewerbsbedingten', 0.5, 2),
  ('wettbewerbs', 0.25, 1),
  ('wettbewerbs', 0.25, 1),
  ('bedingten', 1.0, 1),
  ('preissturz', 0.5, 2),
  ('preis', 0.5, 1),
  ('sturz', 1.0, 1),
)
```

Koehn and Hoang  Machine Translation with Open Source Software  4 November 2010
Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- **Output formats**
- Minimum Bayes risk decoding
- Translation models
- Experiment management system

N-Best List

- **Input**
  
es gibt verschiedene andere Meinungen .

- **Best Translation**
  
  there are various different opinions .

- **Next nine best translations**

  there are various other opinions .
  there are different different opinions .
  there are other different opinions .
  we are various different opinions .
  there are various other opinions of .
  it is various different opinions .
  there are different other opinions .
  it is various other opinions .
  it is a different opinions .
Uses of N-Best Lists

- Let the translator choose from possible translations
- Reranker
  - add more knowledge sources
  - can take global view
  - coherency of whole sentence
  - coherency of document
- Used to tune component weights

N-Best Lists in Moses

Argument to command line

```
./moses -n-bestlist n-best.file.txt [distinct] 100
```

Output

```
0     there are various different opinions .  ||| d: 0 lm: -21.6694 w: -6 ...  ||| -113.734
0     there are various other opinions .  ||| d: 0 lm: -25.3276 w: -6 ...  ||| -114.904
0     there are different different opinions .  ||| d: 0 lm: -27.8429 w: -6 ...  ||| -117.738
0     there are other different opinions .  ||| d: -4 lm: -25.1655 w: -6 ...  ||| -118.007
0     we are various different opinions .  ||| d: 0 lm: -28.1533 w: -6 ...  ||| -118.124
0     there are various other opinions of .  ||| d: 0 lm: -33.7616 w: -7 ...  ||| -118.153
0     it is various different opinions .  ||| d: 0 lm: -29.8191 w: -6 ...  ||| -118.222
0     there are different other opinions .  ||| d: 0 lm: -39.426 w: -6 ...  ||| -118.236
0     it is various other opinions .  ||| d: 0 lm: -32.6824 w: -6 ...  ||| -118.395
0     it is a different opinions .  ||| d: 0 lm: -20.1611 w: -6 ...  ||| -118.434
```
Search Graph

- Input
  
er geht ja nicht nach hause

- Return internal structure from the decoder

- Encode millions of other possible translations (every path through the graph = 1 translation)

Uses of Search Graphs

- Let the translator choose
  
  - Individual words or phrases
  
  - ‘Suggest’ next phrase

- Reranker

- Used to tune component weights

  - More difficult than with n-best list
Search Graphs in Moses

Argument to command line

```
./moses -output-search-graph search-graph.file.txt
```

Argument to command line

```
0 hyp=0 stack=0 forward=36 fscore=-113.734
0 hyp=75 stack=1 back=0 score=-104.943 ... covered=5-5 out=.
0 hyp=72 stack=1 back=0 score=-8.846 ... covered=4-4 out=opinions
0 hyp=73 stack=1 back=0 score=-10.561 ... covered=4-4 out=opinions of
```

- hyp - hypothesis id
- stack - how many words have been translated
- score - total weighted score
- covered - which words were translated by this hypothesis
- out - target phrase

---

Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- Experiment management system
Minimum Bayes Risk Decoding

- Normal (MAP) decoding:
  \[ \hat{t} = \arg\max_t p(t|s) \]

- MBR decoding:
  \[ \hat{t} = \arg\max_t T \sum_{t' \in T} p(t'|s) \times \text{bleu}(t', t) \]

Koehn and Hoang  Machine Translation with Open Source Software  4 November 2010

Minimum Bayes Risk Decoding

- Set of translations \( t' \in T \)
  \[ \hat{t} = \arg\max_t \sum_{t' \in T} p(t'|s) \times \text{bleu}(t', t) \]

- Using n-best list:
  \texttt{moses -f moses.ini -i in.txt -mbr}

- Using lattice:
  \texttt{lmbrgrid ... -f moses.ini -i input.txt}
Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- **Translation models**
- Experiment management system

Phrase-Based Model

- Advantages
  - fast: under half a second per sentence for fast configuration
  - low-memory requirement
    * 200-300MB for lowest configuration
    * suitable for netbooks and mobile devices
  - outperform more complicated models for many language pairs
    * especially for related languages pairs

- Command line
  ```
  ./moses -f moses.ini -i in.txt > out.txt
  ```

- Output
  ```
  there are various different opinions .
  ```
Hierarchical Models

Advantages

- able to model non-contiguous phrases
  - ne...pas → not
- low-memory requirement
  - 200-300MB for lowest configuration
  - suitable for netbooks and mobile devices
- outperform phrase-based models when translating between widely different languages
  - Chinese-English consistently better with hierarchical model
  - better at medium range re-ordering
- Linguistically motivated

Disadvantages

- slower
  - 0.5 - 2 sec for fastest configuration
- more memory requirement
  - 1-2GB ram
- more disk usage
  - translation model ×10 larger than phrase-based

Command line

```
./moses-chart -f moses.ini -i in.txt > out.txt
```

Syntax Models

- Hierarchical model + use of syntactic information
  (constituency parser, chunkers)

- Advantage
  - Can use outside linguistic information
  - promises to solve important problems in SMT, eg. long-range reordering

- Disadvantages
  - difficult to get right
  - for many language pairs still worse than phrase-based and hierarchical models
  - need syntactic parse information
    - unreliable
    - available only for some languages
    - not designed for machine translation
Moses Tree Representation

```
TOP
  NP  PUNC
    NE  ADJA  NN  ?
  Musharraf  letzter  Akt

<tree label="TOP">
  <tree label="NP">
    <tree label="NE">Musharraf</tree>
    <tree label="ADJA">letzter</tree>
    <tree label="NN">Akt</tree>
  </tree>
  <tree label="PUNC">?</tree>
</tree>
```

Phrase-Based Model Training

- Command line

```
train-model.perl ...
```

- Model

```
Bliniere || alliances || 1 1 1 1 2.718 ||| 1 1
General Musharraf betrat am || general Musharraf appeared on ||| 1 1 1 1 2.718 ||| 1 1
```
Hierarchical Model Training

- Command line
  
  `train-model.perl ... -hierarchical`

- Example rule from model
  
  Bündnisse [X][X] Kräften [X] ||| alliances [X][X] forces [X] ||| 1 1 1 2.718 ||| 1-1 ||| 0.0526316 0.0526316

- Visualization of rule

  ![Visualization of rule](image)

Tree-to-String Model Training

- Command line
  
  `train-model.perl ... -source-syntax`

- Example rule from model
  
  Bündnisse [PP][X] [NP] ||| alliances [PP][X] [X] ||| 1 1 1 2.718 ||| 1-1 ||| 1 1

- Visualization of rule

  ![Visualization of rule](image)
String to Tree Model Training

- Command line
  
  \texttt{train-model.perl ... -target-syntax}

- Example rule from model
  
  \texttt{von [X][NPB] und [X][NPB] [X] \rightarrow with [X][NPB] and [X][NPB] [PP] \rightarrow ...}

- Visualization of rule
  
  \[
  \begin{array}{c}
  X \\
  \downarrow_{\text{von}}
  \end{array}
  \begin{array}{c}
  X_1 \\
  \downarrow_{\text{X}_2}
  \end{array}
  \rightarrow
  
  \begin{array}{c}
  PP \\
  \downarrow_{\text{with}}
  \end{array}
  \begin{array}{c}
  \text{NPB}_1 \\
  \downarrow_{\text{and}}
  \end{array}
  \begin{array}{c}
  \text{NPB}_2
  \end{array}
  \]

---

Tree-to-Tree Model Training

- Command line
  
  \texttt{train-model.perl ... -source-syntax -target-syntax}

- Example rule from model
  
  \texttt{seine Stellung und Maßnahmen [CNP] \rightarrow his position and actions [NPB] \rightarrow ...}

- Visualization of rule
  
  \[
  \begin{array}{c}
  \text{CNP} \\
  \downarrow_{\text{seine}}
  \end{array}
  \begin{array}{c}
  \text{Stellung und Massnahmen}
  \end{array}
  \rightarrow
  
  \begin{array}{c}
  \text{NPB} \\
  \downarrow_{\text{his}}
  \end{array}
  \begin{array}{c}
  \text{position and actions}
  \end{array}
  \]
Syntax Models Decoding in Moses

- String-to-string (hierarchical) or string-to-tree

```
./moses-chart -f moses.ini -i in.txt > out.txt
```

- Tree-to-string or tree-to-tree

```
./moses-chart -f moses.ini -i in.txt -inputtype 3 > out.txt
```

Advanced Features

- Data and domain adaptation
- Speed vs. quality
- Speed vs. memory use
- Language models
- Instructions to decoder
- Input formats
- Output formats
- Minimum Bayes risk decoding
- Translation models
- **Experiment management system**
Running Experiments

Execute a lot of scripts

tokenize < corpus.en > corpus.en.tok
lowercase < corpus.en.tok > corpus.en.lc
...
mert.perl ....
moses ...
mteval-v13.pl ...

Change a part of the process, execute everything again

tokenize < corpus.en > corpus.en.tok
lowercase < corpus.en.tok > corpus.en.lc
...
mert.perl ....
moses ...
mteval-v13.pl ...

Experiment Management System

- One configuration file for all settings: record of all experimental details

- Scheduler of individual steps in pipeline
  - automatically keeps track of dependencies
  - on single machine, multi-core machines, GridEngine clusters
  - parallel execution
  - crash detection
  - automatic re-use of prior results

- Fast to use
  - set up a new experiments in minutes
  - set up a variation of an experiment in seconds
How does it work?

- Write a configuration file (typically by adapting an existing file)

- Execute:
  
  ```
  experiment.perl -config config
  ```
Web Interface

All Experimental Setups

<table>
<thead>
<tr>
<th>ID</th>
<th>User</th>
<th>Task</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>pkoehn</td>
<td>Acquis Truecased</td>
<td>/group/project/statmt2/pkoehn/acquis-truecase</td>
</tr>
<tr>
<td>96</td>
<td>pkoehn</td>
<td>Chinese-English AGILE 2008</td>
<td>/group/project/statmt2/pkoehn/agile08-chinese</td>
</tr>
<tr>
<td>95</td>
<td>miles</td>
<td>Random testing</td>
<td>/group/project/statmt7/miles/experiments/ep-enfr/work</td>
</tr>
<tr>
<td>94</td>
<td>joseph</td>
<td>Proj2008 Impl. Adapted experiment (fr-en) for News Comm.</td>
<td>/group/project/statmt2/joseph/experimentJo/task6</td>
</tr>
<tr>
<td>93</td>
<td>joseph</td>
<td>Proj2008 Impl. Baseline experiment (fr-en) for News Comm.</td>
<td>/group/project/statmt2/joseph/experimentJo/task5</td>
</tr>
<tr>
<td>92</td>
<td>jschroe1</td>
<td>FR-EN System Combination Components</td>
<td>/group/project/statmt9/jschroe1/experiments/fr-syscomb/work</td>
</tr>
</tbody>
</table>

List of experiments

Koehn and Hoang

Machine Translation with Open Source Software

4 November 2010

List of Runs

Task: WMT10 German-English (pkoehn)

| Wiki Notes | Overview of experiments | /es/bragi2/pkoehn-experiment/wmt10-de-en |

<table>
<thead>
<tr>
<th>compare</th>
<th>ID</th>
<th>start</th>
<th>end</th>
<th>avg</th>
<th>newtest2009</th>
<th>newtest2010</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16 May</td>
<td>16 May</td>
<td>BLEU-c: 21.74, BLEU: 22.91</td>
<td>21.03 (1.002)</td>
<td>22.90 (1.002)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22.45 (1.041)</td>
<td>23.51 (1.041)</td>
</tr>
<tr>
<td></td>
<td>configparing</td>
<td>11+analysis</td>
<td>11+internal emplas test set</td>
<td>crashed</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>configparing</td>
<td>9+interpolated-tm.lng-weighted</td>
<td>9+only-cp</td>
<td>9+only-nc</td>
<td>18.96 (1.002)</td>
<td>21.19 (1.002)</td>
</tr>
</tbody>
</table>

Koehn and Hoang

Machine Translation with Open Source Software

4 November 2010
Analysis: Basic Statistics

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Phrase Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>corpus</td>
</tr>
<tr>
<td>0</td>
<td>2047 (3.1%)</td>
</tr>
<tr>
<td>1</td>
<td>738 (1.1%)</td>
</tr>
<tr>
<td>2-5</td>
<td>1483 (2.2%)</td>
</tr>
<tr>
<td>6+</td>
<td>61745 (43.5%)</td>
</tr>
<tr>
<td>1</td>
<td>26897 (40.7%)</td>
</tr>
<tr>
<td>2</td>
<td>4144 (6.3%)</td>
</tr>
<tr>
<td>3</td>
<td>639 (1.0%)</td>
</tr>
<tr>
<td>4+</td>
<td>158 (0.2%)</td>
</tr>
</tbody>
</table>

by token / by type /
details

- Basic statistics
  - n-gram precision
  - evaluation metrics
  - coverage of the input in corpus and translation model
  - phrase segmentations used

Analysis: Unknown Words

grouped by frequency in test set

unknown words

18 Eatonville
16 Hurston
12 Barrick
12 Hesse
12 Stewarts
11 Gebrsclassic
10 Flamenco
10 Mango
9 Glitter
9 ŬOHS
9 ČTÚ
8 Coles
8 Deka
8 Garcí
8 ITV
8 Özi
4: Amnil,
3: Abfertigung,
2: Abteilungen,
1: -ach, -minister, -pakets, -weiss, -doc, -zips, -xlsx, -i45,
Arnold, Br23C, Albums, Alondra,
Andeh, Anm, Armínion,
Ashford, BZO, Baloldal,
Bani, Baugesellschaften,
Bento, Bodenkomfort, Bento,
Bentos, Bingleys, Bojen,
Browns, Bowery, Boyd,
Bringley, Browser,
Bélohávé, CBGB,
Hanzelka, Hanze,
Kremer, Kremer,
Lados, Ilhauzem, Iván, Carci, Cera, Charts,
Mundandell, Januña, Chemical, Chugi,
Schpütze, Joanne, Cinest, Contres,
Kemnerá, Kid, Commerzbank, Coppola,
Tolalda, Llamazares, Cortek, Cowon, Df,
Zenobia, Loafs, Mangas, Dinkins, Download,
Son, Medikamentes, Drehbewegung,
Éveredért, Mobiliz, Dezewicki, Dripal,
Özl Mutual, Düsseldorfer, Ella,
Analysis: Output Annotation

This time was the reason for the collapse on Wall Street.
This time the fall in stocks on Wall Street is responsible for the drop.

Color highlighting to indicate n-gram overlap with reference translation

darker bleu = word is part of larger n-gram match

Analysis: Input Annotation

100 occurrences in corpus, 52 distinct translations, translation entropy: 3.08447

diesmal, der Grund lag für den Einbruch an der Wall Street.

- For each word and phrase, color coding and stats on
  - number of occurrences in training corpus
  - number of distinct translations in translation model
  - entropy of conditional translation probability distribution $\phi(e|f)$ (normalized)
Analysis: Bilingual Concordancer

entre autres (560/1554)

...d and trade recommendations, 'inter alia', with respect to the follow...:
...on (EC) No 1995/2000 imposing 'inter alia', a definitive anti-f dumping...:
...devises, this increase, arising 'inter alia', as a result of economic growth...:
...of paragraph 1 the Commission may 'inter alia', bring forward:
...of stocks of absolute pesticides 'inter alia', by supporting projects aimed at s...
...es rules of procedure which shall 'inter alia', contain provision for converting...
...such specific agreements may cover 'inter alia', financing provisions, assignment...
...be internal market and concerning 'inter alia', health and environmental protect...
...e product concerned' originating 'inter alia', in India.

...the EU budget by addressing 'inter alia', the problems of accountability...
...ates, the Commission has adopted 'inter alia', Decision 2003/526 /EC (3) wh...
...ed equitable development involving 'inter alia', access to productive resources...
...ertain products which could be used 'inter alia', as equipment on board ships but w...
...netes, taking into consideration 'inter alia', available scientific, technical...
...so that it is absolutely necessary 'inter alia', because of straightforward, as...:
...paragraphs 1 and 2 as appropriate, 'inter alia', by conducting studies and compil...
...liability and efficiency, caused 'inter alia', by insufficient technical and adm...
...in the Programme shall be pursued 'inter alia', by the following means:

notamment (447/1554)

...get de l'Union, or who pass notamment par la résolution du problème de r...
...est membre, la Commission a notamment arrêté la décision 2003/526/C...
...durable et équitable, impliquant notamment l'accès aux ressources produc...
...susceptible d'être utilisés notamment comme équipements mis à bord, mais:
...ion et à ses annexes, compte tenu notamment des informations scientifiques tec...
...ile, il est absolument nécessaire, notamment en raison de l'étirement ...
...graphes 1 et 2 le cas échéant, notamment en menant des études et en complai...
...et d'efficacité en raison, notamment, d'une interprétabilité too...
...mis dans le programme, il convient notamment de mettre en œuvre les moyens ci @...:

translation of input phrase in training data context

Koehn and Hoang Machine Translation with Open Source Software 4 November 2010

Analysis: Alignment

Phrases alignment of the decoding process
(red border, interactive)
Analysis: Tree Alignment

Uses nested boxes to indicate tree structure
(red border, yellow shaded spans in focus, interactive)
for syntax model, non-terminals are also shown

Analysis: Comparison of 2 Runs

annotated sentences
sorted by order worse display allscreen showing 5 more all
identical same better worse
93% 2% 2% 3%

In Austria, Haider and Co. are ready to govern to prevent a red and black coalition.
Haider and his party are ready to govern in order to avoid red @-@ black coalition.

The SPÖ wants to show that the cooperation of both parties is possible - in some countries and in the social partnership that is already the case.
SPÖ would like to show that the cooperation of the two parties is possible - it does exist in some of the provinces as well as in social partnership.

Different words are highlighted
sortable by most improvement, deterioration
Acknowledgements

Moses Developers

Abhishek Arun
Amplit Agarwal
Barry Haddow
Christian Hardmeier
Edmund Huber
Frederic Blain
Hien Hoang
Jean-Baptiste Fontet
Abby Levenberg
Mauro Cettolo
Mark Fishel
Nicola Bertoldi
Phil Williams
José Luis Rosas
Sara Stymne
Yizhe Ni
Suzy Howlett
Alexander Fraser

Adam Lopez
Ankit Srivastava
Ondřej Bojar
Christian Federmann
Evan Herbst
Brooke Cowan
H. Lea Fontes
Joern Wuebker
Alexandra Breh
Marcello Federico
Gabriele Antonio Musillo
Oliver Wilson
Bruno Pouthier
Rico Sennrich
Steven B. Parks
David Koçurkýn
Wade Shen
Richard Zens

Ales Turchyna
Anthony Rouseau
Chris Callison-Burch
Lane Schwartz
Andreas Eisele
Grace M. Ngai
Holger Schwenk
Jorge Civera
Bo Fu
Michael Auli
Miles Osborne
Pascal Martinez
Raphael Payen
Hervé Saint-Amand
Steven Burjasek Poggio
Sergio Penkale
Yang Gao

Alex
Benjamin Gottesman
Christine Corbett
David Talbot
Eva Hasler
Kenneth Heafield
Josh Schroeder
Konrad Rawlik
M.J. Bellino-Machado
John Joseph Morgan
Nadi Tomin
Philipp Koehn
Chris Dyer
Felipe Sanchez Martinez
Andre Lynum
Stephan
Tsuyoshi Okita