
MOSES

Machine Translation with Open Source Software

Hieu Hoang and Matthias Huck

October 2014



Outline



Slides downloadable from

<http://www.statmt.org/moses/mtsummit.2013.pdf>

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



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
- 2006-2012** Funding by EU projects EuroMatrix, EuroMatrixPlus
- 2009** Tree-based models implemented in Moses
- 2012-2015** MosesCore project. Full-time staff to maintain and enhance 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 github.com
- Main forum: support and developer mailing lists
- Main event: Machine Translation Marathon (next week in Prague)
 - 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, FastAlign
 - language model: SRILM, IRSTLM, RANDLM, KENLM
 - 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/>
- **Phrasal** — Stanford University
<http://nlp.stanford.edu/phrasal/>
- 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

Many others ...

World Trade Organisation, Adobe, Symantec, WIPO, Sybase, Safaba, Bloomberg, Pangeanic, KatanMT, Capita, ...

Phrase-Based Model



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

Phrase Translation Options

| er | geht | ja | nicht | nach | hause |
|------------|--------------|-------------|-----------|--------------|---------|
| he | is | yes | not | after | house |
| it | are | is | do not | to | home |
| , it | goes | , of course | does not | according to | chamber |
| , he | go | , | is not | in | at home |
| it is | | not | | home | |
| he will be | | is not | | under house | |
| it goes | | does not | | return home | |
| he goes | | do not | | do not | |
| | is | | to | | |
| | are | | following | | |
| | is after all | | not after | | |
| | does | | not to | | |
| | not | | | | |
| | is not | | | | |
| | are not | | | | |
| | is not a | | | | |

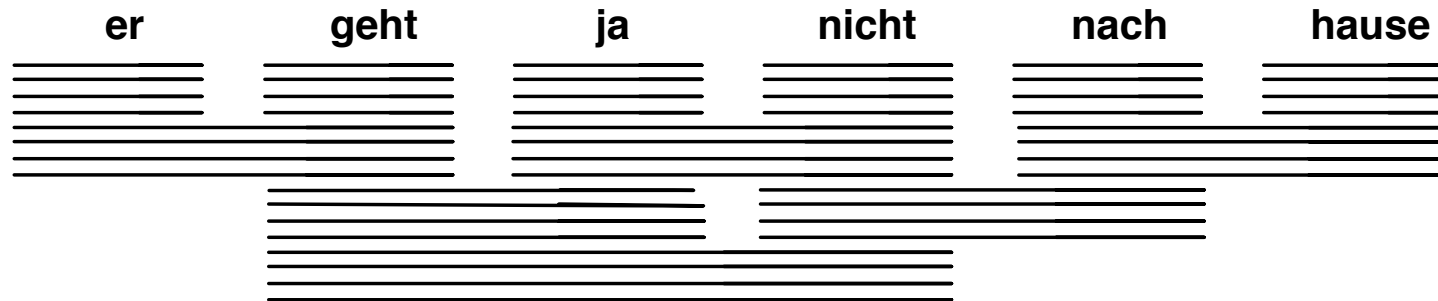
- Many translation options to choose from

Phrase Translation Options

| er | geht | ja | nicht | nach | hause |
|------------|--------------|-------------|-----------|--------------|---------|
| he | is | yes | not | after | house |
| it | are | is | do not | to | home |
| , it | goes | , of course | does not | according to | chamber |
| , he | go | | is not | in | at home |
| it is | | not | | home | |
| he will be | | is not | | under house | |
| it goes | | does not | | return home | |
| he goes | | do not | | do not | |
| | is | | to | | |
| | are | | following | | |
| | is after all | | not after | | |
| | does | | not to | | |
| | not | | | | |
| | is not | | | | |
| | are not | | | | |
| | is not a | | | | |

- 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 Options¹⁴



consult phrase translation table for all input phrases

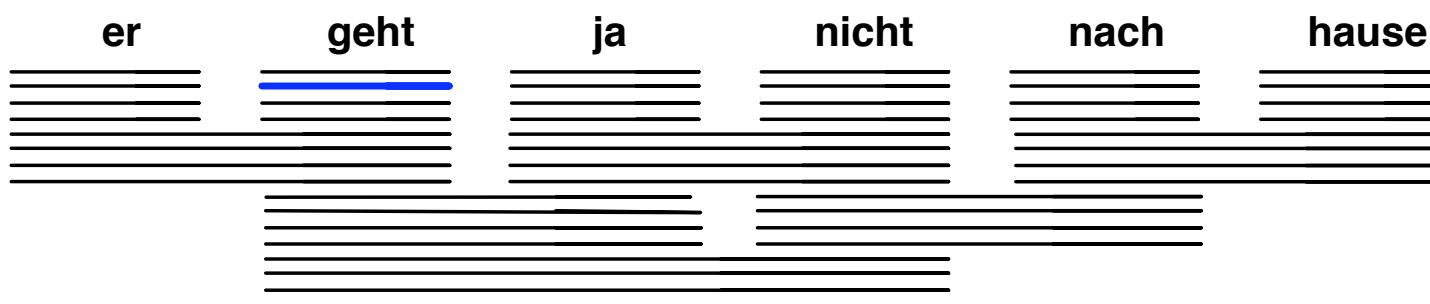
Decoding: Start with Initial Hypothesis

15



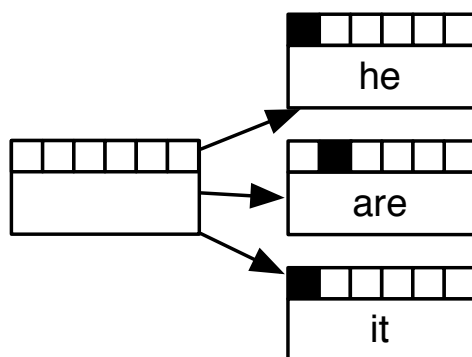
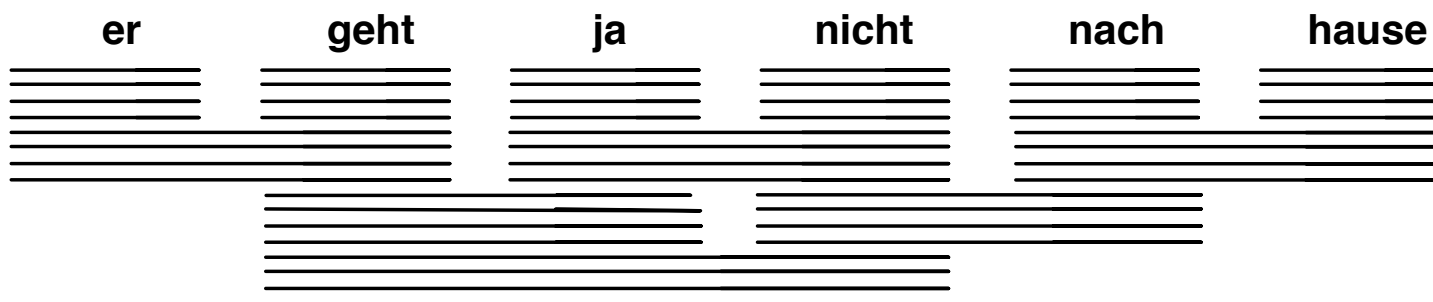
initial hypothesis: no input words covered, no output produced

Decoding: Hypothesis Expansion



pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



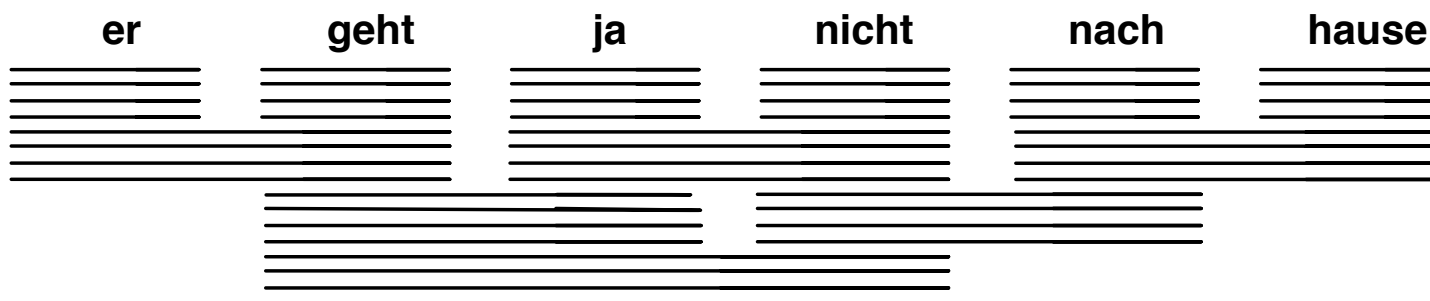
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



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



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

Factored Model

Example:



Decomposing the translation step

Translating lemma and morphological information more robust

Syntax Models

String to String

John misses Mary

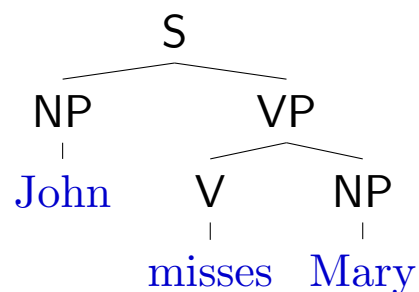
⇒ Marie manque à Jean

String to Tree

John misses Mary

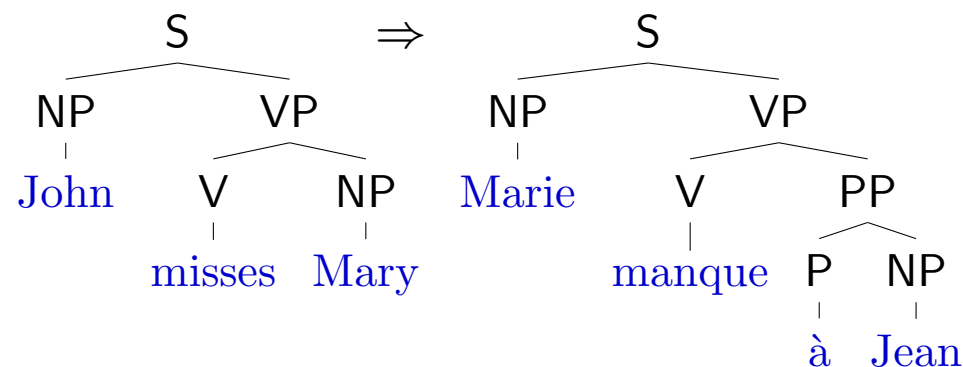


Tree to String

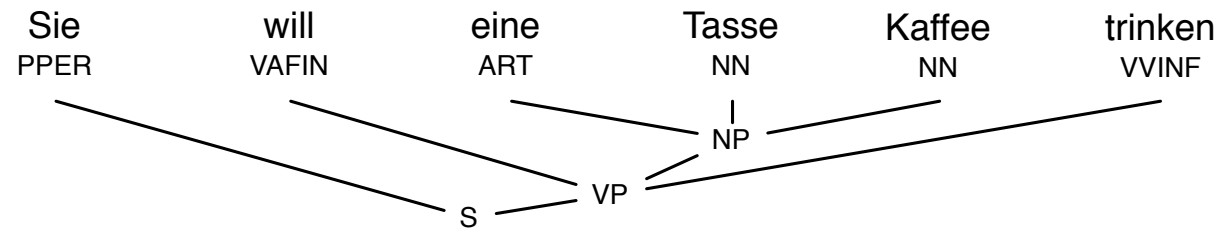


⇒ Marie manque à Jean

Tree to Tree



Syntax Decoding

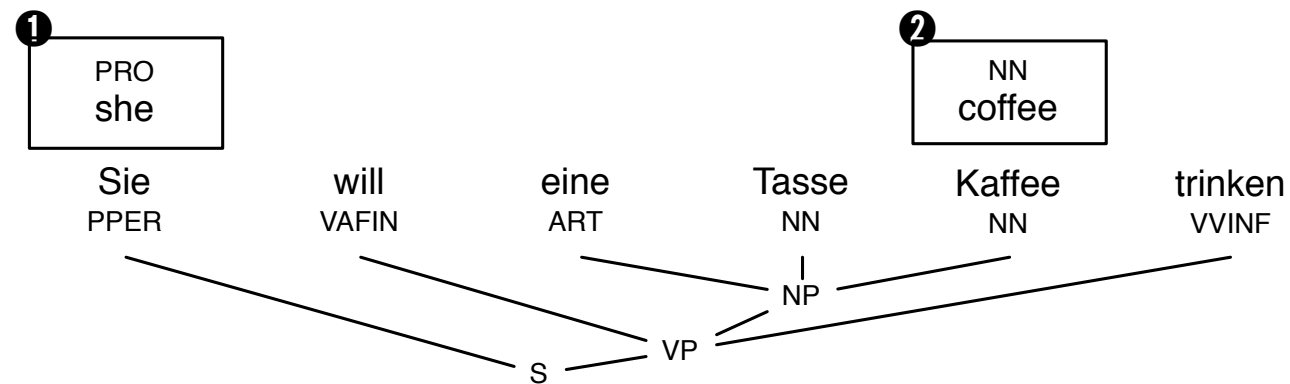


Syntax Decoding



Syntax Decoding

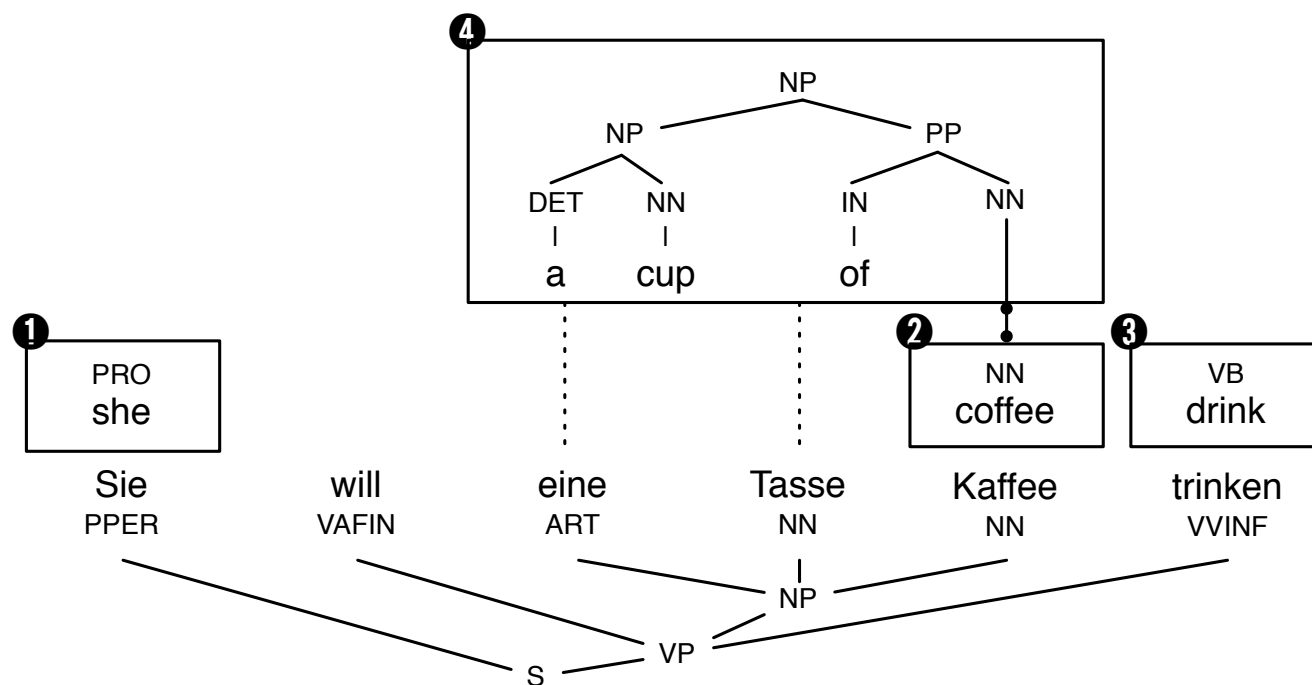
26



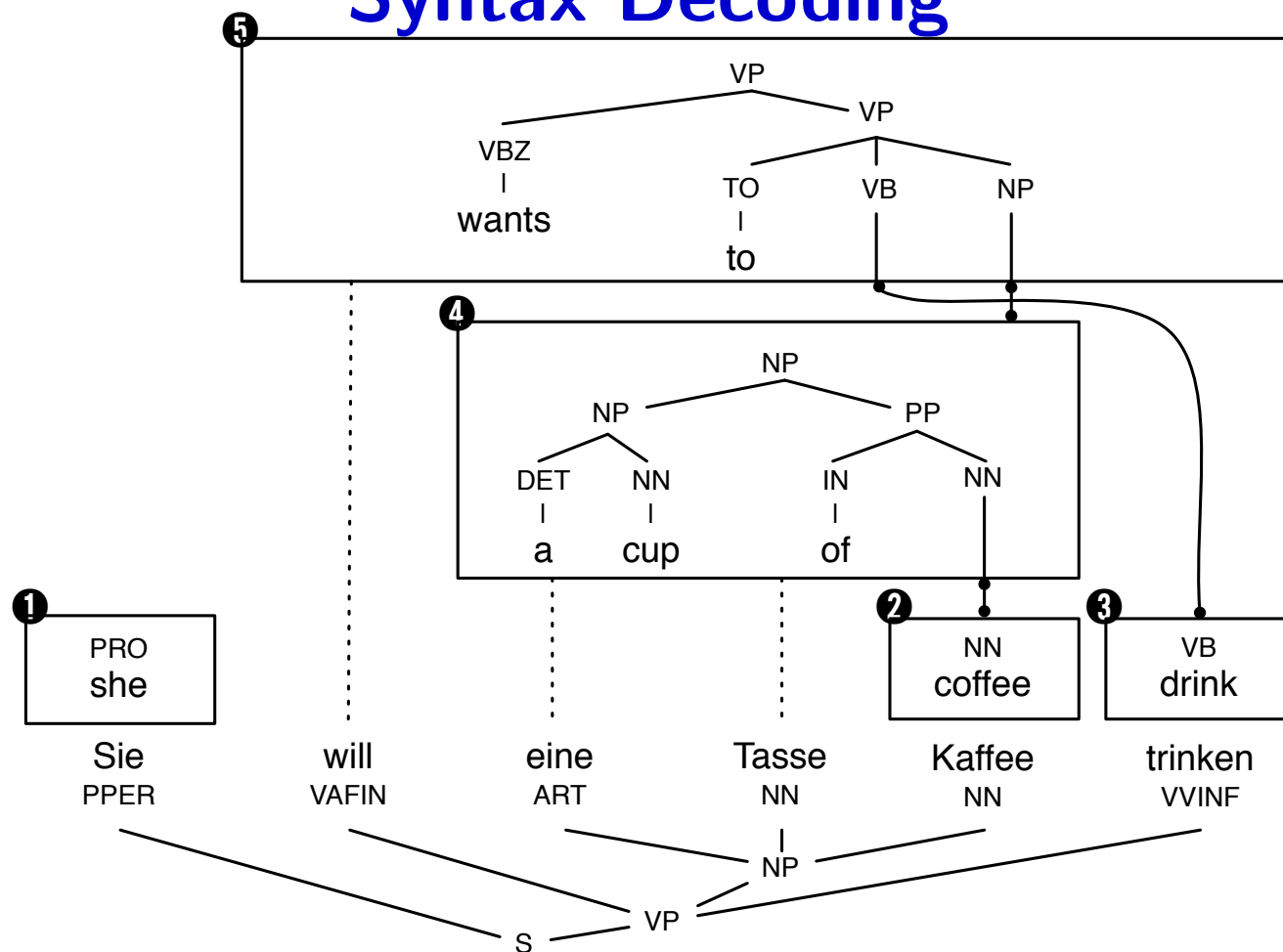
Syntax Decoding



Syntax Decoding



Syntax Decoding





Advanced Features

- **How do I get started?**
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Incremental Training

How do I get started?

- Collect your data
 - Parallel data
 - Translation memories
 - Open-sourced data, eg. Europarl, UN, TAUS Data Association
 - Monolingual data
- Set up Moses
 - Download source code for Moses, GIZA++, MGIZA
 - Compile, install
 - More info: <http://www.statmt.org/moses/>
 - Prepackaged Moses: Precision Tools, MacPorts, Debian packages, M4Loc

How do I get started?

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 ...
```

Advanced Features

- How do I get started?
- **Experiment Management System**
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Incremental Training

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
 - runs on single machine, multi-core machine, GridEngine cluster
 - parallel execution
 - crash detection
 - automatic re-use of prior results
- Fast to use
 - set up a new experiment in minutes
 - set up a variation of an experiment in seconds
- Disadvantage — not all Moses features are integrated

Workflow automatically
generated by
experiment.perl



How does it work?

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

- Test:

```
experiment.perl -config config
```

- Execute:

```
experiment.perl -config config -exec
```

Web Interface

All Experimental Setups

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

List of experiments

List of Runs

Task: WMT10 German-English (pkoehn)

[Wiki Notes](#) | [Overview of experiments](#) | [/fs/bragi2/pkoehn-experiment/wmt10-de-en](#)

| compare | ID | start | end | avg | newstest2009 | | newstest2010 | |
|---|---|--------|---------------------------------------|---------------------------------|--|---|--|---|
| <input type="checkbox"/> cfglparlimg | [1042-16] 11+analysis | 16 May | 16 May | BLEU-c: 21.74 BLEU: 22.91 | 21.03 (1.002) 22.30 (1.002) | A <input type="checkbox"/> | 22.45 (1.041) 23.51 (1.041) | A <input type="checkbox"/> |
| <input type="checkbox"/> cfglparlimg | [1042-15] 11+Internal emplus test set | 21 Apr | crashed | - | - | | - | |
| <input type="checkbox"/> cfglparlimg | [1042-14] 9+interpolated-tm.lm-weighted | 21 Feb | 21 Feb 9: 0.239258 -> 0.239296 | - | 20.81 (1.003) 22.06 (1.003) | A <input type="checkbox"/> | - | |
| <input type="checkbox"/> cfglparlimg | [1042-13] 9+only-ep | 21 Feb | 21 Feb 13: 0.235046 -> 0.235053 | - | 20.42 (1.002) 21.69 (1.002) | A <input type="checkbox"/> | - | |
| <input type="checkbox"/> cfglparlimg | [1042-12] 9+only-nc | 21 Feb | 21 Feb 7: 0.222237 -> | - | 18.96 (1.002) 20.16 | A <input type="checkbox"/> | - | |

Analysis: Basic Statistics

| Coverage | | | Phrase Segmentation | | | | |
|--|---------------|---------------|-------------------------------------|---------------|---------------|-------------|-------------|
| | model | corpus | | 1 | 2 | 3 | 4+ |
| 0 | 2047 (3.1%) | 1708 (2.6%) | 1 to | 26897 (40.7%) | 2145 (3.2%) | 278 (0.4%) | 90 (0.1%) |
| 1 | 738 (1.1%) | 518 (0.8%) | 2 to | 4144 (6.3%) | 14414 (21.8%) | 2518 (3.8%) | 432 (0.7%) |
| 2-5 | 1483 (2.2%) | 818 (1.2%) | 3 to | 639 (1.0%) | 3522 (5.3%) | 4821 (7.3%) | 1272 (1.9%) |
| 6+ | 61745 (93.5%) | 62969 (95.4%) | 4+ to | 158 (0.2%) | 855 (1.3%) | 1693 (2.6%) | 2135 (3.2%) |
| by token / by type / details | | | by word / by phrase | | | | |

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

Analysis: Unknown Words

grouped by count in test set

unknown words

| | | | | |
|-----------------|---------------|-------------------|--------------------------|--|
| 18 Eatonville | 4: | 3: Anmil, | 2: Abfertigungen, | 1: -Ach, -Minister, -Pakets, -weiss, .docx, .pptx, .xlsx, 1,45, |
| 16 Hurston | Eatonvilles, | Atlasz, BR23C, | Albums, Alondra, | 1.106,55, 1.983,73, 10.365,45, 10.579, 10.809,25, 106,85, |
| 12 Barrick | Együtt, | BSA, Bayón, | Andoh, Anm., Armiñon, | 11,9, 11.743,61, 12.595,75, 14,2, 14,7, 145,29, 16,8, 17,9, |
| 12 Hema | Garver, | Biztos, Bt., | Ashford, BZÖ, Baloldal, | 18,6, 18.286,90, 1802, 1834, 1880ern, 1920ern, 1925, |
| 12 Stewards | Harmadik, | Butch, Casado, | Bani, Baugesellschaften, | 19252008, 199,61, 2,178, 2,37, 2.400, 26,3, 270.000, 29,2, |
| 11 Gebrselassie | Hurstons, | Dal, Embraer, | Bedienkomfort, Bento, | 3,30, 3,632, 3,827, 3.0.0, 4,161, 4,357, 42,2, 43,4, 499, |
| 10 Flamenco | Jobb, Jol, | FT, Faymann, | Bentos, Bingleys, Bojen, | 49sten, 5.839, 506,43, 6,98, 684,81, 729,700, 75,5, 777,68, |
| 10 Mango | Jos, Jövőért, | Fiatal, Gregg, | Bowens, Bowery, Boyd, | 8,25, 8,81, 9,14, 99,80, AAC, ADQ, ART, Aareal, |
| 9 Glitter | Kovalev, | Gélineau, HSV, | Bringley, Browser, | Abbremsens, Abhöraktion, Absenzen, Abwesenheiten, |
| 9 ÚOHS | Kreuer, | Hanzelka, | Bělohávek, CBGB, | Abwiegen, Abwärtssog, Achronot, Actor, AdSense, |
| 9 ČTÚ | Lados, | Illhäusern, Iván, | Carci, Cera, Charts, | AdWords, Aday, Adobe, Adressverzeichnisses, Adwards, |
| 8 Coles | Mercandelli, | Jansen, Jančura, | Chemical, Chigi, | Adélar, Agazio, Akku, Akron, Aktuálně.cz, Alameda, |
| 8 Deka | Stehplätze, | Joanne, | Cineast, Comics, | Alatriste, Alcolock, Aleš, Alhambra, Alleinregierer, |
| 8 Garci | Tauro, | Kemrová, Kid, | Commerzbank, Coppola, | Amazonengebiet, Amil, Aminei, Amministrazione, Amway, |
| 8 ITV | Tórtola, | Llamazares, | Corker, Cowon, DF, | Andalusierin, Andik, Android, Anděl, Angeklagtem, Ansa, |
| | Zenobia, | Loafs, Mangas, | Dinkins, Download, | Anthologie, Antiasthmatica, Apnoe, Aquel, Arabija, |
| | fon, | Medikamentes, | Drehbewegung, | Arbeitenehmers, Arcandor, Arriaga, Asiana, Askale, |
| | Évezredért, | Mobil.cz, | Drzewiecki, Drápal, | Astronomen, Aufeislegen, Augäpfel, Ausdrückstärke, |
| | Ózd | Mutual, | Düsseldorfer, Ella, | Ausführungs-, Ausgeruhter, Ausscheidungsspiele, |

Analysis: Output Annotation

[0.2152] This time was the reason for the collapse on Wall Street .
[ref] This time the fall in stocks on Wall Street is responsible for the drop .

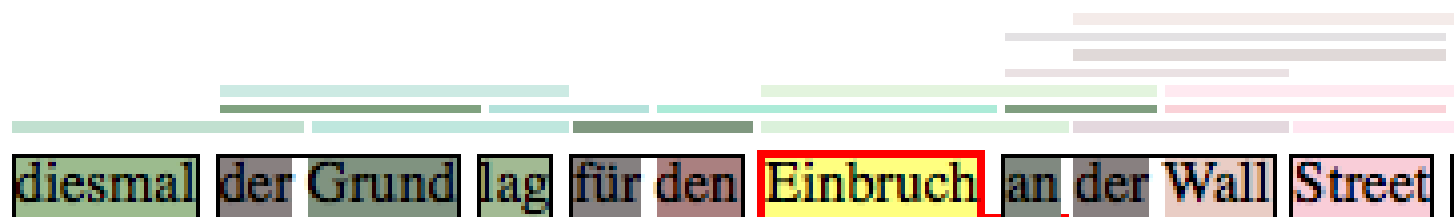
Color highlighting to indicate n-gram overlap with reference translation

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

Analysis: Input Annotation

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

[#4]



- 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 made recommendations , " **inter alia** " , with respect to the follow...
 ...on (EC) No 1995 / 2000 imposing , **inter alia** , a definitive anti @-@ dumping dut...
 ...ervices . this increase , arising , **inter alia** , as a result of economic growth , ...
 ...of paragraph 1 the Commission may , **inter alia** , bring forward :
 ... of stocks of obsolete pesticides , **inter alia** , by supporting projects aimed at s...
 ...wn rules of procedure which shall , **inter alia** , contain provisions for convening ...
 ...uch specific agreements may cover , **inter alia** , financing provisions , assignment...
 ...he internal market and concerning , **inter alia** , health and environmental protecti...
 ...e product concerned) originating , **inter alia** , in Belarus and Russia (the count...
 ...e product concerned) originating , **inter alia** , in India .

... des recommandations concernant , **entre autres** , les questions spécifiques suiva...
 ...995 / 2000 du Conseil instituant , **entre autres** , un droit antidumping définitif ...
 ...nsports . cette augmentation , due **entre autres** facteurs à la croissance économi...
 ...aragraphe 1 , la Commission peut , **entre autres** , présenter :
 ...r les stocks de vieux pesticides , **entre autres** en soutenant des projets à cet ef...
 ...lement intérieur , qui contient , **entre autres** dispositions , les modalités de c...
 ...ords spécifiques peuvent porter , **entre autres** , sur les mécanismes financiers s...
 ...hé intérieur et qui concernent , **entre autres** , la santé et la protection de l&...
 ...it concerné ") originaire , **entre autres** , du Belarus et de Russie (ci @-@ ...
 ...t concerné ") originaires , **entre autres** , de l ' Inde .

notamment(447/1554)

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

...get de l' Union , ce qui passe **notamment** par la résolution du problème de r...
 ...es États membres , la Commission a **notamment** arrêté la décision 2003 / 526 / C...
 ... durable et équitable , impliquant **notamment** l' accès aux ressources produc...
 ...usceptibles d' être utilisés **notamment** comme équipements mis à bord , mai...
 ...ion et à ses annexes , compte tenu **notamment** des informations scientifiques , tec...
 ...os ; il est absolument nécessaire , **notamment** en raison de l' élargissement ...
 ...ragraphes 1 et 2 le cas échéant , **notamment** en menant des études et en compilan...
 ... et d' efficacité en raison , **notamment** , d' une interopérabilité tec...
 ...nis dans le programme , il convient **notamment** de mettre en oeuvre les moyens ci @-...

translation of input phrase in training data context

Analysis: Alignment

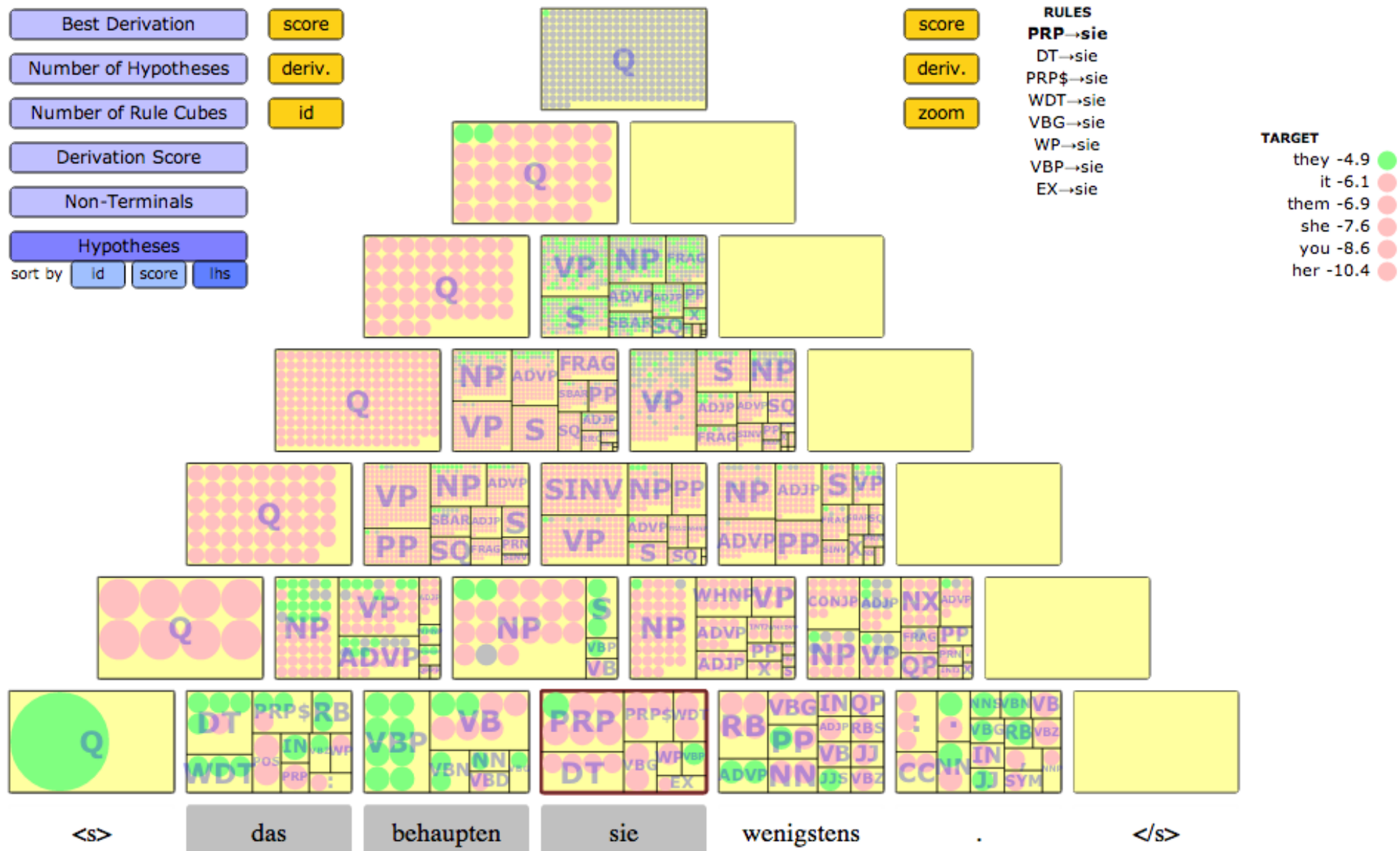
| | | | | | | | | |
|---------------|-----------|------------|---------|----------|--------|------|--------|---|
| diesmal | der Grund | lag | für den | Einbruch | an der | Wall | Street | . |
| 52] This time | was | the reason | for the | collapse | on | Wall | Street | . |

Phrase 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](#) order [worse](#) display [fullscreen](#) showing 5 [more](#) [all](#)

identical same better worse

2348 51 57 69

93% 2% 2% 3%

[2143:0.2974] In Austria , Haider and Co. are ready to govern to prevent a red and black coalition .

[2143:0.1754] In Austria , Haider and Co. are prepared to rule to prevent a red and black coalition .

[ref] Haider and his party are ready to govern Austria in order to avoid red @-@ black coalition .

[2165:0.3174] 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 .

[2165:0.2061] The SPÖ wants to show that a cooperation of both parties is possible - in some countries and in the social partnership that is already the case .

[ref] 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

Hands-On Session

Advanced Features

- How do I get started?
- Experiment Management System
- **Faster Training**
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Incremental Training

Advanced Features

- How do I get started?
- Experiment Management System
- **Faster Training**
 - Tokenization
 - Tuning
 - Alignment
 - Phrase-Table Extraction
 - Train language model

...

Faster Training

- Run steps in parallel (that do not depend on each other)

- Multicore Parallelization

```
.../train-model.perl -parallel
```

- EMS:

```
[TRAINING]  
parallel = yes
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
 - **Tokenization**
 - Tuning
 - Alignment
 - Phrase-Table Extraction
 - Train language model

...

Faster Training

- Multi-threaded tokenization
- Specify number of threads

```
.../tokenizer.perl -threads NUM
```

- EMS:

```
input-tokenizer = "$moses-script-dir/tokenizer/tokenizer.perl  
-threads NUM "
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
 - Tokenization
 - **Tuning**
 - Alignment
 - Phrase-Table Extraction
 - Train language model

...

Faster Training

- Multi-threaded tokenization
- Specify number of threads

```
.../mert -threads NUM
```

- EMS:

```
tuning-settings = "-threads NUM"
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
 - Tokenization
 - Tuning
 - **Alignment**
 - Phrase-Table Extraction
 - Train language model

...

Faster Training

- Word Alignment
- Multi-threaded
 - Use MGIZA, not GIZA++

```
.../train-model.perl -mgiza -mgiza-cpus NUM
```

EMS:

```
training-options = " -mgiza -mgiza-cpus NUM "
```

- On: memory-limited machines
 - snt2cooc program requires 6GB+ memory
 - Reimplementation uses 10MB, but take longer to run

```
.../train-model.perl -snt2cooc snt2cooc.pl
```

EMS:

```
training-options = "-snt2cooc snt2cooc.pl"
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
 - Tokenization
 - Tuning
 - Alignment
 - **Phrase-Table Extraction**
 - Train language model

...

Faster Training

- Phrase-Table Extraction
 - Split training data into NUM equal parts
 - Extract concurrently

```
.../train-model.perl -cores NUM
```

Faster Training

- Sorting
 - Rely heavily on Unix 'sort' command
 - may take 50%+ of translation model build time
 - Need to optimize for
 - * speed
 - * disk usage
 - Dependent on
 - * sort version
 - * Unix version
 - * available memory

Faster Training

- Plain sorted

```
sort < extract.txt > extract.sorted.txt
```

- Optimized for large server

```
sort --buffer-size 10G --parallel 5  
    --batch-size 253 --compress-program [gzip/pigz] ...
```

- Use 10GB of RAM — the more the better
- 5 CPUs — the more the better
- mergesort at most 253 files
- compress intermediate files — less disk i/o

- In Moses:

```
.../train-model.perl -sort-buffer-size 10G -sort-parallel 5  
    -sort-batch-size 253 -sort-compress pigz
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
 - Tokenization
 - Tuning
 - Alignment
 - Phrase-Table Extraction
 - **Train language model**

...

IRSTLM: Training

- Developed by FBK-irst, Trento, Italy
- Specialized training for large corpora
 - parallelization
 - reduce memory usage
- Quantization of probabilities
 - reduces memory but lose accuracy
 - probability stored in 1 byte instead of 4 bytes

IRSTLM: Training

- 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

- EMS:

```
irst-dir = [IRST path]  
lm-training = "$moses-script-dir/generic/trainlm-irst.perl  
              -cores NUM -irst-dir $irst-dir"
```

New: KENLM Training

- Can train very large language models with limited RAM (on disk streaming)

```
lmplz -o [order] -S [memory] < text > text.lm
```

- `-o order` = n-gram order
- `-S memory` = How much memory to use.
 - `NUM%` = percentage of physical memory
 - `NUM[b/K/M/G/T]` = specified amount in bytes, kilo bytes, etc.

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- **Faster Decoding**
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Incremental Training

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- **Faster Decoding**
 - Multi-threading
 - Speed vs. Memory
 - Speed vs. Quality

...

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
 - **Multi-threading**
 - Speed vs. Memory
 - Speed vs. Quality

...

Fast Decoding

- Multi-threaded decoding

```
.../moses --threads NUM
```

- Easy speed-up

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
 - Multi-threading
 - **Speed vs. Memory**
 - Speed vs. Quality

...

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

Process size
(RAM)



Speed vs. Memory Use

- Load into memory
 - long load time
 - large memory usage
 - fast decoding
- Load-on-demand
 - store indexed model on disk
 - binary format
 - minimal start-up time, memory usage
 - slower decoding



Create Binary Tables

Phrase Table:

Phrase-based

```
export LC_ALL=C
cat pt.txt | sort | ./processPhraseTable -ttable 0 0 - \
    -nscores 4 -out out.file
```

```
export LC_ALL=C    ./CreateOnDiskPt 1 1 4 100 2 pt.txt out.folder
```

Hierarchical / Syntax

```
export LC_ALL=C ./CreateOnDiskPt 1 1 4 100 2 pt.txt out.folder
```

Lexical Reordering Table:

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

Language Models (later)

Specify Binary Tables

Change ini file

Phrase Table

```
[feature]
PhraseDictionaryBinary name=TranslationModel0 table-limit=20 \
  num-features=4 path=../../phrase-table
```

Hierarchical / Syntax

```
[feature]
PhraseDictionaryOnDisk name=TranslationModel0 table-limit=20 \
  num-features=4 path=../../phrase-table
```

Lexical Reordering Table
automatically detected

Compact Phrase Table

- Memory-efficient data structure
 - phrase table 6–7 times smaller than on-disk binary table
 - lexical reordering table 12–15 times smaller than on-disk binary table
- Stored in RAM
- May be memory mapped
- Train with `processPhraseTableMin`
- Specify with `PhraseDictionaryCompact`

IRSTLM



77

- 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
- Not multithreaded

IRSTLM in Moses

- Compile the decoder with IRSTLM library

```
./configure --with-irstlm=[root dir of the IRSTLM toolkit]
```

- Create binary format:

```
compile-lm language-model.srilm language-model.blm
```

- Load-on-demand:

```
rename file .mm
```

- Change ini file to use IRSTLM implementation

```
[feature]  
IRSTLM name=LM0 factor=0 path=../../lm order=5
```

KENLM

- Developed by Kenneth Heafield (CMU / Edinburgh / Stanford)
- Fastest and smallest language model implementation
- Compile from LM trained with SRILM

```
build_binary model.lm model.binlm
```

- Specify in decoder

```
[feature]  
KENLM name=LM0 factor=0 path=../../model.binlm order=5
```


New: OSM (Operations Sequence Model)



- A model that
 - considers source and target contextual information across phrases
 - integrates translation and reordering into a single model
- Convert a bilingual sentence to a sequence of operations
 - Translate (Generate a minimal translation unit)
 - Reordering (Insert a gap or Jump)
- $P(e,f,a)$ = N-gram model over resulting operation sequences
- Overcomes phrasal independence assumption
 - Considers source and target contextual information across phrases
- Better reordering model
 - Translation and reordering decisions influence each
 - Handles local and long distance reorderings in a unified manner
- No spurious phrasal segmentation problem
- Average gain of +0.40 on news-test2013 across 10 pairs

New: OSM (Operations Sequence Model)

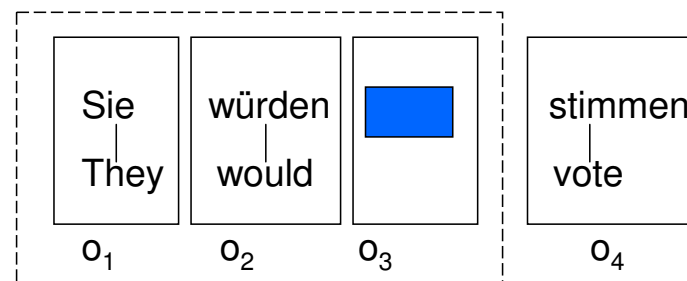


Sie würden gegen Sie stimmen

They would vote against you

Operations

- o_1 Generate (Sie, They)
- o_2 Generate (würden, would)
- o_3 Insert Gap
- o_4 Generate (stimmen, vote)
- o_5 Jump Back (1)
- o_6 Generate (gegen, against)
- o_7 Generate (Sie, you)



Context Window

Model:

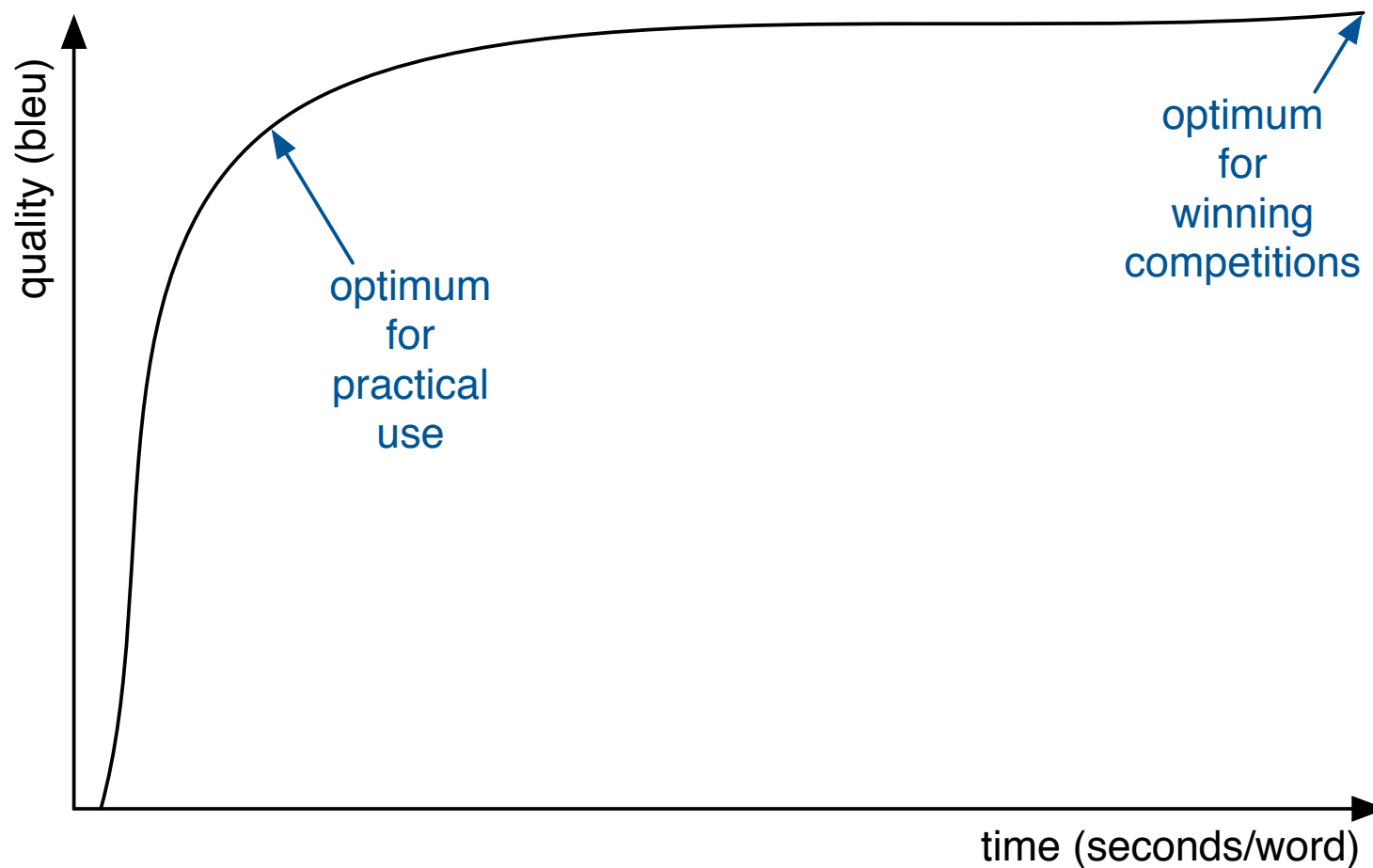
$$p_{\text{osm}}(F, E, A) = p(o_1, \dots, o_N) = \prod_i p(o_i | o_{i-n+1} \dots o_{i-1})$$

Advanced Features

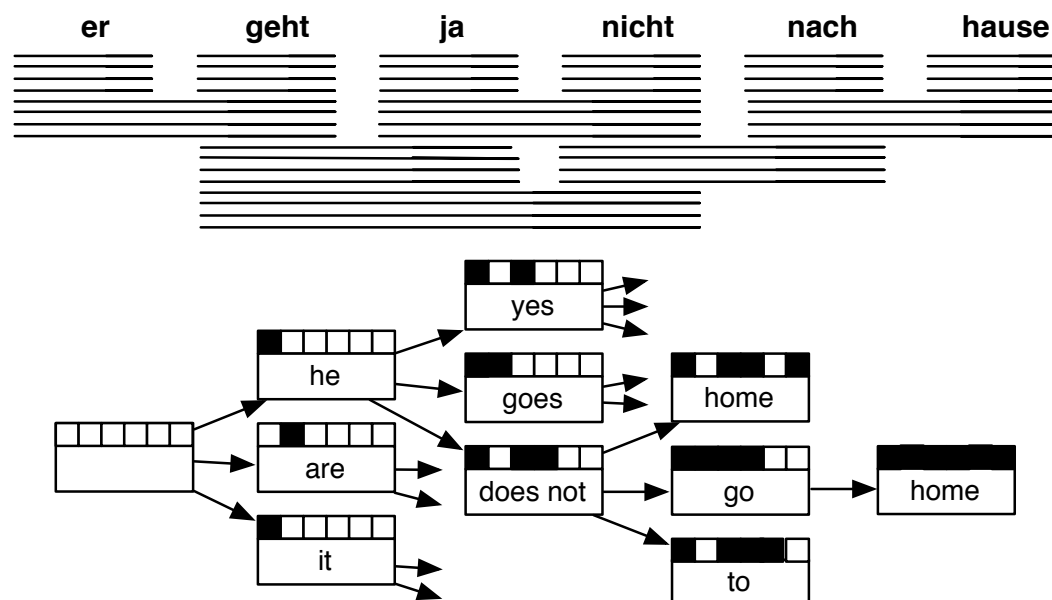
- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
 - Multi-threading
 - Speed vs. Memory
 - **Speed vs. Quality**

...

Speed vs. Quality

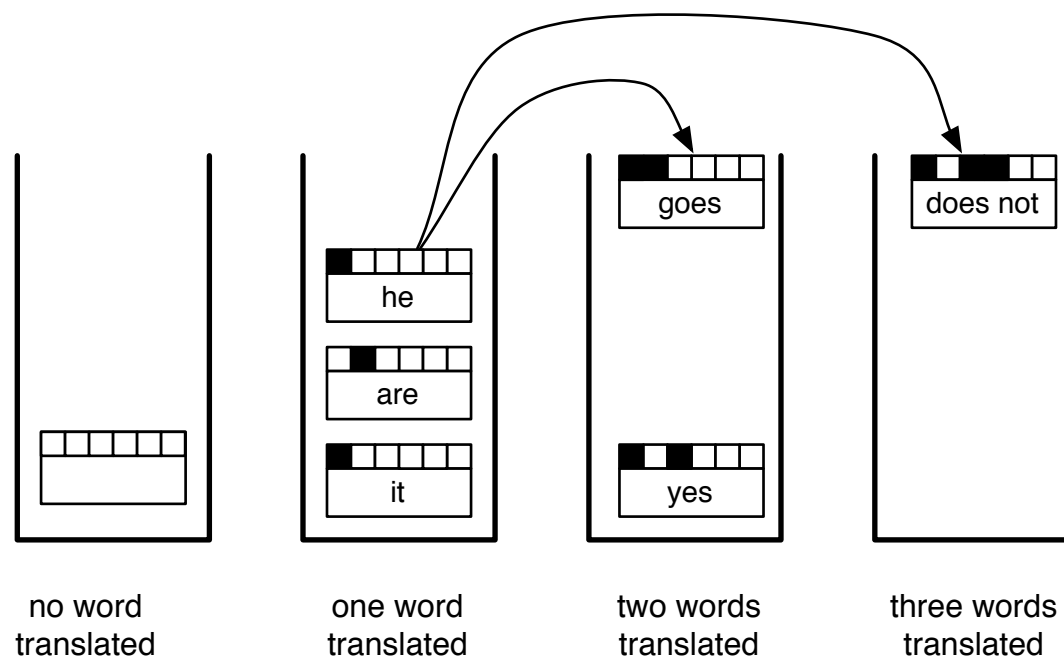


Speed vs. Quality



- 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

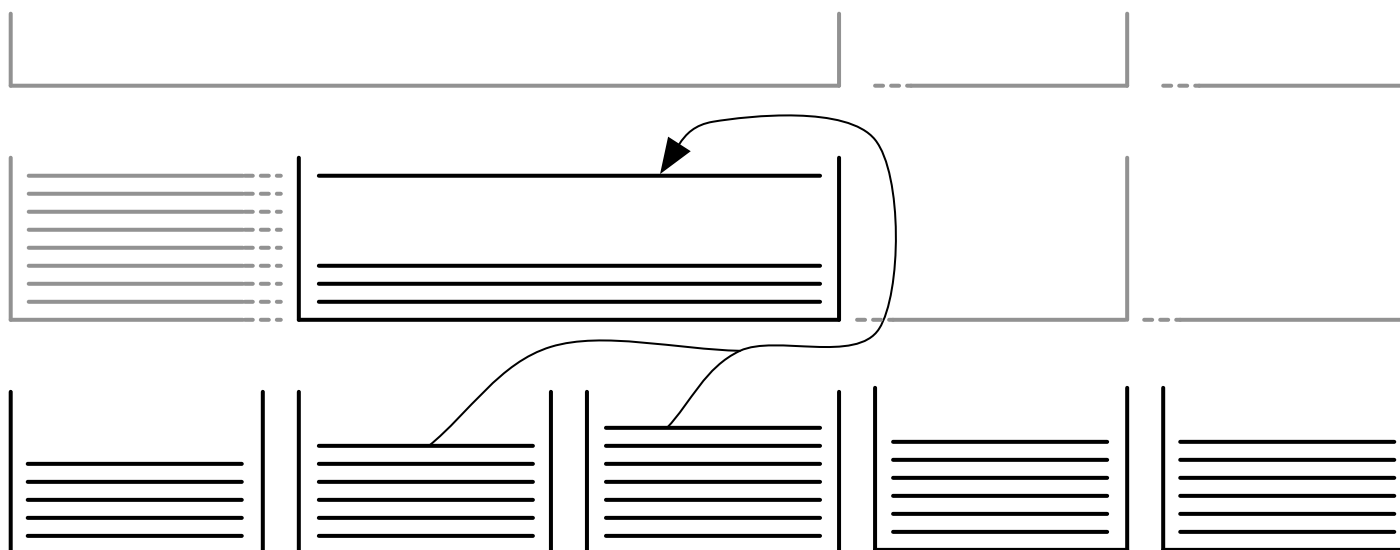


- Phrase-based: One stack per number of input words covered
- Number of hypothesis created =
sentence length \times stack size \times applicable translation options

Pruning Parameters

- Regular beam search
 - `--stack NUM` max. number of hypotheses contained in each stack
 - `--ttable-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 length² × number of hypotheses added to each stack
- `--cube-pruning-pop-limit NUM` number of hypotheses added to each stack

Advanced Features



88

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- **Moses Server**
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- Incremental Training

Moses Server

- Moses command line:

```
.../moses -f [ini] < [input file] > [output file]
```

- Not practical for commercial use

- Moses Server:

```
.../mosesserver -f [ini] --server-port [PORT] --server-log [LOG]
```

- Accept HTTP input. XML SOAP format

- Client:

- Communicate via http
- Example clients in Java and Perl
- Write your own client
- Integrate into your own application

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- **Data and domain adaptation**
 - Train everything together
 - Secondary phrase table
 - Domain indicator features
 - Interpolated language models

Data



91

- 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)
- Always tune towards target domain

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
 - **Train everything together**
 - Secondary phrase table
 - Domain indicator features
 - Interpolated language models

Default: Train Everything Together



- Easy to implement
 - Concatenate new data with existing data
 - Retrain
- Disadvantages:
 - Slower training for large amount of data
 - Cannot weight old and new data separately

Default: Train Everything Together

Specification in EMS:

- Phrase-table

```
[CORPUS]
[CORPUS:in-domain]
raw-stem = ....
[CORPUS:background]
raw-stem = ....
```

- LM

```
[LM]
[LM:in-domain]
raw-corpus = ....
[LM:background]
raw-corpus = ....
```


Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
 - Train everything together
 - **Secondary phrase table**
 - Domain indicator features
 - Interpolated language models

Secondary Phrase Table

- Train initial phrase table and LM on baseline data
- Train secondary phrase table and LM new/in-domain data
- Use both in Moses

– Secondary phrase table

```
[feature]
PhraseDictionaryMemory path=.../path.1
PhraseDictionaryMemory path=.../path.2

[mapping]
0 T 0
1 T 1
```

Secondary Phrase Table

- – Secondary LM

```
[feature]  
KENLM path=.../path.1  
KENLM path=.../path.2
```

- Can give different weights for primary and secondary tables
- Not integrated into the EMS

Secondary Phrase Table

- Terminology/Glossary database
 - fixed translation
 - per client, project, etc
- Primary phrase table
 - backoff to 'normal' phrase-table if no glossary term

```
[feature]
```

```
PhraseDictionaryMemory path=.../glossary
```

```
PhraseDictionaryMemory path=.../normal.phrase.table
```

```
[mapping]
```

```
0 T 0
```

```
1 T 1
```

```
[decoding-graph-backoff]
```

```
0
```

```
1
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
 - Train everything together
 - Secondary phrase table
 - **Domain indicator features**
 - Interpolated language models

Domain Indicator Features

- One translation model
- Flag each phrase pair's origin
 - indicator: binary flag if it occurs in specific domain
 - ratio: how often it occurs in specific domain relative to all
 - subset: similar to indicator, but if in multiple domains, marked with multiple-domain feature
- In EMS:

```
[TRAINING]  
domain-features = "indicator"
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
 - Train everything together
 - Secondary phrase table
 - Domain indicator features
 - **Interpolated language models**

Interpolated Language Models

- Train one language model per corpus
- Combine them by weighting each according to its importance
 - weights obtained by optimizing perplexity of resulting language model on tuning set (not the same as machine translation quality)
 - models are linearly combined
- EMS provides a section [INTERPOLATED-LM] that needs to be commented out
- Alternative: use multiple language models (disadvantage: larger process, slower)

Advanced Features

104



- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- **Instructions to decoder**
- Input formats
- Output formats
- Translation models
- Incremental Training

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

the revenue for <num translation="2003"> 2003 </num> is higher than ...

- Deal with different number formats

er erzielte <num translation="17.55"> 17,55 </num> Punkte .

Specifying Translations with XML

```
./moses -xml-input [exclusive | inclusive | constraint ]
```

the revenue for <num translation="2003"> 2003 </num> is higher than ...

Three types of XML input:

- Exclusive
Only possible translation is given in XML
- Inclusive
Translation is given in XML is in addition to phrase-table
- Constraint
Only use translations from phrase-table if it match XML specification

Constraint XML

- Specifically for translating terminology
 - consistently translate particular phrase in a document
 - may have learned larger phrase pairs that contain terminology term

- Example:

```
Microsoft <option translation="Windows"> Windows </option> 8 ...
```

- Allows use of phrase pair only if maps Windows to Windows

Placeholders

- Translate:
 - You owe me 100 dollars !
 - You owe me 200 dollars !
 - You owe me 9.56 dollars !
- Problem: need translations for
 - 100
 - 200
 - 9.56
- Some things are better off being handled by simple rules:
 - Numbers
 - Dates
 - Currency
 - Named entities

Placeholders

- Input

You owe me 100 dollars !

- Replace numbers with @num@

You owe me @num@ dollars !

- Specification

You owe me <ne translation="@num@" entity="100">@num@</ne> dollars !

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> " </zone> proposal .

Preserving Markup

- How do you translate this:

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

| | | | | | | |
|---|--------|------|----|------------|---------|---|
| I | really | like | to | eat | chicken | ! |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| - | - | - | - | | - | - |

- translate without markup

I really like to eat chicken !

- keep word alignment to source

| | | | | | |
|-----|------|----------|-------|----------|---|
| Ich | esse | wirklich | gerne | Hühnchen | ! |
| 1 | 5 | 2 | 3-4 | 6 | 7 |

- re-insert markup

Ich esse wirklich gerne Hühnchen!

Transliteration

- Languages are written in different scripts
 - Russian, Bulgarian and Serbian - written in Cyrillic script
 - Urdu, Farsi and Pashto - written in Arabic script
 - Hindi, Marathi and Nepalese - written in Devanagari
- Transliteration can be used to translate OOVs and Named Entities
- Problem: Transliteration corpus is not always available
- Naive Solution:
 - Crawl training data from Wikipedia titles
 - Build character-based transliteration model
 - Replace OOV words with 1-best transliteration

Transliteration

- 2 methods to integrate into MT
- Post-decoding method
 - Use language model to pick best transliteration
 - Transliteration features
- In-decoding method
 - Integrate transliteration inside decoder
 - Words can be translated OR transliterated

Transliteration

- EMS:

```
[TRAINING]  
transliteration-module = "yes"
```

- Post-processing method

```
post-decoding-transliteration = "yes"
```

- In-decoding method

```
in-decoding-transliteration = "yes"  
transliteration-file = /list of words to be transliterated/
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- **Input formats**
- Output formats
- Translation models
- Incremental Training



Example: Misspelt Words

- Misspelt sentence:

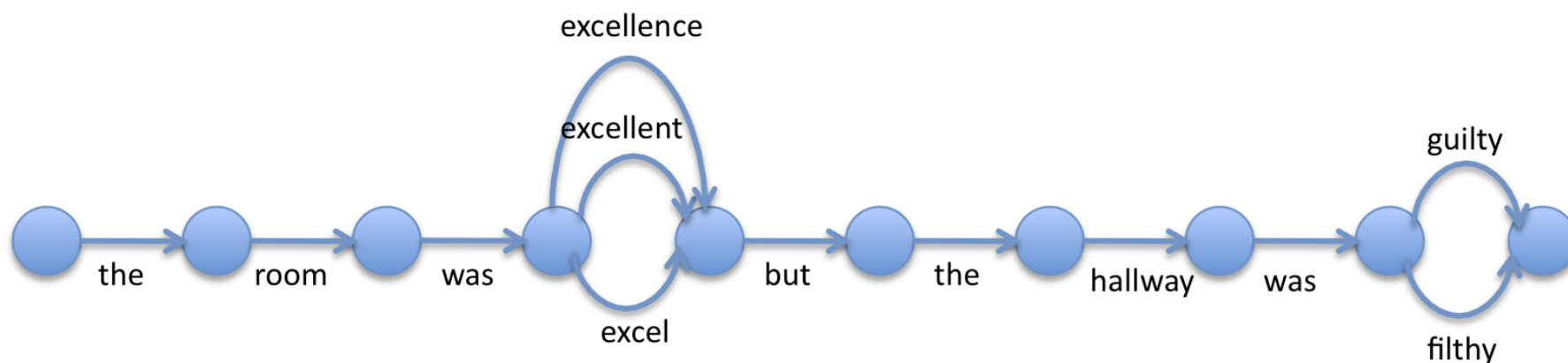
The room was *exellent but the hallway was *filty .

- 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 *excellent but the hallway was *filthy .

Input to decoder:



Let the decoder decide

Example: Diacritics

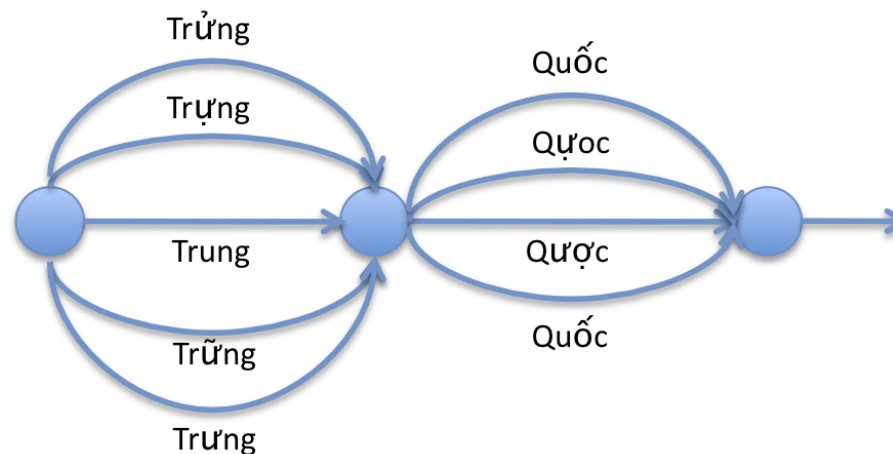
- Correct sentence

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 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
```

Lattice

Example: Chinese Word Segmentation

- Unsegmented sentence

硬质合金号称"工业牙齿"

- Incorrect segmentation

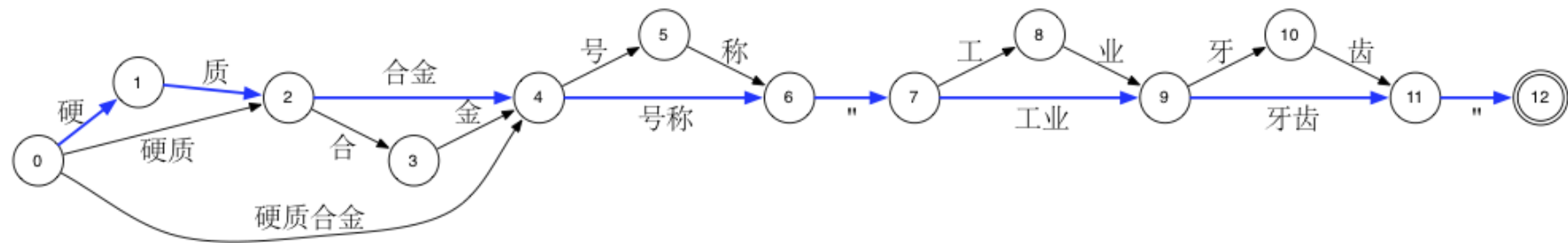
硬质 合 金 号称 " 工 业牙 齿 "

- Correct segmentation

硬 质 合金 号称 " 工业 牙齿 "

Lattice

Input to decoder:



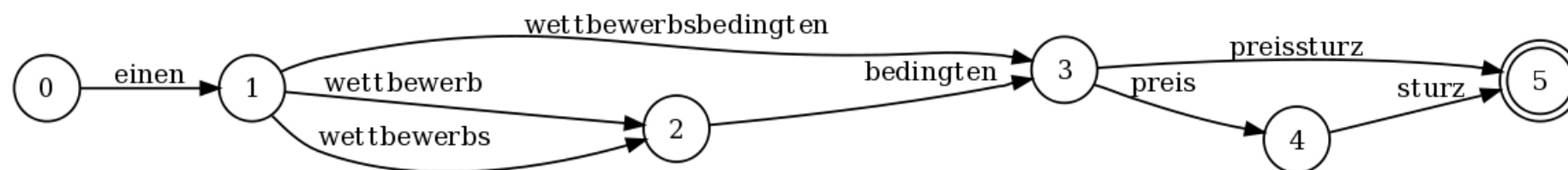
Let the decoder decide

Example: Compound Splitting

- Input sentence

einen wettbewerbsbedingten preissturz

- Different compound splits



- Let the decoder decide

Lattice Specification

Command line argument

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

Input to Moses (PLF format - Python Lattice Format)

```
(  
  (  
    ('einen', 1.0, 1),  
  ),  
  (  
    ('wettbewerbsbedingen', 0.5, 2),  
    ('wettbewerbs', 0.25, 1),  
    ('wettbewerb', 0.25, 1),  
  ),  
  (  
    ('bedingen', 1.0, 1),  
  ),  
  (  
    ('preissturz', 0.5, 2),  
    ('preis', 0.5, 1),  
  ),  
  (  
    ('sturz', 1.0, 1),  
  ),  
)
```

Advanced Features

- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- **Output formats**
- Translation models
- Incremental Training

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

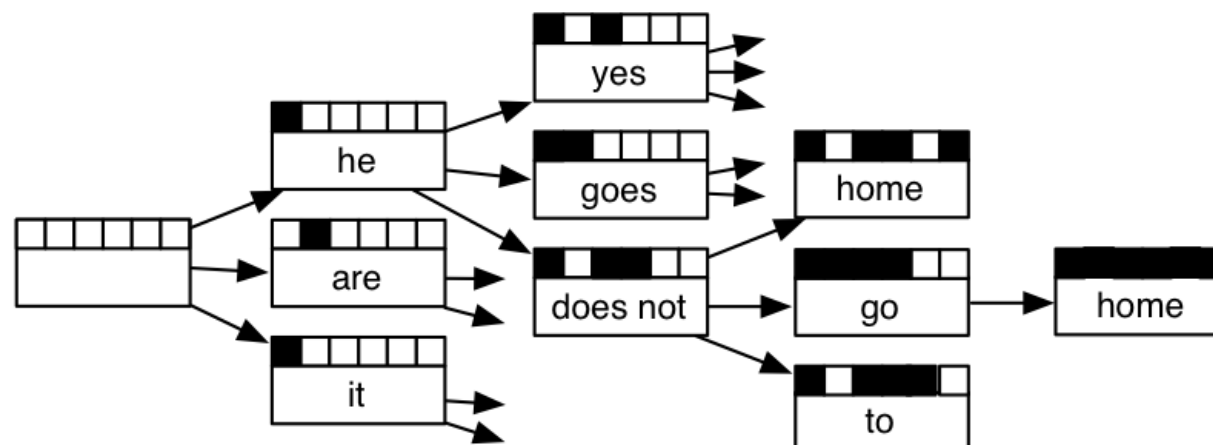
```
0 ||| there are various different opinions . ||| d: 0 lm: -21.6664 w: -6 ... ||| -113.734
0 ||| there are various other opinions . ||| d: 0 lm: -25.3276 w: -6 ... ||| -114.004
0 ||| there are different different opinions . ||| d: 0 lm: -27.8429 w: -6 ... ||| -117.738
0 ||| there are other different opinions . ||| d: -4 lm: -25.1666 w: -6 ... ||| -118.007
0 ||| we are various different opinions . ||| d: 0 lm: -28.1533 w: -6 ... ||| -118.142
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: -30.426 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

[1] New probe into US attorney affair >>
Neuer Vorstoß in den USA Anwalt neue Affäre sonde (9 edits)

neue sonde

enter in

| new | probe | into | US | attorney | affair |
|-------|---------------|------|---------------|-------------------------|---------------|
| neue | Sonde | in | | Anwalt | die |
| die | testet | In | die | Staatsanwalt | Affäre |
| die | prüfen | In | In | Anwälte | die |
| der | Vorstoß | In | die | Testamentsvollstreckers | sie |
| eine | auszuforschen | In | die | Vollmachten | Angelegenheit |
| neuer | prüfen | auch | In | Anwalt | um |
| die | prüfen | In | der | | Sache |
| das | prüfen | zu | amerikanische | | haben |
| neu | prüfen | In | der | | Geschichte |
| In | | nach | die | | das |

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=-10.661 ... 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

133



- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- **Translation models**
- Incremental Training

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/sent. for fastest configuration
- more memory requirement
 - 1-2GB ram
- more disk usage
 - translation model $\times 10$ 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

Phrase-Based Model Training

- Command line

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

- Model

```
Bndnisse ||| 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

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

- Decoding

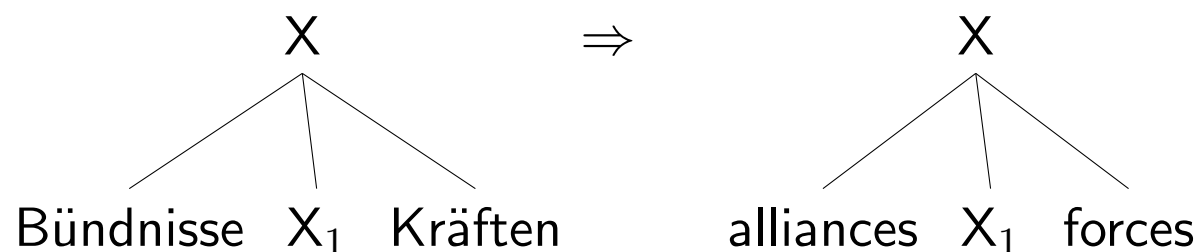
```
./moses-chart -f moses.ini -i in.txt > out.txt
```

Hierarchical Model

- Example rule from model

Bündnisse [X][X] Kräften [X] ||| alliances [X][X] forces [X] ||| 1 1 1 1 2.718 ||| 1-1 ||| 0.0526316 0.0526316

- Visualization of rule



Hierarchical Model

Comparison with phrase-based model:

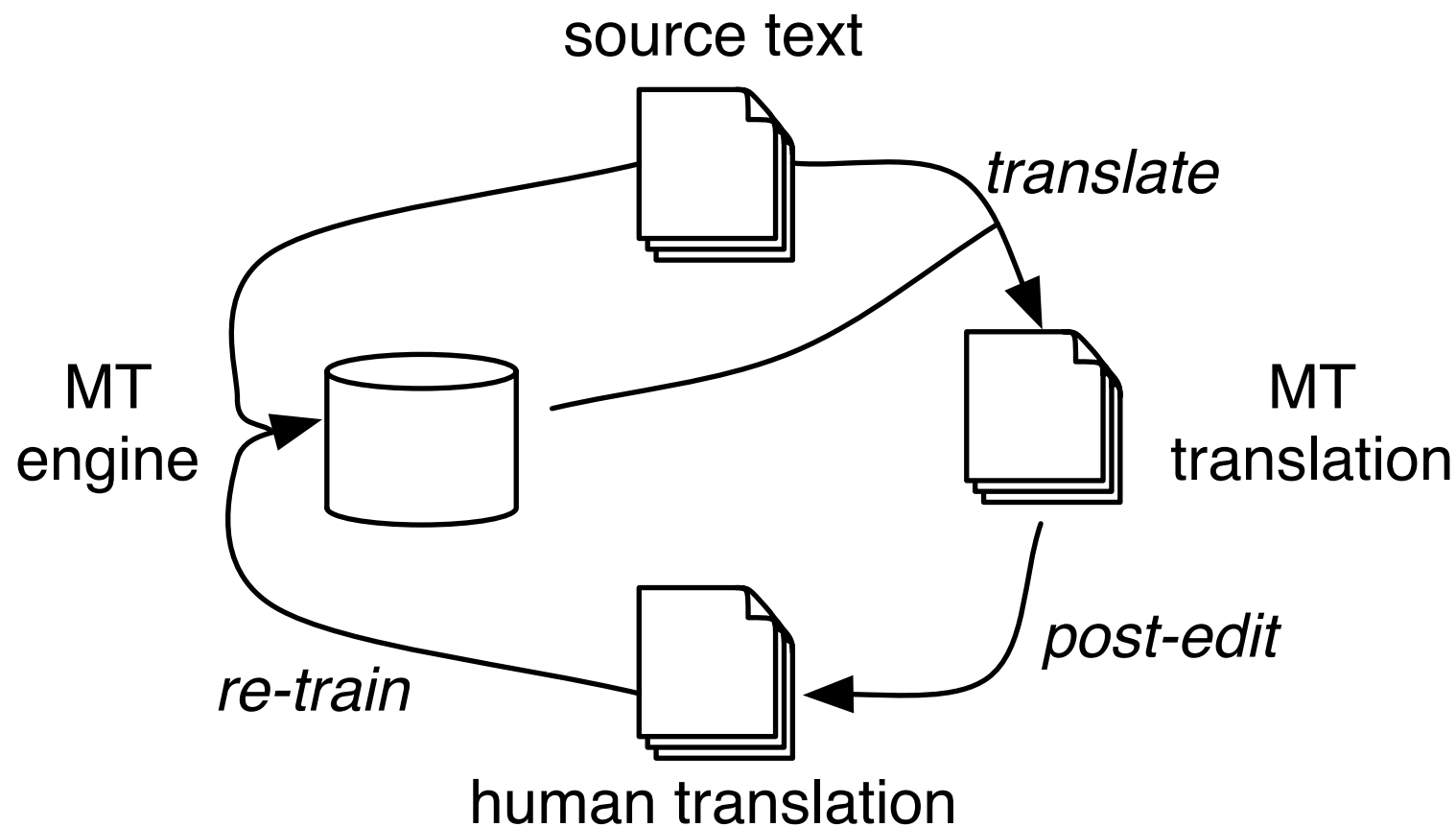
| | | Phrase-based | Hierarchical |
|---------------------|--------------|--------------|--------------|
| BLEU (Europarl) | fr-en | 25.10 | 24.58 |
| | de-en | 18.11 | 17.99 |
| | es-en | 25.81 | 25.17 |
| | de-en | 18.11 | 17.99 |
| | cs-en | 18.00 | 17.86 |
| Phrase-table size | fr-en | 2.5GB | 20.0GB |
| Decoding time (sec) | per sentence | 2.27 | 6.45 |
| | per word | 0.09 | 0.26 |

Advanced Features



- How do I get started?
- Experiment Management System
- Faster Training
- Faster Decoding
- Moses Server
- Data and domain adaptation
- Instructions to decoder
- Input formats
- Output formats
- Translation models
- **Incremental Training**

Incremental Training



Incremental Training

- Incremental word alignment
 - requires modified version of GIZA++
(available at <http://code.google.com/p/inc-giza-pp/>)
 - only works for HMM alignment (not the common IBM Model 4)
- Translation model is defined by parallel corpus

```
PhraseDictionaryBitextSampling \  
  path=/path/to/corpus \  
  L1=source language extension \  
  L2=target language extension
```


Update Word Alignment

144



- Uses original word alignment models
(with additional model files stored after training)
- Incremental GIZA++ loads model
- New sentence pairs is aligned on the fly
- Typically, GIZA++ processes are run in both directions, symmetrized

Update Translation Model

- Translation table is stored as word-aligned parallel corpus
- Update = add word aligned sentence pair
- Updating a running Moses instance via XML RPC

Acknowledgements

146



Moses Developers

| | | | |
|---------------------|--------------------------|----------------------|-------------------------|
| Abhishek Arun | Adam Lopez | Ales Tamchyna | Alex |
| Amittai Axelrod | Ankit Srivastava | Anthony Rousseau | Benjamin Gottesman |
| Barry Haddow | Ondrej Bojar | Chris Callison-Burch | Christine Corbett |
| Christian Hardmeier | Christian Federmann | Lane Schwartz | David Talbot |
| Edmund Huber | Evan Herbst | Andreas Eisele | Eva Hasler |
| Frederic Blain | Brooke Cowan | Grace M. Ngai | Kenneth Heafield |
| Hieu Hoang | H. Leal Fontes | Holger Schwenk | Josh Schroeder |
| Jean-Baptiste Fouet | Joern Wuebker | Jorge Civera | Konrad Rawlik |
| Abby Levenberg | Alexandra Birch | Bo Fu | M.J.Bellino-Machado |
| Mauro Cettolo | Marcello Federico | Michael Auli | John Joseph Morgan |
| Mark Fishel | Gabriele Antonio Musillo | Miles Osborne | Nadi Tomeh |
| Nicola Bertoldi | Oliver Wilson | Pascual Martinez | Philipp Koehn |
| Phil Williams | Bruno Pouliquen | Raphael Payen | Chris Dyer |
| Joao Lus Rosas | Rico Sennrich | Herve Saint-Amand | Felipe Sanchez Martinez |
| Sara Stymne | Steven B. Parks | Steven Buraje Poggel | Andre Lynum |
| Yizhao Ni | David Kolovratnak | Sergio Penkale | Stephan |
| Suzy Howlett | Wade Shen | Yang Gao | Tsuyoshi Okita |
| Alexander Fraser | Richard Zens | | |

FOR ICRON 2014

148

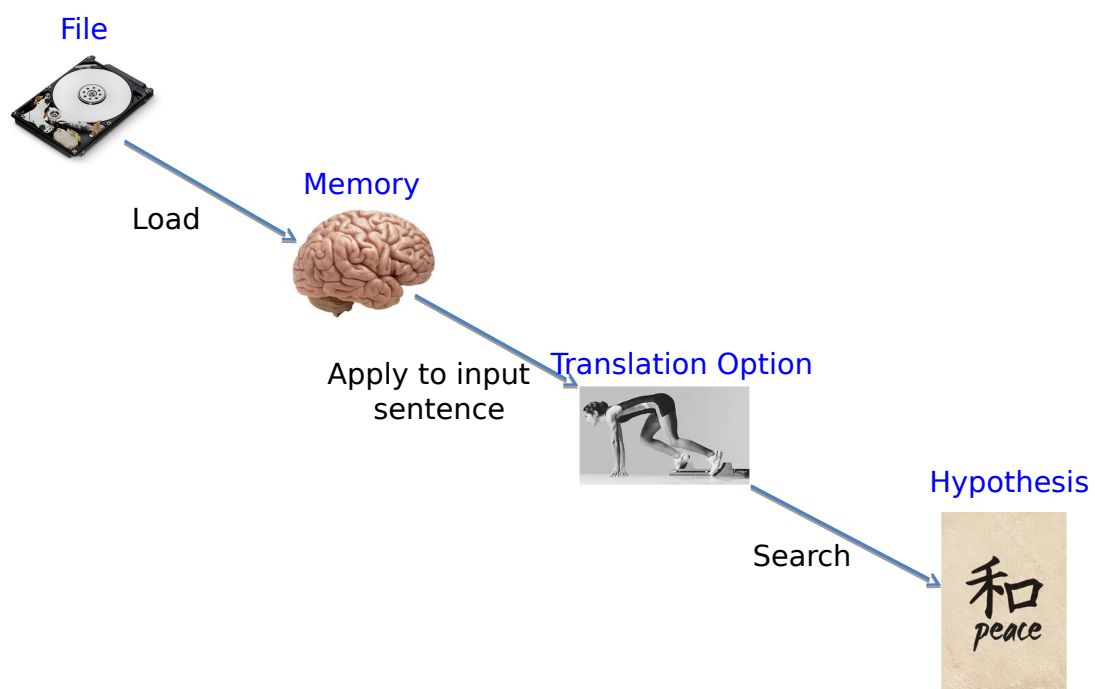


Feature Functions

What is a feature function?

Gives a score to a translation. - partial translation - completed

Timeline of Translation Rule



Feature Functions

