The Foundation for Statistical Machine Translation at MIT

Philipp Koehn

koehn@csail.mit.edu

Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology

Outline

- **Overview: SMT at MIT**
- Phrase Model
- Implementation Issues
- Parameter Tuning
- Arabic
- Chinese
- Analysis

People Working On SMT at MIT

- Michael Collins (Professor)
- Philipp Koehn (Postdoc)
- Brooke Cowan (Graduate Student)
- Luke Zettlemoyer (Graduate Student)
- Ivona Kucerova (Graduate Student)
- Chao Wang (Research Scientist)

MT Eval 2004 Effort

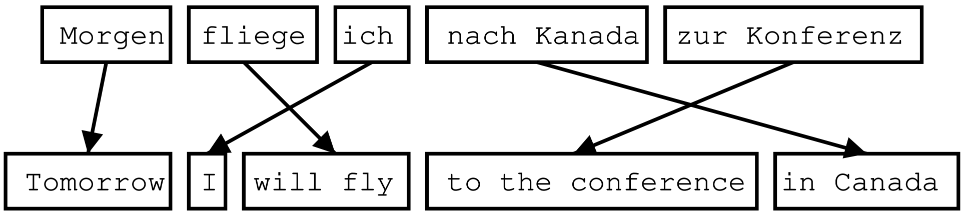
- 3-month effort by Philipp Koehn at MIT
  - partly funded by NTT Corporation, Japan
  - builds on work done as graduate student at ISI
- Focus:
  - make things work with large corpora
  - efficiency (full system train/tune automatically in less than 1 week)

⇒ Effective system as basis for future research
  - in many ways a “first cut”
  - not many alternatives explored

Outline

- Overview: SMT at MIT
- **Phrase Model**
- Implementation Issues
- Parameter Tuning
- Arabic
- Chinese
- Analysis

Phrase-Based Translation



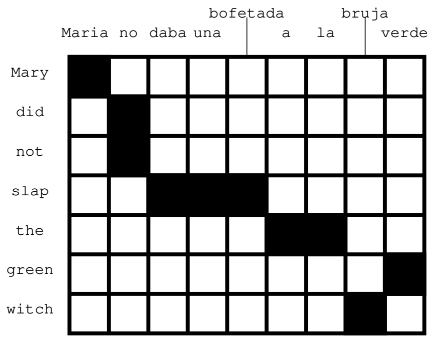
- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

Overview: Training

- Learning the phrase translation table
  - find word alignment with GIZA++
  - heuristically symmetrize word alignments
  - collect phrase pairs consistent with word alignment
  - various methods of scoring
- Other model components
  - reordering model
  - language model
  - word penalty

How to Learn the Phrase Translation Table?

- Start with the word alignment:

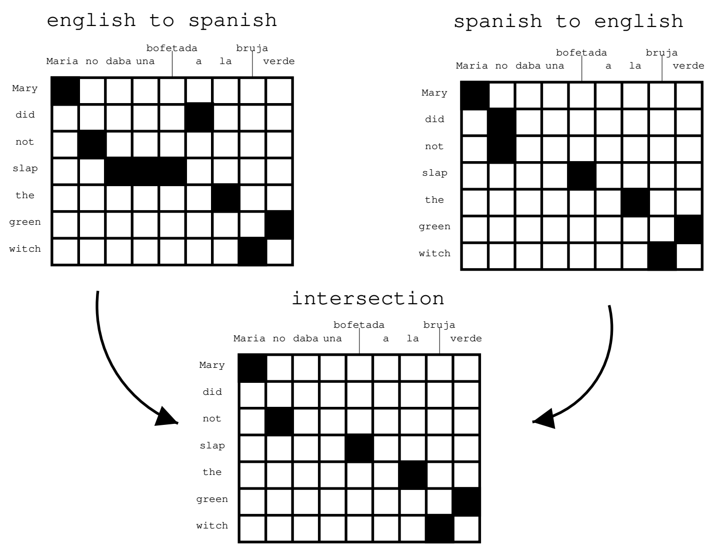


- Collect all phrase pairs that are consistent with the word alignment

Word Alignments from GIZA++

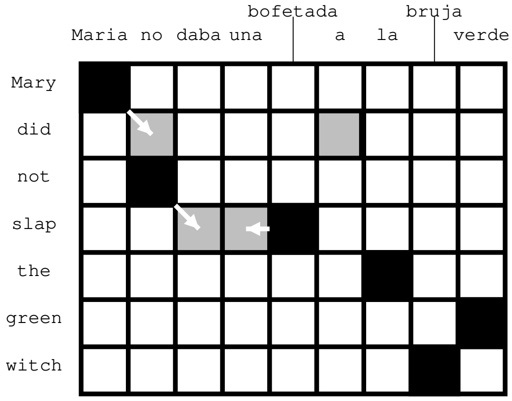
- Use IBM Model 4 implementation GIZA++
- Run GIZA++ bidirectionally
  - foreign to English
  - English to foreign
- Intersect alignments
- Grow additional alignment points

Intersection of GIZA alignments



- Intersection of bidirectional alignments

Improved Word Alignments (2)



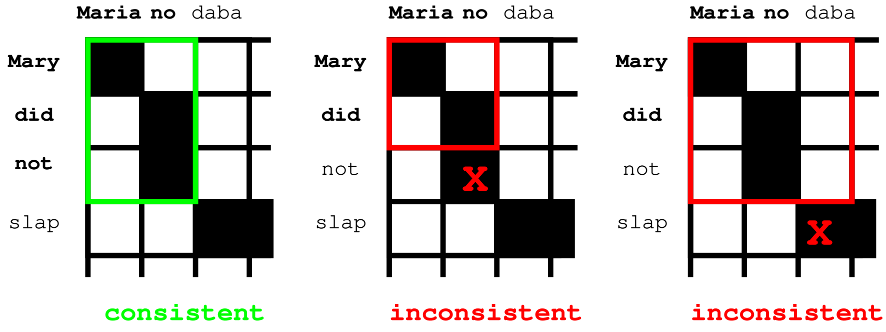
- Grow additional alignment points

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 )
```

Consistent with Word Alignment

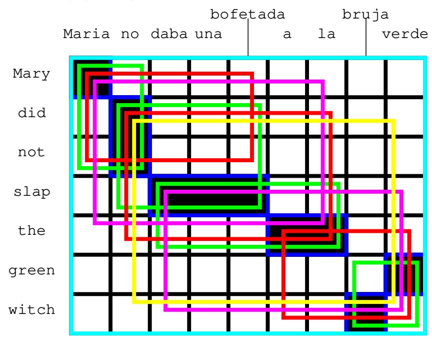


- Consistent with the word alignment :=

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

$$(\bar{e}, \bar{f}) \in BP \Leftrightarrow \quad \forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f}$$
$$\text{AND} \quad \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e}$$

Collect Phrase Pairs

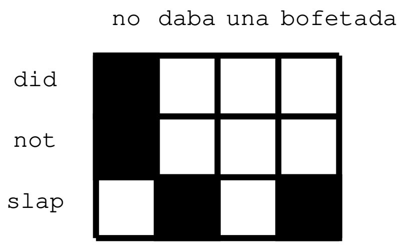


(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap), (no daba una bofetada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

Probability Distribution of Phrase Pairs

- We need a probability distribution over the collected phrase pairs
- One choice: relative frequency of collected phrases: $\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f},\bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f},\bar{e})}$
- or, conversely $\phi(\bar{e}|\bar{f})$
- Also: use lexical translation probabilities

Lexical Translation



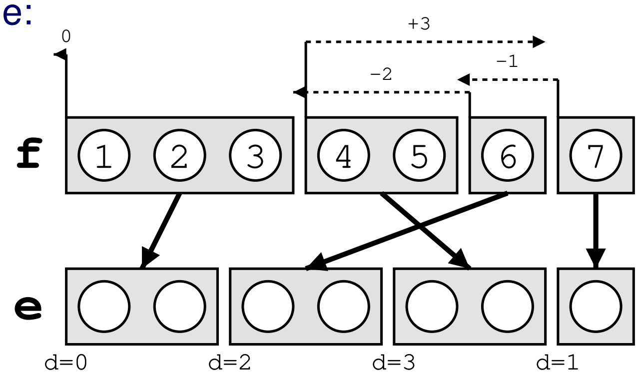
- Lexical weighting:

$$\text{lex}(\bar{f}|\bar{e},a) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j)\in a\}|} \sum_{\forall(i,j)\in a} w(f_i|e_j)$$

$\text{lex}(\text{no daba una bofetada}|\text{did not slap}) =$
$\frac{1}{2}(w(\text{no}|\text{did}) + w(\text{no}|\text{not}))\times$
$\times w(\text{daba}|\text{slap})\times$
$\times w(\text{una}|\text{NULL})\times$
$\times w(\text{bofetada}|\text{slap})$

Reordering Model

- Based on word position of foreign phrases
  - d = abs(last word position of previously translated phrase + 1 - first word position of newly translated phrase)
  - $p_D = \text{exp}(-\sum_i d_i)$
- Example:

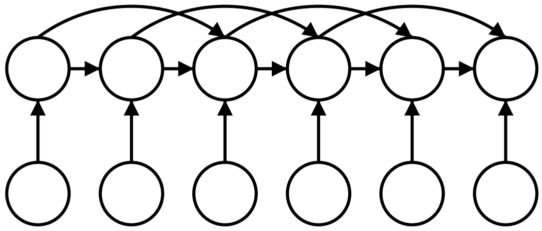


Language Model

- Trigram language model on words
- Used SRI language modeling toolkit
- Kneser-Ney smoothing

Re-Capitalizing

- Graphical Model
  - $p(\text{Egypt}|\text{egypt})$
  - $p(\text{Egypt}|\text{river in})$



- Special handling of headlines

Outline

- Overview: SMT at MIT
- Phrase Model
- **Implementation Issues**
- Parameter Tuning
- Arabic
- Chinese
- Analysis

Size of the Phrase Table

- Phrase table does not fit in memory

$\Rightarrow$ Estimation of $\phi(e|f)$ on disk

  - sort extracted phrase by f
  - only load all pairs involving one f at one time

$\Rightarrow$ Only load partial table for decoding

  - find all foreign phrase f in test corpus
  - only load phrase pairs involving these f

Multiple Phrase Translation Scores

- Many methods to score phrase translation
  - phrase translation $\phi(e|f)$
  - phrase translation $\phi(f|e)$
  - lexical translation lex$(e|f)$
  - lexical translation lex$(f|e)$
  - word penalty
  - phrase penalty
- All scores are combined (and weighted) while loading the phrase translation table into memory of decoder
- Phrase table pruning with the help of unigram LM score

Integration of Number Translator

- Use XML notation to allow alternate translations

```
es kostet <NUMBER english=''3.99''>3,99</NUMBER> euro
```

- Add these to the phrase translation table during run time
- Scoring can also be provided by XML

```
es kostet <NUMBER prob=100 english=''3.99''>3,99</NUMBER> euro
```


Outline

- Overview: SMT at MIT
- Phrase Model
- Implementation Issues
- **Parameter Tuning**
- Arabic
- Chinese
- Analysis

Knowledge Sources

- Many different knowledge sources useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word penalty
  - additional language models
  - additional features

Log-Linear Models

- IBM Models provided mathematical justification for factoring components together

  $$p_{LM} \times p_{TM} \times p_D$$

- These may be weighted

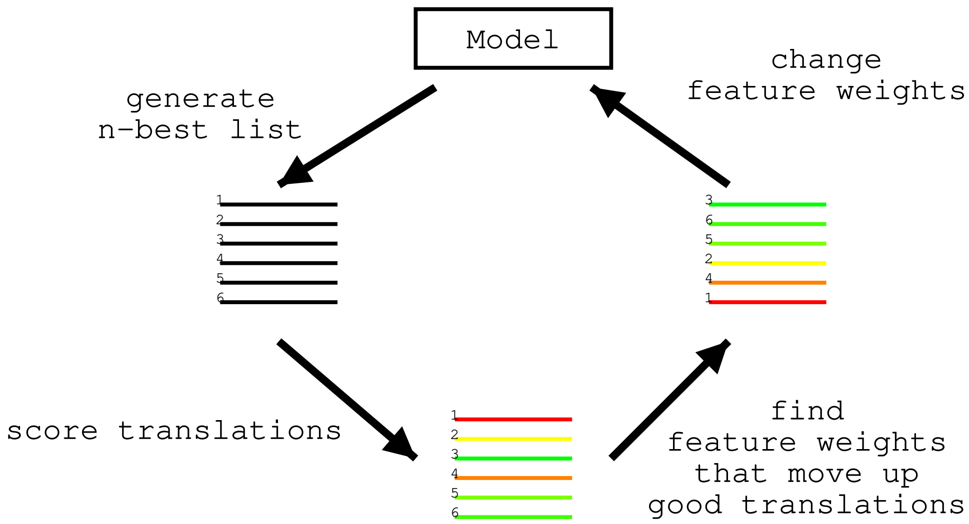
  $$p_{LM}^{\lambda_{LM}} \times p_{TM}^{\lambda_{TM}} \times p_D^{\lambda_D}$$

- Many components p_i with weights λ_i

  $\Rightarrow \prod_i p_i^{\lambda_i} = exp(\sum_i \lambda_i log(p_i))$

  $\Rightarrow log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i log(p_i)$

Learn Feature Weights



Parameter Tuning

- Reimplemented Minimum Error Rate Training

```
while not converged
  for i = 1 .. 20
    set starting point
    while not converged
      find weight change with biggest gain in BLEU
      apply weight change
```

- Some details
  - 300-1000 sentences in development set
  - 19 random starting points, plus best from last iteration
  - typically 10-20 iterations
  - typically 5-10 improvements from each starting point
- Assessment
  - it basically works (better than manually choosing weights)
  - weight settings very unstable
  - we are actively looking for better methods

Outline

- Overview: SMT at MIT
- Phrase Model
- Implementation Issues
- Parameter Tuning
- **Arabic**
- Chinese
- Analysis

Arabic: Data Used

- Arabic news translation (LDC2003E05,LDC2004E11)
- Ummah news corpus (LDC2002E48)
- Arabic treebank translation (LDC2003T07)
- UN corpus (LDC2004E13)
- Multiple translation corpus (LDC2002E54)

Arabic: System Components

- reordering model
- language model trained on all data
- phrase translation model f$\longrightarrow$e
- phrase translation model e$\longrightarrow$f
- word translation model f$\longrightarrow$e
- word translation model e$\longrightarrow$f
- word penalty

Arabic: Language Specific Work

- Conversion of Unicode into Buckwalter format
- Nothing else
- Trained first model two weeks before evaluation

Outline

- Overview: SMT at MIT
- Phrase Model
- Implementation Issues
- Parameter Tuning
- Arabic
- **Chinese**
- Analysis

Chinese: Data Used

- FBIS (LDC2003E14)
- Hongkong Hansards (LDC2000T50)
- Hongkong News (LDC2000T46)
- UN corpus (LDC2003E11)
- LDC Named Entity corpus (LDC2002E01)
- Xinhua News Corpus (LDC2002E18)
- Sinorama (LDC2002E58)
- Translation Lexicon (LDC2002E27)
- Multiple Translation (LDC2003E04,LDC2003E08

Chinese: System Components

- reordering model
- language model trained on all data
- language model trained on news data
- phrase translation model f$\rightarrow$e
- phrase translation model e$\rightarrow$f
- word translation model e$\rightarrow$f
- word penalty

Chinese: Language Specific Work

- Used ISI preprocessor
- Used ISI segmenter (which is based on LDC segmenter)
- Used ISI sentence aligner
- Built a number translator (slightly buggy)

Outline

- Overview: SMT at MIT
- Phrase Model
- Implementation Issues
- Parameter Tuning
- Arabic
- Chinese
- **Analysis**

Analysis: Keys to good performance

- Flexible phrase translation scoring
  - adding $\phi(e|f)$ caused biggest gain from a model change
- Automatic parameter tuning
  - if more than 3, weights can not be effectively optimized by hand

What we tried but did not work

- Not much: we did not try many alternatives
- Using all the data
  - using all UN data decreased score
  - ⇒ corpus weighting needed

What will help

- Bigger language model
- Corpus weighting
- Experimentation with alternative ...
  - GIZA++ training
  - word alignment heuristic
  - phrase scoring (esp. lexical scoring)
  - recapitalizing
- Better reordering model
- Better parameter tuning
- Using syntax for restructuring
- Language-specific preprocessing