Statistical Machine Translation LING-462/COSC-482 Week 4: Phrase-based statistical machine translation

Achim Ruopp

achim.ruopp@Georgetown.edu

Agenda

- Language in ten minutes: Janet Liu: German
- Planning further "Language in ten minutes" presentations
- Step-by-step walk trough IBM Model 1 word alignment

- Break -

- Phrase-based statistical machine translation
- MT internships?

IBM Model 1

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

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

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

  - parameter ϵ is a normalization constant

EM Expectation step ground work

- Deriving the formula to calculate the alignment probability based on word translation probabilities

$$p(a|e,f) = \frac{p(a,e,f)}{p(e,f)} = \frac{p(a,e,f)p(f)}{p(e,f)p(f)}$$

$$= \frac{p(e,a|f)}{p(e|f)} = \ \ldots = \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}$$

IBM Model 1 and EM: Maximization Step

- Now we have to collect counts
- Evidence from a sentence pair **e,f** that word e is a translation of word f:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_a p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)})$$

- With the same simplication as before:

$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

IBM Model 1 and EM: Maximization Step

After collecting these counts over a corpus, we can estimate the model:

$$t(e|f;\mathbf{e},\mathbf{f}) = \frac{\sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}{\sum_e \sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}$$

Walk-trough with Excel sheet

Noisy channel model

$$argmax_f p(f|e) = argmax_f p(e|f)p(f)$$

Language Model p(f)

- Given the target language corpus

  <s> la maison </s>

  <s> la lune </s>

  <s> une maison </s>

Unigrams maximum likelihood probabilities

w	p(w)
<s>	0.25
la	0.167
maison	0.167
lune	0.084
une	0.084
</s>	0.25

Bigram counts

w1/w2	<s>	la	maison	lune	une	</s>	Total
<s>		2			1		3
la			1	1			2
maison						2	2
lune						1	1
une			1				1
</s>							0

Bigram maximum likelihood probabilities

p(w2|w1)	<s>	la	maison	lune	une	</s>	Total
<s>		0.667			0.333		1.000
la			0.500	0.500			1.000
maison						1.000	1.000
lune						1.000	1.000
une			1.000				1.000
</s>							0.000

Bigram maximum likelihood probabilities

- What happens if translation model suggests “une lune” to translate “a moon”?
- Bigram model estimates probability zero!
- Language model smoothing/ interpolation/backoff needed

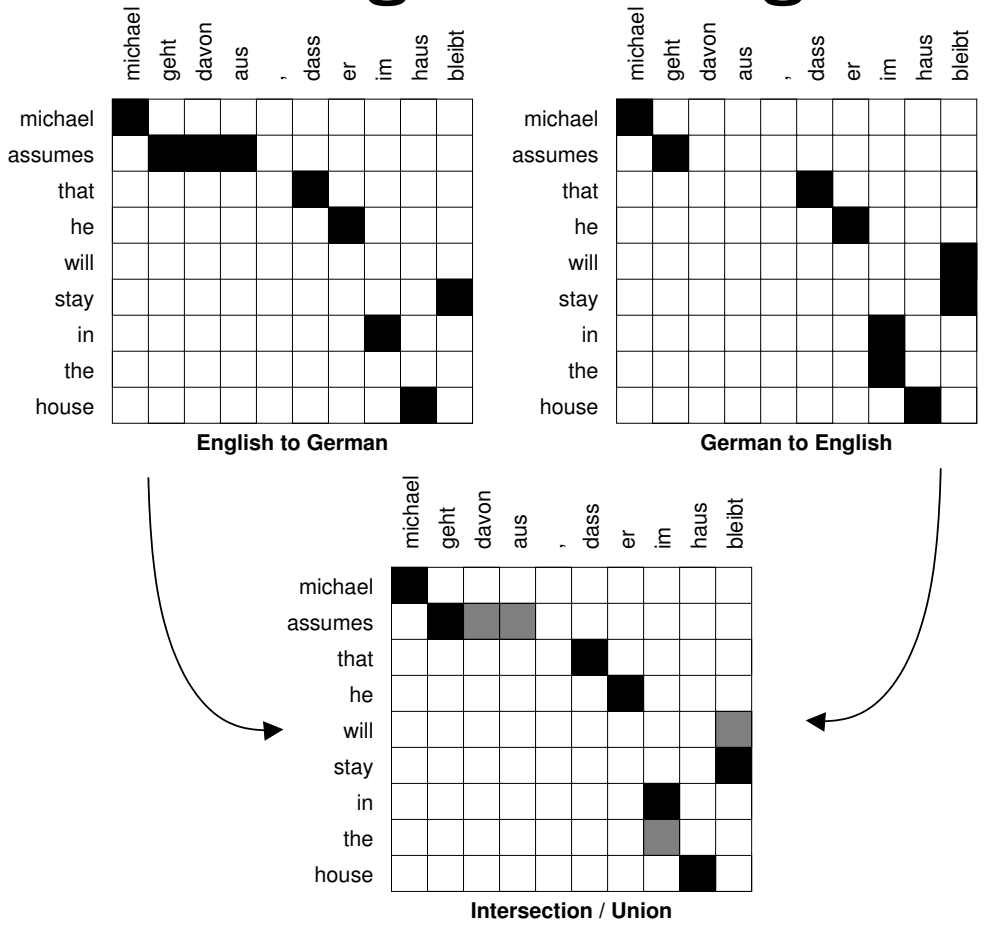
Bigram add-one counts

w1/w2	<s>	la	maison	lune	une	</s>	Total
<s>	1	3	1	1	2	1	9
la	1	1	2	2	1	1	8
maison	1	1	1	1	1	3	8
lune	1	1	1	1	1	2	7
une	1	1	2	1	1	1	7
</s>	1	1	1	1	1	1	6

Bigram add-one probabilities

p(w2|w1)	<s>	la	maison	lune	une	</s>	Total
<s>	0.111	0.333	0.111	0.111	0.222	0.111	**1.000**
la	0.125	0.125	0.250	0.250	0.125	0.125	**1.000**
maison	0.125	0.125	0.125	0.125	0.125	0.375	**1.000**
lune	0.143	0.143	0.143	0.143	0.143	0.286	**1.000**
une	0.143	0.143	0.286	0.143	0.143	0.143	**1.000**
</s>	0.167	0.167	0.167	0.167	0.167	0.167	**1.000**

Symmetrizing Word Alignments



- Intersection of GIZA++ bidirectional alignments
- Grow additional alignment points [Och and Ney, CompLing2003]

Growing heuristic

grow-diag-final(e2f,f2e)

1: neighboring = {(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)}
2: alignment A = intersect(e2f,f2e); grow-diag(); final(e2f); final(f2e);

grow-diag()

1: **while** new points added **do**
2: **for all** English word $e \in [1...e_n]$, foreign word $f \in [1...f_n]$, $(e, f) \in A$ **do**
3: **for all** neighboring alignment points (e_{new}, f_{new}) **do**
4: **if** (e_{new} unaligned OR f_{new} unaligned) AND (e_{new}, f_{new}) $\in$ union(e2f,f2e) **then**
5: add (e_{new}, f_{new}) to A
6: **end if**
7: **end for**
8: **end for**
9: **end while**

final()

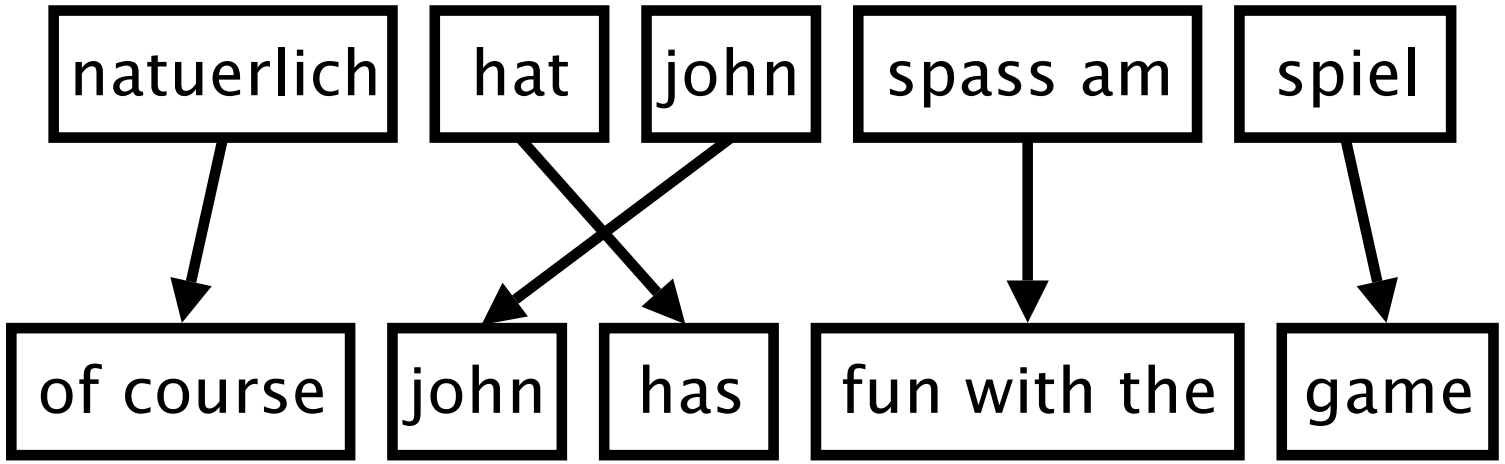
1: **for all** English word $e_{\text{new}} \in [1...e_n]$, foreign word $f_{\text{new}} \in [1...f_n]$ **do**
2: **if** (e_{new} unaligned OR f_{new} unaligned) AND (e_{new}, f_{new}) $\in$ union(e2f,f2e) **then**
3: add (e_{new}, f_{new}) to A
4: **end if**
5: **end for**

PHRASE-BASED STATISTICAL MACHINE TRANSLATION

Motivation

- Word-Based Models translate *words* as atomic units
- Phrase-Based Models translate *phrases* as atomic units
- Advantages:
  - many-to-many translation can handle non-compositional phrases
  - use of local context in translation
  - the more data, the longer phrases can be learned
- "Standard Model", used by Google Translate and others

Phrase-Based Model



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

Phrase Translation Table

- Main knowledge source: table with phrase translations and their probabilities
- Example: phrase translations for natuerlich

Translation	Probability $\phi(\bar{e}\|\bar{f})$
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

Real Example

- Phrase translations for den Vorschlag learned from the Europarl corpus:

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

  - lexical variation (proposal vs suggestions)
  - morphological variation (proposal vs proposals)
  - included function words (the, a, ...)
  - noise (it)

Linguistic Phrases?

- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases, ...)

- Example non-linguistic phrase pair

  spass am → fun with the

- Prior noun often helps with translation of preposition

- Experiments show that limitation to linguistic phrases hurts quality

Probabilistic Model

- Bayes rule

$$\begin{aligned}\mathbf{e}_{\text{best}} &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}|\mathbf{f}) \\ &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{f}|\mathbf{e})\ p_{\text{LM}}(\mathbf{e})\end{aligned}$$

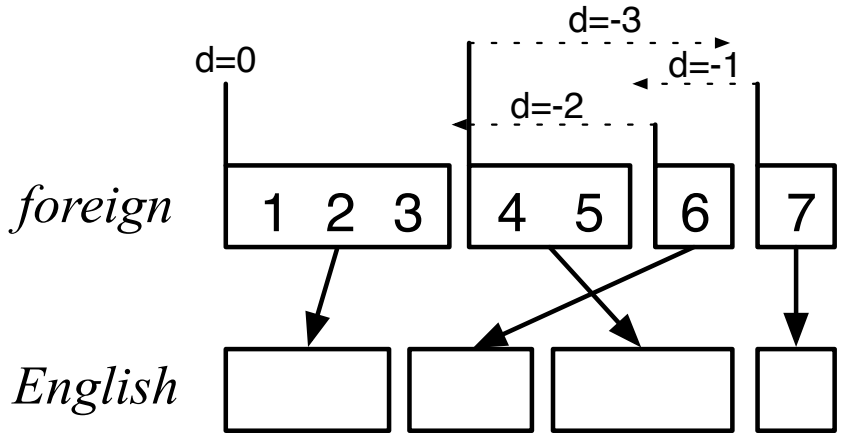
  - translation model $p(\mathbf{e}|\mathbf{f})$
  - language model $p_{\text{LM}}(\mathbf{e})$

- Decomposition of the translation model

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(start_i - end_{i-1} - 1)$$

  - phrase translation probability ϕ
  - reordering probability d

Distance-Based Reordering



phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

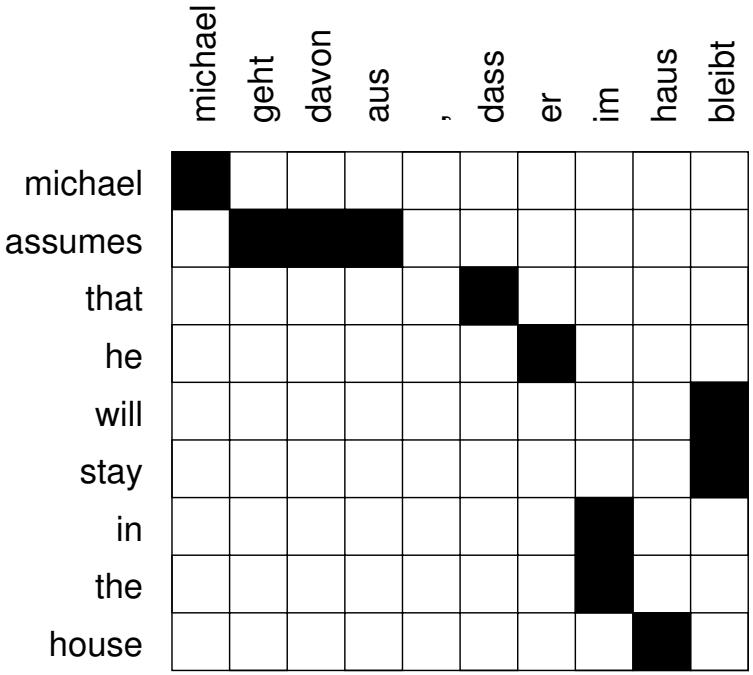
Scoring function: $d(x) = \alpha^{|x|}$ — exponential with distance

Learning a Phrase Translation Table

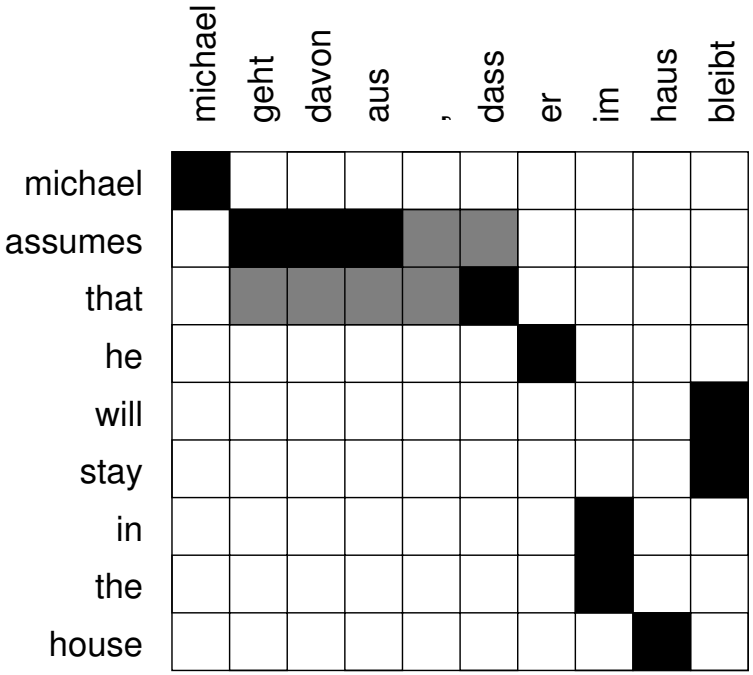
- Task: learn the model from a parallel corpus

- Three stages:
  - word alignment: using IBM models or other method
  - extraction of phrase pairs
  - scoring phrase pairs

Word Alignment



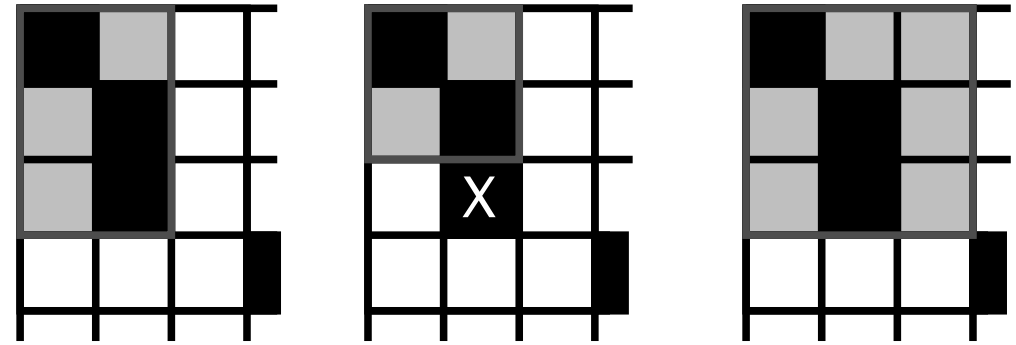
Extracting Phrase Pairs



extract phrase pair consistent with word alignment:

assumes that / geht davon aus , dass

Consistent



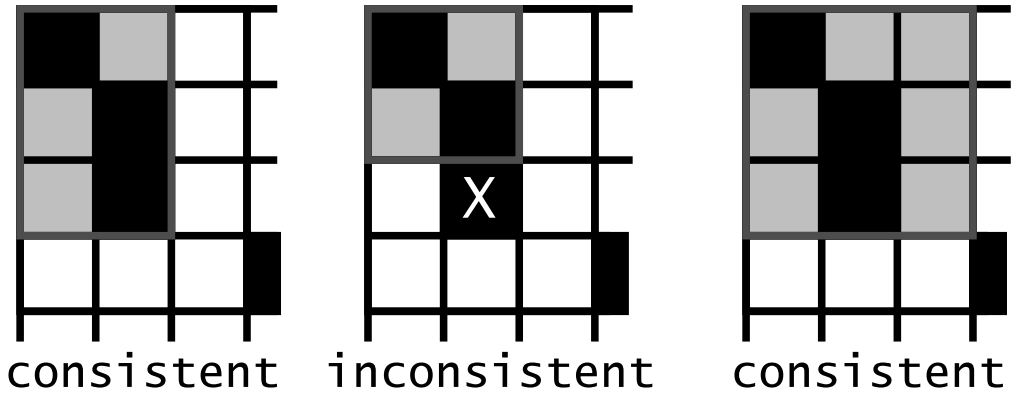
ok

violated
one alignment
point outside

ok
unaligned
word is fine

All words of the phrase pair have to align to each other.

Consistent



Phrase pair $(\bar{e}, \bar{f})$ consistent with an alignment A, if all words $f_1, ..., f_n$ in $\bar{f}$ that have alignment points in A have these with words $e_1, ..., e_n$ in $\bar{e}$ and vice versa:

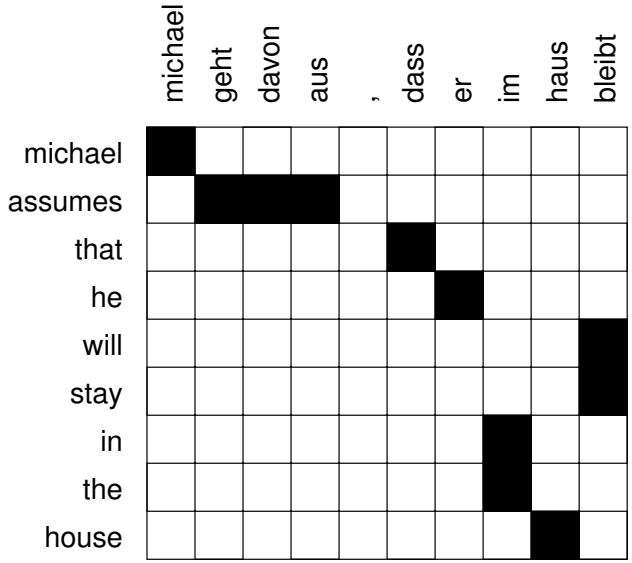
$(\bar{e}, \bar{f})$ consistent with $A \Leftrightarrow$

$$\forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f}$$

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

$$\text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A$$

Phrase Pair Extraction



Smallest phrase pairs:

michael — michael

assumes — geht davon aus / geht davon aus ,

that — dass / , dass

he — er

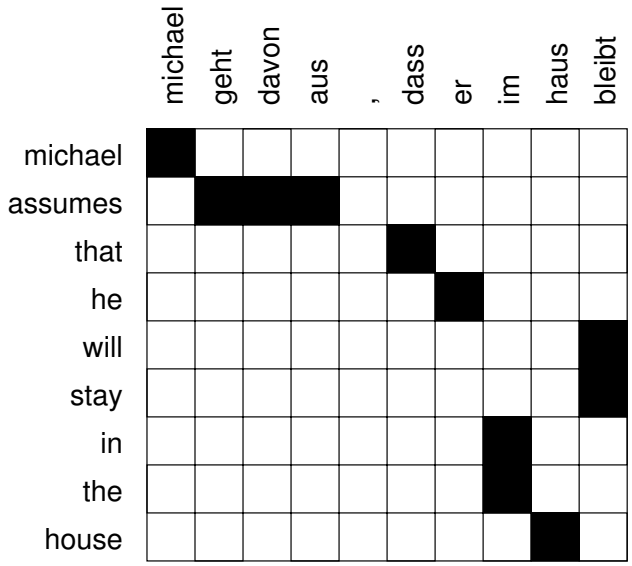
will stay — bleibt

in the — im

house — haus

unaligned words (here: German comma) lead to multiple translations

Larger Phrase Pairs



michael assumes — michael geht davon aus / michael geht davon aus ,
assumes that — geht davon aus , dass ; assumes that he — geht davon aus , dass er
that he — dass er / , dass er ; in the house — im haus
michael assumes that — michael geht davon aus , dass
michael assumes that he — michael geht davon aus , dass er
michael assumes that he will stay in the house — michael geht davon aus , dass er im haus bleibt
assumes that he will stay in the house — geht davon aus , dass er im haus bleibt
that he will stay in the house — dass er im haus bleibt ; dass er im haus bleibt ,
he will stay in the house — er im haus bleibt ; will stay in the house — im haus bleibt

Scoring Phrase Translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e}, \bar{f})}{\sum_{\bar{f}_i} \text{count}(\bar{e}, \bar{f}_i)}$$

Size of the Phrase Table

- Phrase translation table typically bigger than corpus

  ... even with limits on phrase lengths (e.g., max 7 words)

→ Too big to store in memory?

- Solution for training
  - extract to disk, sort, construct for one source phrase at a time
- Solutions for decoding
  - on-disk data structures with index for quick look-ups
  - suffix arrays to create phrase pairs on demand

Weighted Model

- Described standard model consists of three sub-models
  - phrase translation model $\phi(\bar{f}|\bar{e})$
  - reordering model d
  - language model $p_{LM}(e)$

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(start_i - end_{i-1} - 1) \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1...e_{i-1})$$

- Some sub-models may be more important than others

- Add weights λ_ϕ, λ_d, λ_{LM}

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi}\ d(start_i - end_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1...e_{i-1})^{\lambda_{LM}}$$

Log-Linear Model

- Such a weighted model is a log-linear model:

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

- Our feature functions
  - number of feature function $n = 3$
  - random variable $x = (e, f, start, end)$
  - feature function $h_1 = \log \phi$
  - feature function $h_2 = \log d$
  - feature function $h_3 = \log p_{\text{LM}}$

Weighted Model as Log-Linear Model

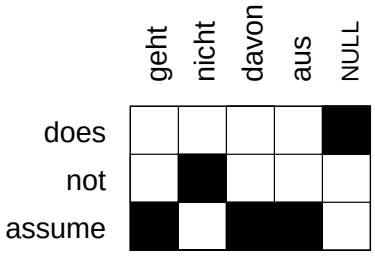
$$p(e,a|f) = \exp(\lambda_\phi \sum_{i=1}^{I} \log \phi(\bar{f}_i|\bar{e}_i) +$$
$$\lambda_d \sum_{i=1}^{I} \log d(a_i - b_{i-1} - 1) +$$
$$\lambda_{LM} \sum_{i=1}^{|e|} \log p_{LM}(e_i|e_1 ... e_{i-1}))$$

More Feature Functions

- Bidirectional alignment probabilities: $\phi(\bar{e}|\bar{f})$ and $\phi(\bar{f}|\bar{e})$

- Rare phrase pairs have unreliable phrase translation probability estimates

  $\rightarrow$ lexical weighting with word translation probabilities



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

More Feature Functions

- Language model has a bias towards short translations

  $\rightarrow$ word count: $\text{wc}(e) = \log |\mathbf{e}|^{\omega}$

- We may prefer finer or coarser segmentation

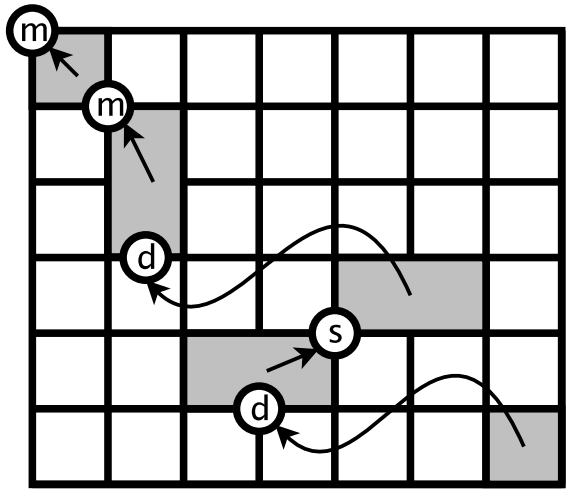
  $\rightarrow$ phrase count $\text{pc}(e) = \log |I|^{\rho}$

- Multiple language models

- Multiple translation models

- Other knowledge sources

Lexicalized Reordering



- Distance-based reordering model is weak

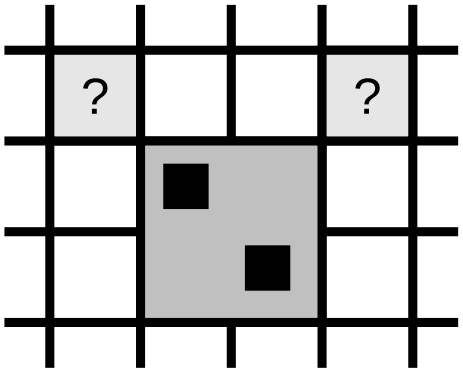
  $\rightarrow$ learn reordering preference for each phrase pair

- Three orientations types: (m) monotone, (s) swap, (d) discontinuous

$$\text{orientation} \in \{m, s, d\}$$

$$p_o(\text{orientation} | \bar{f}, \bar{e})$$

Learning Lexicalized Reordering



- Collect orientation information during phrase pair extraction
  - if word alignment point to the top left exists → **monotone**
  - if a word alignment point to the top right exists→ **swap**
  - if neither a word alignment point to top left nor to the top right exists → neither monotone nor swap → **discontinuous**

Learning Lexicalized Reordering

- Estimation by relative frequency

$$p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} count(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} count(o, \bar{e}, \bar{f})}$$

- Smoothing with unlexicalized orientation model $p(\text{orientation})$ to avoid zero probabilities for unseen orientations

$$p_o(\text{orientation}|\bar{f}, \bar{e}) = \frac{\sigma\ p(\text{orientation}) + count(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o count(o, \bar{e}, \bar{f})}$$

Summary

- What we have now: statistical models
  - Word-based translation models
  - Phrase-based translation models
  - N-gram language models
  - Noisy channel model
  - Log-linear model
- Next: decoding
  - How do we find the most-likely or top-n most likely translations?