USC Viterbi
School of Engineering
Department of Computer Science

CSCI 544
Applied Natural Language Processing

Mohammad Rostami
USC Computer Science Department

Slides adopted from slides by Michael Collins and Philipp Koehn

Logistical Notes

- Quiz8: Oct 7 & Oct 11 sessions
- Midterm:

- All lectures until Oct 21 (including)
- Date: October 28^{th} in class
- Exam length: 100 minutes and ~20 question
- Covers all sessions of the course
- Primarily problems aimed at evaluating your understanding; more challenging than quizzes; written response
- On Blackboard: 3-4 versions for each question in random order without possibility of returning to previous questions; in-person unless approved to participate remotely
- Remote students: be in a quite location; Camera and mic should be open with no virtual background

The Noisy Channel Model for MT

- Goal: translate from French (foreign) to English
- Generate a model $p(e \mid f)$ which estimates conditional probability of any English sentence given the French sentence f.
- Use the training corpus to set the parameters.

- Noisy channel Model:

$$p(e \mid f) = \frac{p(e,f)}{p(f)} = \frac{\overbrace{p(e)}^{\text{Language Model}}\overbrace{p(f \mid e)}^{\text{Translation Model}}}{\sum_e p(e)p(f \mid e)}$$

Decoding Problem

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

Translation Model: IBM Models

- How do we model the translation model?
  - In the parallel corpus, consider that for a pair, the English sentence has l words and the French sentence has m words
  - An alignment map determines which English word each French word originated from
  - An alignment a is $\{a_1, \dots a_m\}$, where $a_j \in \{0 \dots l\}$
  - Hence there are $(l+1)^m$ possible alignments
  - Ex: l = 6, M = 7, a= {2,3,4,5,6,6,0}, a(j) = i

Null Word

i = 1 2 3 4 5 6

$e =$ And the program has been implemented

$f =$ Le programme a ete mis en application

j = 1 2 3 4 5 6 7

Translation Model

- Total probability over all possible alignments

$$p(f \mid e, m) = \sum_{a \in \mathcal{A}} \underbrace{\overbrace{p(a \mid e, m)}^{\text{Alignment Distribution}} \overbrace{p(f \mid a, e, m)}^{\text{Conditional Translation Model}}}_{p(f, a \mid e, m)}$$

- We will model the conditional probabilities:

$p(a \mid e, m)$ and $p(f \mid a, e, m)$

$$p(f, a \mid e, m) = p(a \mid e, m) p(f \mid a, e, m)$$

$$p(f \mid e, m) = \sum_{a \in \mathcal{A}} p(a \mid e, m) p(f \mid a, e, m)$$

- Having computed the conditional probabilities:

$$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\underbrace{\sum_{a \in \mathcal{A}} p(f, a \mid e, m)}_{p(f, a \mid e, m)}}$$

Most Likely Alignment $\quad a^* = \arg\max_a p(a \mid f, e, m)$

IBM Model 1

- Equally likely Alignment Probability

$$p(a \mid e, m) = \frac{1}{(l+1)^m} \qquad p(f \mid e, m) = \sum_{a \in \mathcal{A}} p(a \mid e, m) p(f \mid a, e, m)$$

- Conditional Translation Model: lexical translation

Lexical Translations: Model Parameters

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

- Ex: $l = 6,\ m = 7$ $\qquad a = \{2, 3, 4, 5, 6, 6, 6\}$

$$e = \text{And the program has been implemented}$$
$$f = \text{Le programme a ete mis en application}$$

$$p(f \mid a, e) = t(Le \mid the) \times t(programme \mid program) \times t(a \mid has) \times t(ete \mid been) \times$$
$$t(mis \mid implemented) \times t(en \mid implemented) \times t(application \mid implemented)$$

IBM Model 1

- Lexical probability tables $t(f_j \mid e_{a_j})$

English	French	Probability
position	position	0.756715
position	situation	0.0547918
position	mesure	0.0281663
position	vue	0.0169303
position	point	0.0124795
position	attitude	0.0108907

IBM Model 2

- Non-uniform alignments: distortion parameters

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j}) \qquad p(a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j = i \mid j, l, m)$$

j's French word is connected from i's English word given the lengths

- Conditional Translation Model

$$p(f, a \mid e, m) = p(a \mid e, m) p(f \mid a, e, m)$$

$$p(f, a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m) \mathbf{t}(f_j \mid e_{a_j})$$

$$p(f \mid e, m) = \sum_{a \in \mathcal{A}} p(a \mid e, m) p(f \mid a, e, m)$$

IBM Model Parameter Estimation

- Input: sentence pairs $(e^{(k)}, f^{(k)})$
- Output: parameters $t(f|e)$ and $q(i|j,l,m)$
- Primary Challenge: alignments are not known

- Data annotation is expensive

$$\begin{aligned} e^{(100)} &= \text{And the program has been implemented} \\ f^{(100)} &= \text{Le programme a ete mis en application} \end{aligned}$$

- Expectation Maximization (EM) algorithm

IBM Model Parameter Estimation

- Assume the alignments are accessible

$$
\begin{aligned}
e^{(100)} &= \text{And the program has been implemented} \\
f^{(100)} &= \text{Le programme a ete mis en application} \\
a^{(100)} &= \langle 2, 3, 4, 5, 6, 6, 6 \rangle
\end{aligned}
$$

- We will have triplets $(e^{(k)}, f^{(k)}, a^{(k)})$
- ML estimates for parameters boils down to counting, ex, t(position|position)

$$
t_{ML}(f|e) = \frac{\text{Count}(e,\ f)}{\text{Count}(e)} \qquad q_{ML}(j|i,l,m) = \frac{\text{Count}(j|i,l,m)}{\text{Count}(i,l,m)}
$$

IBM Model Parameter Estimation

Ex:
e= the position
f=La position
a = {1,2}

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$\begin{aligned} c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\ c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\ c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\ c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j) \end{aligned}$$

English Position

French Position

where $\delta(k,i,j) = 1$ if $\boxed{a_i^{(k)} = j,\ 0}$ otherwise.

Pair Index

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Expectation Maximization

- Dempster et al., 1977: An algorithm for computing maximum likelihood from incomplete data:

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

- EM in a nutshell:

1. initialize model parameters, e.g., random
2. assign probabilities to the missing data
3. estimate model parameters from completed data
4. iterate steps 2–3 until convergence

EM Algorithm for MT

- We don't have the alignments:

1. The algorithm is **iterative**: we start with some arbitrary random choice for the q and t parameters. At each iteration we compute the "counts" based on the data together with our current parameter estimates. We then re-estimate the parameters with these counts, and iterate
2. $\delta(k, i, j)$ is defined as follows

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}$$

EM Algorithm for MT

- S ~ 10-20

For $s = 1 \dots S$

- Set all counts $c(\dots) = 0$
- For $k = 1 \dots n$
  - For $i = 1 \dots m_k$, For $j = 0 \dots l_k$

- Delta parameters:

$$\delta(k,i,j) = P(a_j^{(k)} = i | e^{(k)}, f^{(k)})$$

M-Step

$$\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\
c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\
c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j)
\end{aligned}$$

where

- EM would converge to local ML optimums

E-Step

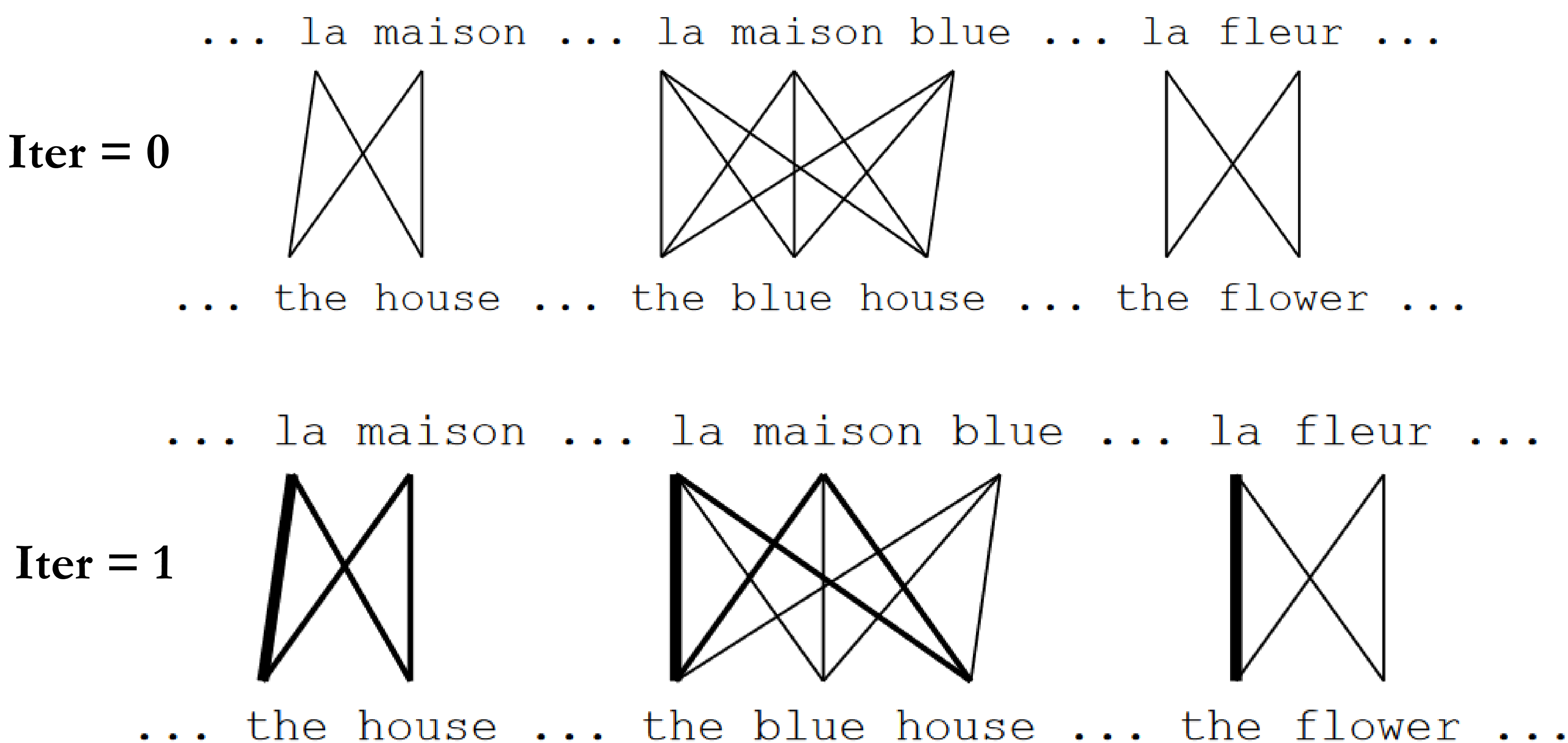
$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

- Recalculate the parameters:

$$t(f|e) = \frac{c(e,f)}{c(e)} \qquad q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$$

EM Algorithm for MT

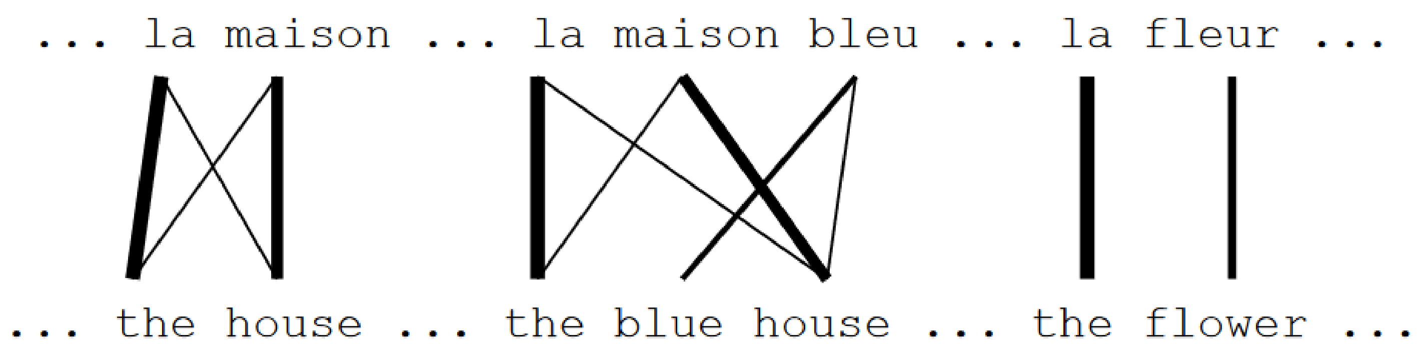
- Initialization: set all assignments equally likely
- Model learns ‘La’ is often aligned with ‘the’



EM Algorithm for MT

- After one more iteration `fleur’ is aligned with ‘flower’
- Convergence: after One more iteration

Iter = 2



Iter = 1

EM Algorithm for MT

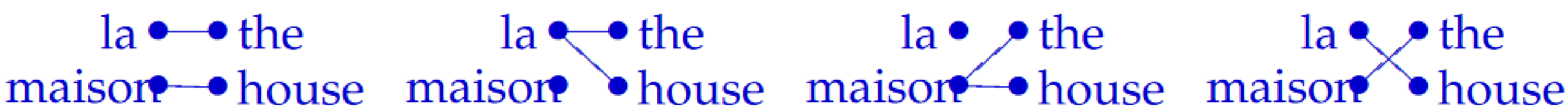
- EX: IBM 1

- **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

- **Alignments**

E-Step



$$p(\mathbf{e}, a|\mathbf{f}) = 0.56 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.035 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.08 \qquad p(\mathbf{e}, a|\mathbf{f}) = 0.005$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.824 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.052 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.118 \qquad p(a|\mathbf{e}, \mathbf{f}) = 0.007$$

- **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \qquad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \qquad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

M-Step

Model Evaluation

- Perplexity: derived from probability of the training data according to the model

$$\log_2 PP = -\sum_s \log_2 p(\mathbf{e}_s|\mathbf{f}_s)$$

- Ex:

	initial	1st it.	2nd it.	3rd it.	...	final
p(the haus|das haus)	0.0625	0.1875	0.1905	0.1913	...	0.1875
p(the book|das buch)	0.0625	0.1406	0.1790	0.2075	...	0.25
p(a book|ein buch)	0.0625	0.1875	0.1907	0.1913	...	0.1875
perplexity	4095	202.3	153.6	131.6	...	113.8

Phrase Based Translation Models

- Translation involves many phrase-based (PB) lexicons
- A PB lexicon pairs strings in one language with strings in another language, e.g.,

nach Kanada	$\leftrightarrow$	in Canada
zur Konferenz	$\leftrightarrow$	to the conference
Morgen	$\leftrightarrow$	tomorrow
fliege	$\leftrightarrow$	will fly
...		

- Improves upon word-to-word MT models of IBM

Building Phrase Level Alignment

- Representing alignments using matrices

English: Mary did not slap the green witch

Spanish: Maria no daba una bofetada a la bruja verde

Sp

En

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

IMB Model Alignment

Building Phrase Level Alignment

- Weaknesses of IBM model's alignments:

1. Noisy: not accurate

2. Many-to-One: many words in the source language can be mapped to a single word, i.e., for each source word we find one target word -> Many-to-Many

- Advantages

– many-to-many translation can handle non-compositional phrases, e.g., hot dog
– use of local context in translation
– the more data, the longer phrases can be learned

- Standard Model", used by Google Translate and others until about 2017

Building Phrase Level Alignment

- Approach

1. Train a model for $p(f|e)$ using IBM 2
2. Train a model for $p(e|f)$ using IBM 2
3. Extracting phrases: take intersection of the two alignments as a starting point and use them to grow alignments on the union of the alignments
4. Score the extracted phrases

Building Phrase Level Alignment

- Example

Alignment from $p(f \mid e)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did									
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Alignment from $p(e \mid f)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap					●				
the							●		
green									●
witch								●	

Building Phrase Level Alignment

- The final alignment, created by taking the intersection of the two alignments, then adding new points using the growing heuristics:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

- Note that the alignment is no longer many-to-one: potentially multiple Spanish words can be aligned to a single English word, and vice versa.

Heuristics for Growing Alignments

- Only explore alignment in union of p(f|e)and p(e|f) alignment
- Add one alignment point at a time
- Only add alignment points which align a word that currently has no alignment
- At first, restrict ourselves to alignment points that are “neighbors” (adjacent or diagonal) of current alignment points
- Later, consider other alignment points

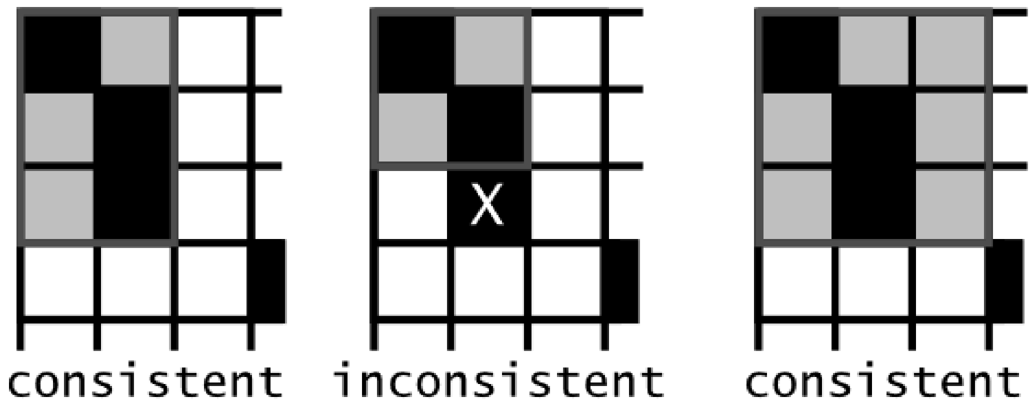
Extracting Phrase Pairs

- A phrase-pair consists of a sequence of English words, e, paired with a sequence of foreign words, f
- A phrase-pair (e,f) is consistent if: 1) there is at least one word in e aligned to a word in f; 2) there are no words in f aligned to words outside e; 3) there are no words in e aligned to words outside f, e.g., (Mary did not, Maria no) is consistent. (Mary did, Maria no) is not consistent
- We extract all consistent phrase pairs from the training example

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Extracting Phrase Pairs

- Consistent Phrases



ok	**violated** one alignment point outside	**ok** unaligned word is fine

Extracting Phrase Pairs

- Scoring Phrase Translations
  - Phrase pair extraction: collect all phrase pairs from the data
  - Phrase pair scoring: assign probabilities to phrase translations
  - Use empirical frequency

Phrase Lexicon Probabilities

- Probabilities for Phrase Pairs

$$t(f|e) = \frac{Count(f,e)}{Count(e)}$$

$$t(\text{daba una bofetada} \mid \text{slap}) = \frac{Count(\text{daba una bofetada}, \text{slap})}{Count(\text{slap})}$$

Phrase Lexicon Probabilities

- Real Example: Koehn, EACL 2006

- Translation table for “den Vorschlag”

English	$t(e\|f)$	English	$t(e\|f)$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

Phrase-Level Bilingual Dictionary

- Model is not limited to linguistic phrases (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair

$$\text{spass am} \rightarrow \text{fun with the}$$

- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality

EM for Phrase Based MT

- Heuristic set-up to build phrase translation table: (word alignment, phrase extraction, phrase scoring)
- Align phrase pairs directly with EM algorithm

– initialization: uniform model, all probabilities are equally likely

– expectation step:

estimate likelihood of all possible phrase alignments for all sentence pairs

– maximization step:

collect counts for phrase pairs, weighted by alignment probability

update phrase translation probabilities