CMPN463: Natural Language Processing

Lecture 10: Phrase-Based Translation

Mohamed Alaa El-Dien Aly
Computer Engineering Department
Cairo University
Fall 2013

Agenda

- Phrases from Alignments
- Phrase-Based Models
- Decoding in Phrase-Based Models
- Evaluation

Acknowledgment:
Most slides adapted from Michael Collins NLP class on Coursera.

Recall: IBM Model 1

- English sentence e has l words $e_1, \ldots, e_l$

  French sentence f has m words $f_1, \ldots, f_m$

- An alignment a identifies the source of each *french* word

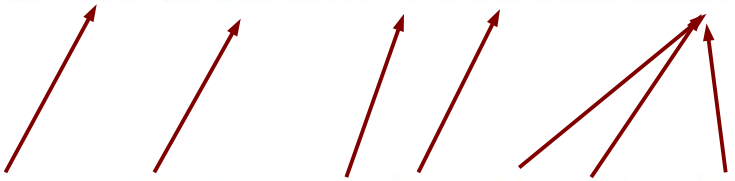
- Final Model:

$$p(f, a|e, m) = p(a|e, m)\, p(f|a, e, m) = \frac{1}{(l+1)^m} \prod_{i=1}^{m} t(f_i|e_{a_i})$$

Recall: IBM Model 1 Example

e.g., $l = 6$, $m = 7$

e = And the program has been implemented



f = Le programme a ete mis en application

$a = \{2, 3, 4, 5, 6, 6, 6\}$

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

Recall: IBM Model 2

- Only difference: *alignment* or *distortion* parameters

$$q(j|i,l,m)$$

Probability that j^{th} French word is connected to i^{th} English word, given sentence lengths of e and f are l and m

- Define $p(a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)$

where $a = \{a_1, \ldots, a_m\}$

- The final result for IBM Model 2:

$$p(f,a|e,m)=\prod_{i=1}^{m} q(a_i|i,l,m)\,t(f_i|e_{a_i})$$

Recall: IBM Model 2 Example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
e &= \text{And the program has been implemented} \\
f &= \text{Le programme a ete mis en application} \\
a &= \{2, 3, 4, 5, 6, 6, 6\}
\end{aligned}
$$

$$
\begin{aligned}
p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\
& \mathbf{q}(3 \mid 2, 6, 7) \times \\
& \mathbf{q}(4 \mid 3, 6, 7) \times \\
& \mathbf{q}(5 \mid 4, 6, 7) \times \\
& \mathbf{q}(6 \mid 5, 6, 7) \times \\
& \mathbf{q}(6 \mid 6, 6, 7) \times \\
& \mathbf{q}(6 \mid 7, 6, 7)
\end{aligned}
$$

Recall: IBM Model 2 Example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
e &= \text{And the program has been implemented} \\
f &= \text{Le programme a ete mis en application} \\
a &= \{2, 3, 4, 5, 6, 6, 6\}
\end{aligned}
$$

$$
\begin{aligned}
p(f \mid a, e, 7) = \; & \mathbf{t}(Le \mid the) \times \\
& \mathbf{t}(programme \mid program) \times \\
& \mathbf{t}(a \mid has) \times \\
& \mathbf{t}(ete \mid been) \times \\
& \mathbf{t}(mis \mid implemented) \times \\
& \mathbf{t}(en \mid implemented) \times \\
& \mathbf{t}(application \mid implemented)
\end{aligned}
$$

Recall: Recovering Alignments

- If we have estimates for the parameters *q* and *t*, we can easily recover the most likely alignment for any sentence pair

- Given a sentence pair $e_1, e_2, \ldots, e_l$ and $f_1, \ldots, f_m$, define

$$a_i = \text{argmax}_{a \in \{0,\ldots,l\}} q(a|i,l,m)\, t(f_i|e_a)$$

for $i = 1, \ldots, m$

Recall: EM Algorithm

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

Initialization: Initialize $t(f|e)$ and $q(j|i,l,m)$ parameters (e.g., to random values).

Recall: EM Algorithm

For $s = 1 \ldots S$

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

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

where

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

- Recalculate the parameters:

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

Summary

- IBM Models 1 & 2 not used for translation but for recovering alignments
- Training done with the EM Algorithm (homework)
- Alignments used to extract phrases in Phrase-Based Models

Phrase-Based Models

- First stage in training a phrase-based model is extraction of a *phrase-based* (PB) *lexicon*
- A *PB lexicon* pairs strings in one language with strings in another language, e.g.,
  - nach Kanada ↔ in Canada
  - zur Konferenz ↔ to the conference
  - Morgen ↔ tomorrow
  - fliege ↔ will fly
  - …

An Example (from tutorial by Koehn and Knight)

- A training example (Spanish/English sentence pair):

  Spanish: Maria no daba una bofetada a la bruja verde

  English: Mary did not slap the green witch

- Some (not all) phrase pairs extracted from this example:

  (Maria ↔ Mary), (bruja ↔ witch), (verde ↔ green),

  (no ↔ did not), (no daba una bofetada ↔ did not slap),

  (daba una bofetada a la ↔ slap the)

- We'll see how to do this using alignments from the IBM models (e.g., from IBM model 2):

  1. Extract alignments
  2. Extract phrase pairs

Alignment Matrix

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

(Note: "bof"' = "bofetada")

In IBM model 2, each foreign (Spanish) word is aligned to exactly one English word. The matrix shows these alignments.

Alignment Matrix

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

- Two problems with these alignments:
  1. They are often *noisy*
  2. They are only many-to-one i.e. each Spanish word is aligned to only *one* English word

Finding Better Alignments

- *Step 1*: train IBM model 2 for $p(f \mid e)$, and come up with most likely alignment for each (*e*, *f*) pair
- *Step 2*: train IBM model 2 for $p(e \mid f)$ and come up with most likely alignment for each (*e*, *f*) pair
- *Step 3*: take intersection of the two alignments as a starting point
- *Step 4*: grow the alignments using heuristics

Better Alignments: Step 1

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

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

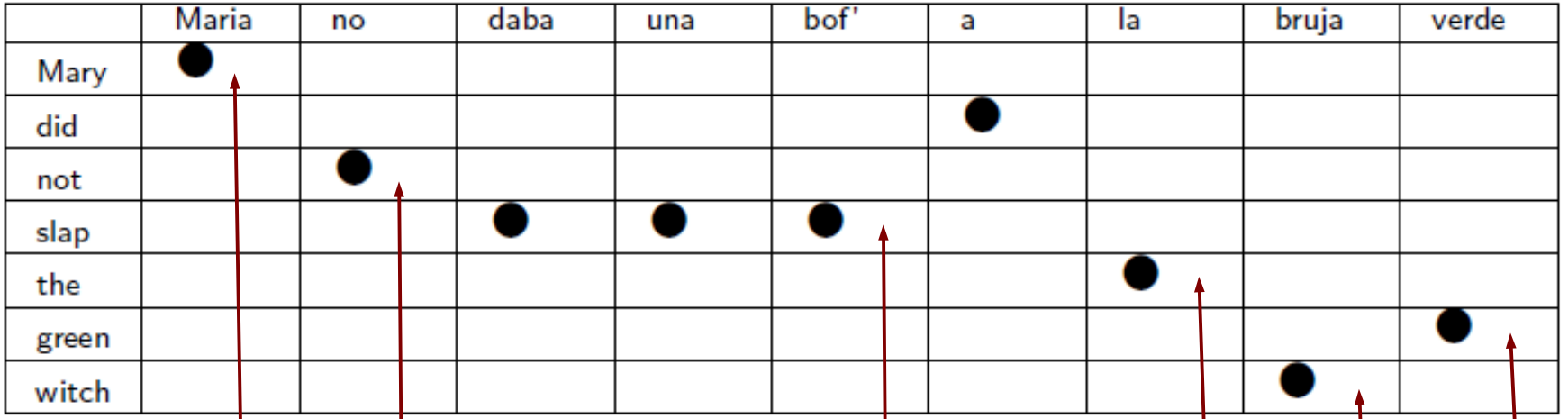
Better Alignments: Step 2

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

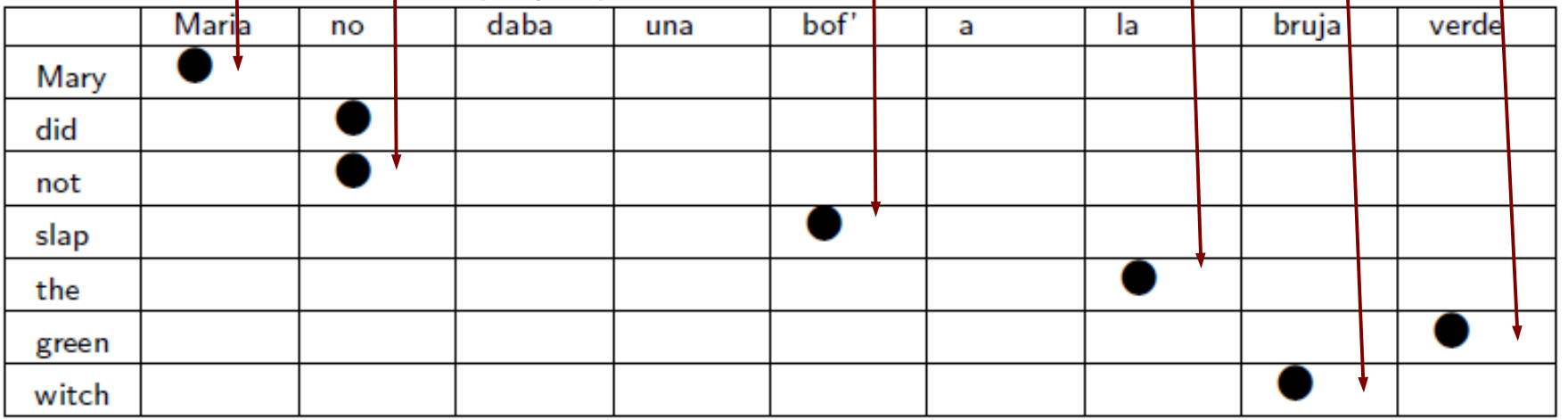
	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap					●				
the							●		
green									●
witch								●	

Better Alignments: Step 3

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



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



Take the intersection

Better Alignments: Step 3

Intersection of the two alignments:

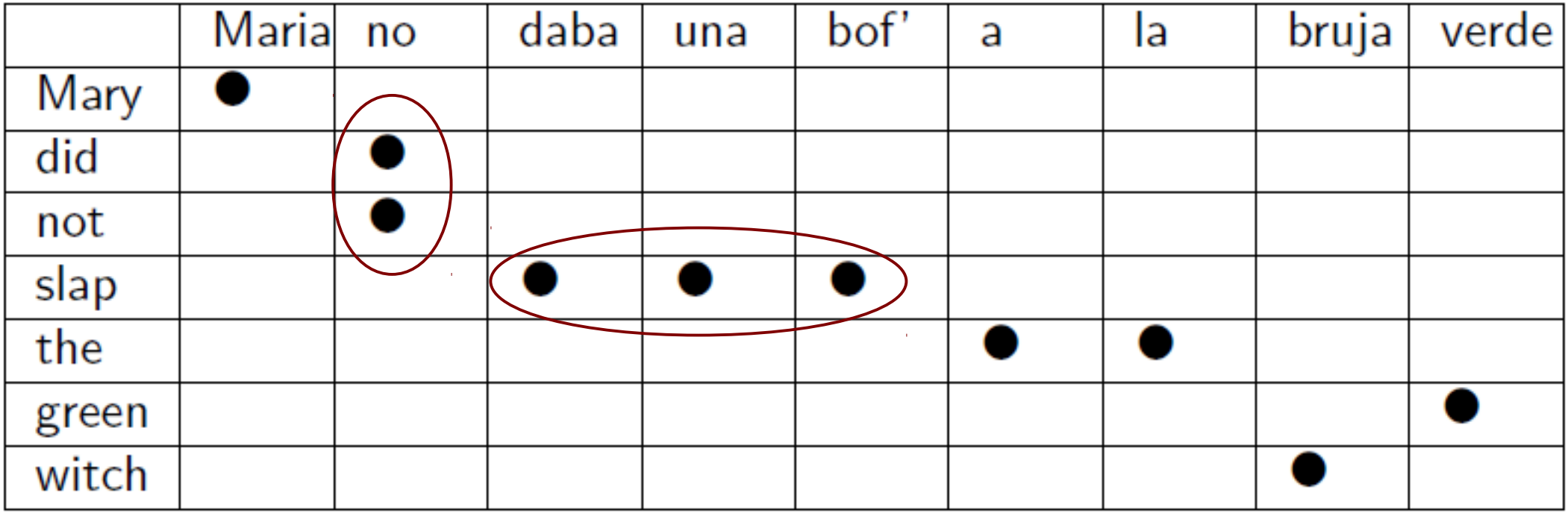
	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did									
not		●							
slap					●				
the							●		
green									●
witch								●	

The intersection has been found to be a very reliable starting point

Better Alignments: Step 4

- Heuristics for growing the alignments:
  - Only explore alignment in union of $p(f \mid e)$ and $p(e \mid f)$ alignments
  - 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

Better Alignments: Step 4



Final alignments

Note: the alignments are no longer many-to-one, but may be now many-to-many, since some Spanish words can be aligned to more one English word, and vice versa.

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. At least one word in *e* is aligned with a word in *f*
  2. No word in *f* is aligned to a word outside *e*
  3. No word in *e* is aligned to a word outside *f*

- Extract all consistent pairs from a training example

Extracting Phrase Pairs

(Mary did ↔ Maria no) is *inconsistent*

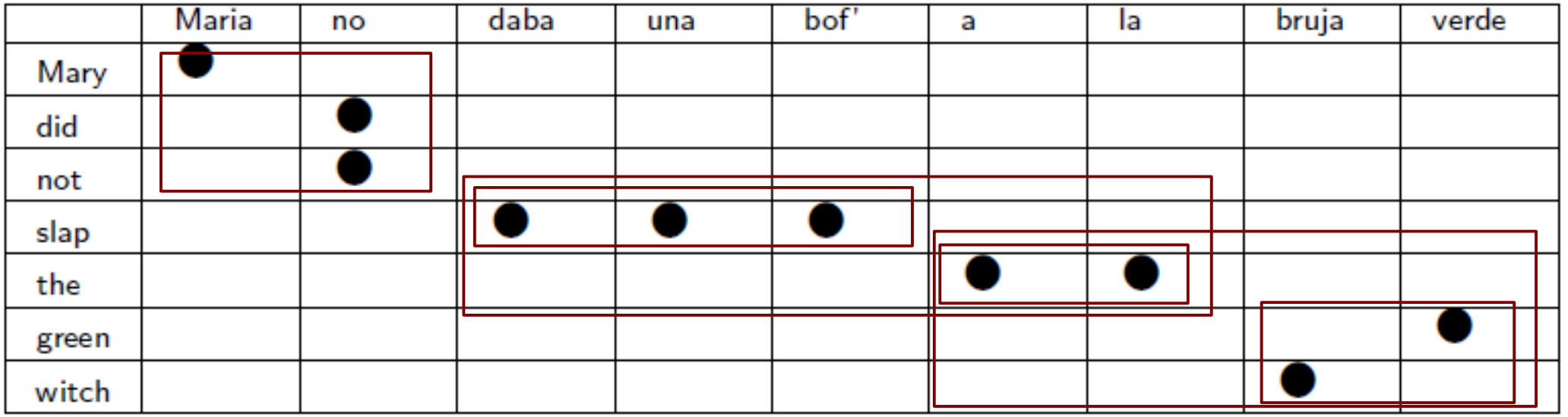
- A phrase pair (*e*, *f*) is *consistent* if:
  1. At least one word in *e* is aligned with a word in *f*
  2. No word in *f* is aligned to a word outside *e*
  3. No word in *e* is aligned to a word outside *f*

Extracting Phrase Pairs

(Mary did not ↔ Maria no) is *consistent*

- A phrase pair (*e*, *f*) is *consistent* if:
  1. At least one word in *e* is aligned with a word in *f*
  2. No word in *f* is aligned to a word outside *e*
  3. No word in *e* is aligned to a word outside *f*

Extracting Phrase Pairs



Extract all *consistent* phrase pairs

- A phrase pair (*e*, *f*) is *consistent* if:
  1. At least one word in *e* is aligned with a word in *f*
  2. No word in *f* is aligned to a word outside *e*
  3. No word in *e* is aligned to a word outside *f*

Phrase Pair Probabilities

- For any phrase pair (f, e) extracted from the training data, we can calculate:

Number of times f was aligned to e

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

Number of times e appeared

- For example:

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

Example Phrase Translation Table

Phrase Translations 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	...	...

An example from Koehn, EACL 2006 tutorial.
Note that we have t(e|f) not t(f|e) in this example.

Phrase-Based Systems

- We want to translate from a Foreign language to English
- Translation is done by choosing a sequence of phrases from the foreign language and outputting their equivalent in English
- Each choice of a phrase has a score with three components:

1. Language model: e.g. a Trigram English model, that measures the correctness of the resulting English

$$\log q(v \mid t, u)$$

2. Phrase model: that measures the correctness of the chosen phrase pairs

$$\log t(f \mid e)$$

3. Distortion model: that enforces the order of words taken from the foreign language (usually negative)

$$\eta \times skip$$

Example

Today

Heute werden wir uber die Wiedereroffnung des Mont-Blanc-Tunnels diskutieren

Start Symbols

$$\text{Score} = \underbrace{\log q(\text{Today} \mid *, *)}_{\text{Language model}} + \underbrace{\log t(\text{Heute} \mid \text{Today})}_{\text{Phrase model}} + \underbrace{\eta \times 0}_{\text{Distortion model}}$$

We did not skip any words in German

Choosing the phrase pair (Heute, Today) has this score

Example

Today we shall be

Heute werden wir uber die Wiedereroffnung

des Mont-Blanc-Tunnels diskutieren

$$\text{Score} = \underbrace{\log q(\text{we}|*, \text{Today}) + \log q(\text{shall}|\text{Today}, \text{we}) + \log q(\text{be}|\text{we}, \text{shall})}_{\text{Language model}}$$

$$+ \underbrace{\log t(\text{werden wir} \mid \text{we shall be})}_{\text{Phrase model}}$$

$$+ \underbrace{\eta \times 0}_{\text{Distortion model}}$$

Choosing the phrase pair (werden wir, we shall be) has this score

Example

Today we shall be debating
Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

$$\text{Score} = \underbrace{\log q(\text{debating}|\text{shall, be})}_{\text{Language model}} + \underbrace{\log t(\text{diskutieren} \mid \text{debating})}_{\text{Phrase model}} + \underbrace{\eta \times 6}_{\text{Distortion model}}$$

We skipped 6 words in German and choose “diskutieren” instead of the next word “uber”

Choosing the phrase pair (diskutieren, debating) has this score

Example

Today we shall be debating the reopening
Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

Choosing the phrase pair (uber die Wiederreroffnung, the reopening)

Today we shall be debating the reopening
of the Mont Blanc tunnel
Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

Choosing the phrase pair (des Mont-Blanc-Tunnels, of the Mont Blanc tunnel)

Phrase-Based Models

- Make a sequence of choices from phrases in the foreign language to phrases in the English language
- Each choice of phrase pair has a score

- Decoding Algorithm: Find the sequence of phrase pairs y that maximizes the resulting score

- There are possibly exponential number of possible sequences to choose from
  - Find approximate solution using, e.g., Beam Search (next)

Phrase-Based Models: Definitions

wir müssen auch diese kritik ernst nehmen

- Phrase-based Lexicon contains entries (f, e) like
  - (wir müssen, we must)
  - (wir, we)
  - (wir müssen auch, we must also)

- Each entry has a score $g(f, e)$, e.g.

$$g(\text{wir müssen, we must}) = \log\left(\frac{\text{Count(wir müssen, we must)}}{\text{Count(we must)}}\right)$$

- A trigram model, with parameters $q(t \mid u, v)$, e.g. $q(\text{also} \mid \text{we must})$
- A distortion parameter η

Phrase-Based Models: Definitions

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

- A phrase p is a tuple (s, t, e) that signifies that the foreign sequence of words $f_s, f_{s+1}, \ldots, f_t$ can be translated as the English sentence e using an entry from the PB lexicon. Example:
  - (1, 2, we must)
  - (1, 1, we)
  - (1, 3, we must also)
- $\mathscr{P}$ is the set of all phrases for a sentence
- For any phrase p:
  - $s(p)$, $t(p)$, and $e(p)$ are its components.
  - $g(p)$ is its score

Phrase-Based Models: Definitions

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

- A derivation y is a finite sequence of phrases $p_1 \ldots p_L$ where each p_1 is a member of the set $\mathscr{P}$

- For any derivation y we use $e(y)$ as its underlying English translation

- For example:
  y = (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

  and

- $e(y)$ = we must also take this criticism seriously

Phrase-Based Models: Definitions

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

- The set of all valid derivations $\mathscr{Y}(f)$ where $f = f_1 \ldots f_n$ is a sequence of foreign words. Usually exponential!
- A derivation $y = p_1 \ldots p_L$ is valid if:
  - Each phrase p_i is a member of $\mathscr{P}$
  - Each word in f is translated only once
  - For all $k \in \{1,\ldots,L\text{-}1\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where d is a parameter of the model called the "distortion limit".

    Example:

    - d = 4, (1, 2, we must) & (3, 3, also) → |2 + 1 – 3| = 0 < d
    - d = 4, (1, 1, we) & (7, 7, take) → |1 + 1 – 7| = 5 > d

Phrase-Based Models: Definitions

- The set of all valid derivations $\mathscr{Y}(f)$ where $f = f_1 \ldots f_n$ is a sequence of foreign words. Usually exponential!
- A derivation $y = p_1 \ldots p_L$ is valid if:
  - Each phrase p_i is a member of $\mathscr{P}$
  - Each word in f is translated only once
  - For all $k \in \{1,\ldots,L\text{-}1\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where d is a parameter of the model called the "distortion limit"
    - Improves the speed of the decoding step by limiting the number of possible translations to search
    - Also improves the quality of the translation
  - We must also have $|1 - s(p_1)| \leq d$

Phrase-Based Models: Definitions

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

- A derivation $y = p_1 \ldots p_L$ is valid if:
  - Each phrase p_i is a member of $\mathscr{P}$
  - Each word in f is translated only once
  - For all $k \in \{1,\ldots,L\text{-}1\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ $d = 4$
  - We must also have $|1 - s(p_1)| \leq d$

$y = (1, 3, \text{we must also}), (7, 7, \text{take}), (4, 5, \text{this criticism}), (6, 6, \text{seriously})$ ✓

$y = (1, 3, \text{we must also}), (1, 2, \text{we must}), (4, 5, \text{this criticism}), (6, 6, \text{seriously})$ ✗

"wir müssen" translated twice!

$y = (1, 2, \text{we must}), (7, 7, \text{take}), (3, 3, \text{also}), (4, 5, \text{this criticism}), (6, 6, \text{seriously})$ ✗

(7, 7, take) & (3, 3, also) → $|7 + 1 - 3| = 5 > d$

Phrase-Based Models: Definitions

- The translation problem: find the *valid derivation* with maximum score
- We need to search the exponential set $\mathcal{Y}(f)$
- The score for a derivation $y = p_1 \ldots p_L$ is defined as

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

Trigram Language Model for English sentence

Phrase model
One term per phrase

Distortion model

Example

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

$y = (1, 3, \text{we must also}), (7, 7, \text{take}), (4, 5, \text{this criticism}), (6, 6, \text{seriously})$

Score = log q(we | **) + log q(must|* we) + log q(also|we must) +
log q(take|must also) + log q(this|also take) + q(criticism|take this) +
log q(seriously|this criticism) +

g(1, 3, we must also) + g(7, 7, take) + g(4, 5, this criticism) +
g(6, 6, seriously) +

η |1 – 1| + η |3+1 – 7| + η |7+1 – 4| + η |5+1 - 6|

Decoding Algorithm

- We need to search an exponential space of possible sequences of phrases (valid derivations)
- Will treat the problem as a graph search, to find a path from the *starting state* to the *goal state* with (approximately) maximum score using beam search
- From every *state* in the graph, explore neighboring reachable states (valid moves), keeping only the top scoring ones
- End when finished translating the whole

Decoding Algorithm: Definitions

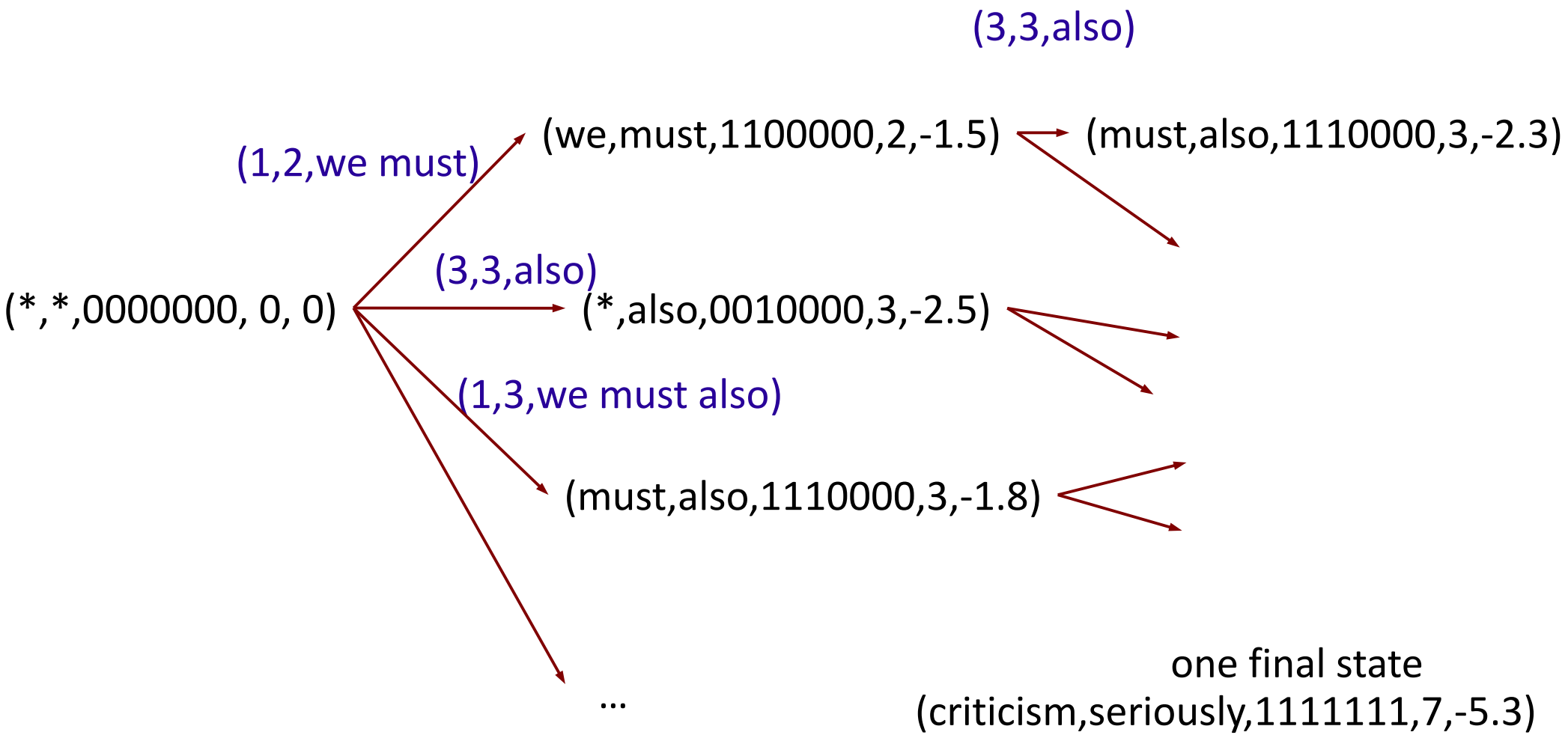
- A state is a tuple (e_1, e_2, b, r, α) where
  - e_1, e_2 are English words
  - b is a bit string of length n specifying which foreign words have been translated
  - r is an integer specifying the end-point of the last phrase
  - α is the score of the state (the sequence of phrases)

- The initial state is $q_0(*, *, 0^n, 0, 0)$
- The final state is $q_f(e_{i\text{-}1}, e_i, 1^n, i, \alpha*)$

Example

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7



Decoding Algorithm: Definitions

- We define a function $ph(q)$ for any state $q=(e_1, e_2, b, r, \alpha)$ that returns the set of phrases that are allowed to follow state q
- For a phrase $p=(s, t, e)$ to be in $ph(q)$, it must satisfy:
  - p must not overlap with the bit string b of q i.e. $b_i = 0$ for $i \in \{s, \ldots, t\}$
  - The distortion limit must not be violated i.e. $|r+1-s| \leq d$

  For example:
  - $ph(q_0) = \{(1,1,\text{we}), (1,2,\text{we must}), (3,3,\text{also}), \ldots.\}$
  - $(7,7,\text{take}) \notin ph(q_0)$

wir müssen auch diese kritik ernst nehmen

1 2 3 4 5 6 7

Decoding Algorithm: Definitions

- We define a function $next(q, p)$ for any state $q=(e_1, e_2, b, r, \alpha)$ and phrase $p=(s, t, \epsilon_1 \ldots \epsilon_M)$ to be the state that results from combining state q with phrase p
- Formally, $next(q, p)$ is the state $q'=(e_1', e_2', b', r', \alpha')$ such that:
  - $e_1' = \epsilon_{M-1}$ *and* $e_2' = \epsilon_M$
  - $b_i'=1$ for $i \in \{s \ldots t\}$ & $b_i'=1$ for $i \notin \{s \ldots t\}$
  - $r' = t$
  - $$\alpha' = \alpha + g(p) + \sum_{i=1}^{M} \log q(\epsilon_i | \epsilon_{i-2}, \epsilon_{i-1}) + \eta \times |r + 1 - s|$$

    Example:

    $next$((must,also,1110000,3,-2.5), (7,7,take)) =

    (also,take,1110001,7,-3.2)

Decoding Algorithm: Definitions

- We define the equality function $eq(q, q')$ for any two states $q=(e_1, e_2, b, r, \alpha)$ and $q'=(e_1', e_2', b', r', \alpha')$
- It returns TRUE if
  - $e_1=e_1'$
  - $e_2=e_2'$
  - $b=b'$
  - $r=r'$

Decoding Algorithm

- *Inputs*: sentence $f_1, \ldots, f_n$ and Phrase-Based Model
- *Initialization*: $Q_0 = \{q_0\}$ and $Q_i = \Phi$ for $i = 1 \ldots n$
- For $i = 0 \ldots n\text{-}1$
  - For each state q in $beam(Q_i)$, for each phrase p in $ph(q)$
    - $q' = next(q, p)$
    - $Add(Q_j, q', q, p)$ where $j = len(q')$ i.e. number of 1's in b'
- *Return*: highest scoring state in Q_n. Back pointers will be used to construct the translation and the phrases chosen.

- Breadth-First search (with a catch)
- Each queue Q_i holds states that have exactly i foreign words translated

Add(Q, q', q, p)

- If there is some q'' in Q such that $eq(q'', q')$
  - If $\alpha(q') > \alpha(q'')$
    - $\alpha(q'') = \alpha(q')$
    - Set $bp(q') = (q, p)$
  - Else return
- Else
  - $Insert(Q, q')$
    - Set $bp(q') = (q, p)$

- If the state exists, then do nothing or update its score if higher
- If the state is new, then add it to the queue

beam(Q)

- Define $\alpha^* = \text{argmax}_{q \in Q}\, \alpha(q)$
- Define $\beta \geq 0$ to be the *beam-width* parameter
- $beam(Q)$ = $\{q \in Q: \alpha(q) \geq \alpha^* - \beta\}$

Summary

- Start with IBM Model 2 to learn alignments
- From alignments learn phrase-based lexicon
- Given a foreign sentence, perform a beam search to find the highest approximate translation

MT Evaluation

- How do we evaluate machine translation?
  - Human Evaluation
  - Automatic Evaluation
    - BLEU (Bi-Lingual Evaluation Understudy)

BLEU

- A number between 0 and 1
- Evaluates the quality of translation of a whole corpus (test set)
- Measures quality by comparing to a set of reference human translations
- It is a *modified* measure of precision i.e. how many of the output n-grams in the machine translated output are in the reference translations
- It computes scores for uni-grams, bi-grams, tri-grams, and usually quadri-grams, and takes their geometric mean
- It also includes a *brevity penalty* to penalize shorter translations since they usually get higher precision

BLEU Calculation

- Candidate: the the the the the the the
- Reference 1: the cat is on the mat
- Reference 2: there is a cat on the mat

Unigram Precision = 7 / 7 = 1 !!

- Modify the precision by setting a *maximum count* for each token.
- The maximum count is the maximum number of times this token appeared in the reference translations.

- Candidate: the the the the the the the — Count(the) = 7
- Reference 1: the cat is on the mat — Count(the) = 2
- Reference 2: there is a cat on the mat — Count(the) = 1

Modified Unigram Precision = min(2, 7) / 7 = 2/7

BLEU Calculation

- Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party
- Candidate 2: It is to insure the troops forever hearing the activity guidebook that party direct

- Reference 1: It is a guide to action that ensures that the military will forever heed Party commands
- Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party
- Reference 3: It is the practical guide for the army always to heed the directions of the party

Correct bi-grams in candidate translations

$$p_2(\text{Candidate } 1) = \frac{10}{17} \qquad p_2(\text{Candidate } 2) = \frac{1}{13}$$

Number of bi-grams in candidate translations

BLEU Calculation

Modified Precision for n-gram

$$p_n = \frac{\sum_{c \in Candidate} \sum_{n-gram \in c} min\left(\text{Count}\left(n-gram\right), \text{MaxCount}\left(n-gram\right)\right)}{\sum_{c' \in Candidate} \sum_{n-gram \in c'} \text{Count}\left(n-gram\right)}$$

BLEU

$$BLEU = BP \times \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Brevity Penalty

Geometric Mean of first *N* n-grams

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leqslant r \end{cases}$$

r: length of reference
c: length of candidate

Recap

- Phrases from Alignments
- Phrase-Based Models
- Decoding in Phrase-Based Models
- Evaluation