Natural Language Processing (CSE 447/547M): Bitext and Machine Translation, Continued

Noah Smith

University of Washington
nasmith@cs.washington.edu

March 4, 2018

IBM Model 1

(Brown et al., 1993)

Let ℓ and m be the (known) lengths of $\boldsymbol{e}$ and $\boldsymbol{f}$.

Latent variable $\boldsymbol{a} = \langle a_1, \ldots, a_m \rangle$, each a_i ranging over $\{0, \ldots, \ell\}$ (positions in $\boldsymbol{e}$).

- $a_4 = 3$ means that f_4 is "aligned" to e_3.
- $a_6 = 0$ means that f_6 is "aligned" to a special NULL symbol, e_0.

$$p(\boldsymbol{f} \mid \boldsymbol{e}, m) = \sum_{a_1=0}^{\ell} \sum_{a_2=0}^{\ell} \cdots \sum_{a_m=0}^{\ell} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m)$$

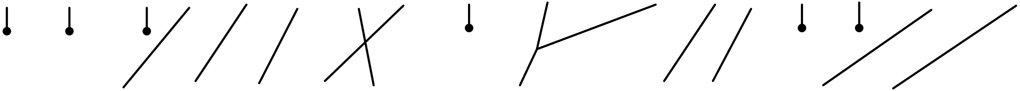
$$= \sum_{\boldsymbol{a} \in \{0,\ldots,\ell\}^m} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m)$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \prod_{i=1}^{m} p(a_i \mid i, \ell, m) \cdot p(f_i \mid e_{a_i})$$

$$= \prod_{i=1}^{m} \frac{1}{\ell + 1} \cdot \theta_{f_i \mid e_{a_i}} = \left(\frac{1}{\ell + 1} \right)^m \prod_{i=1}^{m} \theta_{f_i \mid e_{a_i}}$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .



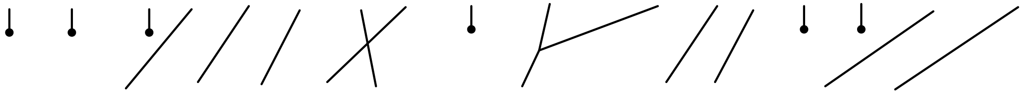
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 4, \ldots \rangle$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{17+1} \cdot \theta_{\textsf{Noahs}|\textsf{Noah's}}$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .



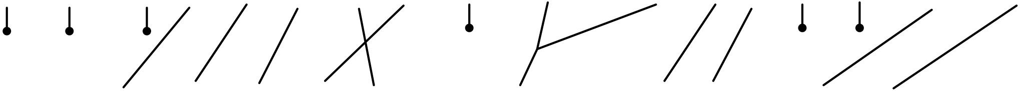
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 4, 5, \ldots \rangle$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{17+1} \cdot \theta_{\textsf{Noahs}|\textsf{Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\textsf{Arche}|\textsf{ark}}$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .

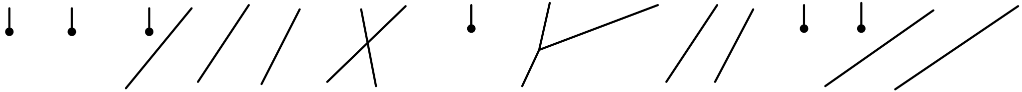


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$
\begin{aligned}
\boldsymbol{a} &= \langle 4, 5, 6, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{17+1} \cdot \theta_{\textsf{Noahs}|\textsf{Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\textsf{Arche}|\textsf{ark}} \\
&\quad \cdot \frac{1}{17+1} \cdot \theta_{\textsf{war}|\textsf{was}}
\end{aligned}
$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .



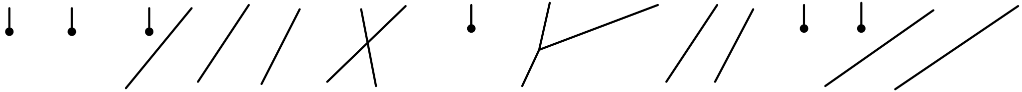
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 4, 5, 6, 8, \ldots \rangle$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{17+1} \cdot \theta_{\textsf{Noahs}|\textsf{Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\textsf{Arche}|\textsf{ark}} \cdot \frac{1}{17+1} \cdot \theta_{\textsf{war}|\textsf{was}} \cdot \frac{1}{17+1} \cdot \theta_{\textsf{nicht}|\textsf{not}}$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .



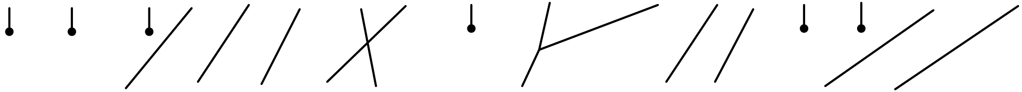
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 4, 5, 6, 8, 7, \ldots \rangle$$

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = {} & \frac{1}{17+1} \cdot \theta_{\mathsf{Noahs|Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{Arche|ark}} \\ & \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{war|was}} \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{nicht|not}} \\ & \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{voller|filled}} \end{aligned}$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .

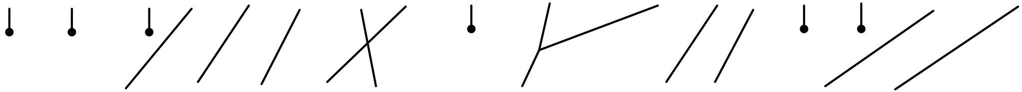


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$
\begin{aligned}
\boldsymbol{a} &= \langle 4, 5, 6, 8, 7, ?, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{17+1} \cdot \theta_{\mathsf{Noahs|Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{Arche|ark}} \\
&\quad \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{war|was}} \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{nicht|not}} \\
&\quad \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{voller|filled}} \cdot \frac{1}{17+1} \cdot \theta_{\mathsf{Productionsfactoren|?}}
\end{aligned}
$$

Example: $\boldsymbol{f}$ is German

Mr President , Noah's ark was filled not with production factors , but with living creatures .

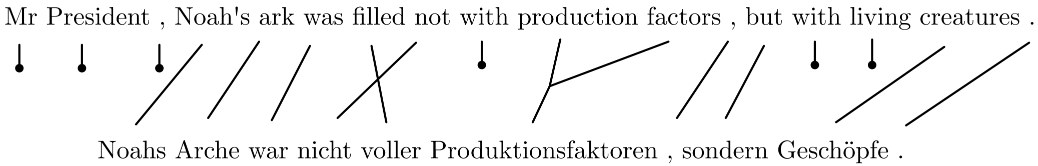


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$
\begin{aligned}
\boldsymbol{a} &= \langle 4, 5, 6, 8, 7, ?, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{17+1} \cdot \theta_{\text{Noahs}|\text{Noah's}} \cdot \frac{1}{17+1} \cdot \theta_{\text{Arche}|\text{ark}} \\
&\cdot \frac{1}{17+1} \cdot \theta_{\text{war}|\text{was}} \cdot \frac{1}{17+1} \cdot \theta_{\text{nicht}|\text{not}} \\
&\cdot \frac{1}{17+1} \cdot \theta_{\text{voller}|\text{filled}} \cdot \frac{1}{17+1} \cdot \theta_{\text{Productionsfactoren}|?}
\end{aligned}
$$

Problem: This alignment isn't possible with IBM Model 1! Each f_i is aligned to at most *one* e_{a_i}!

Example: $\boldsymbol{f}$ is English

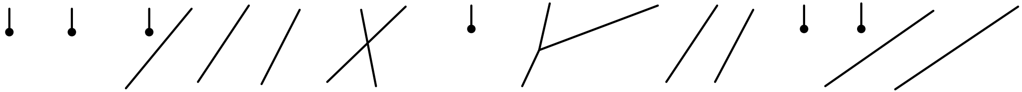


$$\boldsymbol{a} = \langle 0, \ldots \rangle$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{10+1} \cdot \theta_{\mathsf{Mr}|\textsc{null}}$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .

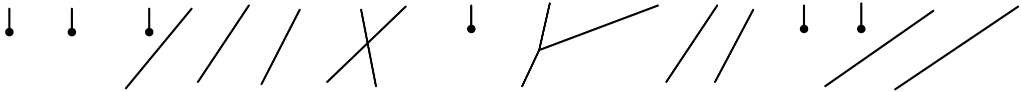


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$
\begin{aligned}
\boldsymbol{a} &= \langle 0, 0, 0, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{10+1} \cdot \theta_{\textsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{President}|\textsc{null}} \\
&\cdot \frac{1}{10+1} \cdot \theta_{,|\textsc{null}}
\end{aligned}
$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .



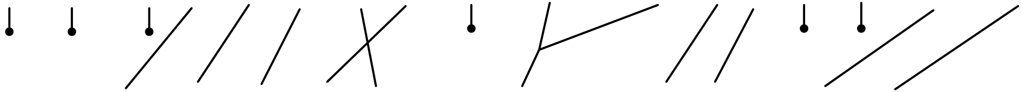
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$
\boldsymbol{a} = \langle 0, 0, 0, 1, \ldots \rangle
$$

$$
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{10+1} \cdot \theta_{\mathsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{President}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{,|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{Noah's}|\mathsf{Noahs}}
$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .



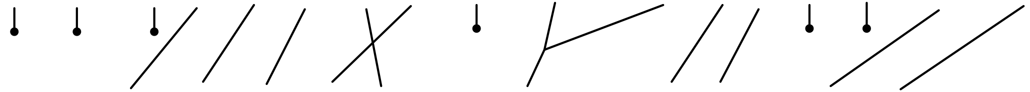
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 0, 0, 0, 1, 2, \ldots \rangle$$

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = {} & \frac{1}{10+1} \cdot \theta_{\text{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\text{President}|\textsc{null}} \\ & \cdot \frac{1}{10+1} \cdot \theta_{,|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\text{Noah's}|\text{Noahs}} \\ & \cdot \frac{1}{10+1} \cdot \theta_{\text{ark}|\text{Arche}} \end{aligned}$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .



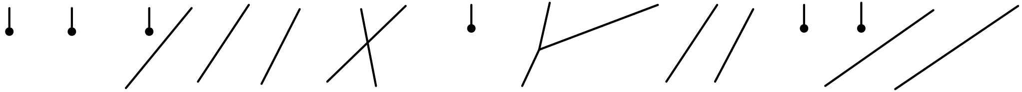
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 0, 0, 0, 1, 2, 3, \ldots \rangle$$

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = &\frac{1}{10+1} \cdot \theta_{\mathsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{President}|\textsc{null}} \\ &\cdot \frac{1}{10+1} \cdot \theta_{\mathsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{Noah's}|\mathsf{Noahs}} \\ &\cdot \frac{1}{10+1} \cdot \theta_{\mathsf{ark}|\mathsf{Arche}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{was}|\mathsf{war}} \end{aligned}$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .



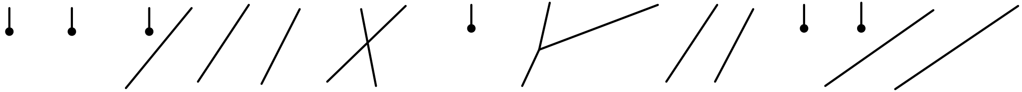
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 0, 0, 0, 1, 2, 3, 5, \ldots \rangle$$

$$\begin{aligned}
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = & \frac{1}{10+1} \cdot \theta_{\textsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{President}|\textsc{null}} \\
& \cdot \frac{1}{10+1} \cdot \theta_{\textsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{Noah's}|\textsf{Noahs}} \\
& \cdot \frac{1}{10+1} \cdot \theta_{\textsf{ark}|\textsf{Arche}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{was}|\textsf{war}} \\
& \cdot \frac{1}{10+1} \cdot \theta_{\textsf{filled}|\textsf{voller}}
\end{aligned}$$

Example: $\boldsymbol{f}$ is English

Mr President , Noah's ark was filled not with production factors , but with living creatures .



Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$\boldsymbol{a} = \langle 0, 0, 0, 1, 2, 3, 5, 4, \ldots \rangle$$

$$\begin{aligned} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = &\frac{1}{10+1} \cdot \theta_{\mathsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{President}|\textsc{null}} \\ &\cdot \frac{1}{10+1} \cdot \theta_{\mathsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{Noah's}|\mathsf{Noahs}} \\ &\cdot \frac{1}{10+1} \cdot \theta_{\mathsf{ark}|\mathsf{Arche}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{was}|\mathsf{war}} \\ &\cdot \frac{1}{10+1} \cdot \theta_{\mathsf{filled}|\mathsf{voller}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{not}|\mathsf{nicht}} \end{aligned}$$

How to Estimate Translation Distributions?

This is a problem of **incomplete data**: at training time, we see $\boldsymbol{e}$ and $\boldsymbol{f}$, but not $\boldsymbol{a}$.

How to Estimate Translation Distributions?

This is a problem of **incomplete data**: at training time, we see $\boldsymbol{e}$ and $\boldsymbol{f}$, but not $\boldsymbol{a}$.

Classical solution is to *alternate*:

- Given a parameter estimate for $\boldsymbol{\theta}$, align the words.
- Given aligned words, re-estimate $\boldsymbol{\theta}$.

Traditional approach uses "soft" alignment.

"Complete Data" IBM Model 1

Let the training data consist of N word-aligned sentence pairs:
$\langle \boldsymbol{e}_1^{(1)}, \boldsymbol{f}^{(1)}, \boldsymbol{a}^{(1)} \rangle, \ldots, \langle \boldsymbol{e}^{(N)}, \boldsymbol{f}^{(N)}, \boldsymbol{a}^{(N)} \rangle$.
Define:

$$\iota(k,i,j) = \begin{cases} 1 & \text{if } a_i^{(k)} = j \\ 0 & \text{otherwise} \end{cases}$$

Maximum likelihood estimate for $\theta_{f|e}$:

$$\frac{c(e,f)}{c(e)} = \frac{\displaystyle\sum_{k=1}^{N} \sum_{i:f_i^{(k)}=f} \sum_{j:e_j^{(k)}=e} \iota(k,i,j)}{\displaystyle\sum_{k=1}^{N} \sum_{i=1}^{m^{(k)}} \sum_{j:e_j^{(k)}=e} \iota(k,i,j)}$$

MLE with "Soft" Counts for IBM Model 1

Let the training data consist of N "softly" aligned sentence pairs, $\langle \boldsymbol{e}_1^{(1)}, \boldsymbol{f}^{(1)}, \rangle, \ldots, \langle \boldsymbol{e}^{(N)}, \boldsymbol{f}^{(N)} \rangle$.

Now, let $\iota(k, i, j)$ be "soft," interpreted as:

$$\iota(k, i, j) = p(a_i^{(k)} = j)$$

Maximum likelihood estimate for $\theta_{f|e}$:

$$\frac{\displaystyle\sum_{k=1}^{N} \sum_{i:f_i^{(k)}=f} \sum_{j:e_j^{(k)}=e} \iota(k, i, j)}{\displaystyle\sum_{k=1}^{N} \sum_{i=1}^{m^{(k)}} \sum_{j:e_j^{(k)}=e} \iota(k, i, j)}$$

Expectation Maximization Algorithm for IBM Model 1

1. Initialize $\boldsymbol{\theta}$ to some arbitrary values.
2. E step: use current $\boldsymbol{\theta}$ to estimate expected ("soft") counts.

$$\iota(k,i,j) \leftarrow \frac{\theta_{f_i^{(k)}|e_j^{(k)}}}{\sum_{j'=0}^{\ell^{(k)}} \theta_{f_i^{(k)}|e_{j'}^{(k)}}}$$

3. M step: carry out "soft" MLE.

$$\theta_{f|e} \leftarrow \frac{\sum_{k=1}^{N} \sum_{i:f_i^{(k)}=f} \sum_{j:e_j^{(k)}=e} \iota(k,i,j)}{\sum_{k=1}^{N} \sum_{i=1}^{m^{(k)}} \sum_{j:e_j^{(k)}=e} \iota(k,i,j)}$$

4. Go to 2 until converged.

Expectation Maximization

- Originally introduced in the 1960s for estimating HMMs when the states really are "hidden."
- Can be applied to any generative model with hidden variables.
- Greedily attempts to maximize probability of the observable data, marginalizing over latent variables. For IBM Model 1, that means:

$$\max_{\boldsymbol{\theta}} \prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{f}^{(k)} \mid \boldsymbol{e}^{(k)}) = \max_{\boldsymbol{\theta}} \prod_{k=1}^{N} \sum_{\boldsymbol{a}} p_{\boldsymbol{\theta}}(\boldsymbol{f}^{(k)}, \boldsymbol{a} \mid \boldsymbol{e}^{(k)})$$

- Usually converges only to a *local* optimum of the above, which is in general not convex.
- Strangely, for IBM Model 1 (and very few other models), it *is* convex!

IBM Model 2

(Brown et al., 1993)

Let ℓ and m be the (known) lengths of $\boldsymbol{e}$ and $\boldsymbol{f}$.

Latent variable $\boldsymbol{a} = \langle a_1, \ldots, a_m \rangle$, each a_i ranging over $\{0, \ldots, \ell\}$ (positions in $\boldsymbol{e}$).

- E.g., $a_4 = 3$ means that f_4 is "aligned" to e_3.

$$p(\boldsymbol{f} \mid \boldsymbol{e}, m) = \sum_{\boldsymbol{a} \in \{0,\ldots,n\}^m} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m)$$

$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \prod_{i=1}^{m} p(a_i \mid i, \ell, m) \cdot p(f_i \mid e_{a_i})$$

$$= \delta_{a_i \mid i, \ell, m} \cdot \theta_{f_i \mid e_{a_i}}$$

Variations

- Dyer et al. (2013) introduced a new parameterization:

$$\delta_{j|i,\ell,m} \propto \exp -\lambda \left| \frac{i}{m} - \frac{j}{\ell} \right|$$

  (This is called `fast_align`.)
- IBM Models 3–5 (Brown et al., 1993) introduced increasingly more powerful ideas, such as "fertility" and "distortion."

From Alignment to (Phrase-Based) Translation

Obtaining word alignments in a parallel corpus is a common first step in building a machine translation system.

1. Align the words.
2. Extract and score **phrase pairs**.
3. Estimate a global scoring function to optimize (a proxy for) translation quality.
4. Decode French sentences into English ones.

(We'll discuss 2–4.)

The noisy channel pattern isn't taken quite so seriously when we build real systems, but **language models** are really, really important nonetheless.

Phrases?

Phrase-based translation uses automatically-induced phrases . . . not the ones given by a phrase-structure parser.

Examples of Phrases

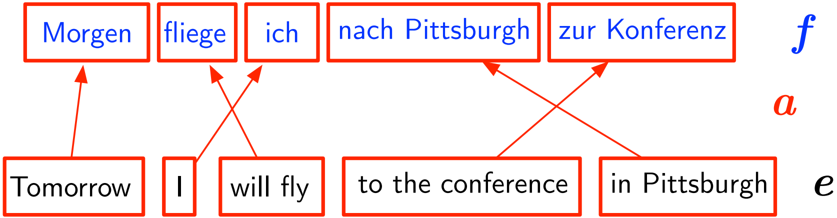
Courtesy of Chris Dyer.

German	English	$p(\bar{\boldsymbol{f}} \mid \bar{\boldsymbol{e}})$
das Thema	the issue	0.41
	the point	0.72
	the subject	0.47
	the thema	0.99
es gibt	there is	0.96
	there are	0.72
morgen	tomorrow	0.90
fliege ich	will I fly	0.63
	will fly	0.17
	I will fly	0.13

Phrase-Based Translation Model

Originated by Koehn et al. (2003).

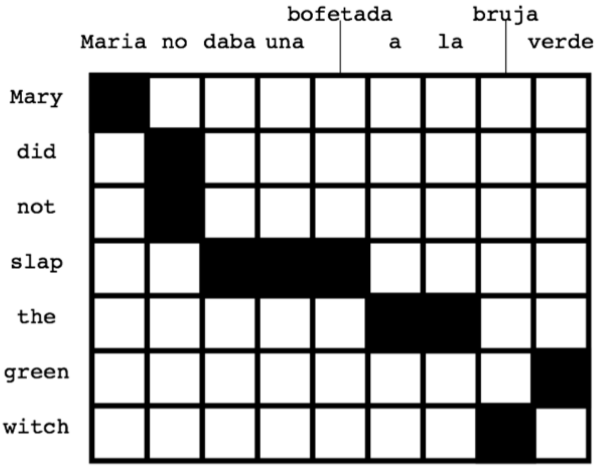
R.v. $\boldsymbol{A}$ captures segmentation of sentences into phrases, alignment between them, and reordering.



$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}) = p(\boldsymbol{a} \mid \boldsymbol{e}) \cdot \prod_{i=1}^{|\boldsymbol{a}|} p(\bar{\boldsymbol{f}}_i \mid \bar{\boldsymbol{e}}_i)$$

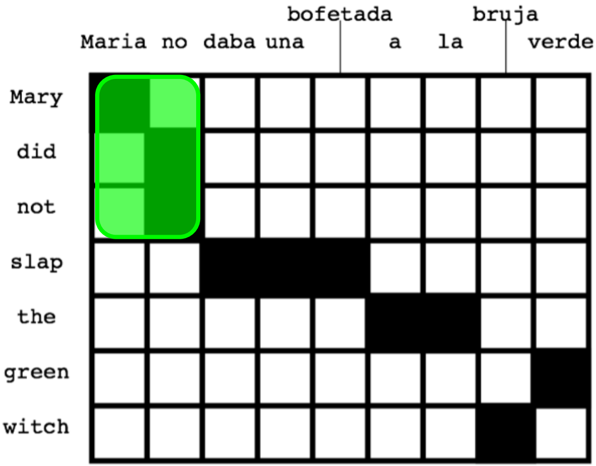
Extracting Phrases

After inferring word alignments, apply heuristics.



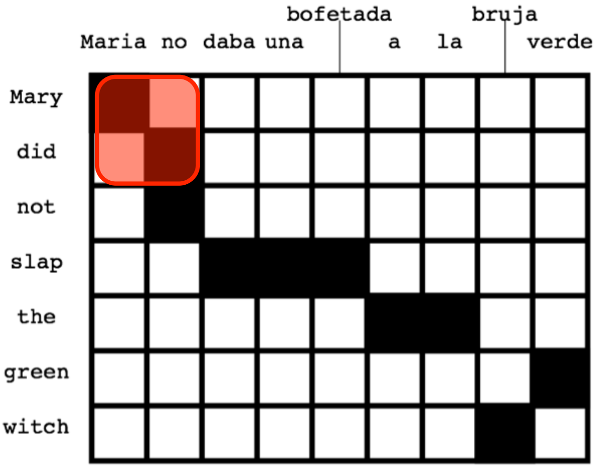
Extracting Phrases

After inferring word alignments, apply heuristics.



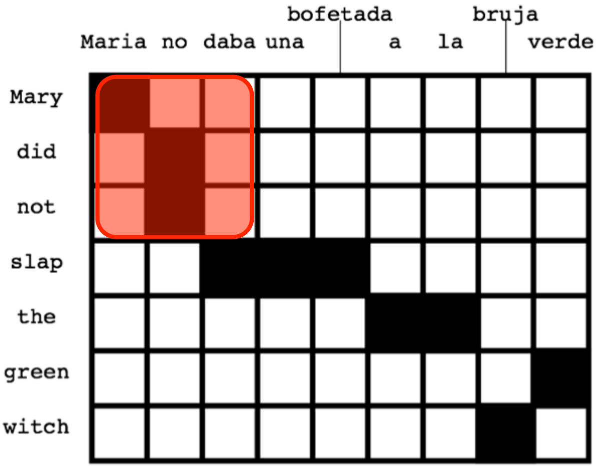
Extracting Phrases

After inferring word alignments, apply heuristics.



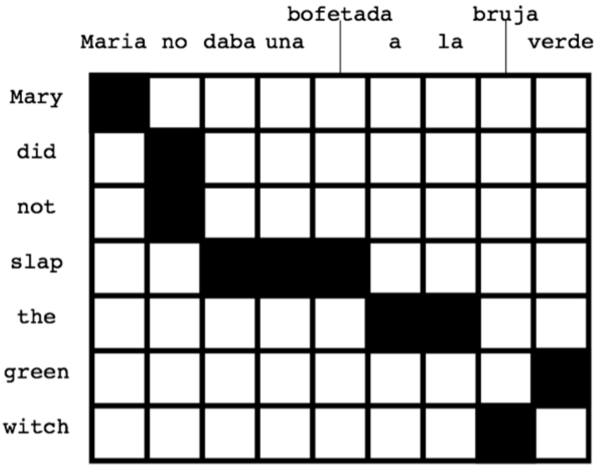
Extracting Phrases

After inferring word alignments, apply heuristics.



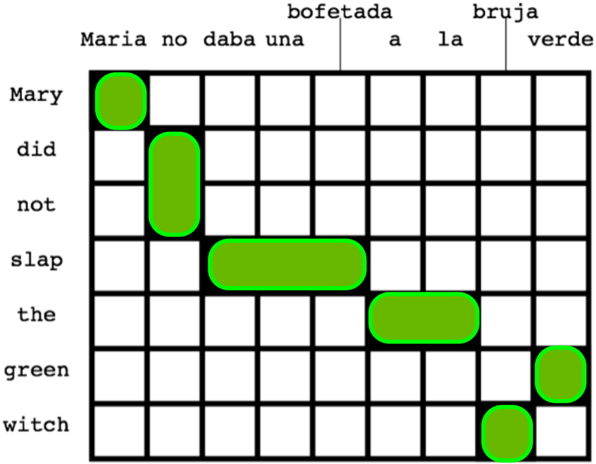
Extracting Phrases

After inferring word alignments, apply heuristics.



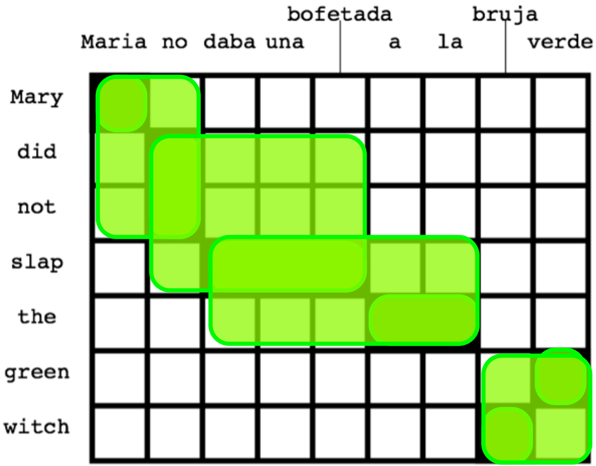
Extracting Phrases

After inferring word alignments, apply heuristics.



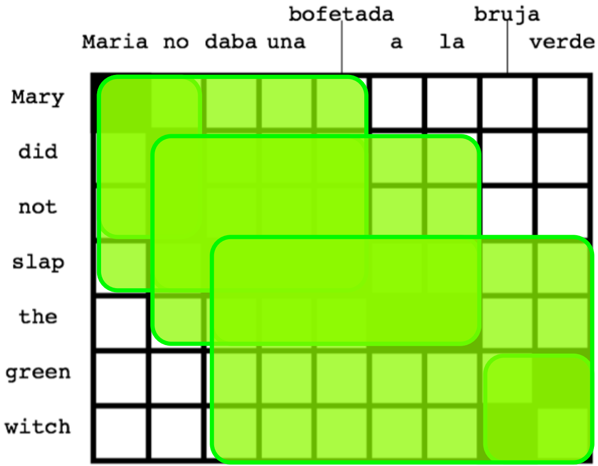
Extracting Phrases

After inferring word alignments, apply heuristics.



Extracting Phrases

After inferring word alignments, apply heuristics.



Scoring Whole Translations

$$s(\boldsymbol{e}, \boldsymbol{a}; \boldsymbol{f}) = \underbrace{\log p(\boldsymbol{e})}_{\text{language model}} + \underbrace{\log p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e})}_{\text{translation model}}$$

Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.

Scoring Whole Translations

$$s(\boldsymbol{e}, \boldsymbol{a}; \boldsymbol{f}) = \underbrace{\log p(\boldsymbol{e})}_{\text{language model}} + \underbrace{\log p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e})}_{\text{translation model}} + \underbrace{\log p(\boldsymbol{e}, \boldsymbol{a} \mid \boldsymbol{f})}_{\text{reverse t.m.}}$$

Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.
- I am simplifying!
  - Reverse translation model typically included.

Scoring Whole Translations

$$s(\boldsymbol{e}, \boldsymbol{a}; \boldsymbol{f}) = \beta_{\text{l.m.}} \underbrace{\log p(\boldsymbol{e})}_{\text{language model}} + \beta_{\text{t.m.}} \underbrace{\log p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e})}_{\text{translation model}}$$
$$+ \beta_{\text{r.t.m.}} \underbrace{\log p(\boldsymbol{e}, \boldsymbol{a} \mid \boldsymbol{f})}_{\text{reverse t.m.}}$$

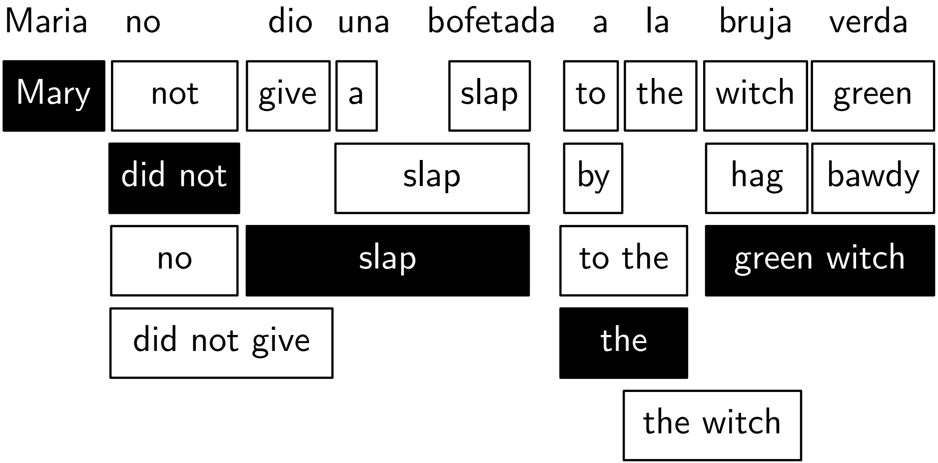
Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.
- I am simplifying!
  - Reverse translation model typically included.
  - Each log-probability is treated as a “feature” and weights are optimized for Bleu performance.

Decoding: Example

Maria	no	dio	una	bofetada	a	la	bruja	verda
Mary	not	give	a	slap	to	the	witch	green
	did not	slap			by		hag	bawdy
	no	slap			to the		green witch	
	did not give				the			
						the witch		

Decoding: Example



Decoding: Example

Maria no dio una bofetada a la bruja verda

Mary | not | give | a | slap | to | the | witch | green

did not | slap | by | hag | bawdy

no | slap | to the | green witch

did not give | the

the witch

Decoding

Adapted from Koehn et al. (2006).

Typically accomplished with **beam** search.

Initial state: $\langle \underbrace{\circ \circ \ldots \circ}_{|\boldsymbol{f}|}, \text{``''} \rangle$ with score 0

Goal state: $\langle \underbrace{\bullet \bullet \ldots \bullet}_{|\boldsymbol{f}|}, \boldsymbol{e}^* \rangle$ with (approximately) the highest score

Reaching a new state:

- Find an uncovered span of $\boldsymbol{f}$ for which a phrasal translation exists in the input $(\bar{\boldsymbol{f}}, \bar{\boldsymbol{e}})$
- New state appends $\bar{\boldsymbol{e}}$ to the output and "covers" $\bar{\boldsymbol{f}}$.
- Score of new state includes additional language model, translation model components for the global score.

Decoding Example

Maria	no	dio	una	bofetada	a	la	bruja	verda
Mary	not	give	a	slap	to	the	witch	green
	did not		slap		by		hag	bawdy
	no	slap			to the		green witch	
	did not give				the			
						the witch		

$$\langle \circ\circ\circ\circ\circ\circ\circ\circ\circ, \text{``''} \rangle, 0$$

Decoding Example

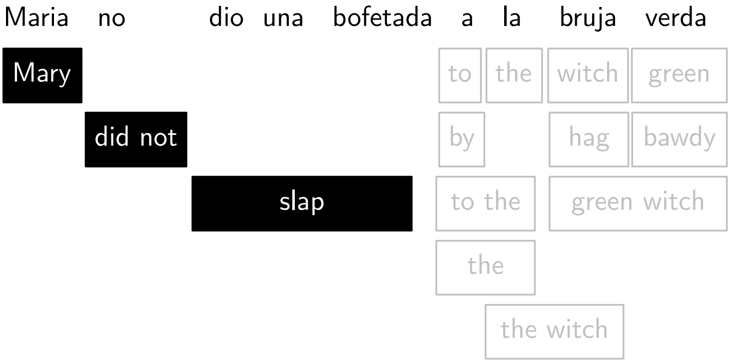
Maria	no	dio	una	bofetada	a	la	bruja	verda
Mary	not	give	a	slap	to	the	witch	green
	did not		slap		by		hag	bawdy
	no	slap			to the		green witch	
	did not give				the			
						the witch		

$$\langle \bullet \circ \circ \circ \circ \circ \circ \circ \circ, \text{“Mary”} \rangle, \ \log p_{\text{l.m.}}(\text{Mary}) + \log p_{\text{t.m.}}(\text{Maria} \mid \text{Mary})$$

Decoding Example

$$\langle \bullet\,\bullet\,\circ\,\circ\,\circ\,\circ\,\circ\,\circ\,\circ, \text{“Mary did not”} \rangle,$$
$$\log p_{\text{l.m.}}(\text{Mary did not}) + \log p_{\text{t.m.}}(\text{Maria} \mid \text{Mary})$$
$$+ \log p_{\text{t.m.}}(\text{no} \mid \text{did not})$$

Decoding Example



$$\langle \bullet\bullet\bullet\bullet\bullet\circ\circ\circ\circ, \text{“Mary did not slap”}\rangle,$$
$$\log p_{\text{l.m.}}(\text{Mary did not slap}) + \log p_{\text{t.m.}}(\text{Maria} \mid \text{Mary})$$
$$+ \log p_{\text{t.m.}}(\text{no} \mid \text{did not}) + \log p_{\text{t.m.}}(\text{dio una bofetada} \mid \text{slap})$$

Machine Translation: Remarks

Sometimes phrases are organized hierarchically (Chiang, 2007).

Extensive research on syntax-based machine translation (Galley et al., 2004), but requires considerable engineering to match phrase-based systems.

Recent work on semantics-based machine translation (Jones et al., 2012); remains to be seen!

Some good pre-neural overviews: Lopez (2008); Koehn (2009)

Neural Machine Translation

Original idea proposed by Forcada and Ñeco (1997); resurgence in interest starting around 2013.

Strong starting point for current work: Bahdanau et al. (2014). (My exposition is borrowed with gratitude from a lecture by Chris Dyer.)

This approach eliminates (hard) alignment and phrases.

Take care: here, the terminology “encoder” and “decoder” are used differently than in the noisy-channel pattern.

High-Level Model

$$
\begin{aligned}
p(\boldsymbol{E} = \boldsymbol{e} \mid \boldsymbol{f}) &= p(\boldsymbol{E} = \boldsymbol{e} \mid \mathsf{encode}(\boldsymbol{f})) \\
&= \prod_{j=1}^{\ell} p(e_j \mid e_0, \ldots, e_{j-1}, \mathsf{encode}(\boldsymbol{f}))
\end{aligned}
$$

The encoding of the source sentence is a *deterministic* function of the words in that sentence.

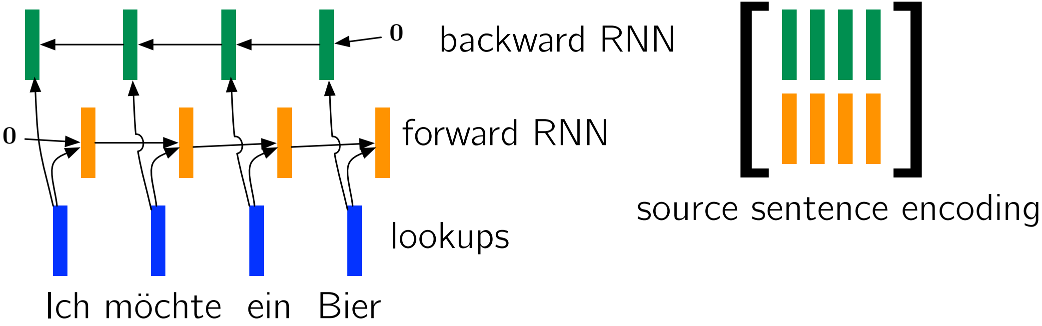
Building Block: Recurrent Neural Network

Review from earlier in the course!

- Each input element is understood to be an element of a sequence: $\langle \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell \rangle$
- At each timestep t:
  - The tth input element $\mathbf{x}_t$ is processed alongside the previous state $\mathbf{s}_{t-1}$ to calculate the new **state** $(\mathbf{s}_t)$.
  - The tth output is a function of the state $\mathbf{s}_t$.
  - The *same functions* are applied at each iteration:

$$\begin{aligned} \mathbf{s}_t &= g_{\text{recurrent}}(\mathbf{x}_t, \mathbf{s}_{t-1}) \\ \mathbf{y}_t &= g_{\text{output}}(\mathbf{s}_t) \end{aligned}$$

Neural MT Source-Sentence Encoder



$\mathbf{F}$ is a $d \times m$ matrix encoding the source sentence $\boldsymbol{f}$ (length m).

Decoder: Contextual Language Model

Two inputs, the previous word and the source sentence context.

$$\mathbf{s}_t = g_{\text{recurrent}}(\mathbf{e}_{e_{t-1}}, \underbrace{\mathbf{F}\mathbf{a}_t}_{\text{“context”}}, \mathbf{s}_{t-1})$$

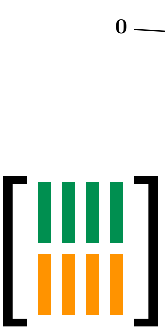
$$\mathbf{y}_t = g_{\text{output}}(\mathbf{s}_t)$$

$$p(E_t = v \mid e_1, \ldots, e_{t-1}, \boldsymbol{f}) = [\mathbf{y}_t]_v$$

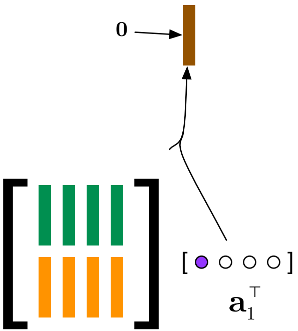
(The forms of the two component gs are suppressed; just remember that they (i) have parameters and (ii) are differentiable with respect to those parameters.)

The neural language model we discussed earlier (Mikolov et al., 2010) didn't have the context as an input to $g_{\text{recurrent}}$.

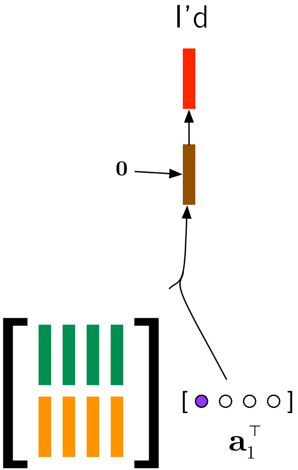
Neural MT Decoder



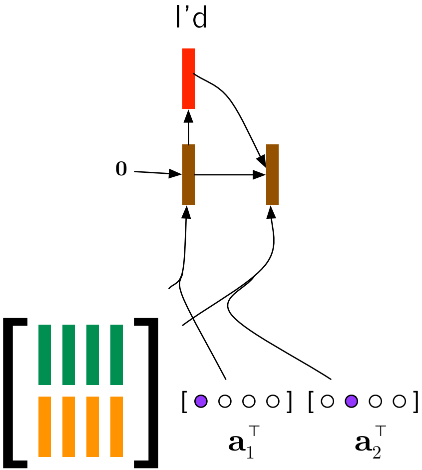
Neural MT Decoder



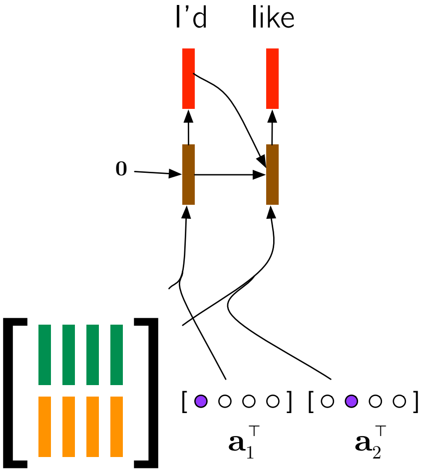
Neural MT Decoder



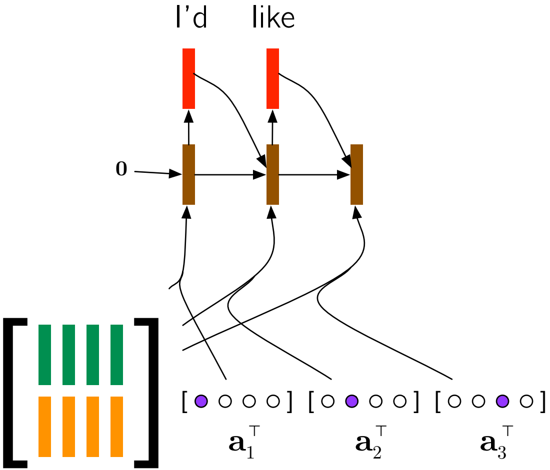
Neural MT Decoder



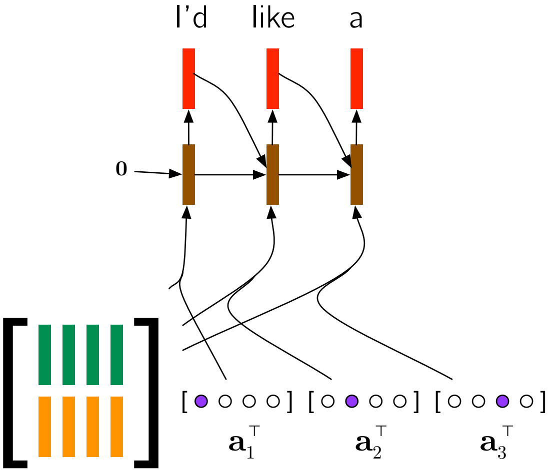
Neural MT Decoder



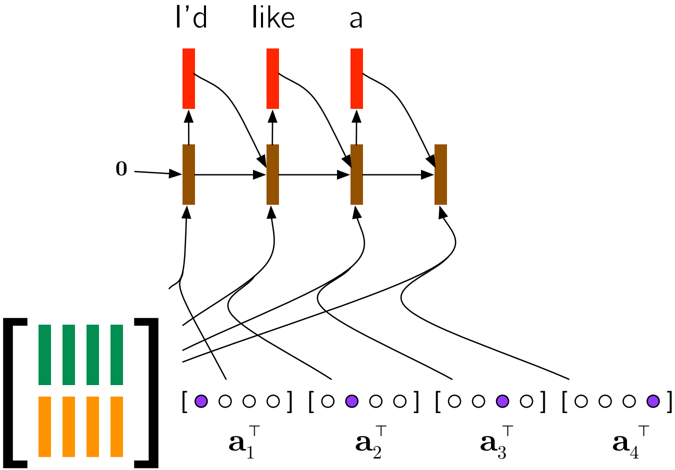
Neural MT Decoder



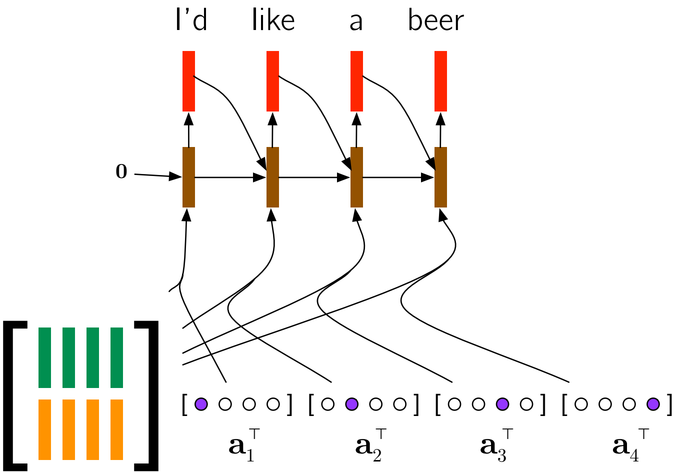
Neural MT Decoder



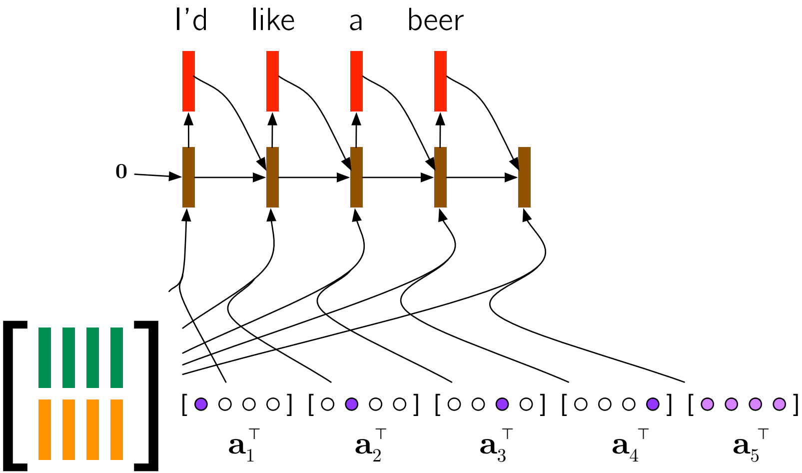
Neural MT Decoder



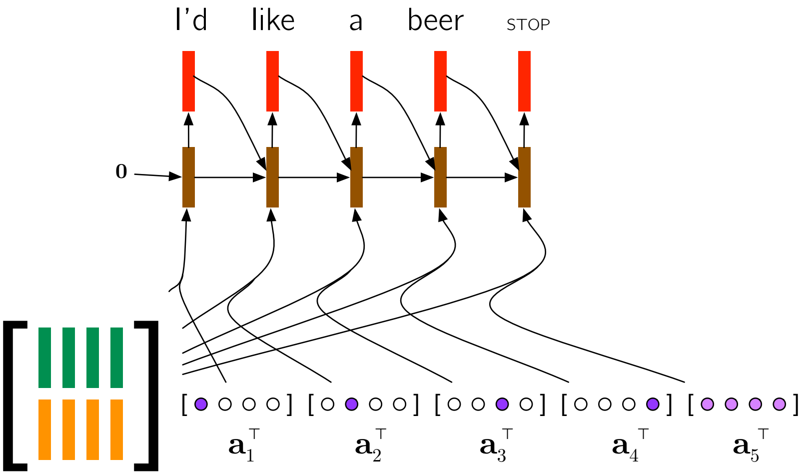
Neural MT Decoder



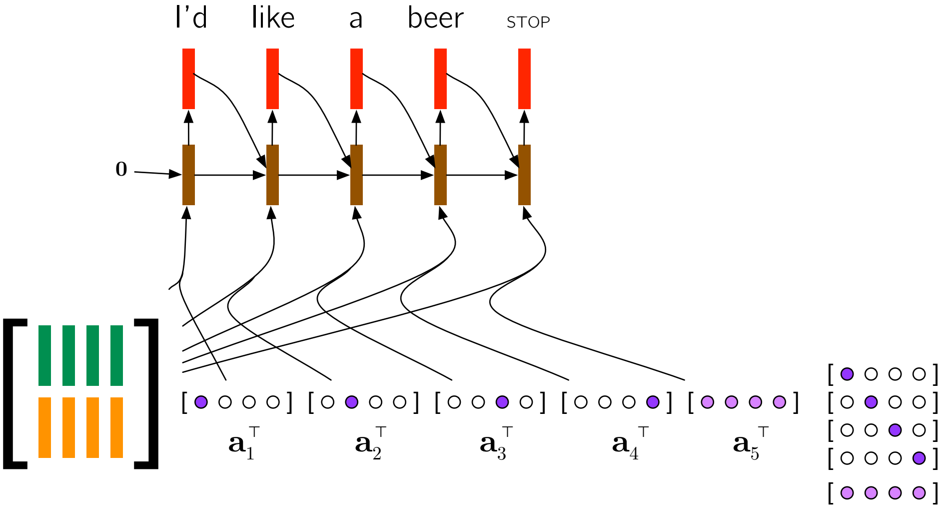
Neural MT Decoder



Neural MT Decoder



Neural MT Decoder



Computing "Attention"

Let $\mathbf{V}\mathbf{s}_{t-1}$ be the "expected" input embedding for timestep t.
(Parameters: $\mathbf{V}$.)

Attention is $\mathbf{a}_t = \text{softmax}\left(\mathbf{F}^\top \mathbf{V}\mathbf{s}_{t-1}\right)$.

Context is $\mathbf{F}\mathbf{a}_t$, i.e., a weighted sum of the source words' in-context representations.

Learning and Decoding

$$\log p(\boldsymbol{e} \mid \mathsf{encode}(\boldsymbol{f})) = \sum_{i=1}^{m} \log p(e_i \mid \boldsymbol{e}_{0:i-1}, \mathsf{encode}(\boldsymbol{f}))$$

is differentiable with respect to all parameters of the neural network, allowing "end-to-end" training.

Trick: train on shorter sentences first, then add in longer ones.

Decoding typically uses beam search.

Remarks

We covered two approaches to machine translation:

- Phrase-based statistical MT following Koehn et al. (2003), including probabilistic noisy-channel models for alignment (a key preprocessing step; Brown et al., 1993), and
- Neural MT with attention, following Bahdanau et al. (2014).

Note two key differences:

- Noisy channel $p(\boldsymbol{e}) \times p(\boldsymbol{f} \mid \boldsymbol{e})$ vs. "direct" model $p(\boldsymbol{e} \mid \boldsymbol{f})$
- Alignment as a discrete random variable vs. attention as a deterministic, differentiable function

At the moment, neural MT is winning, at least when you have enough data.

When monolingual target-language data is plentiful, we'd like to use it! Recent neural models try (Sennrich et al., 2016; Xia et al., 2016; Yu et al., 2017).

References I

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*, 2014. URL `https://arxiv.org/abs/1409.0473`.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

David Chiang. Hierarchical phrase-based translation. *computational Linguistics*, 33(2):201–228, 2007.

Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. of NAACL*, 2013.

Mikel L. Forcada and Ramón P. Ñeco. Recursive hetero-associative memories for translation. In *International Work-Conference on Artificial Neural Networks*, 1997.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *Proc. of NAACL*, 2004.

Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. Semantics-based machine translation with hyperedge replacement grammars. In *Proc. of COLING*, 2012.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proc. of NAACL*, 2003.

References II

Philipp Koehn, Marcello Federico, Wade Shen, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, and Richard Zens. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding, 2006. Final report of the 2006 JHU summer workshop.

Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3):8, 2008.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proc. of Interspeech*, 2010. URL `http://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proc. of ACL*, 2016. URL `http://www.aclweb.org/anthology/P16-1009`.

Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *NIPS*, 2016.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky. The neural noisy channel. In *Proc. of ICLR*, 2017.