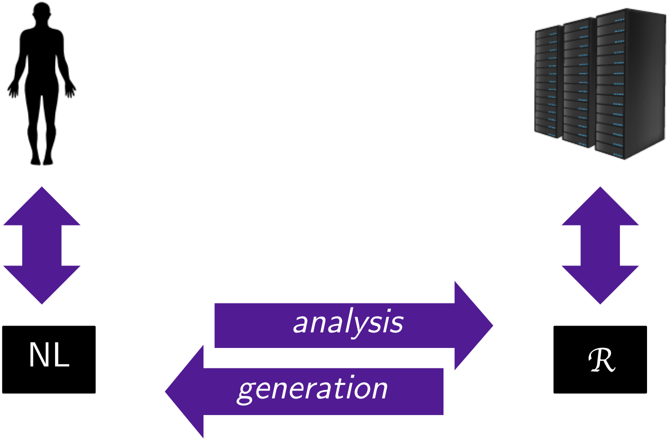
Natural Language Processing (CSE 490U): Generation: Translation & Summarization

Noah Smith

University of Washington
nasmith@cs.washington.edu

March 6–8, 2017

No office hours Thursday.


NL
analysis
generation
$\mathcal{R}$

Natural Language Generation

The classical view: $\mathcal{R}$ is a meaning representation language.

- Often very specific to the domain.
- For a breakdown of the problem space and a survey, see Reiter and Dale (1997).

Today: considerable emphasis on **text-to-text** generation, i.e., transformations:

- Translating a sentence in one language into another language
- Summarizing a long piece of text by a shorter one
- Paraphrase generation (Barzilay and Lee, 2003; Quirk et al., 2004)

Machine Translation

Warren Weaver to Norbert Wiener, 1947

One naturally wonders if the problem of translation could be conceivably treated as a problem in cryptography. When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'

Evaluation

Intuition: good translations are **fluent** in the target language and **faithful** to the original meaning.

Bleu score (Papineni et al., 2002):

- Compare to a human-generated reference translation
- Or, better: multiple references
- Weighted average of n-gram precision (across different n)

There are some alternatives; most papers that use them report Bleu, too.

Noisy Channel Models

Review

A pattern for modeling a pair of random variables, X and Y:

$$\boxed{\text{source}} \longrightarrow Y \longrightarrow \boxed{\text{channel}} \longrightarrow X$$

Noisy Channel Models

Review

A pattern for modeling a pair of random variables, X and Y:

$$\boxed{\text{source}} \longrightarrow Y \longrightarrow \boxed{\text{channel}} \longrightarrow X$$

- Y is the plaintext, the true message, the missing information, the output

Noisy Channel Models

Review

A pattern for modeling a pair of random variables, X and Y:

$$\boxed{\text{source}} \longrightarrow Y \longrightarrow \boxed{\text{channel}} \longrightarrow X$$

- Y is the plaintext, the true message, the missing information, the output
- X is the ciphertext, the garbled message, the observable evidence, the input

Noisy Channel Models

Review

A pattern for modeling a pair of random variables, X and Y:

$$\boxed{\text{source}} \longrightarrow Y \longrightarrow \boxed{\text{channel}} \longrightarrow X$$

- Y is the plaintext, the true message, the missing information, the output
- X is the ciphertext, the garbled message, the observable evidence, the input
- Decoding: select y given $X = x$.

$$\begin{aligned} y^* &= \operatorname*{argmax}_y p(y \mid x) \\ &= \operatorname*{argmax}_y \frac{p(x \mid y) \cdot p(y)}{p(x)} \\ &= \operatorname*{argmax}_y \underbrace{p(x \mid y)}_{\text{channel model}} \cdot \underbrace{p(y)}_{\text{source model}} \end{aligned}$$

Bitext/Parallel Text

Let $\boldsymbol{f}$ and $\boldsymbol{e}$ be two sequences in $\mathcal{V}^\dagger$ (French) and $\bar{\mathcal{V}}^\dagger$ (English), respectively.

We're going to define $p(\boldsymbol{F} \mid \boldsymbol{e})$, the probability over French translations of English sentence $\boldsymbol{e}$.

In a noisy channel machine translation system, we could use this together with source/language model $p(\boldsymbol{e})$ to "decode" $\boldsymbol{f}$ into an English translation.

Where does the data to estimate this come from?

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.

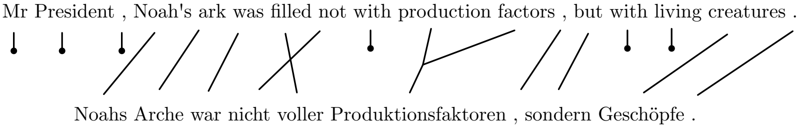
$$p(\boldsymbol{f} \mid \boldsymbol{e}, m) = \sum_{a_1=0}^{\ell} \sum_{a_2=0}^{\ell} \cdots \sum_{a_m=0}^{\ell} p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m)$$

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

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

$$= \frac{1}{\ell + 1} \cdot \theta_{f_i \mid e_{a_i}}$$

Example: $\boldsymbol{f}$ is German



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

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

Example: $\boldsymbol{f}$ is German

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

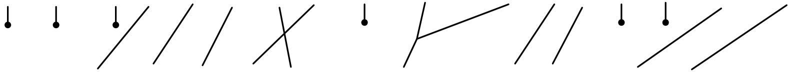
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

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

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

Example: $\boldsymbol{f}$ is German

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

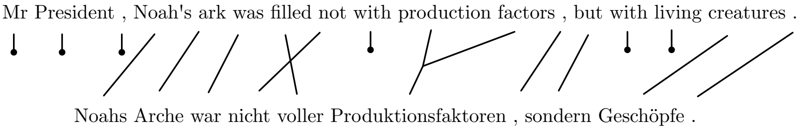


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

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

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

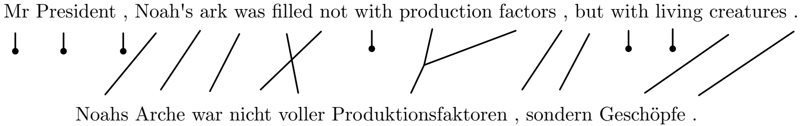
Example: $\boldsymbol{f}$ is German



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

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

Example: $\boldsymbol{f}$ is German

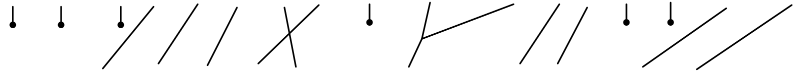


$$\boldsymbol{a} = \langle 4, 5, 6, 8, 7, \ldots \rangle$$

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

Example: $\boldsymbol{f}$ is German

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

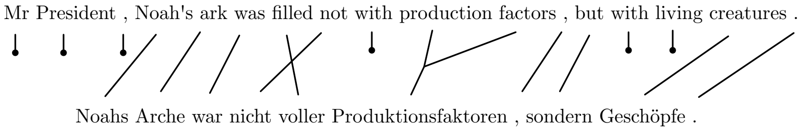


Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

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

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

Example: $\boldsymbol{f}$ is German

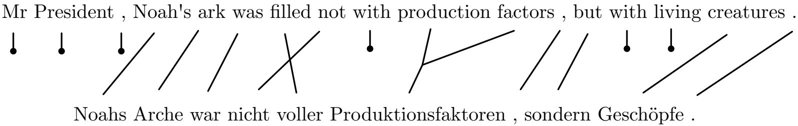


$$\boldsymbol{a} = \langle 4, 5, 6, 8, 7, ?, \ldots \rangle$$

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

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

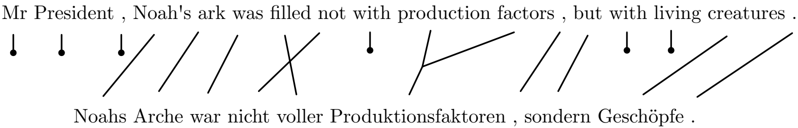
Example: $\boldsymbol{f}$ is English



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

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

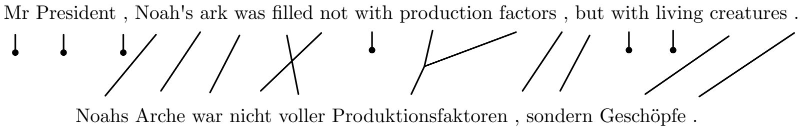
Example: $\boldsymbol{f}$ is English



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

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

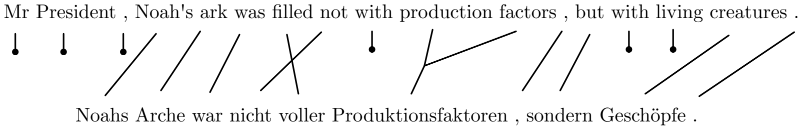
Example: $\boldsymbol{f}$ is English



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

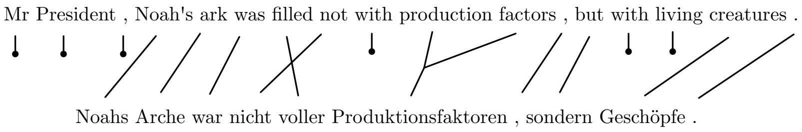
$$p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) = \frac{1}{10+1} \cdot \theta_{\textsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{President}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{Noah's}|\textsf{Noahs}}$$

Example: $\boldsymbol{f}$ is English



$$
\begin{aligned}
\boldsymbol{a} &= \langle 0, 0, 0, 1, 2, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{10+1} \cdot \theta_{\textsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{President}|\textsc{null}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\textsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{Noah's}|\textsf{Noahs}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\textsf{ark}|\textsf{Arche}}
\end{aligned}
$$

Example: $\boldsymbol{f}$ is English



$$
\begin{aligned}
\boldsymbol{a} &= \langle 0, 0, 0, 1, 2, 3, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{10+1} \cdot \theta_{\mathsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{President}|\textsc{null}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{Noah's}|\mathsf{Noahs}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{ark}|\mathsf{Arche}} \cdot \frac{1}{10+1} \cdot \theta_{\mathsf{was}|\mathsf{war}}
\end{aligned}
$$

Example: $\boldsymbol{f}$ is English

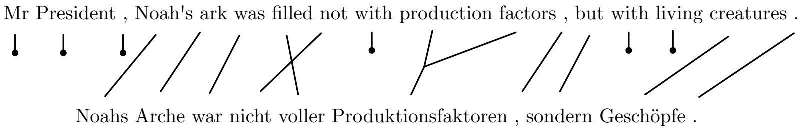
Mr President , Noah's ark was filled not with production factors , but with living creatures .

Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

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

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

Example: $\boldsymbol{f}$ is English



$$
\begin{aligned}
\boldsymbol{a} &= \langle 0, 0, 0, 1, 2, 3, 5, 4, \ldots \rangle \\
p(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{e}, m) &= \frac{1}{10+1} \cdot \theta_{\textsf{Mr}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{President}|\textsc{null}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\textsf{,}|\textsc{null}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{Noah's}|\textsf{Noahs}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\textsf{ark}|\textsf{Arche}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{was}|\textsf{war}} \\
&\quad \cdot \frac{1}{10+1} \cdot \theta_{\textsf{filled}|\textsf{voller}} \cdot \frac{1}{10+1} \cdot \theta_{\textsf{not}|\textsf{nicht}}
\end{aligned}
$$

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 $\theta_{f|e}$:

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

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.

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

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

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

Expectation Maximization

- 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 $\boldsymbol{f}$.

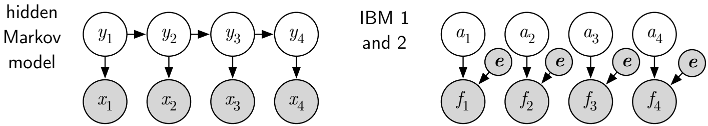
Latent variable $\boldsymbol{a} = \langle a_1, \ldots, a_m \rangle$, each a_i ranging over $\{0, \ldots, \ell\}$ (positions in $\boldsymbol{e}$).

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

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

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

IBM Models 1 and 2, Depicted



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

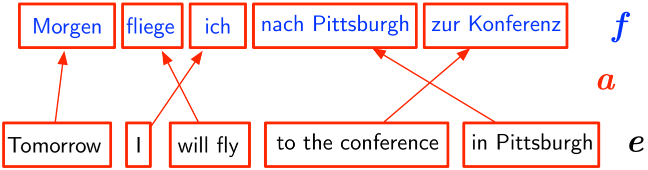
Courtesy of Chris Dyer.

German	English	$p(\bar{f} \mid \bar{e})$
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).

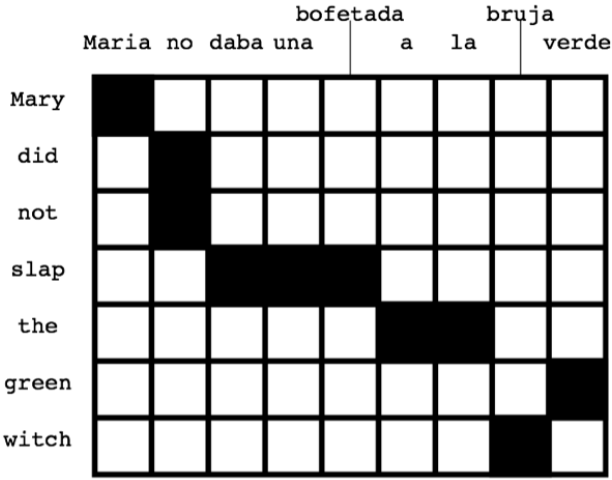
R.v. $\boldsymbol{A}$ captures segmentation of sentences into phrases, alignment between them, and reordering.



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

Extracting Phrases

After inferring word alignments, apply heuristics.



Extracting Phrases

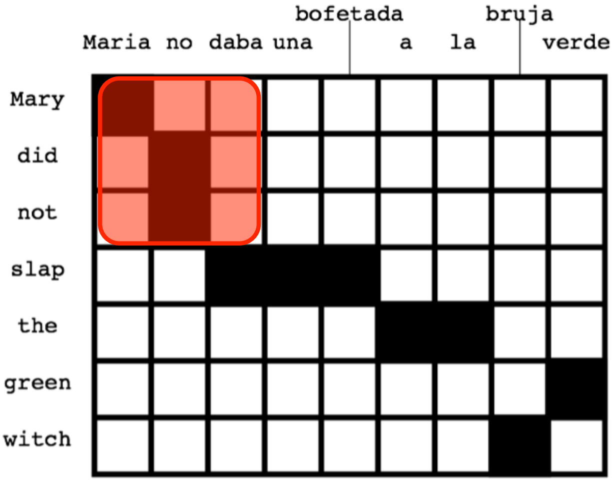
After inferring word alignments, apply heuristics.

Extracting Phrases

After inferring word alignments, apply heuristics.

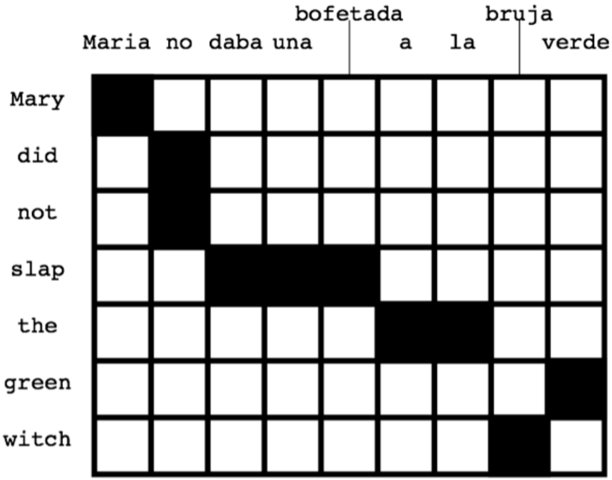
Extracting Phrases

After inferring word alignments, apply heuristics.



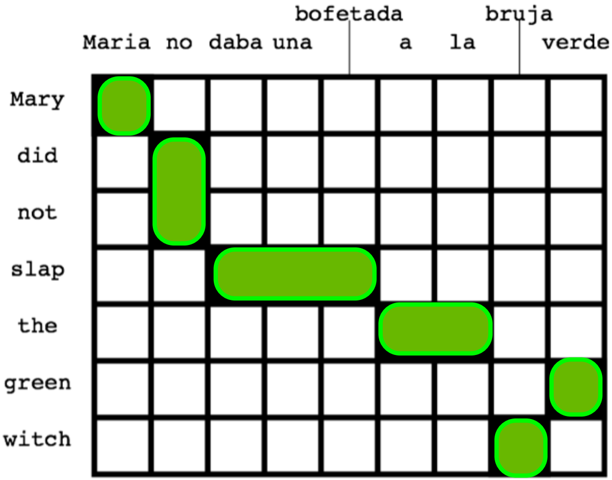
Extracting Phrases

After inferring word alignments, apply heuristics.



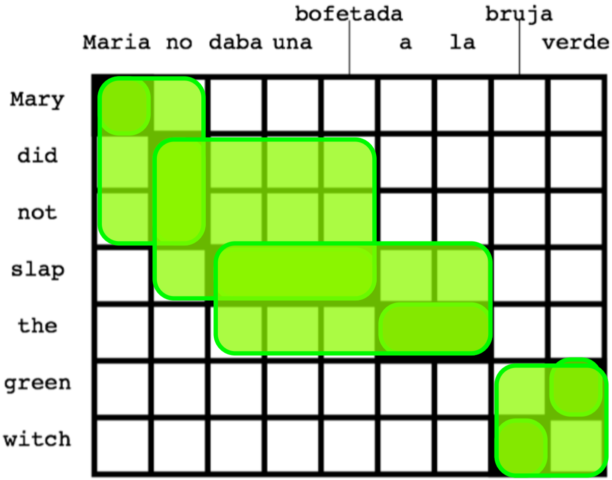
Extracting Phrases

After inferring word alignments, apply heuristics.



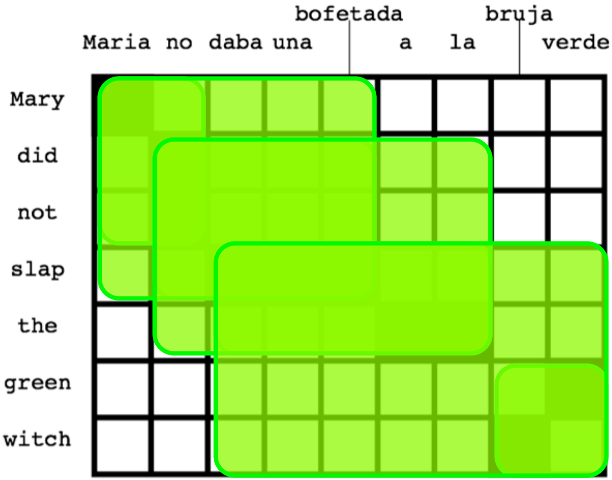
Extracting Phrases

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.

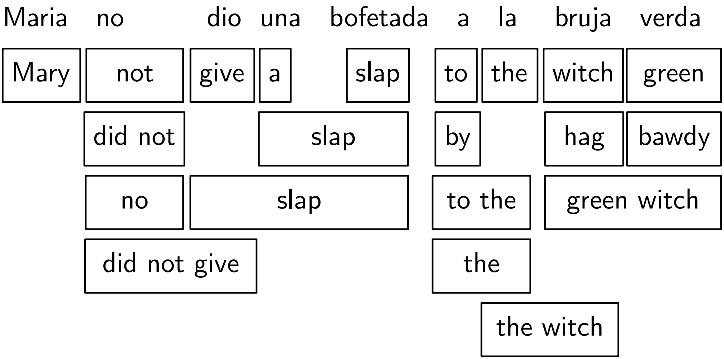
Scoring Whole Translations

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

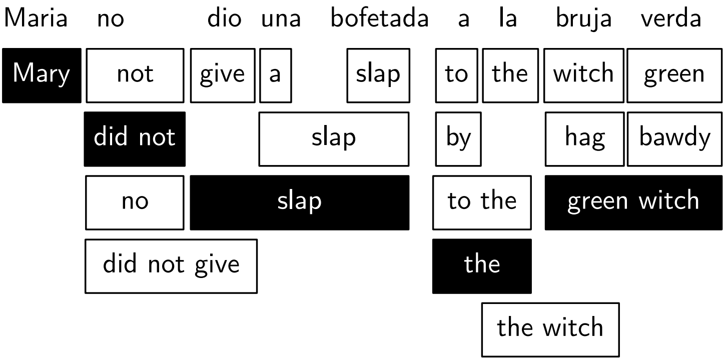
Remarks:

- 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



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

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

Decoding Example

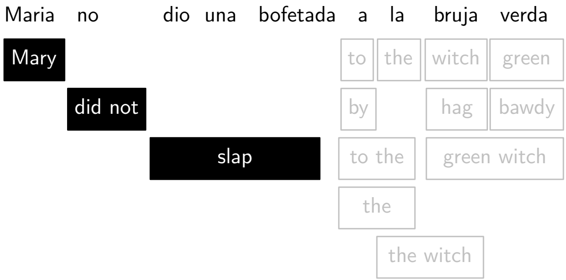
Maria	no	dio	una	bofetada	a	la	bruja	verda
Mary	not	give	a	slap	to	the	witch	green
	did not		slap		by		hag	bawdy
	no	slap			to the		green witch	
	did not give				the			
						the witch		

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

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!

Neural models have become popular and are competitive (e.g., Devlin et al., 2014); impact remains to be seen!

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

Summarization

Automatic Text Summarization

Survey from before statistical methods came to dominate: Mani, 2001

Parallel history to machine translation:

- Noisy channel view (Knight and Marcu, 2002)
- Automatic evaluation (Lin, 2004)

Differences:

- Natural data sources are less obvious
- Human information needs are less obvious

We'll briefly consider two subtasks: **compression** and **selection**

Sentence Compression as Structured Prediction

(McDonald, 2006)

Input: a sentence

Output: the same sentence, with some words deleted

McDonald's approach:

- Define a scoring function for compressed sentences that factors locally in the output.
  - He factored into *bigrams* but considered input parse tree features.
- Decoding is dynamic programming (not unlike Viterbi).
- Learn feature weights from a corpus of compressed sentences, using structured perceptron or similar.

Sentence Selection

Input: one or more documents and a "budget"

Output: a within-budget subset of sentences (or passages) from the input

Challenge: **diminishing returns** as more sentences are added to the summary.

Classical greedy method: "maximum marginal relevance" (Carbonell and Goldstein, 1998)

Casting the problem as **submodular optimization**: Lin and Bilmes (2009)

Joint selection and compression: Martins and Smith (2009)

To-Do List

- Course evaluation due March 12!
- Collins (2011, 2013)
- Assignment 5 due Friday

References I

Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proc. of NAACL*, 2003.

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.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR*, 1998.

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

Michael Collins. Statistical machine translation: IBM models 1 and 2, 2011. URL `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf`.

Michael Collins. Phrase-based translation models, 2013. URL `http://www.cs.columbia.edu/~mcollins/pb.pdf`.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M. Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proc. of ACL*, 2014.

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

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

References II

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.

Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1): 91–107, 2002.

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.

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.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. of ACL Workshop: Text Summarization Branches Out*, 2004.

Hui Lin and Jeff A. Bilmes. How to select a good training-data subset for transcription: Submodular active selection for sequences. In *Proc. of Interspeech*, 2009.

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

Inderjeet Mani. *Automatic Summarization*. John Benjamins Publishing, 2001.

References III

André FT Martins and Noah A Smith. Summarization with a joint model for sentence extraction and compression. In *Proc. of the ACL Workshop on Integer Linear Programming for Natural Langauge Processing*, 2009.

Ryan T. McDonald. Discriminative sentence compression with soft syntactic evidence. In *Proc. of EACL*, 2006.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, 2002.

Chris Quirk, Chris Brockett, and William B. Dolan. Monolingual machine translation for paraphrase generation. In *Proc. of EMNLP*, 2004.

Ehud Reiter and Robert Dale. Building applied natural-language generation systems. *Journal of Natural-Language Engineering*, 3:57–87, 1997. URL http://homepages.abdn.ac.uk/e.reiter/pages/papers/jnle97.pdf.