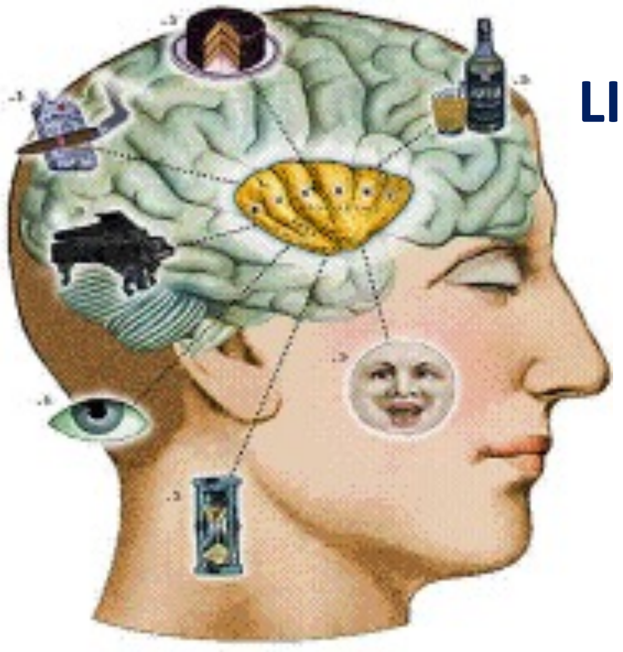
Generative Classification



LING83800: METHODS IN COMPUTATIONAL LINGUISTICS II
April 15, 2024
Spencer Caplan

Today

1. Expectation Maximization

2. POS Induction

3. Text Classification

4. Naïve Bayes Model

Warren Weaver: 1949 Memorandum

- Proposes Machine Translation using Information Theory!

"It is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the "Chinese code." If we have useful methods for solving almost any cryptographic problem, may it not be that with proper interpretation we already have useful methods for translation?"

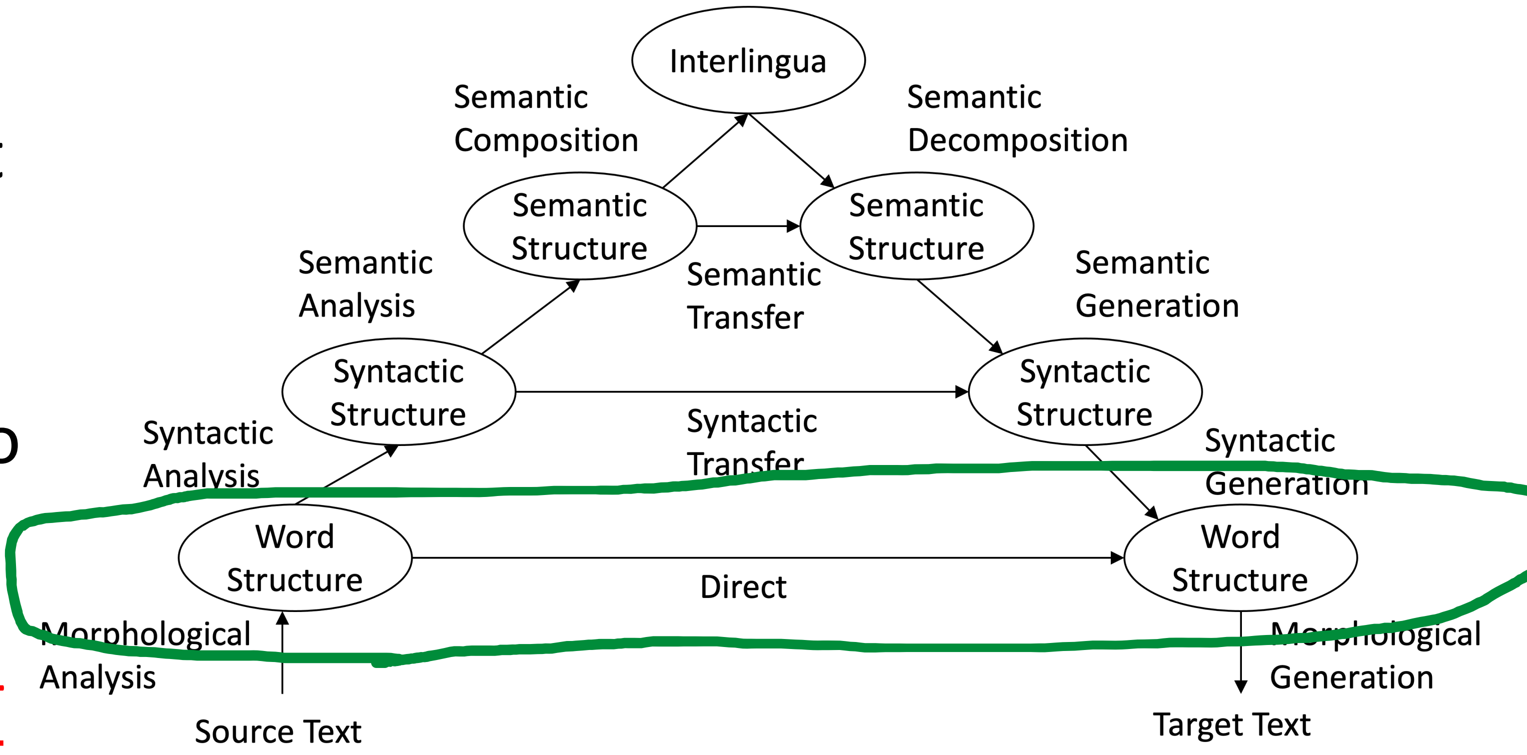
$\longrightarrow$ **Core idea of the current approach!**

Weaver, W. (1949): 'Translation'. Repr. in: Locke, W.N. and Booth, A.D. (eds.) *Machine translation of languages: fourteen essays* (Cambridge, Mass.: Technology Press of the Massachusetts Institute of Technology, 1955), pp. 15-23.

Don’t let the perfect MT be the enemy of the good

- Bar-Hillel: “MT requires a machine to understand the sentence to be translated, but we are so far from designing programs that could *understand* human language that we should put off MT into the indefinite future”

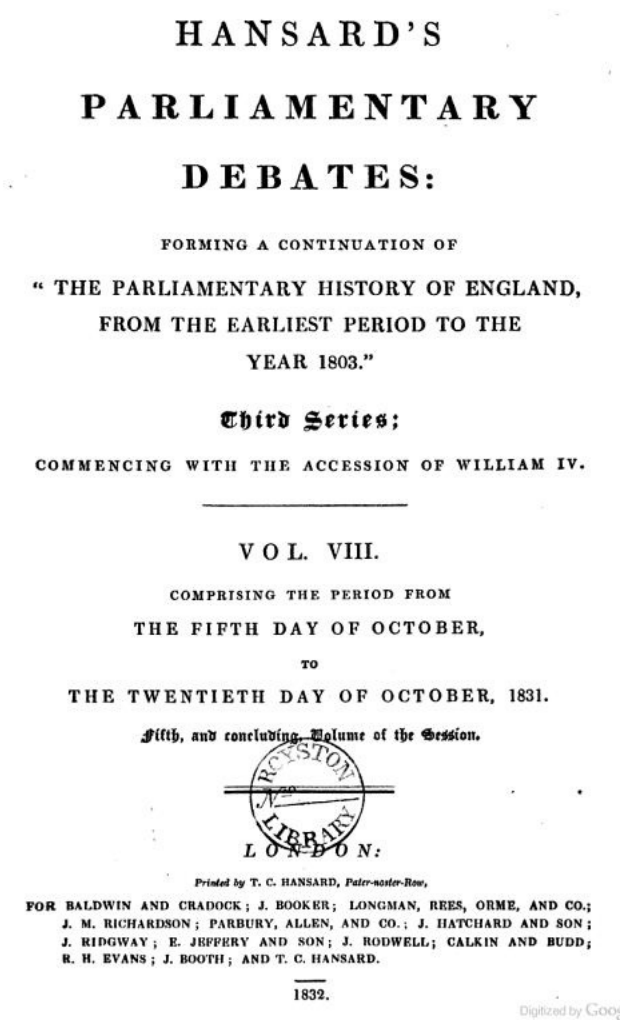
- How far can raw statistical MT get us?



Parallel Corpora for MT

- E.g. *Canadian Hansard's corpus*

- Sentence alignment vs. word alignment



HANSARD'S

PARLIAMENTARY

DEBATES:

FORMING A CONTINUATION OF

" THE PARLIAMENTARY HISTORY OF ENGLAND, FROM THE EARLIEST PERIOD TO THE YEAR 1803."

Third Series;

COMMENCING WITH THE ACCESSION OF WILLIAM IV.

VOL. VIII.

COMPRISING THE PERIOD FROM

THE FIFTH DAY OF OCTOBER,

TO

THE TWENTIETH DAY OF OCTOBER, 1831.

Fifth, and concluding, Volume of the Seſsion.

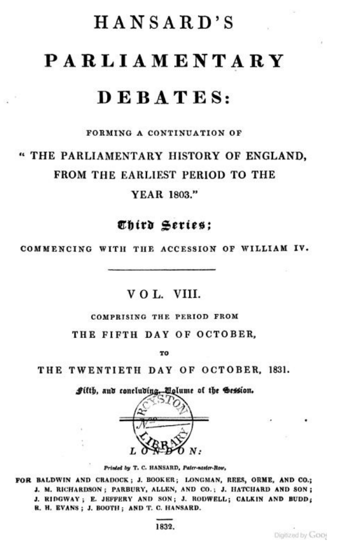
LONDON:

Printed by T. C. HANSARD, *Pater-noster-Row,*

FOR BALDWIN AND CRADOCK; J. BOOKER; LONGMAN, REES, ORME, AND CO.; J. M. RICHARDSON; PARBURY, ALLEN, AND CO.; J. HATCHARD AND SON; J. RIDGWAY; E. JEFFERY AND SON; J. RODWELL; CALKIN AND BUDD; R. H. EVANS; J. BOOTH; AND T. C. HANSARD.

1832.

Parallel Corpora for MT


HANSARD'S

PARLIAMENTARY

DEBATES:

FORMING A CONTINUATION OF

" THE PARLIAMENTARY HISTORY OF ENGLAND, FROM THE EARLIEST PERIOD TO THE YEAR 1803."

Third Series;

COMMENCING WITH THE ACCESSION OF WILLIAM IV.

VOL. VIII.

COMPRISING THE PERIOD FROM THE FIFTH DAY OF OCTOBER, TO THE TWENTIETH DAY OF OCTOBER, 1831.

Fifth, and concluding, Volume of the Session.

LONDON:

Printed by T. C. HANSARD, Pater-noster-Row,

FOR BALDWIN AND CRADOCK; J. BOOKER; LONGMAN, REES, ORME, AND CO.; J. M. RICHARDSON; PARBURY, ALLEN, AND CO.; J. HATCHARD AND SON; J. RIDGWAY; E. JEFFERY AND SON; J. RODWELL; CALKIN AND BUDD; R. H. EVANS; J. BOOTH; AND T. C. HANSARD.

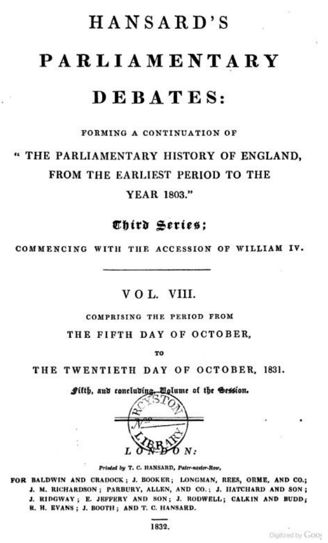
1832.

- E.g. *Canadian Hansard's corpus*
- Sentence alignment vs. word alignment

Much like our tokenization scheme for English: splitting "doesn't" into "does + n't"

J 'ai acheté du pain	I bought some bread
J 'ai acheté du beurre	I bought some butter
Nous devons manger le pain blanc	We must eat the white bread

Parallel Corpora for MT


HANSARD'S

PARLIAMENTARY

DEBATES:

FORMING A CONTINUATION OF

" THE PARLIAMENTARY HISTORY OF ENGLAND,
FROM THE EARLIEST PERIOD TO THE
YEAR 1803."

Third Series;

COMMENCING WITH THE ACCESSION OF WILLIAM IV.

VOL. VIII.

COMPRISING THE PERIOD FROM
THE FIFTH DAY OF OCTOBER,
TO
THE TWENTIETH DAY OF OCTOBER, 1831.

Fifth, and concluding, Volume of the Session.

LONDON:

Printed by T. C. HANSARD, Pater-noster-Row,

FOR BALDWIN AND CRADOCK; J. BOOKER; LONGMAN, REES, ORME, AND CO.; J. M. RICHARDSON; PARBURY, ALLEN, AND CO.; J. HATCHARD AND SON; J. RIDGWAY; E. JEFFERY AND SON; J. RODWELL; CALKIN AND BUDD; R. H. EVANS; J. BOOTH; AND T. C. HANSARD.

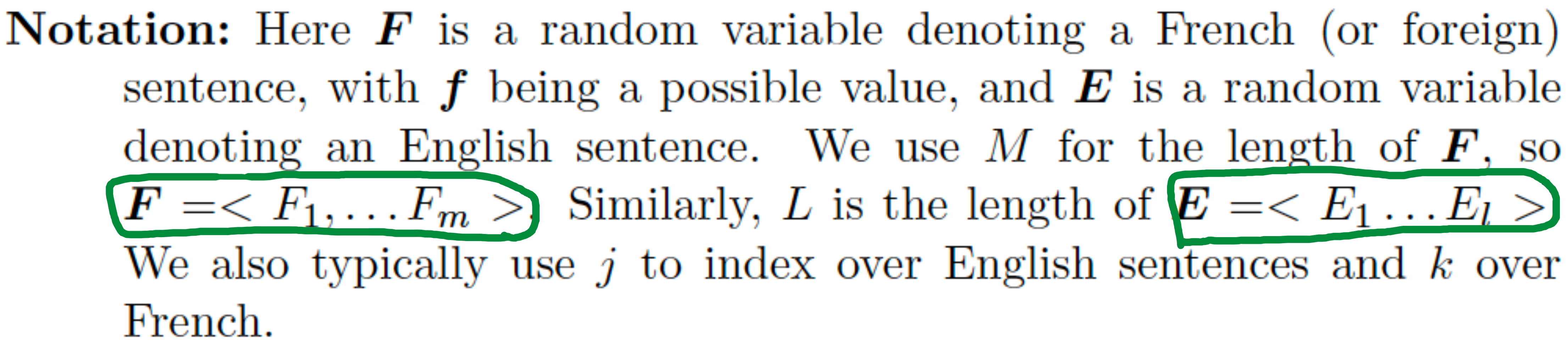
1832.

- E.g. *Canadian Hansard's corpus*
- Sentence alignment vs. word alignment

This kind of distributional regularity is the simple idea behind statistical MT

J 'ai acheté du pain	I bought some bread
J 'ai acheté du beurre	I bought some butter
Nous devons manger le pain blanc	We must eat the white bread

Formalizing the problem



Notation: Here $\boldsymbol{F}$ is a random variable denoting a French (or foreign) sentence, with $\boldsymbol{f}$ being a possible value, and $\boldsymbol{E}$ is a random variable denoting an English sentence. We use M for the length of $\boldsymbol{F}$, so $\boxed{\boldsymbol{F} =< F_1, \ldots F_m >}$ Similarly, L is the length of $\boxed{\boldsymbol{E} =< E_1 \ldots E_l >}$ We also typically use j to index over English sentences and k over French.

(Variable naming convention is just alphabetical here)
E before F
l before m
j before k

Fundamental Theorem of MT

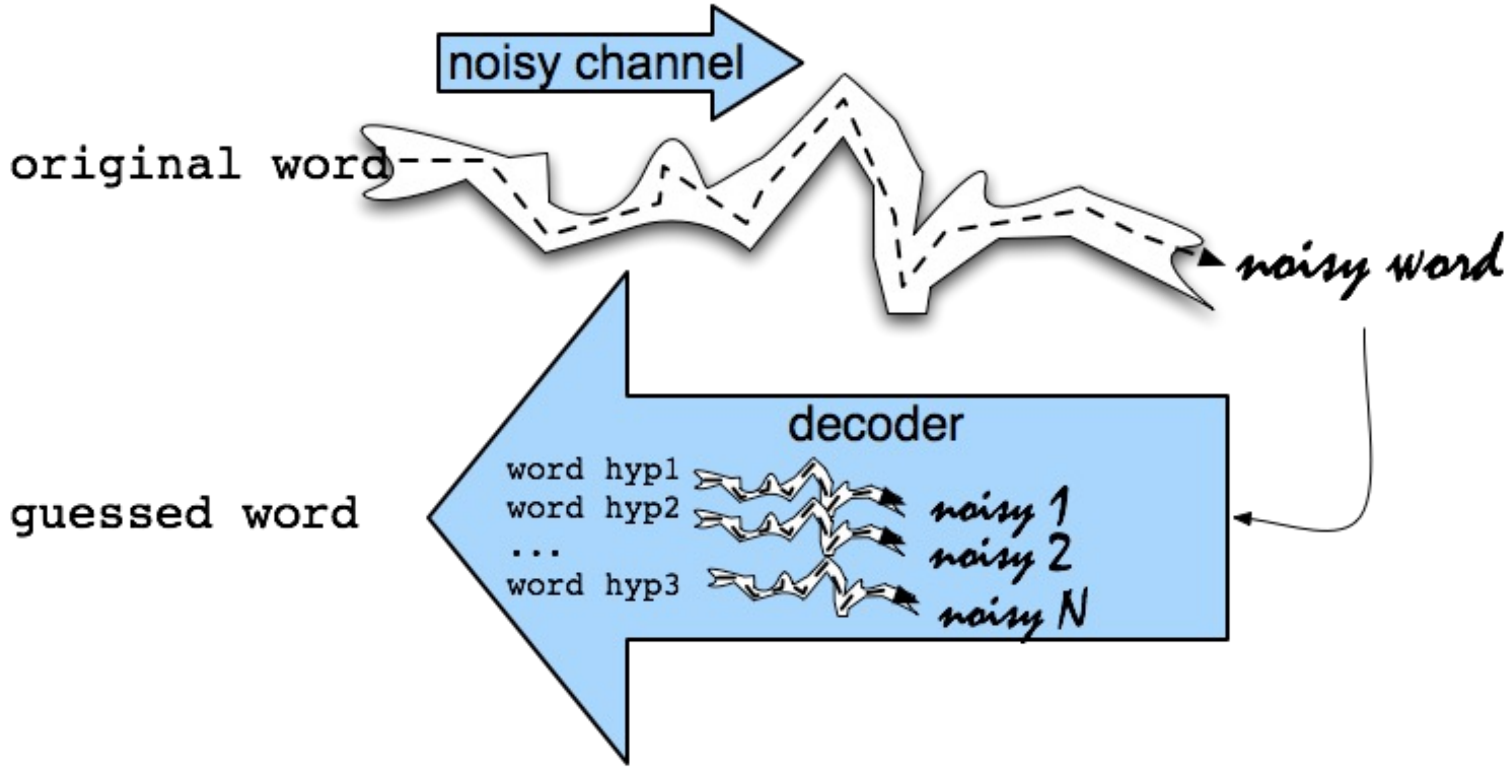
- From a probabilistic point of view, MT can be formalized as finding the most probable translation ***e*** of a foreign language string ***f***, which is

$$\arg\max_{\boldsymbol{e}} \mathrm{P}(\boldsymbol{e} \mid \boldsymbol{f})$$

French Sentence:
"Nous devons manger le pain blanc"

Hyp1	We must eat the white bread
Hyp2	We must eat the bread white
Hyp3	We eat must the bread white

Noisy Chanel Model



Noisy Channel Model

- We see an observation x of a ~~misspelled word~~ french sentence
- Find the correct ~~word~~ w

English sentence

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w \mid x)$$

$$= \underset{w \in V}{\operatorname{argmax}} \frac{P(x \mid w)\boxed{P(w)}}{P(x)}$$

Language model

spelling

Noisy Channel Model

- We see an observation x of a ~~misspelled word~~ french sentence
- Find the correct ~~word~~ w

English sentence

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w \mid x)$$

$$= \underset{w \in V}{\operatorname{argmax}} \frac{P(x \mid w) P(w)}{P(x)}$$

$$= \underset{w \in V}{\operatorname{argmax}} \underbrace{P(x \mid w)}_{\text{Channel model (Translation Model)}} \underbrace{P(w)}_{\text{Prior probability (Language Model)}}$$

Prior probability
(Language Model)

Channel model
(Translation Model)

Fundamental Theorem of MT

$$\arg\max_{\boldsymbol{e}} \mathrm{P}(\boldsymbol{e} \mid \boldsymbol{f}) = \arg\max_{\boxed{\boldsymbol{e}}} \boxed{\mathrm{P}(\boldsymbol{e})}\boxed{\mathrm{P}(\boldsymbol{f} \mid \boldsymbol{e})}$$

Iterate over candidate set

Language model

Translation model

Benefits of noisy channel factorization

- The translation and language models capture different kinds of dependencies
- The fundamental theorem of MT tells us how these should be combined

French Sentence: “Nous devons manger le pain blanc”

Hyp1	We must eat the white bread
Hyp2	We must eat the bread white
Hyp3	We eat must the bread white

Benefits of noisy channel factorization

- Word reordering in translation handled by P(E)
  - P(E) factor frees P(F | E) from worrying about word order in the “Source” language
- Word choice in translation handled by P (F|E)
  - P(F| E) factor frees P(E) from worrying about picking the right translation

Benefits of noisy channel factorization

- The translation and language models capture different kinds of dependencies
- The fundamental theorem of MT tells us how these should be combined

$$\arg\max_{\boldsymbol{e}} \mathrm{P}(\boldsymbol{e} \mid \boldsymbol{f}) = \arg\max_{\boldsymbol{e}} \underbrace{\boxed{\mathrm{P}(\boldsymbol{e})}}_{\text{Language model}} \underbrace{\boxed{\mathrm{P}(\boldsymbol{f} \mid \boldsymbol{e})}}_{\text{Translation model}}$$

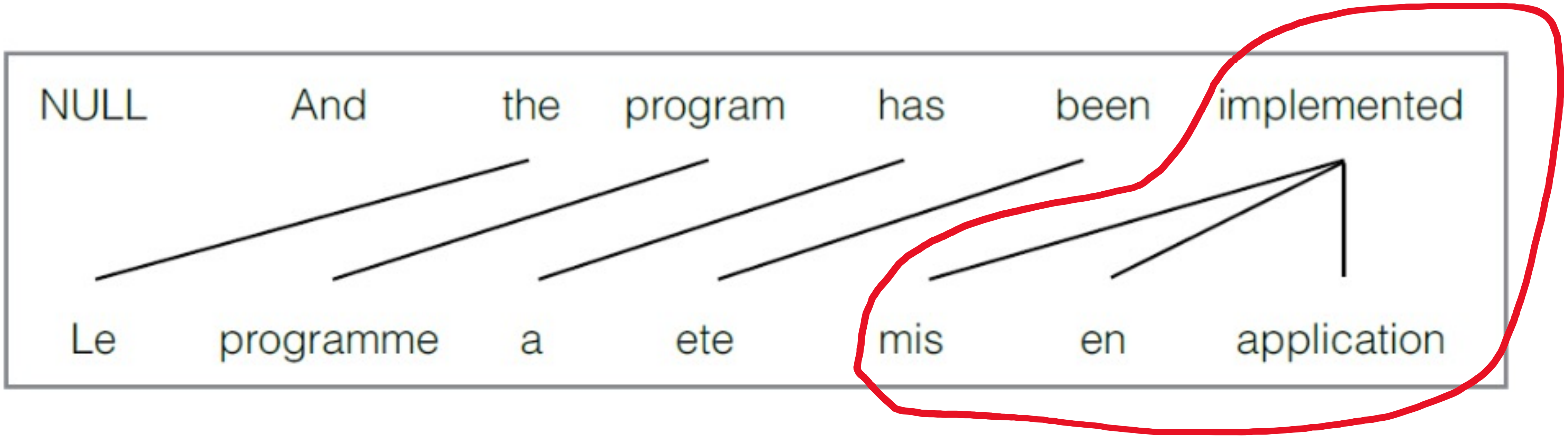
Only the translation model requires parallel training data, language model can be trained on a monolingual corpus

IBM Model 1: Assumptions

Assumption 1: each French word f_k is aligned to exactly one English word e_j

- (including a special NULL token)

But not necessarily one-to-one



IBM Model 1: Assumptions

Assumption 1: each French word f_k is aligned to exactly one English word e_j

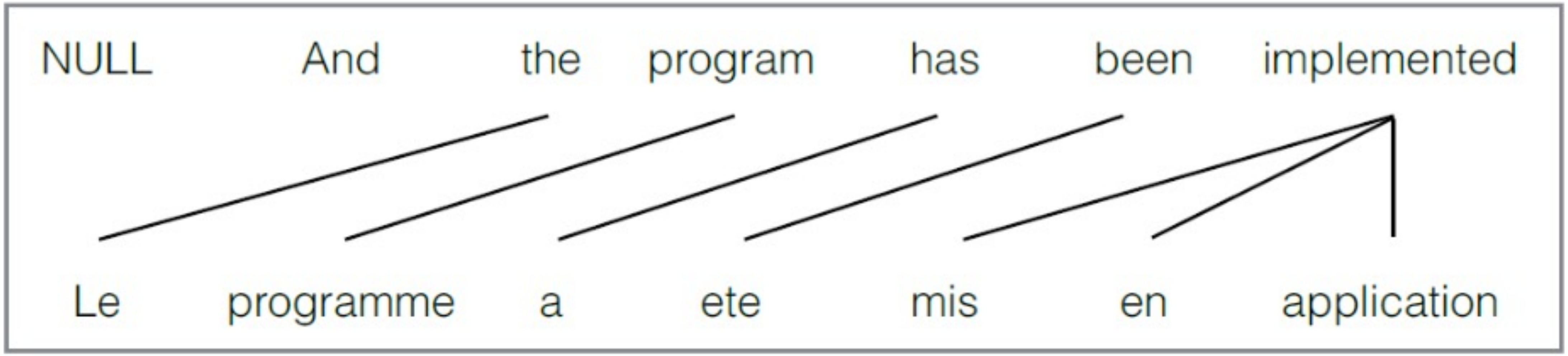
- (including a special NULL token)

But k does not need to equal j

Assumption 2: f_k is independent of all the other words in e, given the word e_j

We formalize the "word-to-word" translation idea using **alignments**

Word Alignment Vectors



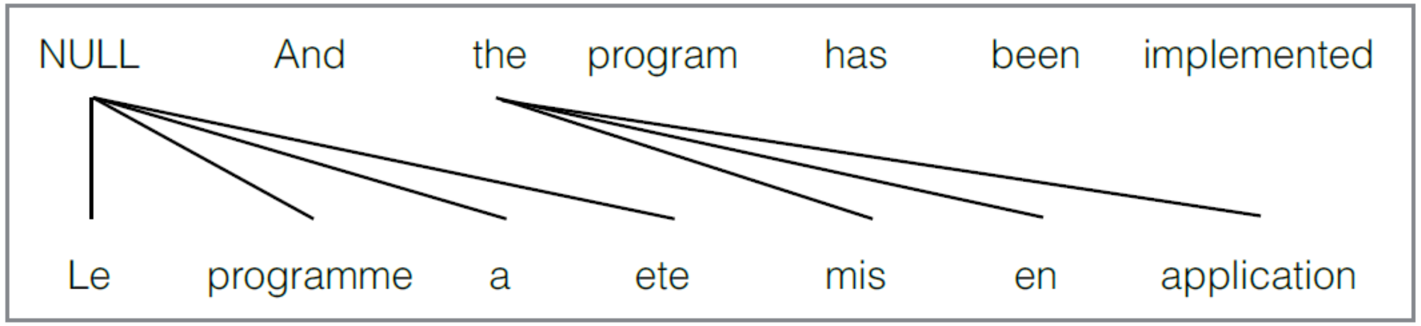
- Alignment vector a = <2,3,4,5,6,6,6>
  - Length of a = length of sentence f
  - $a_i = j$ if French position I is aligned to English position j

Word Alignment Vectors

J 'ai acheté du pain	I bought some bread
J 'ai acheté du beurre	I bought some butter
Nous devons manger le pain blanc	We must eat the white bread

<1,0,2,3,4>

<1,2,3,4,6,5>



Alignment vector a = <0,0,0,0,2,2,2>

Word-aligned data

J 'ai acheté du pain	I bought some bread
J 'ai acheté du beurre	I bought some butter
Nous devons manger le pain blanc	We must eat the white bread

A word-aligned parallel corpus would specify the alignment **a** for the English-French sentences.

J 'ai acheté du pain	$< 1, 0, 2, 3, 4 >$	I bought some bread
J 'ai acheté du beurre	$< 1, 0, 2, 3, 4 >$	I bought some butter
Nous devons manger le pain blanc	$< 1, 2, 3, 4, 6, 5 >$	We must eat the white bread

MLE for word-aligned data

$$n_{e,f}(\boldsymbol{a}) = \sum_{k:f_k=f} \underbrace{[\![e_{a_k} = e]\!]}_{\text{“Condition function”}}$$

$$\hat{\tau}_{e,f} = \frac{n_{e,f}(\boldsymbol{a})}{n_{e,\circ}(\boldsymbol{a})}$$

$$n_{e,\circ} = \sum_f n_{e,f}(\boldsymbol{a})$$

“Go through the corpus counting how often **e** aligns with **f** and then take the maximum likelihood estimate to get the corresponding probability.”

What if we introduced some uncertainty to the alignments?

It would be helpful to let the annotators of our word-aligned corpus to encode their confidence

"e_2 probably aligned with f_2 but it might rather be e_1…"

"I'll assign a probability of 0.9 to the first guess and 0.1 to my second"

How should we incorporate this information when the alignments are "more or less confident" rather than "yes or no"?

Handling probabilistic alignments

1. We could imply ignore any alignment with confidence less than some threshold, say 0.8
   - If we had a surplus of word-aligned data this might be reasonable…..
   - But that’s never going to happen!

2. Assign ”partial counts” based on the probabilities
   - Given a 10-word sentence, even if an annotator were maximally uncertain between two choices, then they both get 0.5
   - Consider how much signal this already carries!
   - In Model 1, since all alignments are equally probably then they would originally have had probability of 0.1 (and shared 0.8 of the mass with definitely wrong choices)

Getting Taus from partial counts

- Let’s say I pay some (very bad) annotators to word-align my three sentence corpus
  - Sentence one: they align ‘achete’ with equal probability to both ‘bought’ and ‘bread’
  - Sentence two: ‘pain' is also aligned with equal probability to ‘bought' and `bread’
  - Sentence three: same confusion but now with ‘manger/pain' and ‘eat/bread'.

English	French	Sentences		
		1	2	3
bought	pain	1/2		
bought	acheté	1/2	1/2	
bread	pain	1/2		1/2
bread	acheté	1/2		
bought	beurre		1/2	
butter	acheté		1/2	
butter	beurre		1/2	
eat	pain			1/2
eat	manger			1/2
bread	manger			1/2

Getting Taus from partial counts

Before looking at any word-aligned data, then best we can do is set Taus to all be equal. But now….

$$\tau^2_{\text{`bought',`pain'}} = \frac{n_{\text{`bought',`pain'}}}{n_{\text{`bought'},\circ}} = \frac{1/2}{1/2+1/2+1/2+1/2} = 1/4.$$

Now our Tau parameters prefer the 'bought/achete' translation over other translation of 'bought' even though the annotators did not specify this!

(But 'butter' and 'eat' have not been clarified)

English	French	Sentences 1	2	3
bought	pain	1/2		
bought	acheté	1/2	1/2	
bread	pain	1/2		1/2
bread	acheté	1/2		
bought	beurre		1/2	
butter	acheté		1/2	
butter	beurre		1/2	
eat	pain			1/2
eat	manger			1/2
bread	manger			1/2

Expectation-Maximization

- Let’s take this idea to an extreme:
  - Pretend that we have very bad initial annotators (in reality we don’t even has *any*) who give each **f-e** alignment equal probability
  - Go through the corpus to sum up the partial counts
  - Set the Tau parameters to the maximum likelihood estimate from these counts (as if they were “real”)

Second big idea

- These Tau estimates should be much better than our original assumption of equal probabilities
- So just repeat the process, but using our new probabilities

Expectation-Maximization

- We just need an equation for computing fractional counts from probabilistic information

$$n_{e_j,f_k} += \frac{\tau_{e_j,f_k}}{p_k}$$

Estimate of probability that fk is the translation of ej

Total probability that fk translates *any* word in the current sentence

The expected number of times our generative model aligned fk with ej given our data:

$$E[n_{e,f} \mid \boldsymbol{e}, \boldsymbol{f}]$$

$$p_k = \sum_j \tau_{e_j,f_k}$$

Getting Taus from partial counts (round two)

English	French	Sentences 1	2	3	$\tau^2_{e,f}$
bought	pain	1/2			1/4
bought	acheté	1/2	1/2		$\boxed{1/2}$
bread	pain	1/2		1/2	1/2
bread	acheté	1/2			$\boxed{1/4}$
bought	beurre		1/2		1/4
butter	acheté		1/2		1/2
butter	beurre		1/2		1/2
eat	pain			1/2	1/2
eat	manger			1/2	1/2
bread	manger			1/2	1/4

- The previous example in effect walked through one iteration of EM, culminating in a new set of Taus.
- Now we'd need to go through each sentence to tally up the partial counts $n_{e,f}$ for each word pair
- To take a single example, consider $n_{bought,achete}$ for the first sentence

$$P(\text{acheté}) = \sum_j \tau_{j,\text{acheté}} = \frac{3}{4}$$

Getting Taus from partial counts (round two)

English	French	Sentences 1	2	3	$\tau^2_{e,f}$
bought	pain	1/2			1/4
bought	acheté	1/2	1/2		$\boxed{1/2}$
bread	pain	1/2		1/2	1/2
bread	acheté	1/2			1/4
bought	beurre		1/2		1/4
butter	acheté		1/2		1/2
butter	beurre		1/2		1/2
eat	pain			1/2	1/2
eat	manger			1/2	1/2
bread	manger			1/2	1/4

- The previous example in effect walked through one iteration of EM, culminating in a new set of Taus.
- Now we'd need to go through each sentence to tally up the partial counts $n_{e,f}$ for each word pair
- To take a single example, consider $n_{bought,achete}$ for the first sentence

$$n_{\text{'bought','achete'}} = \frac{\tau_{\text{'bought','acheté'}}}{p_{\text{'acheté'}}}$$

$$\begin{aligned} p_{\text{'acheté'}} &= \sum_j \tau_{\text{'j','acheté'}} \\ &= \tau_{\text{'bread','acheté'}} + \tau_{\text{'bought','acheté'}} \\ &= 1/4 + 1/2 \\ &= 3/4 \end{aligned}$$

Getting Taus from partial counts (round two)

$$\begin{aligned}
p_{\text{`acheté'}} &= \sum_j \tau_{\text{`}j\text{',`acheté'}} \\
&= \tau_{\text{`bread',`acheté'}} + \tau_{\text{`bought',`acheté'}} \\
&= 1/4 + 1/2 \\
&= 3/4
\end{aligned}$$

$$\begin{aligned}
n_{\text{`bought',`achete'}} &= \frac{\tau_{\text{`bought',`acheté'}}}{p_{\text{`acheté'}}} \\
&= \frac{1/2}{3/4} \\
&= 2/3
\end{aligned}$$

English	French	Sentences		
		1	2	3
bought	pain	1/3		
bought	acheté	2/3	1/2	
bread	pain	2/3		1/2
bread	acheté	1/3		
bought	beurre		1/3	
butter	acheté		1/2	
butter	beurre		2/3	
eat	pain			1/2
eat	manger			2/3
bread	manger			1/3

Expectation-Maximization (full algorithm)

1. Pick positive initial values for $\tau_{e,f}$ for all English words e and all French words f (equal is best).

2. For $i = 1, 2, \ldots$ until convergence (see below) do:

   (a) *E-step:*
   Set $n_{e,f} = 0$ for all English words e and French words f.
   For each E/F sentence pair and for each French word position $k = 1, \ldots, m$ do:

      i. Set $p_k = \sum_{j=0}^{l} \tau_{e_j, f_k}$, where j are the positions of the English words in the same sentence pair as f_k.

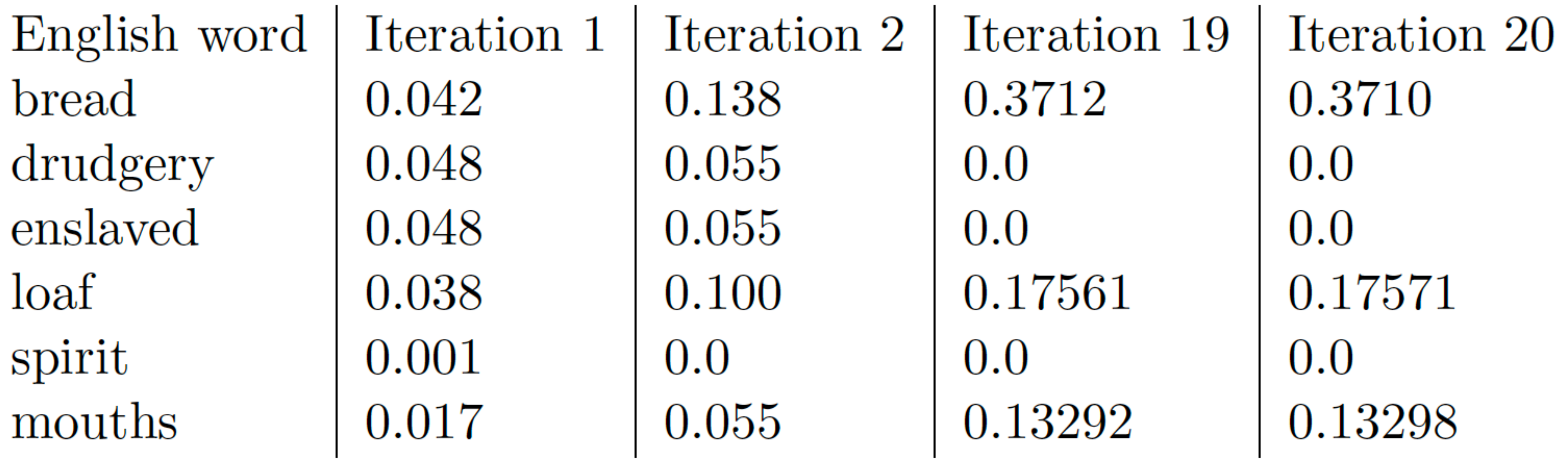
      ii. For each $0 \leq j \leq l$, increment $n_{e_j, f_k} \mathrel{+}= \tau_{e_j, f_k} / p_k$

   ($n_{e,f}$ now contains the expected number of times e aligns with f)

   (b) *M-step:*
   Set $\tau_{e,f} = n_{e,f} / n_{e,\circ}$, where $n_{e,\circ} = \sum_f n_{e,f}$.

EM Convergence

- We simply set a threshold, say 1%, and when the likelihood changes less than that threshold then we stop



English word	Iteration 1	Iteration 2	Iteration 19	Iteration 20
bread	0.042	0.138	0.3712	0.3710
drudgery	0.048	0.055	0.0	0.0
enslaved	0.048	0.055	0.0	0.0
loaf	0.038	0.100	0.17561	0.17571
spirit	0.001	0.0	0.0	0.0
mouths	0.017	0.055	0.13292	0.13298

Figure 2.6: Probabilities for English words translating as '*pain*'

The Three Basic HMM Problems

- *Problem 1 (Evaluation):* Given the observation sequence $O = o_1, \ldots, o_T$ and an HMM model λ, how do we compute the probability of O given the model?

- *Problem 2 (Decoding):* Given the observation sequence O and an HMM model λ, how do we find the state sequence that best explains the observations?

- *Problem 3 (Learning):* How do we adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P(O|\lambda)$?

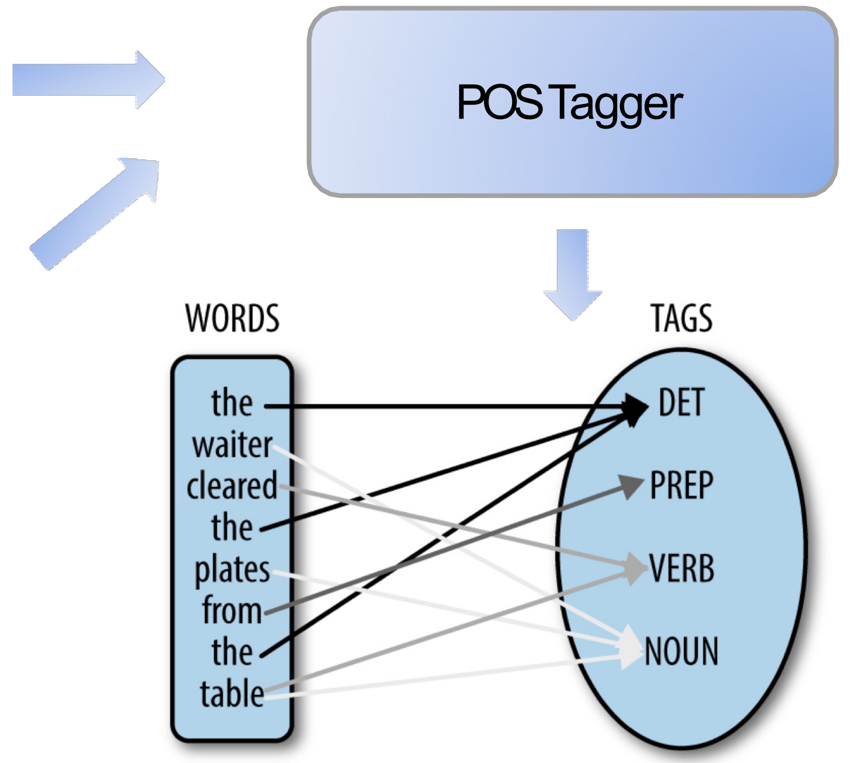
How to get our estimates?

Supervised Learning

Assume linguistic annotators have labeled training examples

Tag set:
DT, JJ, NN, VBD…

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.



How to get our estimates?

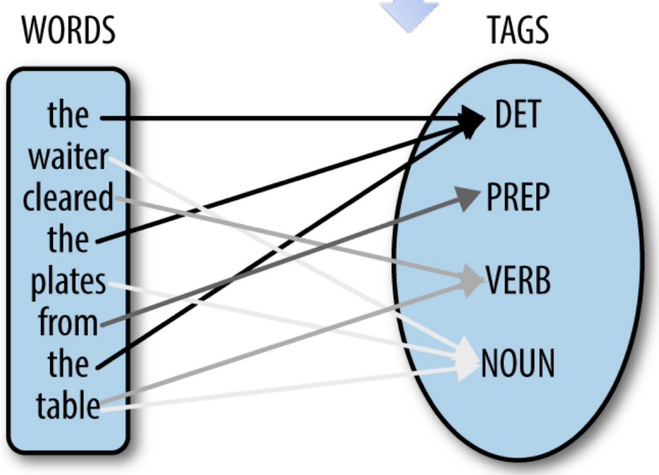
Unsupervised Learning

Assume we only have an unannotated corpus

Tag set:
DT, JJ, NN, VBD…

POS Tagger

The grand jury commented on a number of other topics.



Problem 3: Learning

- Up to now we've assumed that we *know* the underlying model
- Often these parameters are estimated on annotated training data, but:
  - **Annotation is often difficult and/or expensive**
  - **Training data is different from the current data**
- We want to maximize the parameters with respect to the current data

  i.e., we're looking for a model λ', such that $\lambda' = \underset{\lambda}{\arg\max}\ P(O \mid \lambda)$

Forward-Backward (Baum-Welch) algorithm

- We won't be implementing Forward-Backward, so I'll skip the derivation
- It's essentially a version of EM applied to HMMs
  - you start with uniform λ (transition and emission probabilities)
  - Estimate $P(O\,|\,\lambda)$, filling out the trellis as we go
  - Reestimate λ using the trellis, yielding a new estimate λ^'
  - …
  - repeat
- BUT, we can take a look at some of the potential results: **POS induction**

POS Induction

- Initialize our HMM with our Sigma’s and Tau’s approximately equal
- Set the number of word classes to match our desired output (If Penn Treebank uses 45 tags, then we have 45 tags. If Universal tag set uses 12 then we’ll have 12).
- Run Forward-Backward to maximize the likelihood of the development data

- Since we have a Tau emission probability for each word in our vocabulary being generated by each POS tag. We can group them into clusters
  - Which words have high Tau probabilities of being generated by the same POS tag?

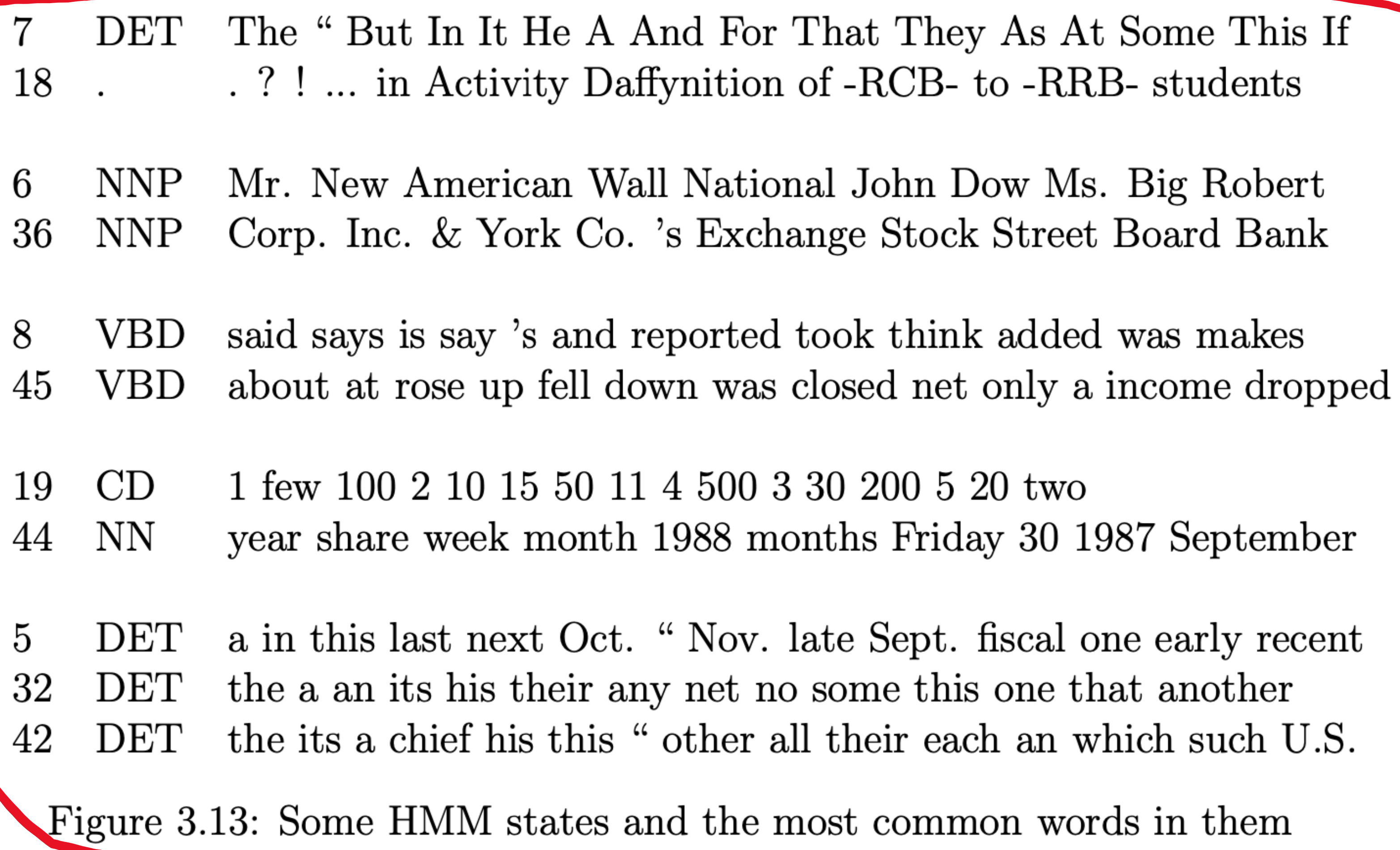
POS Induction

- Following table shows 11 of the 45 resulting states using forward-backward

- The first column is the state number —this is arbitrary.
- Second column shows the actual POS tag that is most common for the words in this numbered class

POS Induction

what's going on!??

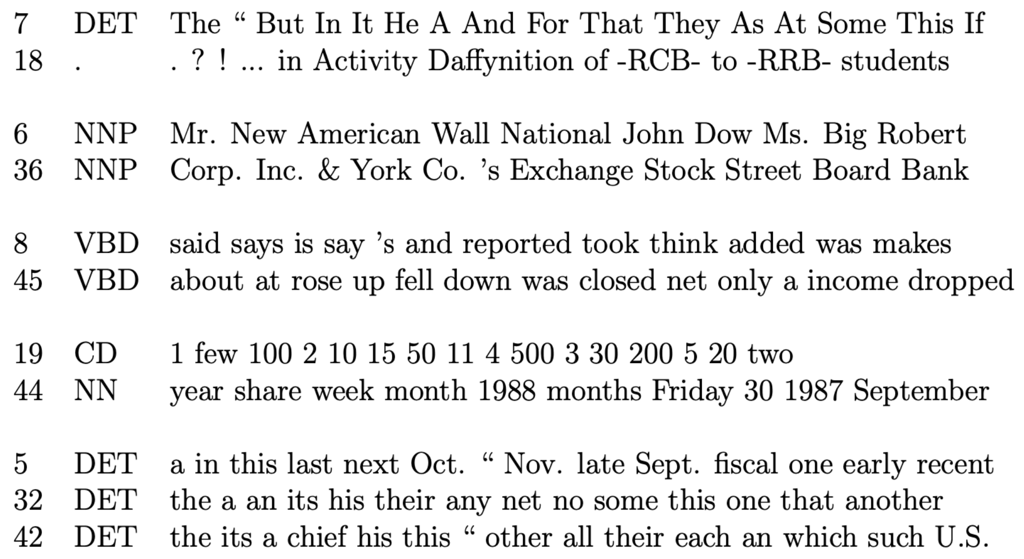


7	DET	The “ But In It He A And For That They As At Some This If
18	.	. ? ! ... in Activity Daffynition of -RCB- to -RRB- students
6	NNP	Mr. New American Wall National John Dow Ms. Big Robert
36	NNP	Corp. Inc. & York Co. ’s Exchange Stock Street Board Bank
8	VBD	said says is say ’s and reported took think added was makes
45	VBD	about at rose up fell down was closed net only a income dropped
19	CD	1 few 100 2 10 15 50 11 4 500 3 30 200 5 20 two
44	NN	year share week month 1988 months Friday 30 1987 September
5	DET	a in this last next Oct. “ Nov. late Sept. fiscal one early recent
32	DET	the a an its his their any net no some this one that another
42	DET	the its a chief his this “ other all their each an which such U.S.

Figure 3.13: Some HMM states and the most common words in them

POS Induction

- Class 7 is made up of words and punctuation that begin sentences
  - Words “w” for which $\sigma_{\rhd,w}$ (ie π_w) is large
  - It’s assigned “DET” just because “The” happens to be the most common word
- Class 18 ends sentences
  - The Penn Treebank tag for this class is “.” – so that’s good!

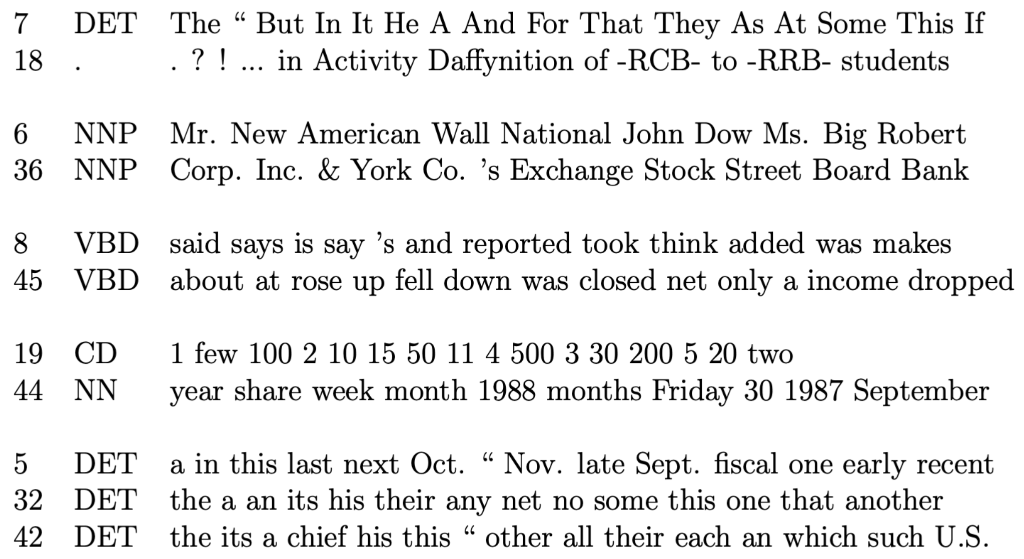


7	DET	The “ But In It He A And For That They As At Some This If
18	.	. ? ! ... in Activity Daffynition of -RCB- to -RRB- students
6	NNP	Mr. New American Wall National John Dow Ms. Big Robert
36	NNP	Corp. Inc. & York Co. ’s Exchange Stock Street Board Bank
8	VBD	said says is say ’s and reported took think added was makes
45	VBD	about at rose up fell down was closed net only a income dropped
19	CD	1 few 100 2 10 15 50 11 4 500 3 30 200 5 20 two
44	NN	year share week month 1988 months Friday 30 1987 September
5	DET	a in this last next Oct. “ Nov. late Sept. fiscal one early recent
32	DET	the a an its his their any net no some this one that another
42	DET	the its a chief his this “ other all their each an which such U.S.

Figure 3.13: Some HMM states and the most common words in them

POS Induction

- Classes 6 and 36 are both assigned to proper nouns (NNP)
  - but 6 is made up of words that typically start names, while 36 end names
- We have a fixed number (45 here) of POS tag classes
  - If we erroneously split NNPs into two then we'll have to erroneously combine some other class!



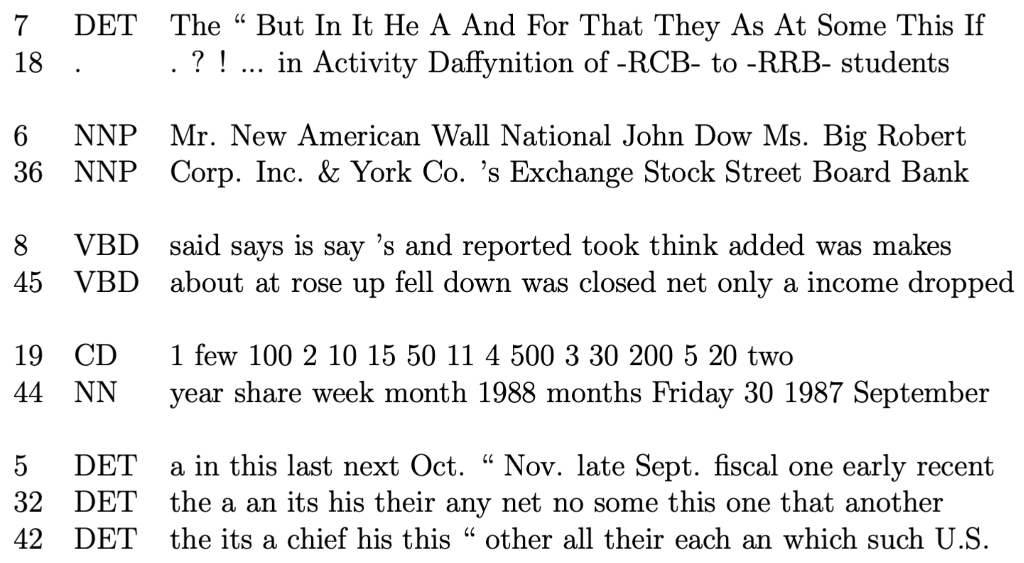
7	DET	The “ But In It He A And For That They As At Some This If
18	.	. ? ! ... in Activity Daffynition of -RCB- to -RRB- students
6	NNP	Mr. New American Wall National John Dow Ms. Big Robert
36	NNP	Corp. Inc. & York Co. 's Exchange Stock Street Board Bank
8	VBD	said says is say 's and reported took think added was makes
45	VBD	about at rose up fell down was closed net only a income dropped
19	CD	1 few 100 2 10 15 50 11 4 500 3 30 200 5 20 two
44	NN	year share week month 1988 months Friday 30 1987 September
5	DET	a in this last next Oct. “ Nov. late Sept. fiscal one early recent
32	DET	the a an its his their any net no some this one that another
42	DET	the its a chief his this “ other all their each an which such U.S.

Figure 3.13: Some HMM states and the most common words in them

POS Induction

- What’s going on with 5, 32, 42?

Simply maximizing the likelihood of the data is not necessarily make good things happen!



7	DET	The “ But In It He A And For That They As At Some This If
18	.	. ? ! ... in Activity Daffynition of -RCB- to -RRB- students
6	NNP	Mr. New American Wall National John Dow Ms. Big Robert
36	NNP	Corp. Inc. & York Co. ’s Exchange Stock Street Board Bank
8	VBD	said says is say ’s and reported took think added was makes
45	VBD	about at rose up fell down was closed net only a income dropped
19	CD	1 few 100 2 10 15 50 11 4 500 3 30 200 5 20 two
44	NN	year share week month 1988 months Friday 30 1987 September
5	DET	a in this last next Oct. “ Nov. late Sept. fiscal one early recent
32	DET	the a an its his their any net no some this one that another
42	DET	the its a chief his this “ other all their each an which such U.S.

Figure 3.13: Some HMM states and the most common words in them

Don’t make life harder than it needs to be

- Unsupervised part-of-speech tagging has been studied for 25 years (Merialdo 1994), but the best results are quite a bit worse than can be obtained with as little as two hours of human annotation (Garrette & Baldridge 2013).

Today

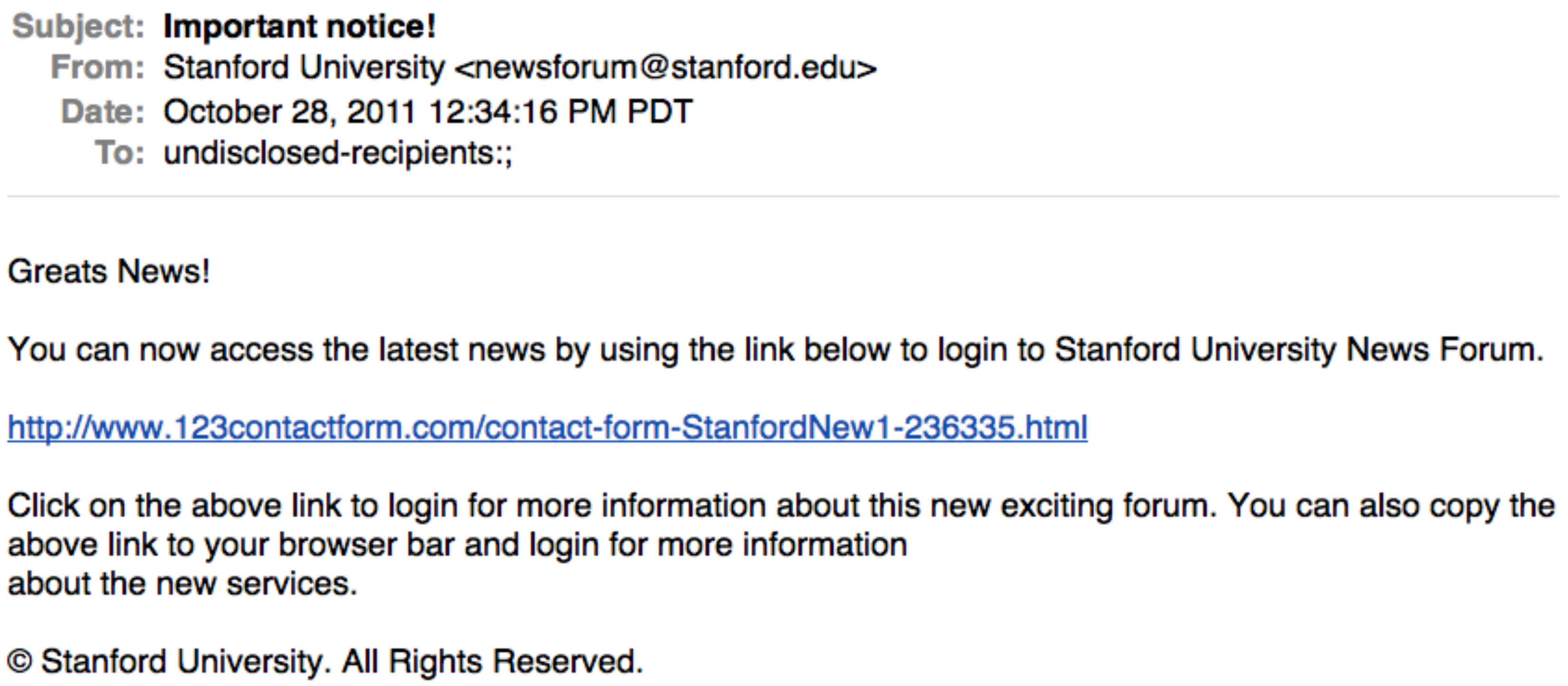
1. ~~Expectation Maximization~~

2. ~~POS Induction~~

3. **Text Classification**

4. Naïve Bayes Model

Is this spam?



Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

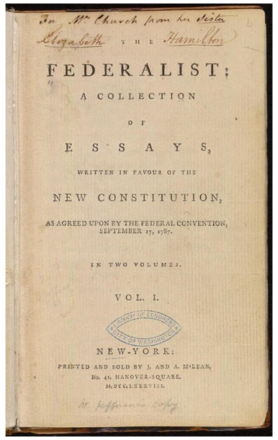
You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

© Stanford University. All Rights Reserved.

Who wrote the Federalist papers?


For Mrs Church from her Sister
Elizabeth THE Hamilton

FEDERALIST:

A COLLECTION

OF

ESSAYS,

WRITTEN IN FAVOUR OF THE

NEW CONSTITUTION,

AS AGREED UPON BY THE FEDERAL CONVENTION, SEPTEMBER 17, 1787.

IN TWO VOLUMES.

VOL. I.

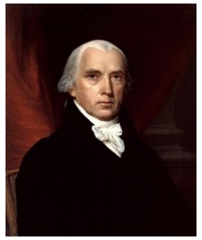
LIBRARY OF CONGRESS CITY OF WASHINGTON

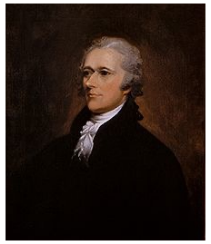
NEW-YORK:

PRINTED AND SOLD BY J. AND A. M'LEAN,
No. 41, HANOVER-SQUARE.
M,DCC,LXXXVIII.

Mr Jefferson's copy

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace


James Madison


Alexander Hamilton

Positive or negative movie review?

- 👎 unbelievably disappointing
- 👍 Full of zany characters and richly applied satire, and some great plot twists
- 👍 this is the greatest screwball comedy ever filmed
- 👎 It was pathetic. The worst part about it was the boxing scenes.

What is the subject of this article?

MEDLINE Article

Available online at www.sciencedirect.com

Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

Susanne Gahl, Lise Menn, Gail Ramsberger, Daniel S. Jurafsky, Elizabeth Elder, Molly Rewega, and L. Holland Audrey

[illegible]

?

MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- …

Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

- *Output*: a predicted class $c \in C$

Classification Methods: Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR (“dollars” AND “have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

Classification Methods: Supervised Machine Learning

- *Input:*
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
- *Output:*
  - a learned classifier $\gamma: d \rightarrow c$

Classification Methods: Supervised Machine Learning

- Any kind of classifier
    - Naïve Bayes
    - Logistic regression
    - Support-vector machines
    - k-Nearest Neighbors

    - …

Naïve Bayes Intuition

- Simple (“naïve”) classification method based on Bayes rule
- Relies on very simple representation of document
    - Bag of words

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

fairy always love to it
it whimsical it I
and seen are anyone
friend happy dialogue
adventure recommend
who sweet of satirical it
it I but to movie romantic I
several yet
again it the humor
the seen would
to scenes I the manages
fun I and the times and
about while
whenever have
conventions
with

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

The Bag of Words Representation

$\gamma($

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

$)=c$

👍
👎

Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Naïve Bayes Classifier (i)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

Naïve Bayes Classifier (ii)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Document d represented as features x1..xn

Naïve Bayes Classifier (iii)

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

O(|*X*|^n•|*C*|) parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Multinomial Naïve Bayes Independence Assumptions

$$P(x_1, x_2, \ldots, x_n \mid c)$$

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class c.

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

Multinomial Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c_j) \prod_{x \in X} P(x \mid c)$$

Applying Multinomial Naïve Bayes Classifiers to Text Classification

$\text{positions} \leftarrow$ all word positions in test document

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

Probabilistic Classification

- We can pick the argmax when we need to make a discrete classification decision

- But really Naïve Bayes is a probabilistic classifier:
  - Provides a probability distribution over all possible classes
  - Often useful to avoid making discrete decisions early on when combining systems downstream

Learning the Naïve Bayes Model

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Parameter Estimation

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use frequency of *w* in mega-document

Problem with raw Maximum Likelihood

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \text{argmax}_c\, \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c)\right) + 1}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Multinomial Naïve Bayes: Learning

- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $= c_j$

    $$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

    $$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

Underflow Prevention: log space

- Multiplying lots of probabilities can result in floating-point underflow.
- Since log(*xy*) = log(*x*) + log(*y*)
  - Better to sum logs of probabilities instead of multiplying probabilities.
- Class with highest un-normalized log probability score is still most probable.

$$c_{NB} = \operatorname*{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

- Model is now just max of sum of weights

Naïve Bayes and its Relationship to Language Modeling

Naïve Bayes and Language Modeling

- Naïve bayes classifiers can use any sort of feature
  - URL, email address, dictionaries, network features
- But if, as in the previous slides
  - We use **only** word features
  - we use **all** of the words in the text (not a subset)
- Then
  - Naïve bayes has an important similarity to language modeling.

Each class = a unigram language model

- Assigning each word: P(word | c)
- Assigning each sentence: $P(s|c)=\Pi\ P(word|c)$

Class	*pos*
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film
…	

I	love	this	fun	film
0.1	0.1	.05	0.01	0.1

P(s | pos) = 0.0000005

Naïve Bayes as a Language Model

- Which class assigns the higher probability to s?

Model pos	
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg	
0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

I	love	this	fun	film
0.1	0.1	0.01	0.05	0.1
0.2	0.001	0.01	0.005	0.1

$P(s|pos) > P(s|neg)$

Naïve Bayes for Language ID

- Rather than run a bunch of LMs over all our data (words)….
  - … how about just look at character n-grams
  - (which is an approximation of language-specific phonotactics)
  - E.g. do we get string like “zw”, “nya”, “thei”, etc.

- Could scrape data from Wikipedia and train a NB classifier on each language as a “class”

Naïve Bayes in Spam Filtering

- SpamAssassin Features:
  - Mentions Generic Viagra
  - Online Pharmacy
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
  - Phrase: impress ... girl
  - From: starts with many numbers
  - Subject is all capitals
  - HTML has a low ratio of text to image area
  - One hundred percent guaranteed
  - Claims you can be removed from the list
  - 'Prestigious Non-Accredited Universities'
  - http://spamassassin.apache.org/tests_3_3_x.html

Naïve Bayes is not so Naïve

- Very Fast, low storage requirements
- Robust to Irrelevant Features

    Irrelevant Features cancel each other without affecting results

- Very good in domains with many equally important features

    Decision Trees suffer from *fragmentation* in such cases – especially if little data

- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
    - **But we will see other classifiers that give better accuracy**

What gets to be a feature?

As alluded to, we could include far more (or far less) than just the unigram counts of words as classifier features

Binarized (Boolean feature) Multinomial Naïve Bayes

- Intuition:
  - For sentiment (and probably for other text classification domains)
  - Word occurrence may matter more than word frequency
    - The occurrence of the word *fantastic* tells us a lot
    - The fact that it occurs 5 times may not tell us much more.
  - Boolean Multinomial Naïve Bayes
    - Clips all the word counts in each document at 1

How about counting other features instead?

- Binary seems to work better than full word counts (at least for some tasks like sentiment of movie reviews)

- Other possibility: log(freq(*w*))

What gets to be a feature?

- STOP words
- Distribution over UNKs
- Features for Spam Detection:
  - Subject in all caps
  - Unbalanced HTML tags