CS447: Natural Language Processing
http://courses.engr.illinois.edu/cs447

Lecture 22: Statistical Machine Translation

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center

Projects and Literature Reviews

First report due Nov 26
(PDF written in LaTeX; no length restrictions; submission through Compass)

Purpose of this first report:
Check-in to make sure that you're on track (or, if not, that we can spot problems)

Rubrics for the *final* reports (due on Reading Day):
https://courses.engr.illinois.edu/CS447/LiteratureReviewRubric.pdf
https://courses.engr.illinois.edu/CS447/FinalProjectRubric.pdf

Projects and Literature Reviews

Guidelines for first Project Report:
What is your project about?
What are the relevant papers you are building on?
What data are you using?
What evaluation metric will you be using?
What models will you implement/evaluate?
What is your to-do list?

Guidelines for first Literature Review Report:
What is your literature review about?
(What task or what kind of models?
Do you have any specific questions or focus?)
What are the papers you will review?
(If you already have it, give a brief summary of each of them)
What's your to-do list?

Statistical Machine Translation

Statistical Machine Translation

We want the best (most likely) [English] translation for the [Chinese] input:

$$\text{argmax}_{\text{English}}\ P(\text{ English} \mid \text{Chinese })$$

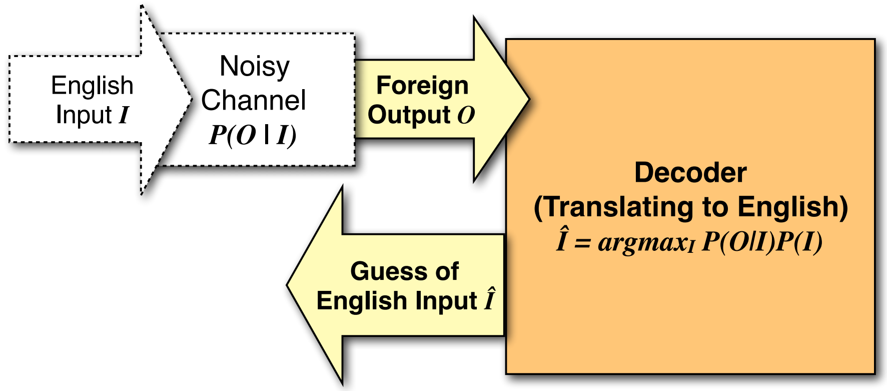
We can either model this probability directly,
or we can apply Bayes Rule.
Using Bayes Rule leads to the “noisy channel” model.

As with sequence labeling, Bayes Rule simplifies the modeling task, so this was the first approach for statistical MT.

The noisy channel model

Translating from Chinese to English:

$$\text{argmax}_{Eng} P(Eng|Chin) \;=\; \text{argmax}_{Eng} \underbrace{P(Chin|Eng)}_{\text{Translation Model}} \times \underbrace{P(Eng)}_{\text{LanguageModel}}$$


English Input *I*
Noisy Channel *P(O | I)*
Foreign Output *O*
Decoder (Translating to English) $\hat{I} = \text{argmax}_I\, P(O|I)P(I)$
Guess of English Input $\hat{I}$

The noisy channel model

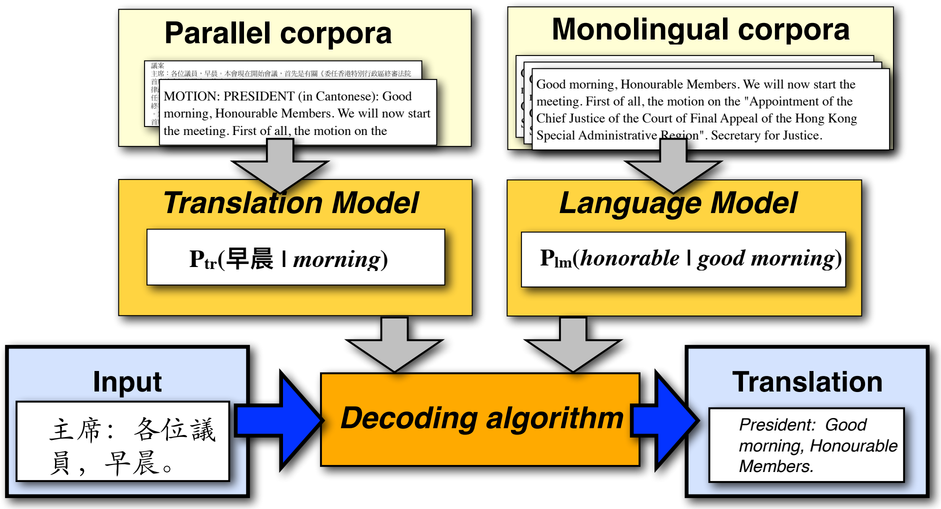
This is really just an application of **Bayes’ rule**:

$$\begin{aligned}\hat{E} &= \arg\max_E P(E|F)\\ &= \arg\max_E \frac{P(F|E)\times P(E)}{P(F)}\\ &= \arg\max_E \underbrace{P(F|E)}_{\text{Translation Model}} \times \underbrace{P(E)}_{\text{Language Model}}\end{aligned}$$

The **translation model** $P(F \mid E)$ is intended to capture the **faithfulness of the translation.**
It needs to be trained on a **parallel corpus**

The **language model** $P(E)$ is intended to capture the **fluency of the translation.**
It can be trained on a (very large) **monolingual corpus**

Statistical MT with the noisy channel model



議案
主席：各位議員，早晨。本會現在開始會議，首先是有關《委任香港特別行政區終審法院

MOTION: PRESIDENT (in Cantonese): Good morning, Honourable Members. We will now start the meeting. First of all, the motion on the

Good morning, Honourable Members. We will now start the meeting. First of all, the motion on the "Appointment of the Chief Justice of the Court of Final Appeal of the Hong Kong Special Administrative Region". Secretary for Justice.

主席：各位議員，早晨。

President: Good morning, Honourable Members.

n-gram language models for MT

With training on data from the web and clever parallel processing (MapReduce/Bloom filters), *n* can be quite large

- Google (2007) uses 5-grams to 7-grams,
- This results in huge models, but the effect on translation quality levels off quickly:

Size of models

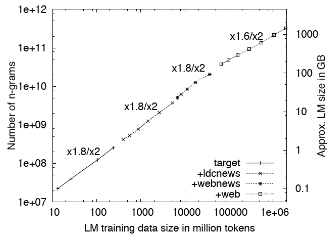


Figure 3: Number of n-grams (sum of unigrams to 5-grams) for varying amounts of training data.

Effect on translation quality

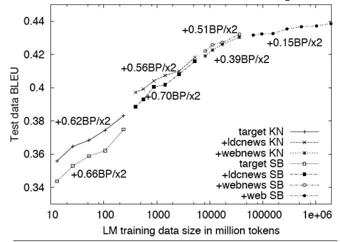


Figure 5: BLEU scores for varying amounts of data using Kneser-Ney (KN) and Stupid Backoff (SB).

Translation probability $P(fp_i \mid ep_i)$

Phrase translation probabilities can be obtained from a **phrase table:**

EP	FP	count
green witch	grüne Hexe	…
at home	zuhause	10534
at home	daheim	9890
is	ist	598012
this week	diese Woche	….

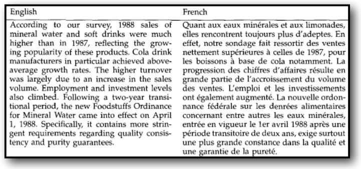
This requires **phrase alignment** on a **parallel corpus**.

Getting translation probabilities

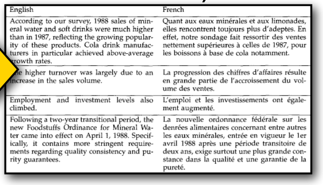
A **parallel corpus** consists of the same text in two (or more) languages.

Examples: Parliamentary debates: Canadian Hansards; Hong Kong Hansards, Europarl; Movie subtitles (OpenSubtitles)

In order to train translation models, we need to **align the sentences** (Church & Gale '93)



English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates. The higher turnover was largely due to an increase in the sales volume. Employment and investment levels also climbed. Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment. La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes. L'emploi et les investissements ont également augmenté. La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.



English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment.
The higher turnover was largely due to an increase in the sales volume.	La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes.
Employment and investment levels also climbed.	L'emploi et les investissements ont également augmenté.
Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

We can learn **word** and **phrase alignments** from these aligned sentences

IBM models

First statistical MT models, based on noisy channel:

Translate from source $\mathbf{f}$ to target $\mathbf{e}$
via a **translation model** $P(\mathbf{f} \mid \mathbf{e})$ and a **language model** $P(\mathbf{e})$
The translation model goes **from target $\mathbf{e}$ to source $\mathbf{f}$**
via **word alignments $\mathbf{a}$**: $P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$

Original purpose: Word-based translation models
Today: Can be used to obtain word alignments, which are then used to obtain phrase alignments for phrase-based translation models

Sequence of 5 translation models

Model 1 is too simple to be used by itself, but can be trained very easily on parallel data.

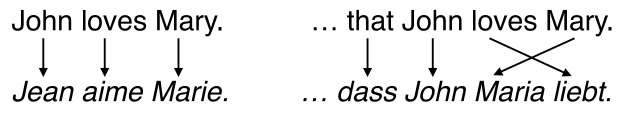
IBM translation models: assumptions

The model "generates" the 'foreign' source sentence $\mathbf{f}$
conditioned on the 'English' target sentence $\mathbf{e}$
by the following stochastic process:

1. Generate the **length** of the source $\mathbf{f}$ with probability p = ...
2. Generate the **alignment** of the source $\mathbf{f}$ to the target $\mathbf{e}$ with probability p = ...
3. Generate the **words** of the source $\mathbf{f}$ with probability p = ...

Word alignments in the IBM models

Word alignment



	Jean	aime	Marie
John	■		
loves		■	
Mary			■

	dass	John	Maria	liebt
that	■			
John		■		
loves				■
Mary			■	

Word alignment

	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap			■	■	■				
the						■	■		
green									■
witch								■	

Word alignment

	Marie	a	traversé	le	lac	à	la	nage
Mary	■							
swam						■	■	■
across		■	■					
the				■				
lake					■			

Word alignment

Source

Target

	Marie	a	traversé	le	lac	à	la	nage
Mary	■							
swam						■	■	■
across		■	■					
the				■				
lake					■			

One target word can be aligned to **many source words**.

Word alignment

Source

Target

	Marie	a	traversé	le	lac	à	la	nage
Mary	■							
swam						■	■	■
across		■	■					
the				■				
lake					■			

One target word can be aligned to **many source words**.
But **each source word** can only be aligned to **one target word.**
This allows us to model ***P*****(source | target)**

Word alignment

Source

Target

	Marie	a	traversé	le	lac	à	la	nage
Mary	■							
swam								■
across		■	■					
the				■				
lake					■			

Some source words may **not align** to ***any* target words**.

Word alignment

Source

Target

	Marie	a	traversé	le	lac	à	la	nage
NULL						■	■	
Mary	■							
swam								■
across		■	■					
the				■				
lake					■			

Some source words may **not align** to ***any* target words**.
To handle this we assume a NULL word in the target sentence.

Representing word alignments

		1	2	3	4	5	6	7	8
		Marie	a	traversé	le	lac	à	la	nage
0	NULL						■	■	
1	Mary	■							
2	swam								■
3	across		■	■					
4	the				■				
5	lake					■			

Position	1	2	3	4	5	6	7	8
Foreign	Marie	a	traversé	le	lac	à	la	nage
Alignment	1	3	3	4	5	0	0	2

Every source word **f[i]** is aligned to **one** target word **e[j]** (incl. NULL).
We represent alignments as a vector **a** (of the same length as the source) with **a[i] = j**

The IBM alignment models

The IBM models

Use the noisy channel (Bayes rule) to get the best (most likely) target translation $\mathbf{e}$ for source sentence $\mathbf{f}$:

$$\boxed{\arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) \quad = \quad \arg\max_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e})}$$

noisy channel

The translation model $P(\mathbf{f} \mid \mathbf{e})$ requires alignments $\mathbf{a}$

$$\boxed{P(\mathbf{f}|\mathbf{e}) \quad = \sum_{\mathbf{a}\in\mathcal{A}(\mathbf{e},\mathbf{f})} P(\mathbf{f},\mathbf{a}|\mathbf{e})}$$

marginalize (=sum) over all alignments a

Generate $\mathbf{f}$ and the alignment $\mathbf{a}$ with $P(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$:

$$P(\mathbf{f},\mathbf{a}|\mathbf{e}) \quad = \boxed{\underbrace{P(m|\mathbf{e})}_{\text{Length: }|\mathbf{f}|=m}} \prod_{j=1}^{m} \boxed{\underbrace{P(a_j|a_{1..j-1}, f_{1..j-1}, m, \mathbf{e})}_{\text{Word alignment } a_j}} \boxed{\underbrace{P(f_j|a_{1..j} f_{1..j-1}, \mathbf{e}, m)}_{\text{Translation } f_j}}$$

m = #words in fj

probability of alignment aj

probability of word fj

Model parameters

Length probability $P(m \mid n)$:

What's the probability of generating a source sentence of length m given a target sentence of length n?
Count in training data

Alignment probability: $P(\mathbf{a} \mid m, n)$:

Model 1 assumes all alignments have the same probability:
For each position $a_1 ... a_m$, pick one of the $n+1$ target positions uniformly at random

Translation probability: $P(f_j = lac \mid a_j = i, e_i = lake)$:

In Model 1, these are the only parameters we have to learn.

IBM model 1: Generative process

For each target sentence $\mathbf{e} = e_1..e_n$ of length n:

0	1	2	3	4	5
NULL	Mary	swam	across	the	lake

1. **Choose a length** m for the source sentence (e.g $m = 8$)

Position	1	2	3	4	5	6	7	8

2. **Choose an alignment** $\mathbf{a} = a_1...a_m$ for the source sentence

Each a_j corresponds to a word e_i in $\mathbf{e}$: $0 \le a_j \le n$

Position	1	2	3	4	5	6	7	8
Alignment	1	3	3	4	5	0	0	2

3. **Translate each target word** e_{a_j} into the source language

Position	1	2	3	4	5	6	7	8
Alignment	1	3	3	4	5	0	0	2
Translation	Marie	a	traversé	le	lac	à	la	nage

IBM model 1: details

The length probability is constant: $P(m \mid \mathrm{e}) = \varepsilon$

The alignment probability is uniform
(n = length of target string): $P(a_i \mid \mathrm{e}) = 1/(n+1)$

The translation probability depends only on e_{ai}
(the corresponding target word): $P(f_i \mid e_{ai})$

$$\begin{aligned} P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) &= \underbrace{P(m \mid \mathbf{e})}_{\text{Length: } |\mathbf{f}|=m} \prod_{j=1}^{m} \underbrace{P(a_j \mid a_{1..j-1}, f_{1..j-1}, m, \mathbf{e})}_{\text{Word alignment } a_j} \underbrace{P(f_j \mid a_{1..j} f_{1..j-1}, \mathbf{e}, m)}_{\text{Translation } f_j} \\ &= \epsilon \prod_{j=1}^{m} \frac{1}{n+1} P(f_j \mid e_{a_j}) \\ &= \frac{\epsilon}{(n+1)^m} \prod_{j=1}^{m} P(f_j \mid e_{a_j}) \end{aligned}$$

Finding the best alignment

How do we find the **best alignment** between $\mathbf{e}$ and $\mathbf{f}$?

$$\begin{aligned} \hat{\mathbf{a}} &= \arg\max_{\mathbf{a}} P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \\ &= \arg\max_{\mathbf{a}} \frac{\epsilon}{(n+1)^m} \prod_{j=1}^{m} P(f_j \mid e_{a_j}) \\ &= \arg\max_{\mathbf{a}} \prod_{j=1}^{m} P(f_j \mid e_{a_j}) \end{aligned}$$

$$\boxed{\hat{a}_j = \arg\max_{a_j} P(f_j \mid e_{a_j})}$$

Learning translation probabilities

The only parameters that need to be learned are the **translation probabilities** $P(\mathrm{f} \mid \mathrm{e})$

$$P(f_j = lac \mid e_i = lake)$$

If the training corpus had word alignments, we could simply count how often 'lake' is aligned to 'lac':

$$P(lac \mid lake) = \mathrm{count}(lac, lake) / \textstyle\sum_w \mathrm{count}(w, lake)$$

But we don't have gold word alignments.
So, instead of relative frequencies, we have to use *expected* relative frequencies:

$$P(lac \mid lake) = \langle \mathrm{count}(lac, lake) \rangle / \langle \textstyle\sum_w \mathrm{count}(w, lake) \rangle$$

Training Model 1 with EM

The only parameters that need to be learned are the **translation probabilities** $P(\mathrm{f} \mid \mathrm{e})$

We use the **EM algorithm** to estimate these parameters from a corpus with S sentence pairs $s = \langle \boldsymbol{f}^{(s)}, \boldsymbol{e}^{(s)} \rangle$ with alignments $\mathcal{A}(\mathbf{f}^{(s)}, \mathbf{e}^{(s)})$

- **Initialization:** guess $P(\mathrm{f} \mid \mathrm{e})$
- **Expectation step:** compute expected counts

$$\langle c(f,e) \rangle = \sum_{s \in S} \langle c(f,e \mid \mathbf{e}^{(s)}, \mathbf{f}^{(s)}) \rangle$$

- **Maximization step:** recompute probabilities $P(\mathrm{f} \mid \mathrm{e})$

$$\hat{P}(f \mid e) = \frac{\langle c(f,e) \rangle}{\sum_{f'} \langle c(f',e) \rangle}$$

Expectation-Maximization (EM)

1. Initialize a first model, M_0

2. Expectation (E) step:
   Go through training data to gather expected counts
   $\langle \mathrm{count}(lac, lake) \rangle$
3. Maximization (M) step:
   Use expected counts to compute a new model M_{i+1}
   $P_{i+1}(lac \mid lake) = \langle \mathrm{count}(lac, lake) \rangle / \langle \sum w\ \mathrm{count}(w, lake) \rangle$
4. Check for convergence:
   Compute log-likelihood of training data with M_{i+1}
   If the difference between new and old log-likelihood smaller than a threshold, stop. Else go to 2.

The E-step

Compute the expected count $\langle c(f,e \mid \mathbf{f}, \mathbf{e}) \rangle$:

$$\langle c(f,e \mid \mathbf{f}, \mathbf{e}) \rangle = \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f},\mathbf{e})} \boxed{P(\mathbf{a} \mid \mathbf{f}, \mathbf{e})} \cdot \underbrace{\boxed{c(f,e \mid \mathbf{a}, \mathbf{e}, \mathbf{f})}}_{\text{How often are } f,e \text{ aligned in } \mathbf{a}?}$$

$$\boxed{P(\mathbf{a} \mid \mathbf{f}, \mathbf{e})} = \frac{P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})}{P(\mathbf{f} \mid \mathbf{e})} = \frac{\boxed{P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})}}{\sum_{\mathbf{a}'} \boxed{P(\mathbf{a}', \mathbf{f} \mid \mathbf{e})}}$$

$$\boxed{P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})} = \prod_j \boxed{P(f_j \mid e_{a_j})}$$

$$\boxed{\langle c(f,e \mid \mathbf{f}, \mathbf{e}) \rangle} = \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{f},\mathbf{e})} \frac{\prod_j \boxed{P(f_j \mid e_{a_j})}}{\sum_{a'} \prod_j \boxed{P(f_j \mid e_{a'_j})}} \cdot c(f,e \mid \mathbf{a}, \mathbf{e}, \mathbf{f})$$

We need to know $P(f_j \mid e_{a_j})$, the probability that word f_j is aligned to word e_{aj} under the alignment a

Other translation models

Model 1 is a very simple (and not very good) translation model.

IBM models 2-5 are more complex. They take into account:

- **“fertility”:** the number of foreign words generated by each target word
- the **word order** and **string position** of the aligned words

Today’s key concepts

Why is machine translation hard?

Linguistic divergences: morphology, syntax, semantics

Different approaches to machine translation:

Vauquois triangle

Statistical MT: Noisy Channel, IBM Model 1 (more on this next time)