Translation Models

David Kauchak

CS159 – Fall 2014

Some slides adapted from

Philipp Koehn
School of Informatics
University of Edinburgh

Kevin Knight
USC/Information Sciences Institute
USC/Computer Science Department

Dan Klein
Computer Science Department
UC Berkeley

Admin

Assignment 5

- extended to Thursday at 2:45
  - Should have worked on it some already!
  - Extra credit (5%) if submitted by Tuesday (i.e. now)
  - No late days

Language translation

Noisy channel model

model $$p(e \mid f) \propto p(f \mid e)\, p(e)$$

translation model: how do English sentences get translated to foreign?

language model: what do English sentences look like?

Problems for Statistical MT

Preprocessing
- How do we get aligned bilingual text?
- Tokenization
- Segmentation (document, sentence, word)

Language modeling
- Given an English string e, assigns P(e) by formula

Translation modeling
- Given a pair of strings <f,e>, assigns P(f | e) by formula

Decoding
- Given a language model, a translation model, and a new sentence f … find translation e maximizing P(e) * P(f | e)

Parameter optimization
- Given a model with multiple feature functions, how are they related? What are the optimal parameters?

Evaluation
- How well is a system doing? How can we compare two systems?

Problems for Statistical MT

Preprocessing

Language modeling

Translation modeling

Decoding

Parameter optimization

Evaluation

From No Data to Sentence Pairs

Easy way: Linguistic Data Consortium (LDC)

Really hard way: pay $$$
- Suppose one billion words of parallel data were sufficient
- At 20 cents/word, that's $200 million

Pretty hard way: Find it, and then earn it!

How would you obtain data?

What are the challenges?

From No Data to Sentence Pairs

Easy way: Linguistic Data Consortium (LDC)

Really hard way: pay $$$
- Suppose one billion words of parallel data were sufficient
- At 20 cents/word, that's $200 million

Pretty hard way: Find it, and then earn it!
- De-formatting
- Remove strange characters
- Character code conversion
- **Document alignment**
- **Sentence alignment**
- **Tokenization (also called Segmentation)**

What is this?

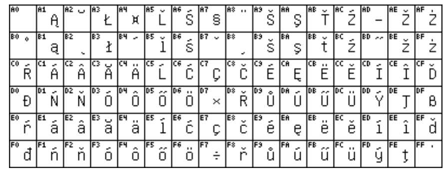
49 20 6c 69 6b 65 20 4e 4c 50 20 63 6c 61 73 73

I _ l i k e _ N L P _ c l a s s

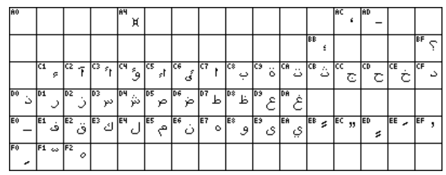
Hexadecimal file contents

Totally depends on file encoding!

ISO-8859-2 (Latin2)



ISO-8859-6 (Arabic)



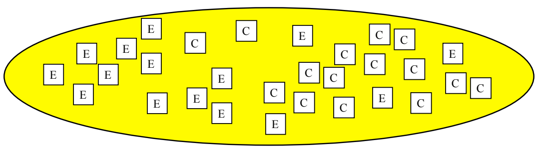
Chinese?

- **GB Code**
- **GBK Code**
- **Big 5 Code**
- **CNS-11643-1992**
- …

If you don't get the characters right…

Document Alignment

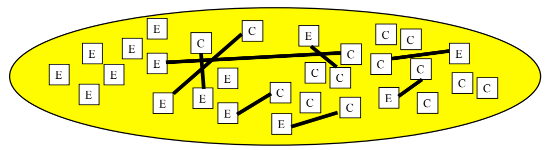
Input:
- Big bag of files obtained from somewhere, believed to contain pairs of files that are translations of each other.



Output:
- List of pairs of files that are actually translations.

Document Alignment

Input:
- Big bag of files obtained from somewhere, believed to contain pairs of files that are translations of each other.



Output:
- List of pairs of files that are actually translations.

Sentence Alignment

The old man is happy. He has fished many times. His wife talks to him. The fish are jumping. The sharks await.

El viejo está feliz porque ha pescado muchos veces. Su mujer habla con él. Los tiburones esperan.

Sentence Alignment

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

What should be aligned?

Sentence Alignment

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

Sentence Alignment

1. The old man is happy. He has fished many times.	1. El viejo está feliz porque ha pescado muchos veces.
2. His wife talks to him.	2. Su mujer habla con él.
3. The sharks await.	3. Los tiburones esperan.

Note that unaligned sentences are thrown out, and sentences are merged in n-to-m alignments (n, m > 0).

Tokenization (or Segmentation)

English
- Input (some byte stream):

```
"There," said Bob.
```

- Output (7 “tokens” or “words”):

```
" There , " said Bob .
```

Chinese
- Input (byte stream): 美国关岛国际机场及其办公室均接获一名自称沙地阿拉伯富商拉登等发出的电子邮件。
- Output: 美国 关岛国 际机 场 及其 办公 室 均接获 一名 自称 沙地 阿拉 伯富 商拉登 等发 出 的 电子邮件。

Problems for Statistical MT

Preprocessing

Language modeling

Translation modeling

Decoding

Parameter optimization

Evaluation

Language Modeling

Most common: n-gram language models

More data the better (Google n-grams)

Domain is important

Problems for Statistical MT

Preprocessing

Language modeling

Translation modeling

Decoding

Parameter optimization

Evaluation

Translation Model

Want: probabilistic model gives us how likely one sentence is to be a translation of another, i.e *p(foreign | english)*

Mary did not slap the green witch

↓

Maria no dió una botefada a la bruja verde

Can we just model this directly, i.e. p(foreign | english) ?
How would we estimate these probabilities, e.g.
p(“Maria …” | “Mary …”)?

Translation Model

Want: probabilistic model gives us how likely one sentence is to be a translation of another, i.e *p(foreign | english)*

Mary did not slap the green witch

↓

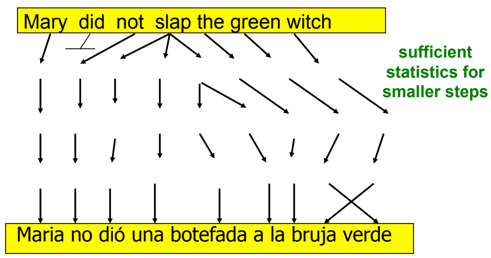
Maria no dió una botefada a la bruja verde

$$p(\text{“Maria…”} \mid \text{“Mary…”}) = \frac{\text{count(“Mary…” aligned-to “Maria…”)}}{\text{count(“Mary…”)}}$$

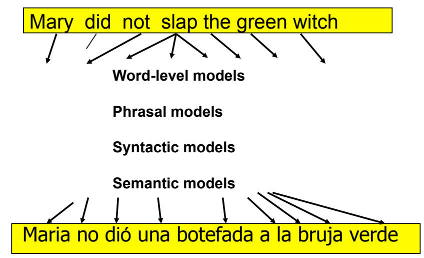
Not enough data for most sentences!

Translation Model

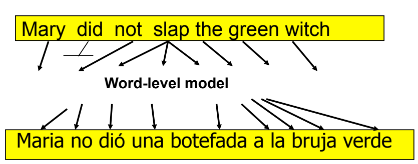
Key: break up process into smaller steps



What kind of Translation Model?



IBM Word-level models



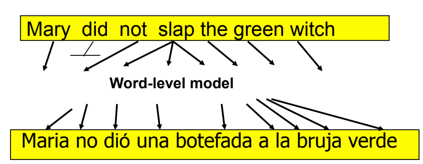
Generative story: description of how the translation happens

1. Each English word gets translated as 0 or more Foreign words

2. Some additional foreign words get inserted

3. Foreign words then get shuffled

IBM Word-level models



Each foreign word is *aligned* to exactly one English word.

Key idea: decompose *p(foreign* | *english)* into word translation probabilities of the form *p(foreign_word* | *english_word)*

IBM described 5 different levels of models with increasing complexity (and decreasing independence assumptions)

Some notation

$E = e_1 e_2 \ldots e_{|E|}$ English sentence with length |E|

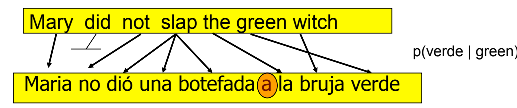
$F = f_1 f_2 \ldots f_{|F|}$ Foreign sentence with length |F|

Mary did not slap the green witch
e_1 e_2 e_3 e_4 e_5 e_6 e_7

f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9
Maria no dió una botefada a la bruja verde

Translation model: $p(F \mid E) = p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|})$

Word models: IBM Model 1

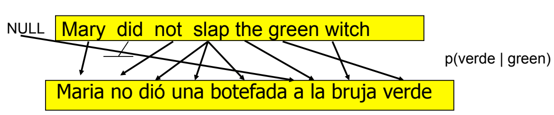


Each foreign word is aligned to exactly one English word

This is the **ONLY** thing we model!

Does the model handle foreign words that are not aligned, e.g. “a”?

Word models: IBM Model 1



Each foreign word is aligned to exactly one English word

This is the **ONLY** thing we model!

Include a “NULL” English word and align to this to account for deletion

Word models: IBM Model 1

generative story -> probabilistic model
- Key idea: introduce “hidden variables” to model the word alignment

$$p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|})$$

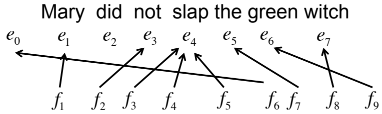
$$\Downarrow$$

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|})$$

- one variable for each foreign word
- a_i corresponds to the ith foreign word
- each a_i can take a value 0…|E|

Alignment variables

Mary did not slap the green witch



Maria no dió una botefada a la bruja verde

a_1	1
a_2	3
a_3	4
a_4	4
a_5	4
a_6	0
a_7	5
a_8	7
a_9	6

Alignment variables

And the program has been implemented

e_0 e_1 e_2 e_3 e_4 e_5 e_6

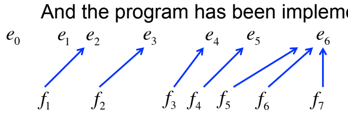
Alignment?

f_1 f_2 f_3 f_4 f_5 f_6 f_7

Le programme a ete mis en application

Alignment variables

And the program has been implemented

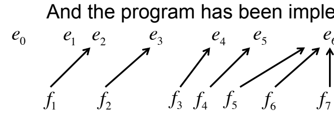


Le programme a ete mis en application

a_1	?
a_2	?
a_3	?
a_4	?
a_5	?
a_6	?
a_7	?

Alignment variables

And the program has been implemented



Le programme a ete mis en application

a_1	2
a_2	3
a_3	4
a_4	5
a_5	6
a_6	6
a_7	6

Probabilistic model

$$p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|}) \overset{?}{=} p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|})$$

NO!

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) \Rightarrow p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|})$$

How do we get rid of variables?

Joint distribution

NLPPass, EngPass	P(NLPPass, EngPass)
true, true	.88
true, false	.01
false, true	.04
false, false	.07

What is P(ENGPass)?

Joint distribution

NLPPass, EngPass	P(NLPPass, EngPass)
true, true	.88
true, false	.01
false, true	.04
false, false	.07

0.92

How did you figure that out?

Joint distribution

$$P(x) = \sum_{y \in Y} p(x,y)$$

Called "marginalization", aka summing over a variable

NLPPass, EngPass	P(NLPPass, EngPass)
true, true	.88
true, false	.01
false, true	.04
false, false	.07

Probabilistic model

$$p(f_1 f_2 \ldots f_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \sum_{a_1} \sum_{a_2} \ldots \sum_{a_{|F|}} p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|})$$

Sum over all possible values, i.e. marginalize out the alignment variables

Independence assumptions

IBM Model 1:

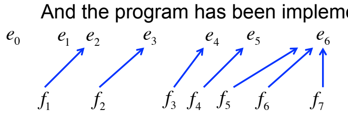
$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

What independence assumptions are we making?

What information is lost?

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

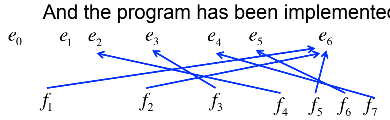
And the program has been implemented



Le programme a ete mis en application

Are the probabilities any different under model 1?

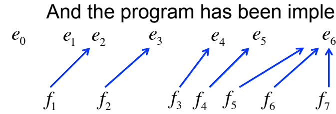
And the program has been implemented



application en programme Le mis ete a

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

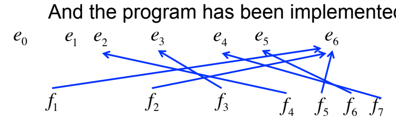
And the program has been implemented



Le programme a ete mis en application

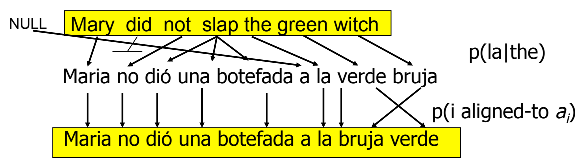
No. Model 1 ignores word order!

And the program has been implemented



application en programme Le mis ete a

IBM Model 2

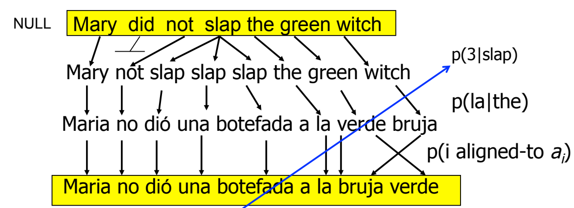


$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(i\ aligned\text{-}to\ a_i)\, p(f_i \mid e_{a_i})$$

Models word movement by position, e.g.

- Words don't tend to move too much
- Words at the beginning move less than words at the end

IBM Model 3



Incorporates "fertility": how likely a particular English word is to produce multiple foreign words

Word-level models

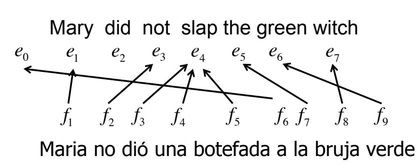
Problems/concerns?

- Multiple English words for one French word
  - IBM models can do one-to-many (fertility) but not many-to-one
- Phrasal Translation
  - "real estate", "note that", "interest in"
- Syntactic Transformations
  - Verb at the beginning in Arabic
  - Translation model penalizes any proposed re-ordering
  - Language model not strong enough to force the verb to move to the right place

Benefits of word-level model

Rarely used in practice for modern MT systems

Why talk about them?



Two key side effects of training a word-level model:

- Word-level alignment
- p(f | e): translation dictionary

Training a word-level model

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

Where do these come from?

Have to learn them!

The old man is happy. He has fished many times.	El viejo está feliz porque ha pescado muchos veces.
His wife talks to him.	Su mujer habla con él.
The sharks await.	Los tiburones esperan.
…	…

Training a word-level model

The old man is happy. He has fished many times.	El viejo está feliz porque ha pescado muchos veces.
His wife talks to him.	Su mujer habla con él.
The sharks await.	Los tiburones esperan.
…	…

$$p(f_1 f_2 \ldots f_{|F|}, a_1 a_2 \ldots a_{|F|} \mid e_1 e_2 \ldots e_{|E|}) = \prod_{i=1}^{|F|} p(f_i \mid e_{a_i})$$

$p(f_i \mid e_{a_i})$: probability that *e* is translated as *f*

How do we learn these?

What data would be useful?

Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$p(f_i \mid e_{a_i}) =$?

Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

$$p(f_i \mid e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

p(el | the) = 0.5
p(Los | the) = 0.5

Any problems concerns?

Thought experiment

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

His wife talks to him.

Su mujer habla con él.

The sharks await.

Los tiburones esperan.

Getting data like this is expensive!

Even if we had it, what happens when we switch to a new domain/corpus

Training without alignments

a b

x y

c b

z x

How should these be aligned?

There is some information!
(Think of the alien translation task last time)

Thought experiment #2

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Annotator 1

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

Annotator 2

$$p(f_i \mid e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

What do we do?

Thought experiment #2

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

80 annotators

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

20 annotators

$$p(f_i \mid e_{a_i}) = \frac{count(f\ aligned\text{-}to\ e)}{count(e)}$$

What do we do?

Thought experiment #2

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

80 annotators

The old man is happy. He has fished many times.

El viejo está feliz porque ha pescado muchos veces.

20 annotators

$$p(f_i \mid e_{a_i}) = \frac{count(f \ aligned\text{-}to \ e)}{count(e)}$$

Use partial counts:
- count(viejo | man) 0.8
- count(viejo | old) 0.2