Translation Models

David Kauchak
CS159 – Fall 2014

Language translation

Yo quiero Taco Bell

Noisy channel model

\[ p(e \mid f) \propto p(f \mid e)p(e) \]

- translation model
- language model

how do English sentences get translated to foreign?
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?

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?
What is this?

Hexadecimal file contents

Totally depends on file encoding!

Chinese?
- GB Code
- GBK Code
- Big 5 Code
- CNS-11643-1992
- ...

If you don’t get the characters right…

ISO-8859-2 (Latin2)

ISO-8859-6 (Arabic)

Chinese?
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 muchas veces. Su mujer habla con él. 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 muchas veces.
2. Su mujer habla con él.
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. \( p(\text{foreign} | \text{english}) \)

Mary did not slap the green witch

Maria no dio una botefada a la bruja verde

Can we just model this directly, i.e. \( p(\text{foreign} | \text{english}) \)?

How would we estimate these probabilities, e.g.

\( p( "\text{Maria} \ldots " | "\text{Mary} \ldots " ) \)?

Translation Model

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

\[
p( "\text{Maria} \ldots " | "\text{Mary} \ldots " ) = \frac{\text{count("Maria" aligned-to "Maria")}}{\text{count( "Mary" )}}
\]

Not enough data for most sentences!
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

Each foreign word is aligned to exactly one English word.

Key idea: decompose \( p(\text{foreign} \mid \text{english}) \) into word translation probabilities of the form \( p(\text{foreign word} \mid \text{english word}) \)

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

- $E = e_1, e_2, ..., e_E$: English sentence with length $|E|
- $F = f_1, f_2, ..., f_F$: Foreign sentence with length $|F|

Mary did not slap the green witch

$e_1, e_2, e_3, e_4, e_5, e_6, e_7$

Maria no dió una botefada a la bruja verde

Translation model: $p(F | E) = p(f_1, f_2, ..., f_F | e_1, e_2, ..., e_E)$

Word models: IBM Model 1

null

Mary did not slap the green witch

$P$ (verde | green)

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

- Generative story $\rightarrow$ probabilistic model
- Key idea: introduce "hidden variables" to model the word alignment

- $p(f_1, f_2, ..., f_F | e_1, e_2, ..., e_E)$
- $p(f_1, f_2, ..., f_F | a_1, a_2, ..., a_F | e_1, e_2, ..., e_E)$

- one variable for each foreign word
- $a_i$ corresponds to the $i$th foreign word
- each $a_i$ can take a value $0, ..., |E|$
Alignment variables

Mary did not slap the green witch
Maria no dio una botefada a la bruja verde

Alignment variables

And the program has been implemented
Le programme a ete mis en application

Alignment variables

And the program has been implemented
Le programme a ete mis en application

Alignment variables

And the program has been implemented
Le programme a ete mis en application
Probabilistic model

\[ p(f_1, f_2, \ldots, f_F | e_1, e_2, \ldots, e_E) \]

\[ \neq \]

\[ p(f_1, f_2, \ldots, f_F, a_1, a_2, \ldots, a_a | e_1, e_2, \ldots, e_E) \]

How do we get rid of variables?

Joint distribution

<table>
<thead>
<tr>
<th>NLPPass, EngPass</th>
<th>P(NLPPass, EngPass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>true, true</td>
<td>0.88</td>
</tr>
<tr>
<td>true, false</td>
<td>0.01</td>
</tr>
<tr>
<td>false, true</td>
<td>0.04</td>
</tr>
<tr>
<td>false, false</td>
<td>0.07</td>
</tr>
</tbody>
</table>

What is \( P(\text{ENGPass}) \)?

Joint distribution

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

Called "marginalization", aka summing over a variable

Joint distribution

0.92

How did you figure that out?
Probabilistic model

\[ p(f_1, f_2, \ldots, f_T | e_0, e_1, \ldots, e_T) = \sum_{a_1} \sum_{a_2} \ldots \sum_{a_T} p(f_1, f_2, \ldots, f_T | a_1, a_2, \ldots, a_T | e_0, e_1, \ldots, e_T) \]

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

Independence assumptions

IBM Model 1:

\[ p(f_1, f_2, \ldots, f_T | a_1, a_2, \ldots, a_T | e_0, e_1, \ldots, e_T) = \prod_{i=1}^{T} p(f_i | e_{a_i}) \]

What independence assumptions are we making?

What information is lost?

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

No. Model 1 ignores word order!
IBM Model 2

Mary did not slap the green witch.

Maria no dió una botefada a la bruja verde

$P(f | e) = \prod_{i} p(f_i | e_i) p(l | e_i)$

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

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

$$p(f_i | e) = \prod_{i=1}^{F} p(f_i | e)$$

Where do these come from?

Have to learn them!

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

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

$$p(f_i | e)$$: 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.

$$p(f_i | e_a) = ?$$

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.

$$p(f_i | e_a) = \frac{\text{count}(f \text{ aligned-to } e)}{\text{count}(e)}$$

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

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.

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

El viejo está feliz porque ha pescado muchos veces.

\[
p(f_i \mid e_n) = \frac{\text{count}(f \text{ aligned-to } e)}{\text{count}(e)}
\]

What do we do?

80 annotators

20 annotators

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.

p(f | eᵦ) = \frac{count(f \text{ aligned-to } e)}{count(e)}

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