Lecture 10: Machine Translation I

Alan Ritter

(many slides from Greg Durrett)
This Lecture

- MT and evaluation
- Word alignment
- Language models
- Phrase-based decoders
- Syntax-based decoders (probably next time)
MT Basics
MT Basics
Trump Pope family watch a hundred years a year in the White House balcony
People’s Daily, August 30, 2017

Trump Pope family watch a hundred years a year in the White House balcony
Ideally
MT Ideally

- I have a friend => ∃x friend(x, self)
I have a friend => ∃x friend(x, self) => J’ai un ami
I have a friend => \( \exists x \text{ friend}(x, \text{self}) \) => J’ai un ami

J’ai une amie
MT Ideally

- I have a friend => ∃x friend(x, self) => J’ai un ami
  - J’ai une amie

- May need information you didn’t think about in your representation
MT Ideally

- I have a friend => ∃x friend(x,self) => J’ai un ami
  - J’ai une amie

- May need information you didn’t think about in your representation
- Hard for semantic representations to cover everything
MT Ideally

- I have a friend => $\exists x \ friend(x,自我) => J'ai un ami$
  
  J’ai une amie

- May need information you didn’t think about in your representation

- Hard for semantic representations to cover everything

- Everyone has a friend =>
MT Ideally

- I have a friend => $\exists x\ friend(x, \text{self})$ => J’ai un ami
  J’ai une amie

- May need information you didn’t think about in your representation

- Hard for semantic representations to cover everything

- Everyone has a friend => $\exists x \forall y\ friend(x, y)$
  $\forall x \exists y\ friend(x, y)$
Ideally

- I have a friend => ∃x friend(x,self) => J’ai un ami
  J’ai une amie

- May need information you didn’t think about in your representation

- Hard for semantic representations to cover everything

- Everyone has a friend => ∃x∀y friend(x,y) => Tous a un ami
  ∀x∃y friend(x,y)
MT Ideally

- I have a friend => ∃x friend(x, self) => J’ai un ami
  J’ai une amie

- May need information you didn’t think about in your representation

- Hard for semantic representations to cover everything

- Everyone has a friend => ∀x∀y friend(x, y) => Tous a un ami
  ∀x∃y friend(x, y)

- Can often get away without doing all disambiguation — same ambiguities may exist in both languages
Today: mostly phrase-based, some syntax
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use

- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate

  - How to identify phrases? Word alignment over source-target bitext
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use

- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations
Phrase-Based MT

- Key idea: translation works better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
- Decoder takes phrases and a language model and searches over possible translations
  - NOT like standard discriminative models (take a bunch of translation pairs, learn a ton of parameters in an end-to-end way)
Phrase-Based MT

Phrase table $P(f|e)$

Unlabeled English data

Language model $P(e)$

Noisy channel model: combine scores from translation model + language model to translate foreign to English

"Translate faithfully but make fluent English"
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>I am exhausted</th>
<th>1-gram</th>
<th>2-gram</th>
<th>3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypothesis 1</td>
<td>___</td>
<td>3/3</td>
<td>1/2</td>
<td>0/1</td>
</tr>
<tr>
<td>hypothesis 2</td>
<td>Tired is I</td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td>hypothesis 3</td>
<td>I I I</td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td>reference 1</td>
<td>I am tired</td>
<td>___</td>
<td>___</td>
<td>___</td>
</tr>
<tr>
<td>reference 2</td>
<td>I am ready to sleep now and so exhausted</td>
<td>___</td>
<td>___</td>
<td>___</td>
</tr>
</tbody>
</table>
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

\[
\text{BLEU} = BP \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)
\]

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>1-gram</th>
<th>2-gram</th>
<th>3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hypothesis 1</strong></td>
<td>3/3</td>
<td>1/2</td>
<td>0/1</td>
</tr>
<tr>
<td><strong>hypothesis 2</strong></td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td><strong>hypothesis 3</strong></td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td><strong>reference 1</strong></td>
<td>I am exhausted</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>reference 2</strong></td>
<td>I am ready to sleep now and so exhausted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples:
- I am exhausted
- Tired is I
- I am tired
- I am ready to sleep now and so exhausted
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

\[ \text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right). \]
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

$$\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right).$$

Typically $n = 4$, $w_i = 1/4$
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

$$\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right).$$

- Typically $n = 4$, $w_i = 1/4$

$$\text{BP} = \begin{cases} 
1 & \text{if } c > r \\
\frac{1}{e^{(1-r/c)}} & \text{if } c \leq r
\end{cases}.$$
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

\[
\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right).
\]

Typically \( n = 4, \ w_i = 1/4 \)

\[
\text{BP} = \begin{cases} 
1 & \text{if } c > r \\
\frac{1}{e^{(1-r/c)}} & \text{if } c \leq r
\end{cases}
\]

\( r = \text{length of reference} \)
\( c = \text{length of prediction} \)
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

\[
\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)
\]

\[
\text{BP} = \begin{cases} 
1 & \text{if } c > r \\
\frac{1}{1 - c/r} & \text{if } c \leq r
\end{cases}
\]

Typically \( n = 4, w_i = 1/4 \)

\( r = \text{length of reference} \)
\( c = \text{length of prediction} \)

- Does this capture fluency and adequacy?
Better methods with human-in-the-loop

HTER: human-assisted translation error rate

If you’re building real MT systems, you do user studies. In academia, you mostly use BLEU
Word Alignment
Word Alignment

- Input: a bitext, pairs of translated sentences
  - nous acceptons votre opinion . ||| we accept your view
  - nous allons changer d’avis ||| we are going to change our minds
Word Alignment

- Input: a bitext, pairs of translated sentences

  nous acceptons votre opinion . ||| we accept your view
  nous allons changer d’avis ||| we are going to change our minds
Word Alignment

- Input: a bitext, pairs of translated sentences
  - nous acceptons votre opinion.
  - nous allons changer d’avis

- Output: alignments between words in each sentence
  - nous acceptons votre opinion.
  - nous allons changer d’avis
Word Alignment

- Input: a bitext, pairs of translated sentences
  - *nous acceptons votre opinion.* | | | we accept your view
  - *nous allons changer d’avis* | | | we are going to change our minds

- Output: alignments between words in each sentence
  - “accept and acceptons are aligned”
Word Alignment

- Input: a bitext, pairs of translated sentences
  - nous acceptons votre opinion. ||| we accept your view
  - nous allons changer d’avis ||| we are going to change our minds

- Output: alignments between words in each sentence
  - We will see how to turn these into phrases
  - “accept and acceptons are aligned”
1-to-Many Alignments

And₁ the₂ program₃ has₄ been₅ implemented₆

Le₁ programme₂ a₃ été₄ mis₅ en₆ application₇
Word Alignment

- Models $P(f|e)$: probability of “French” sentence being generated from “English” sentence according to a model
Models $P(f|e)$: probability of “French” sentence being generated from “English” sentence according to a model

Latent variable model: $P(f|e) = \sum_a P(f, a|e) = \sum_a P(f|a, e)P(a)$
Word Alignment

- Models $P(f|e)$: probability of “French” sentence being generated from “English” sentence according to a model

- Latent variable model: $P(f|e) = \sum_a P(f, a|e) = \sum_a P(f|a, e)P(a)$

- Correct alignments should lead to higher-likelihood generations, so by optimizing this objective we will learn correct alignments
IBM Model 1

- Each French word is aligned to \textit{at most} one English word

\[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i}) P(a_i) \]

Brown et al. (1993)
IBM Model 1

- Each French word is aligned to at most one English word

\[ P(f, a|e) = \prod_{i=1}^{n} P(f_i|e_{a_i}) P(a_i) \]

e Thank you, I shall do so gladly.

Brown et al. (1993)
IBM Model 1

- Each French word is aligned to at most one English word

\[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i}) P(a_i) \]

\[ e \quad \text{Thank you, I shall do so gladly.} \]

\[ a \quad 0 \quad 2 \quad 6 \quad 5 \quad 7 \quad 7 \quad 7 \quad 7 \quad 7 \quad 8 \]
IBM Model 1

- Each French word is aligned to at most one English word

\[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i})P(a_i) \]

- Thank you, I shall do so gladly.

- Gracias, lo hare de muy buen grado.
IBM Model 1

- Each French word is aligned to at most one English word

\[
P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i})P(a_i)
\]

- Thank you, I shall do so gladly.

- Gracias, lo hare de muy buen grado.

- Set \( P(a) \) uniformly (no prior over good alignments)

Brown et al. (1993)
IBM Model 1

- Each French word is aligned to *at most* one English word

\[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i}) P(a_i) \]

- **e** Thank you , I shall do so gladly .

- **a**
  - 0
  - 2
  - 6
  - 5
  - 7
  - 7
  - 7
  - 7
  - 8

- **f** Gracias , lo hare de muy buen grado .

- Set \( P(a) \) uniformly (no prior over good alignments)

- \( P(f_i | e_{a_i}) \): word translation probability table

Brown et al. (1993)
HMM for Alignment

- Sequential dependence between a’s to capture monotonicity

\[
P(f, a|e) = \prod_{i=1}^{n} P(f_i|e_{a_i}) P(a_i|a_{i-1})
\]

- Thank you, I shall do so gladly.

- Gracias, lo hare de muy buen grado.

Brown et al. (1993)
Sequential dependence between a’s to capture monotonicity

\[ P(f, a|e) = \prod_{i=1}^{n} P(f_i|e_{a_i})P(a_i|a_{i-1}) \]

**e** Thank you, I shall do so gladly.

**f** Gracias, lo hare de muy buen grado.

Alignment dist parameterized by jump size: \( P(a_j - a_{j-1}) \)

Brown et al. (1993)
HMM for Alignment

- Sequential dependence between a’s to capture monotonicity
  
  \[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i})P(a_i | a_{i-1}) \]

  e  Thank you, I shall do so gladly.

  f  Gracias, lo hare de muy buen grado.

- Alignment dist parameterized by jump size: \( P(a_j - a_{j-1}) \)

- \( P(f_i | e_{a_i}) \): same as before

Brown et al. (1993)
Which direction is this?
- Which direction is this?

- Alignments are generally monotonic (along diagonal)
Which direction is this?

Alignments are generally monotonic (along diagonal)

Some mistakes, especially when you have rare words (garbage collection)
Evaluating Word Alignment

- “Alignment error rate”: use labeled alignments on small corpus

<table>
<thead>
<tr>
<th>Model</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 INT</td>
<td>19.5</td>
</tr>
<tr>
<td>HMM E→F</td>
<td>11.4</td>
</tr>
<tr>
<td>HMM F→E</td>
<td>10.8</td>
</tr>
<tr>
<td>HMM AND</td>
<td>7.1</td>
</tr>
<tr>
<td>HMM INT</td>
<td>4.7</td>
</tr>
<tr>
<td>GIZA M4 AND</td>
<td>6.9</td>
</tr>
</tbody>
</table>

- Run Model 1 in both directions and intersect “intelligently”
- Run HMM model in both directions and intersect “intelligently”
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words.
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words

d’assister à la réunion et ||| to attend the meeting and
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words

d’assister à la réunion et ||| to attend the meeting and

assister à la réunion ||| attend the meeting
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words

d’assister à la réunion et | | | to attend the meeting and

assister à la réunion | | | attend the meeting

la réunion and | | | the meeting and
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words

d’assister à la réunion et ||| to attend the meeting and

assister à la réunion ||| attend the meeting

la réunion and ||| the meeting and

nous ||| we
Find contiguous sets of aligned words in the two languages that don’t have alignments to other words

- d’assister à la réunion et ||| to attend the meeting and
- assister à la réunion ||| attend the meeting
- la réunion and ||| the meeting and
- nous ||| we
- ...
Phrase Extraction

- Find contiguous sets of aligned words in the two languages that don’t have alignments to other words
  
  d’assister à la réunion et ||| to attend the meeting and
  assister à la réunion ||| attend the meeting
  la réunion and ||| the meeting and
  nous ||| we

- Lots of phrases possible, count across all sentences and score by frequency
Language Modeling
Phrase-Based MT

Phrase table $P(f|e)$

Unlabeled English data

Language model $P(e)$

Noisy channel model:
combine scores from translation model + language model to translate foreign to English

"Translate faithfully but make fluent English"
N-gram Language Models

I visited San ______ put a distribution over the next word
N-gram Language Models

I visited San _____ put a distribution over the next word

- Simple generative model: distribution of next word is a multinomial distribution conditioned on previous n-1 words
N-gram Language Models

I visited San _____ put a distribution over the next word

Simple generative model: distribution of next word is a multinomial distribution conditioned on previous n-1 words

\[
P(x|\text{visited San}) = \frac{\text{count(visited San, } x)}{\text{count(visited San)}}
\]
N-gram Language Models

I visited San _____ put a distribution over the next word

- Simple generative model: distribution of next word is a multinomial distribution conditioned on previous n-1 words

\[ P(x|\text{visited San}) = \frac{\text{count}(\text{visited San}, x)}{\text{count}(\text{visited San})} \]

Maximum likelihood estimate of this probability from a corpus
N-gram Language Models

I visited San _____ put a distribution over the next word

- Simple generative model: distribution of next word is a multinomial distribution conditioned on previous n-1 words

\[
P(x|\text{visited San}) = \frac{\text{count(visited San, } x)}{\text{count(visited San)}}
\]

Maximum likelihood estimate of this probability from a corpus

- Just relies on counts, even in 2008 could scale up to 1.3M word types, 4B n-grams (all 5-grams occurring >40 times on the Web)
I visited San _____ put a distribution over the next word!
Smoothing N-gram Language Models

I visited San _____ put a distribution over the next word!

- Smoothing is very important, particularly when using 4+ gram models
Smoothing N-gram Language Models

I visited San _____ put a distribution over the next word!

- Smoothing is very important, particularly when using 4+ gram models

\[ P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count}(\text{visited San}, x)}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})} \]
Smoothing is very important, particularly when using 4+ gram models.

\[
P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count(visited San, } x)}{\text{count(visited San)}} + \lambda \frac{\text{count( San, } x)}{\text{count(San)}}
\]
Smoothing N-gram Language Models

I visited San _____ put a distribution over the next word!

- Smoothing is very important, particularly when using 4+ gram models
  
  \[ P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count(visited San, } x\text{)}}{\text{count(visited San)}} + \lambda \frac{\text{count(San, } x\text{)}}{\text{count(San)}} \]

- One technique is “absolute discounting:” subtract off constant $k$ from numerator, set lambda to make this normalize ($k=1$ is like leave-one-out)

smooth this too!
Smoothing N-gram Language Models

I visited San ____ put a distribution over the next word!

- Smoothing is very important, particularly when using 4+ gram models

\[
P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count}(\text{visited San}, x)}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})}
\]

- One technique is “absolute discounting:” subtract off constant \( k \) from numerator, set lambda to make this normalize (\( k=1 \) is like leave-one-out)

\[
P(x|\text{visited San}) = \frac{\text{count}(\text{visited San}, x) - k}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})}
\]
Smoothing N-gram Language Models

I visited San _____ put a distribution over the next word!

- Smoothing is very important, particularly when using 4+ gram models

\[ P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count}(\text{visited San}, x)}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})} \]

- One technique is “absolute discounting:” subtract off constant \( k \) from numerator, set lambda to make this normalize (\( k=1 \) is like leave-one-out)

\[ P(x|\text{visited San}) = \frac{\text{count}(\text{visited San}, x) - k}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})} \]

- Kneser-Ney smoothing: this trick, plus low-order distributions modified to capture fertilities (how many distinct words appear in a context)
Engineering N-gram Models

- For 5+-gram models, need to store between 100M and 10B context-word-count triples

- Make it fit in memory by *delta encoding* scheme: store deltas instead of values and use variable-length encoding

<table>
<thead>
<tr>
<th>(a) Context-Encoding</th>
<th>(b) Context Deltas</th>
<th>(c) Bits Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>c</td>
<td>val</td>
</tr>
<tr>
<td>1933</td>
<td>15176585</td>
<td>3</td>
</tr>
<tr>
<td>1933</td>
<td>15176587</td>
<td>2</td>
</tr>
<tr>
<td>1933</td>
<td>15176593</td>
<td>1</td>
</tr>
<tr>
<td>1933</td>
<td>15176613</td>
<td>8</td>
</tr>
<tr>
<td>1933</td>
<td>15179801</td>
<td>1</td>
</tr>
<tr>
<td>1935</td>
<td>15176585</td>
<td>298</td>
</tr>
<tr>
<td>1935</td>
<td>15176589</td>
<td>1</td>
</tr>
</tbody>
</table>

Pauls and Klein (2011), Heafield (2011)
Neural Language Models

- Early work: feedforward neural networks looking at context

Mnih and Hinton (2003)
Neural Language Models

- Early work: feedforward neural networks looking at context

\[ P(w_i|w_{i-n}, \ldots, w_{i-1}) \]

Mnih and Hinton (2003)
Neural Language Models

- Early work: feedforward neural networks looking at context

\[ P(w_i | w_{i-n}, \ldots, w_{i-1}) \]

- Variable length context with RNNs:

\[ P(w_i | w_1, \ldots, w_{i-1}) \]

Mnih and Hinton (2003)
Neural Language Models

- Early work: feedforward neural networks looking at context
  \[ P(w_i|w_{i-n}, \ldots, w_{i-1}) \]

- Variable length context with RNNs:
  - Works like a decoder with no encoder
  \[ P(w_i|w_1, \ldots, w_{i-1}) \]

Mnih and Hinton (2003)
Neural Language Models

- Early work: feedforward neural networks looking at context
  
  \[
  P(w_i|w_{i-n}, \ldots, w_{i-1})
  \]

- Variable length context with RNNs:
  - Works like a decoder with no encoder
  
  \[
  P(w_i|w_1, \ldots, w_{i-1})
  \]

- Slow to train over lots of data!

Mnih and Hinton (2003)
Evaluation
Evaluation

(One sentence) negative log likelihood: \[ \sum_{i=1}^{n} \log p(x_i | x_1, \ldots, x_{i-1}) \]
(One sentence) negative log likelihood: $\sum_{i=1}^{n} \log p(x_i | x_1, \ldots, x_{i-1})$

Perplexity: $2^{- \frac{1}{n} \sum_{i=1}^{n} \log_2 p(x_i | x_1, \ldots, x_{i-1})}$
Evaluation

- (One sentence) negative log likelihood: \( \sum_{i=1}^{n} \log p(x_i|x_1, \ldots, x_{i-1}) \)

- Perplexity: \( 2^{-\frac{1}{n} \sum_{i=1}^{n} \log_2 p(x_i|x_1, \ldots, x_{i-1})} \)

  - NLL (base 2) averaged over the sentence, exponentiated
Evaluation

- (One sentence) negative log likelihood: \( \sum_{i=1}^{n} \log p(x_i|x_1, \ldots, x_{i-1}) \)

- Perplexity: \( 2^{\frac{1}{n} \sum_{i=1}^{n} \log_2 p(x_i|x_1, \ldots, x_{i-1})} \)
  - NLL (base 2) averaged over the sentence, exponentiated
  - NLL = -2 -> on average, correct thing has prob 1/4 -> PPL = 4. PPL is sort of like branching factor
Results

Merity et al. (2017), Melis et al. (2017)
Results

- Evaluate on Penn Treebank: small dataset (1M words) compared to what’s used in MT, but common benchmark

Merity et al. (2017), Melis et al. (2017)
Results

- Evaluate on Penn Treebank: small dataset (1M words) compared to what’s used in MT, but common benchmark
- Kneser-Ney 5-gram model with cache: PPL = 125.7

Merity et al. (2017), Melis et al. (2017)
Results

- Evaluate on Penn Treebank: small dataset (1M words) compared to what’s used in MT, but common benchmark
- Kneser-Ney 5-gram model with cache: PPL = 125.7
- LSTM: PPL ~ 60-80 (depending on how much you optimize it)

Merity et al. (2017), Melis et al. (2017)
Results

- Evaluate on Penn Treebank: small dataset (1M words) compared to what’s used in MT, but common benchmark
- Kneser-Ney 5-gram model with cache: PPL = 125.7
- LSTM: PPL ~ 60-80 (depending on how much you optimize it)
- Melis et al.: many neural LM improvements from 2014-2017 are subsumed by just using the right regularization (right dropout settings). So LSTMs are pretty good

Merity et al. (2017), Melis et al. (2017)
Decoding
Phrase-Based Decoding

- **Inputs:**
  - Language model that scores $P(e_i|e_1, \ldots, e_{i-1}) \approx P(e_i|e_{i-n-1}, \ldots, e_{i-1})$
  - Phrase table: set of phrase pairs $(e, f)$ with probabilities $P(f|e)$
Phrase-Based Decoding

- **Inputs:**
  - Language model that scores $P(e_i|e_1, \ldots, e_{i-1}) \approx P(e_i|e_{i-n-1}, \ldots, e_{i-1})$
  - Phrase table: set of phrase pairs $(e, f)$ with probabilities $P(f|e)$

- What we want to find: $e$ produced by a series of phrase-by-phrase translations from an input $f$, possibly with reordering:
Phrase-Based Decoding

- Inputs:
  - Language model that scores $P(e_i|e_1, \ldots, e_{i-1}) \approx P(e_i|e_{i-n-1}, \ldots, e_{i-1})$
  - Phrase table: set of phrase pairs $(e, f)$ with probabilities $P(f|e)$

- What we want to find: $e$ produced by a series of phrase-by-phrase translations from an input $f$, possibly with reordering:
Phrase lattices are big!

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Lawces are big!</th>
</tr>
</thead>
<tbody>
<tr>
<td>the 7 people</td>
<td>including by some and the russian the astronauts</td>
</tr>
<tr>
<td>it 7 people</td>
<td>included by france and the russian international astronomical of rapporteur</td>
</tr>
<tr>
<td>this 7 out</td>
<td>including the french and the russian fifth</td>
</tr>
<tr>
<td>these 7 among</td>
<td>including from the french and the russian of space members</td>
</tr>
<tr>
<td>that 7 persons</td>
<td>including from the french and to russian of the aerospace members</td>
</tr>
<tr>
<td>7 include</td>
<td>from the french and russian astronauts</td>
</tr>
<tr>
<td>7 numbers</td>
<td>include from france and russian of astronauts who</td>
</tr>
<tr>
<td>7 populations</td>
<td>include those from france and russian astronauts</td>
</tr>
<tr>
<td>7 deportees</td>
<td>included come from france and russia in astronomical personnel</td>
</tr>
<tr>
<td>7 philtrum</td>
<td>including those from france and russia a space member</td>
</tr>
<tr>
<td>include</td>
<td>came from france and russia by cosmonauts</td>
</tr>
<tr>
<td>include</td>
<td>came from french and russia cosmonauts</td>
</tr>
<tr>
<td>includes</td>
<td>coming from french and russia ’s cosmonaut</td>
</tr>
<tr>
<td>french</td>
<td>and russia ’s astronauts</td>
</tr>
<tr>
<td>and russia</td>
<td>special rapporteur</td>
</tr>
<tr>
<td>and russia</td>
<td>rapporteur</td>
</tr>
<tr>
<td>and russia</td>
<td>rapporteur</td>
</tr>
<tr>
<td>or russia ’s</td>
<td></td>
</tr>
</tbody>
</table>
Phrase-Based Decoding

- Input
  - lo haré | rápidamente |

- Translations
  - I’ll do it | quickly |
  - quickly | I’ll do it |

- Decoding objective (for 3-gram LM)
  \[
  \arg \max_{\mathbf{e}} \left[ \prod_{\langle e, f \rangle} P(f|e) \cdot \prod_{i=1}^{|e|} P(e_i|e_{i-1}, e_{i-2}) \right]
  \]

The decoder...
tries different segmentations,
translates phrase by phrase,
and considers reorderings.

Slide credit: Dan Klein
Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

Mary did not give a slap to the witch green
did not a slap by green witch
no slap to the
did not give
to
__________________________ slap __________ the witch

\[ \text{arg max}_e \left[ \prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{\text{|e|}} P(e_i|e_{i-1}, e_{i-2}) \right] \]
If we translate with beam search, what state do we need to keep in the beam?

\[
\text{arg max}_e \left[ \prod_{\langle \overline{e}, \overline{f} \rangle} P(\overline{f}|\overline{e}) \cdot \prod_{i=1}^{\mid e \mid} P(e_i|e_{i-1}, e_{i-2}) \right]
\]
Monotonic Translation

If we translate with beam search, what state do we need to keep in the beam?

What have we translated so far?

$$\arg \max_e \left[ \prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{\left| e \right|} P(e_i|e_{i-1}, e_{i-2}) \right]$$
If we translate with beam search, what state do we need to keep in the beam?

What have we translated so far?

What words have we produced so far?

<table>
<thead>
<tr>
<th>Monotonic Translation</th>
</tr>
</thead>
</table>

- **Mary**
- **did not give a slap to the witch**
- **green witch**

\[
\text{arg max}_{e} \left[ \prod_{\langle \bar{e}, \bar{f} \rangle} P(\bar{f}|\bar{e}) \cdot \prod_{i=1}^{\text{|e|}} P(e_i|e_{i-1}, e_{i-2}) \right]
\]
If we translate with beam search, what state do we need to keep in the beam?

- What have we translated so far?
- What words have we produced so far?
- When using a 3-gram LM, only need to remember the last 2 words!
### Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not give a slap to the witch green</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not</td>
<td>a slap by green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no slap to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

This translation activity involves matching words from the top row to complete the sentence. The goal is to ensure that the sentence flows naturally and logically.

- **Maria** does not give a slap to the witch green.
- **Did not** give a slap by green witch.
- No slap to the.
- Did not give to the.
- Slap the witch.
Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
</table>

Mary did not give a slap to the witch green

...did not give a slap to the green witch

idx = 2

Mary not idx = 2

-1.2

Mary no idx = 2

-2.9

idx = 2

4.2
# Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td>a slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...did not

idx = 2

<table>
<thead>
<tr>
<th>Mary not</th>
<th>idx = 2</th>
</tr>
</thead>
</table>
| score = log [P(Mary) P(not | Mary) P(Mary | Maria) P(not | no)]

LM

TM
### Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td>a slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>slap</td>
<td>the</td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In reality:

\[
\text{score} = \alpha \log P(LM) + \beta \log P(TM)
\]

...and TM is broken down into several features

\[
\begin{align*}
\text{score} &= \log [P(Mary) P(\text{not} | \text{Mary}) P(Mary | \text{Maria}) P(\text{not} | \text{no})] \\
\end{align*}
\]
### Monotonic Translation

<table>
<thead>
<tr>
<th>María</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td>a slap</td>
<td>by</td>
<td>green witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>slap</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td>to the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td>the</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...not slap
idx = 5

...a slap
idx = 5

...no slap
idx = 5

<table>
<thead>
<tr>
<th>8.7</th>
<th>-2.4</th>
<th>-1.1</th>
</tr>
</thead>
</table>
### Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not give a slap to the witch green</td>
<td>did not a slap by green witch</td>
<td>no slap to the</td>
<td>did not give</td>
<td>slap to the</td>
<td>the witch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Several paths can get us to this state, max over them (like Viterbi)

<table>
<thead>
<tr>
<th>...not slap</th>
<th>idx = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td>...a slap</td>
<td>idx = 5</td>
</tr>
<tr>
<td>-2.4</td>
<td></td>
</tr>
<tr>
<td>...no slap</td>
<td>idx = 5</td>
</tr>
<tr>
<td>-1.1</td>
<td></td>
</tr>
</tbody>
</table>
### Monotonic Translation

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
<td></td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td></td>
<td>a slap</td>
<td>by</td>
<td>to the green witch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Several paths can get us to this state, max over them (like Viterbi)

**Scores**

- not give: idx = 3, score = 8.7
- not slap: idx = 5, score = -2.4
- a slap: idx = 5, score = -1.1
- no slap: idx = 5, score = -1.1
Monotonic Translation

Several paths can get us to this state, max over them (like Viterbi)
Monotonic Translation

- Several paths can get us to this state, max over them (like Viterbi)
- Variable-length translation pieces = semi-HMM
Non-Monotonic Translation

Non-monotonic translation: can visit source sentence “out of order”
Non-Monotonic Translation

Non-monotonic translation: can visit source sentence “out of order”

State needs to describe which words have been translated and which haven’t
Non-Monotonic Translation

Non-monotonic translation: can visit source sentence “out of order”

State needs to describe which words have been translated and which haven’t
Non-Monotonic Translation

- Non-monotonic translation: can visit source sentence “out of order”
- State needs to describe which words have been translated and which haven’t
Non-Monotonic Translation

- Non-monotonic translation: can visit source sentence “out of order”
- State needs to describe which words have been translated and which haven’t
- Big enough phrases already capture lots of reorderings, so this isn’t as important as you think
score = \alpha \log P(LM) + \beta \log P(TM)

...and TM is broken down into several features
score = $\alpha \log P(LM) + \beta \log P(TM)$

...and TM is broken down into several feature
score = α log P(LM) + β log P(TM)

...and TM is broken down into several feature

- Usually 5-20 feature weights to set, want to optimize for BLEU score which is not differentiable

- MERT (Och 2003): decode to get 1000-best translations for each sentence in a small training set (<1000 sentences), do line search on parameters to directly optimize for BLEU
Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn’s thesis
Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn’s thesis

- Moses implements word alignment, language models, and this decoder, plus *a ton* more stuff
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2013
Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn’s thesis

- Moses implements word alignment, language models, and this decoder, plus *a ton* more stuff
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2013

- Next time: results on these and comparisons to neural methods
Syntax
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

\[ NP \rightarrow [DT_1 \ JJ_2 \ NN_3; \ DT_1 \ NN_3 \ JJ_2] \]
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

\[ NP \rightarrow [DT_1 \ JJ_2 \ NN_3; \ DT_1 \ NN_3 \ JJ_2] \]
\[ DT \rightarrow [the, la] \]
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

\[
\text{NP} \rightarrow [\text{DT}_1 \text{JJ}_2 \text{NN}_3; \text{DT}_1 \text{NN}_3 \text{JJ}_2]
\]

\[
\text{DT} \rightarrow [\text{the, la}]
\]

\[
\text{DT} \rightarrow [\text{the, le}]
\]
Syntactic MT

- Rather than use phrases, use a synchronous context-free grammar

\[ NP \rightarrow [DT_1 \ JJ_2 \ NN_3; \ DT_1 \ NN_3 \ JJ_2] \]

\[ DT \rightarrow [the, \ la] \]

\[ DT \rightarrow [the, \ le] \]

\[ NN \rightarrow [car, \ voiture] \]
Rather than use phrases, use a *synchronous context-free grammar*

NP → [DT₁ JJ₂ NN₃; DT₁ NN₃ JJ₂]
DT → [the, la]
DT → [the, le]
NN → [car, voiture]
JJ → [yellow, jaune]
Rather than use phrases, use a *synchronous context-free grammar*

\[ NP \rightarrow [DT_1 \ JJ_2 \ NN_3; \ DT_1 \ NN_3 \ JJ_2] \]
\[ DT \rightarrow [the, la] \]
\[ DT \rightarrow [the, le] \]
\[ NN \rightarrow [car, voiture] \]
\[ JJ \rightarrow [yellow, jaune] \]
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

\[
\begin{align*}
\text{NP} &\rightarrow [\text{DT}_1 \text{JJ}_2 \text{NN}_3; \text{DT}_1 \text{NN}_3 \text{JJ}_2] \\
\text{DT} &\rightarrow [\text{the, la}] \\
\text{DT} &\rightarrow [\text{the, le}] \\
\text{NN} &\rightarrow [\text{car, voiture}] \\
\text{JJ} &\rightarrow [\text{yellow, jaune}]
\end{align*}
\]
Rather than use phrases, use a synchronous context-free grammar

NP → [DT₁ JJ₂ NN₃; DT₁ NN₃ JJ₂]

DT → [the, la]
DT → [the, le]
NN → [car, voiture]
JJ → [yellow, jaune]

the yellow car  la voiture jaune
Rather than use phrases, use a **synchronous context-free grammar**

\[
\begin{align*}
\text{NP} & \rightarrow [\text{DT}_1 \text{JJ}_2 \text{NN}_3; \text{DT}_1 \text{NN}_3 \text{JJ}_2] \\
\text{DT} & \rightarrow [\text{the, la}] \\
\text{DT} & \rightarrow [\text{the, le}] \\
\text{NN} & \rightarrow [\text{car, voiture}] \\
\text{JJ} & \rightarrow [\text{yellow, jaune}]
\end{align*}
\]

Translation = parse the input with "half" of the grammar, read off the other half
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

\[ NP \rightarrow [DT_1 \ JJ_2 \ NN_3; \ DT_1 \ NN_3 \ JJ_2] \]

\[ DT \rightarrow [the, \ la] \]

\[ DT \rightarrow [the, \ le] \]

\[ NN \rightarrow [car, \ voiture] \]

\[ JJ \rightarrow [yellow, \ jaune] \]

- Translation = parse the input with “half” of the grammar, read off the other half

- Assumes parallel syntax up to reordering
Use lexicalized rules, look like “syntactic phrases”

Leads to HUGE grammars, parsing is slow
Takeaways

- Phrase-based systems consist of 3 pieces: aligner, language model, decoder
  - HMMs work well for alignment
  - N-gram language models are scalable and historically worked well
  - Decoder requires searching through a complex state space
- Lots of system variants incorporating syntax
- Next time: neural MT