Phrase-based Statistical Machine Translation

Reza Haffari
gholamreza.haffari@monash.edu
References/Acknowledgment

• Readings:
  – J&M: Chapter 25
  – Kevin Knight’s MT workbook

• Some slides are adapted from:
  – Raymond Mooney, Phillip Koehn, Stephan Vogel, Jason Eisner, Aria Haghighi
Outline

• Basic Concepts

• Phrase-based SMT Models
  – Model Components
  – Word Alignment
  – Learning Phrase-Pairs

• Decoding

• Evaluation
Outline

• Basic Concepts
  • Phrase-based SMT Models
    – Model Components
    – Word Alignment
    – Learning Phrase-Pairs

• Decoding

• Evaluation
Machine Translation

- Automatically translate one language into another.

Mary didn’t slap the green witch.

Maria no dió una bofetada a la bruja verde.
Linguistic Issues Making MT Difficult

- Lexical Divergence
- Syntactical Divergence
- Morphology
Lexical Divergence

- Some words in one language do not have a corresponding term in the other.
  - Rivière (river that flows into ocean) and fleuve (river that does not flow into ocean) in French
Syntactical Divergence

- Syntactic variation between **SVO** (e.g. English), **SOV** (e.g. Hindi), and **VSO** (e.g. Arabic) languages.
  - SVO languages use prepositions
  - SOV languages use postpositions
Morphology

• Morphological issues with **polysynthetic**, **agglutinative**, and **fusional** languages with complex word structure
  - *Polysynthetic*: A single word may have many morphemes, corresponding to a complete English sentence!
  - *Agglutinative*: Clear morpheme boundaries (form words through the combination of smaller morphemes to express compound ideas)
  - *Fusional*: Morphemes boundaries not clear
Vauquois Triangle

Semantic structure

Semantic Parsing

Semantic Transfer

Syntactic structure

SRL & WSD

parsing

Direct translation

Tactical Generation

Interlingua

Source Language

Target Language

Words

Words

Syntactic transfer

Syntactic structure

Semantic structure

Semantic structure
Example: Direct Transfer

- **Morphological Analysis**
  - Mary didn’t slap the green witch. →
    Mary DO:PAST not slap the green witch.

- **Lexical Transfer**
  - Mary DO:PAST not slap the green witch.
    → Maria no dar:PAST una bofetada a la verde bruja.

- **Lexical Reordering**
  - Maria no dar:PAST una bofetada a la bruja verde.

- **Morphological generation**
  - Maria no dió una bofetada a la bruja verde.
Outline

• Basic Concepts

• Phrase-based SMT Models
  – Model Components
  – Word Alignment
  – Learning Phrase-Pairs

• Decoding

• Evaluation
Statistical MT

• Manually encoding comprehensive bilingual lexicons and transfer rules is difficult.

• SMT learns knowledge needed for translation from a parallel corpus or bitext that contains the same set of documents in two languages.
  • Eg the Canadian Hansards (parliamentary proceedings in French and English) is a well-known parallel corpus.
Picking a Good Translation

• A good translation should be:
  – *faithful*: correctly convey the information and tone of the original source sentence.
  – *fluent*: grammatically well structured and readable in the target language.

• A potential formalization:

\[ E_{\text{best}} = \arg\max_{E} \text{faithfulness}(E, F) \times \text{fluency}(E) \]
Bayesian Perspective

\[ \hat{E} = \arg\max_{E \in \text{English}} P(E | F) \]

\[ = \arg\max_{E \in \text{English}} \frac{P(F | E)P(E)}{P(F)} \]

Translation Model    Language Model

A decoder determines the most probable translation \( \hat{E} \) given \( F \)

(Translating from a foreign language \( F \) into English \( E \))
The Noisy Channel Model

Want to recover $E$ from $F$
Choose $E$ that maximizes $P(E|F)$
Language Model

• Use a standard $n$-gram language model for $P(E)$
  – Can be trained on a large mono-lingual corpus for the target language $E$
Phrase-Based Translation Model

• Base $P(F \mid E)$ on translating phrases in $E$ to phrases in $F$.
  1. Segment $E$ into a sequence of phrases $\bar{e}_1, \bar{e}_1, \ldots, \bar{e}_I$
  2. Translate each phrase $\bar{e}_i$, into $f_i$, based on translation probability $\phi(f_i \mid \bar{e}_i)$
  3. Reorder translated phrases based on distortion probability $d(i)$ for the $i$th phrase.

$$P(F \mid E) = \prod_{i=1}^{I} \varphi(f_i \mid \bar{e}_i) d(i)$$

<table>
<thead>
<tr>
<th>Position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Mary</td>
<td>did not</td>
<td>slap</td>
<td>the</td>
<td>green</td>
<td>witch</td>
</tr>
<tr>
<td>Spanish</td>
<td>Maria</td>
<td>no</td>
<td>dió una bofetada a</td>
<td>la</td>
<td>bruja</td>
<td>verde</td>
</tr>
</tbody>
</table>
Translation Probabilities

• Assuming a *phrase aligned* parallel corpus is available or constructed that shows matching between phrases in $E$ and $F$

• Then compute (MLE) estimate of $\varphi$ based on simple frequency counts

$$\varphi(f | e) = \frac{\text{count}(f, e)}{\sum_{\bar{f}} \text{count}(f, e)}$$
Distortion Probability

• Measure distortion of phrase $i$ as the distance between
  – the start of the $F$ phrase generated by $\bar{e}_i$, denoted by $a_i$
  – the end of the $F$ phrase generated by the previous phrase $\bar{e}_{i-1}$, denoted by $b_{i-1}$

• Typically assume the probability of a distortion decreases exponentially with the distance of the movement: $d(i) = c \alpha^{|a_i - b_{i-1}|}$

  Set $0 < \alpha < 1$ based on fit to phrase-aligned training data
  Then set $c$ to normalize $d(i)$ so it sums to 1.
## Sample Translation Model

<table>
<thead>
<tr>
<th>Position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>English</strong></td>
<td></td>
<td>Mary</td>
<td>did not</td>
<td>slap</td>
<td>the</td>
<td>green witch</td>
</tr>
<tr>
<td><strong>Spanish</strong></td>
<td></td>
<td>Maria</td>
<td>no</td>
<td>dió una bofetada a</td>
<td>la</td>
<td>bruja verde</td>
</tr>
</tbody>
</table>

$|a_i - b_{i-1}| = 1, 1, 1, 1, 2, 1$

\[
p(F \mid E) = \varphi(Maria \mid Mary) c\alpha^1 \varphi(no \mid did\ not) c\alpha^1 \varphi(dio\ una\ bofetada\ a \mid slap) c\alpha^1 \\
\varphi(la \mid the) c\alpha^1 \varphi(verte \mid green) c\alpha^2 \varphi(bruja \mid witch) c\alpha^1
\]
Challenges in Phrase-based SMT

• **Learning** the translation model $P(F|E)$:
  – How to induce phrase-alignments to learn $\varphi(f|e)$
  – Need to induce word alignments (will see shortly), then extract the phrases.

• **Decoding** $E$ based on $P(F|E)$ and $P(E)$:
  – How to find a translation that maximizes the product of the scores of translation model and language model

\[
\arg\max_{E \in \text{English}} P(F \mid E)P(E)
\]
Outline

- Basic Concepts

- Phrase-based SMT Models
  - Model Components
  - Word Alignment
  - Learning Phrase-Pairs

- Decoding

- Evaluation
Word Alignment

• Shows mapping between words in one language and the other.

Mary didn’t slap the green witch.

Maria no dió una bofetada a la bruja verde.
Word Alignment

• Directly constructing *phrase alignments* is difficult, so rely on first constructing *word alignments*
  – Can learn to align from supervised word alignments, but human-aligned bitexts are rare and expensive to construct.

• Typically use an unsupervised EM-based approach to compute a word alignment from unannotated parallel corpus.
One to Many Alignment

• To simplify the problem, typically assume:
  – Each word in $F$ aligns to 1 word in $E$ (but assume each word in $E$ may generate more than one word in $F$).
  – Some words in $F$ may be generated by the NULL element of $E$.

• Hence, an alignment can be specified by a vector $A$ giving, for each word in $F$, the index of the word in $E$ which generated it.

Mary didn’t slap the green witch.

Maria no dió una bofetada a la bruja verde.
IBM Model 1

• Assumess following simple generative model of producing $F$ from $E=e_1, e_2, \ldots e_I$
  – Choose length, $J$, of $F$ sentence: $F=f_1, f_2, \ldots f_J$
  – Choose a 1 to many alignment $A=a_1, a_2, \ldots a_J$
  – For each position in $F$, generate a word $f_j$ from the aligned word in $E$: $e_{a_j}$
Sample IBM Model 1 Generation

Mary didn’t slap the green witch.

María no dio una bofetada a la bruja verde.
Computing $P(F \mid E)$ in IBM Model 1

- Assume some length distribution $P(J \mid E)$
- Assume all alignments are equally likely. Since there are $(I + 1)^J$ possible alignments:
  \[
P(A \mid E) = P(A \mid E, J)P(J \mid E) = \frac{P(J \mid E)}{(I + 1)^J}
  \]
- Assume $t(f_x, e_y)$ is the probability of translating $e_y$ as $f_x$, therefore:
  \[
P(F \mid E, A) = \prod_{j=1}^{J} t(f_j, e_{a_j})
  \]
- Determine $P(F \mid E)$ by summing over all alignments:
  \[
P(F \mid E) = \sum_{A} P(F \mid E, A)P(A \mid E) = \sum_{A} \frac{P(J \mid E)}{(I + 1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j})
  \]
Decoding for IBM Model 1

- Goal is to find the most probable alignment given a parameterized model.

\[ \hat{A} = \arg\max_{A} P(F, A | E) \]
\[ = \arg\max_{A} \frac{P(J | E)}{(I + 1)^J} \prod_{j=1}^{J} t(f_j, e_{a_j}) \]
\[ = \arg\max_{A} \prod_{j=1}^{J} t(f_j, e_{a_j}) \]

Since translation choice for each position \( j \) is independent, the product is maximized by maximizing each term:

\[ a_j = \arg\max_{0 \leq i \leq I} t(f_j, e_i) \quad 1 \leq j \leq J \]
Training Word Alignment Models

• IBM model 1 can be trained on a parallel corpus to set the required parameters.
  – For supervised (hand-aligned) training data, parameters can be estimated directly using frequency counts.
  – For unsupervised training data, the EM algorithm can be used to estimate parameters.
Sketch of EM Algorithm for Word Alignment

Randomly set model parameters. 
(making sure they represent legal distributions)

Until converge (i.e. parameters no longer change) do:

**E Step:** Compute the probability of all possible alignments of the training data using the current model.

**M Step:** Use these alignment probability estimates to re-estimate values for all of the parameters.

*Note: Use dynamic programming to avoid explicitly enumerating all possible alignments*
### Sample EM Trace for Alignment

(IBM Model 1 with no NULL Generation)

<table>
<thead>
<tr>
<th>Training Corpus</th>
<th>green house</th>
<th>casa verde</th>
<th>the house</th>
<th>la casa</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Translation Probabilities</th>
<th>verde</th>
<th>casa</th>
<th>la</th>
</tr>
</thead>
<tbody>
<tr>
<td>green house</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>casa verde</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>the house</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
</tr>
</tbody>
</table>

Assume uniform initial probabilities

**Compute Alignment Probabilities**

| P(A, F | E)   | green house | green house | the house | the house |
|--------|------------|-------------|-----------|-----------|
|        | casa verde | casa verde  | la casa   | la casa   |
| 1/3 X 1/3 = 1/9 | 1/3 X 1/3 = 1/9 | 1/3 X 1/3 = 1/9 | 1/3 X 1/3 = 1/9 |

**Normalize to get**

| P(A | F, E) | 1/9 = 1 | 1/9 = 1 | 1/9 = 1 | 1/9 = 1 |
|--------|---------|---------|---------|---------|
|        | 2/9 = 2 | 2/9 = 2 | 2/9 = 2 | 2/9 = 2 |
Example cont.

<table>
<thead>
<tr>
<th>green house</th>
<th>green house</th>
<th>the house</th>
<th>the house</th>
</tr>
</thead>
<tbody>
<tr>
<td>casa verde</td>
<td>casa verde</td>
<td>la casa</td>
<td>la casa</td>
</tr>
<tr>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>verde</th>
<th>casa</th>
<th>la</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>house</td>
<td>1/2</td>
<td>1/2 + 1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>the</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Compute weighted translation counts

Normalize rows to sum to one to estimate $P(f | e)$
Example cont.

<table>
<thead>
<tr>
<th>Translation Probabilities</th>
<th>verde</th>
<th>casa</th>
<th>la</th>
</tr>
</thead>
<tbody>
<tr>
<td>green house</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Recompute
Alignment Probabilities

\[
\begin{align*}
\text{green house} & \quad \text{green house} & \quad \text{the house} & \quad \text{the house} \\
\text{casa verde} & \quad \text{casa verde} & \quad \text{la casa} & \quad \text{la casa} \\
\end{align*}
\]

\[
\begin{align*}
\frac{1}{2} \times \frac{1}{4} &= \frac{1}{8} \\
\frac{1}{2} \times \frac{1}{2} &= \frac{1}{4} \\
\frac{1}{2} \times \frac{1}{2} &= \frac{1}{4} \\
\frac{1}{2} \times \frac{1}{4} &= \frac{1}{8} \\
\end{align*}
\]

Normalize to get

\[
\begin{align*}
\frac{1}{8} &= \frac{1}{3} \\
\frac{1}{4} &= \frac{2}{3} \\
\frac{1}{4} &= \frac{2}{3} \\
\frac{1}{8} &= \frac{1}{3} \\
\end{align*}
\]

Continue EM iterations until translation parameters converge
Outline

• Basic Concepts

• Phrase-based SMT Models
  – Model Components
  – Word Alignment
  – Learning Phrase-Pairs

• Decoding

• Evaluation
Phrase Alignments from Word Alignments in a Nutshell

• Recall that we have estimated word alignments to be able to extract phrase-pairs.

• However, alignment algorithms produce one to many word translations rather than many to many phrase translations.

• Combine $E \rightarrow F$ and $F \rightarrow E$ word alignments to produce a phrase alignment.
Phrase Pairs from Viterbi Path

• Train your favorite word alignment
  – IBM Model 1, HMM, …

• Calculate Viterbi path (i.e. path with highest probability or best score)

• Read-off the phrase-pairs
Word Alignment Matrix

- Alignment probabilities according to lexicon
Viterbi Path

- Calculate Viterbi path (i.e. path with highest probability)
Phrases from Viterbi Path

- Read off source phrase – target phrase pairs
Over-generation

- Extract all $n:m$ blocks (phrase-pairs) which:
  - Have at least one link inside
  - No crossing links outside (i.e. in same rows and columns)
- Will extract many overlapping phrase-pairs
  - Note: not all possible blocks have been shown
Dealing with Asymmetry

• Word alignment models are asymmetric
  – Viterbi path has:
    multiple source words – one target word alignments

• Train alignment model in reverse direction as well

• Using both Viterbi paths:
  – Simply extract phrases from both directions and merge them
  – Merge Viterbi paths and extract phrase pairs according to resulting pattern
Combine Viterbi Paths

\[ e_i \]

\[ f_1 \]

\[ f_J \]

- F->E
- E->F
- Intersect.
Combine Viterbi Paths

- Viterbi path combination methods:
  - Intersections: high precision, but low recall
  - Union: lower precision, but higher recall

- Quality of phrase translation pairs depends on:
  - Quality of word alignment
  - Quality of combination of Viterbi paths
Translation Probabilities

- After extracting the phrase-pairs, the maximum likelihood estimate of the translation probabilities is based on simple frequency counts:

\[
\phi(f, e) = \frac{\text{count}(f, e)}{\sum_{\tilde{f}} \text{count}(\tilde{f}, \tilde{e})}
\]

- Various smoothing techniques have been investigated.
Recap: Learning Phrase-based SMT Models

Sentence-aligned corpus

Directional word alignments

Intersected and grown word alignments

Phrase table (translation model)
Phrase Alignment as Sentence Splitting

- A radically different approach to phrase alignment is to directly define a **phrase-level generative story**

- This approach is not covered here
Outline

• Basic Concepts

• Phrase-based SMT Models
  – Model Components
  – Word Alignment
  – Learning Phrase-Pairs

• Decoding

• Evaluation
Decoding

• Goal is to find a translation that maximizes the product of the translation and language models

\[
\text{argmax}_{E \in \text{English}} P(F | E)P(E)
\]

• Cannot explicitly enumerate and test the combinatorial space of all possible translations.
  - The optimal decoding problem for all reasonable model’s is NP-complete
  - Must efficiently (heuristically) search the space of translations that approximates the solution to this difficult optimization problem
Space of Translations

• The phrase translation defines the space of all possible translations

<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></td>
<td>to</td>
<td>the</td>
<td>green witch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td></td>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td>to</td>
<td>the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slap</td>
<td></td>
<td></td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the witch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stack Decoding (Pharaoh/Moses)

- Use a version of heuristic A* search to explore the space of phrase translations to find the best scoring subset that covers the source sentence.
Stack Decoding (Pharaoh/Moses)

1. Initialize priority queue $Q$ (stack) to empty translation.
2. Loop:
   2.1 $s = \text{pop}(Q)$
   2.2 If $s$ is a complete translation, exit loop and return it.
   2.3 For each refinement $s'$ of $s$ created by adding a phrase translation
      2.3.1 Compute score $f(s')$
      2.3.2 Add $s'$ to $Q$  // $Q$ is sorted by score $f$
A* is best-first search using the function $f$ to sort the search queue:

- $f(s) = g(s) + h(s)$
- $g(s)$: Cost of existing partial solution
- $h(s)$: Estimated cost of completion of solution

If $h(s)$ is an underestimate of the true remaining cost (admissible heuristic) then A* is guaranteed to return an optimal solution.
Current Cost: $g(s)$

- Known quality of partial translation, $E$
  - Selected phrase translations
  - Distortion
  - Language model

$$g(s) = \log \left( \frac{1}{\prod_{i \in S} \varphi(\bar{f}_i, \bar{e}_i) d(i)} \right) P(E)$$
Estimated Future Cost: $h(s)$

- $h(s)$ requires knowing the optimal way of translating the remainder of the sentence.
- But this is not computationally tractable.
- Therefore under-estimate the cost of remaining translation:
  - Computing the most probable remaining translation ignoring distortion
  - Efficiently computable using the Viterbi algorithm
Beam Search

• But $Q$ grows too large to be efficient and guarantee an optimal result with full A* search.
• So always cut $Q$ back to only the best (lowest cost) $K$ items to approximate the best translation

1. Initialize priority queue $Q$ (stack) to empty translation.
2. Loop:
   2.1 $s = \text{pop}(Q)$
   2.2 If $s$ is a complete translation, exit loop and return it.
   2.3 For each refinement $s'$ of $s$ created by adding a phrase translation
      2.3.1 Compute score $f(s')$
      2.3.2 Add $s'$ to $Q$ \(\text{// } Q \text{ is sorted by score } f\)
3. Prune $Q$ back to only the first (lowest cost) $K$ items
Multistack Decoding

- Translations that cover different fractions of the foreign sentence are not on the same scale to compare.

- So maintain multiple priority queues (stacks), one for each number of foreign words currently translated.
  - Finally, return best scoring translation in the queue of translations that cover all of the words in $F$. 
Outline

• Basic Concepts

• Phrase-based SMT Models
  – Model Components
  – Word Alignment
  – Learning Phrase-Pairs

• Decoding

• Evaluation
Evaluating MT

• Human evaluation is the best but is time-consuming, expensive, and subjective.

• Automated evaluation:
  – Collect one or more human reference translations of the source.
  – Compare MT output to these reference translations.
  – Score result based on similarity to the reference translations (BLEU, METEOR etc)
BLEU

- Determine number of \( n \)-grams of various sizes that the MT output shares with the reference translations.

- Compute a modified precision measure of the \( n \)-grams in MT result.
BLEU Example

Cand 1: Mary no slap the witch\textcolor{green}{green}
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not \textcolor{green}{slap the green witch}.
Ref 2: Mary did not smack \textcolor{green}{the green witch}.
Ref 3: Mary did not hit a \textcolor{green}{green} sorceress.

Cand 1 Unigram Precision: \textcolor{green}{5/6}
BLEU Example

Cand 1: Mary \textcolor{red}{no} slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not \textcolor{red}{slap the} green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Cand 1 Bigram Precision: 1/5
BLEU Example

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.

Cand 2 Unigram Precision: 7/10
BLEU Example

Cand 1: Mary no slap the witch green.
Cand 2: Mary did not give a smack to a green witch.
Cand 2 Bigram Precision: 4/9

Ref 1: Mary did not slap the green witch.
Ref 2: Mary did not smack the green witch.
Ref 3: Mary did not hit a green sorceress.
Modified \( n \)-gram Precision

- Average \( n \)-gram precision over all \( n \)-grams up to size \( N \) (typically 4) using geometric mean:

\[
p = \sqrt[N]{\prod_{n=1}^{N} p_n}
\]

Cand 1: \( p = \sqrt[2]{\frac{5}{6}} \times \frac{1}{5} = 0.408 \)

Cand 2: \( p = \sqrt[2]{\frac{7}{10}} \times \frac{4}{9} = 0.558 \)
Brevity Penalty

• Average precision is biased towards short translations
  – Penalize translations that are shorter than the reference

• Define effective reference length, $r$, for each sentence as the length of the reference sentence with the largest number of $n$-gram matches. Let $c$ be the candidate sentence length.

$$BP = \begin{cases} 
1 & \text{if } c > r \\
\exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r
\end{cases}$$
BLEU Score

• BLEU Score: $BLEU = BreavityPenalty \times p$

Cand 1: Mary no slap the witch green.

Best Ref: Mary did not slap the green witch.

$$c = 6, \quad r = 7, \quad BP = e^{(1-7/6)} = 0.846$$

$$BLEU = 0.846 \times 0.408 = 0.345$$

Cand 2: Mary did not give a smack to a green witch.

Best Ref: Mary did not smack the green witch.

$$c = 10, \quad r = 7, \quad BP = 1$$

$$BLEU = 1 \times 0.558 = 0.558$$
Summary of Part 1

• Statistical MT methods can automatically learn a translation system from a parallel corpus.
  – Word alignment
  – Phrase-pair extraction
  – Language Model
  – Distortion Model

• Decoding is usually done by A* (and its variants)

• Next: Log-linear models to combines various features, discriminative training, etc