CS5740: Natural Language Processing
Spring 2017

Phrase-based Translation

Instructor: Yoav Artzi

Slides adapted from Michael Collins and Yejin Choi

Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based model
- MT evaluation

Phrase-based Models

- First stage in training a phrase-based (PB) model is extraction of PB lexicon
- A PB lexicon pairs strings in one language with string in another language, e.g.,

nach Kanada	$\leftrightarrow$	in Canada
zur Konferenz	$\leftrightarrow$	to the conference
Morgen	$\leftrightarrow$	tomorrow
fliege	$\leftrightarrow$	will fly
...		

An Example

- A training example:

  Spanish: Maria no daba una bofetada a la bruja verde

  English: Mary did not slap the green witch

- Some (not all) phrase pairs extracted from this example:

  (Maria $\leftrightarrow$ Mary), (bruja $\leftrightarrow$ witch), (verde $\leftrightarrow$ green),
  (no $\leftrightarrow$ did not), (no daba una bofetada $\leftrightarrow$ did not slap),
  (daba una bofetada a la $\leftrightarrow$ slap the)

- We will see how to do this using alignments from IBM models (e.g., IBM Model 2)

[From tutorial by Koehn and Knight]

Recap: IBM Model 2

- IBM Model 2 defines a distribution $p(a, f \mid e, m)$ where f is a target (French) sentence, e is an source (English) sentence, a is an alignment, m is the length of the foreign sentence
- A useful by-product: for any pair (f, e), can calculate
$$a^* = \arg\max_a p(a|f, e, m) = \arg\max_a p(a, f|e, m)$$
where a^* is the most likely alignment

English: Mary did not slap the green witch

Spanish: Maria no daba una bofetada a la bruja verde

Recap: IBM Model 2

- IBM Model 2 defines a distribution $p(a, f \mid e, m)$ where f is a target (French) sentence, e is an source (English) sentence, a is an alignment, m is the length of the foreign sentence
- A useful by-product: for any pair (f, e), can calculate

$$a^* = \arg\max_a p(a|f, e, m) = \arg\max_a p(a, f|e, m)$$

where a^* is the most likely alignment

English: Mary did not slap the green witch

Spanish: Maria no daba una bofetada a la bruja verde

Representation as Alignment Matrix

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

bof' = bofetada

- In IBM Model 2, each target (Spanish) word is aligned to exactly one English word. The matrix shows these alignments.

Finding Alignment Matrices

- Step 1: train IBM Model 2 for $p(f|e)$, and find the most likely alignment for each (e, f) pair
- Step 2: train IBM Model 2 for $p(e|f)$, and find the most likely alignment for each (e, f) pair
- Given the two alignments, take the intersection of the two as a starting point

Alignment from $p(f \mid e)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

Alignment from $p(e \mid f)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap					●				
the							●		
green									●
witch								●	

Intersection of the two alignments:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did									
not		●							
slap					●				
the							●		
green									●
witch								●	

The intersection of the two alignments has been found to be a very reliable starting point

Heuristics for Growing Alignments

- Only explore alignment in **union** of $p(f|e)$ and $p(e|f)$ alignments
- Add one alignment point at a time
- Only add alignment points which align a word that currently has no alignment
- At first, restrict to alignment points that are "neighbors" (adjacent or diagonal) of current alignment points
- Later, consider other alignment points

The final alignment, created by taking the intersection of the two alignments, then adding new points using the growing heuristics:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Note that the alignment is no longer many-to-one: potentially multiple Spanish words can be aligned to a single English word, and vice versa.

Extracting Phrase Pairs from the Alignment Matrix

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

- A phrase-pair consists of a sequence of source (English) words, e, paired with a sequence of target (French) words, f
- A phrase-pair (e, f) is **consistent** if:
  - There is at least one word in e aligned to a word in f
  - There are no words in f aligned to words outside e
  - There are no words in e aligned to words outside f
- Extract all consistent phrase pairs from the training example

Extracting Phrase Pairs from the Alignment Matrix

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

- A phrase-pair consists of a sequence of source (English) words, e, paired with a sequence of target (French) words, f
- A phrase-pair (e, f) is **consistent** if:
  - There is at least one word in e aligned to a word in f
  - There are no words in f aligned to words outside e
  - There are no words in e aligned to words outside f
- Extract all consistent phrase pairs from the training example

(Maria, Mary)
(no, did not)
(Maria no, Mary did not)
X (no daba, did not slap)
(no daba una bof', did not slap)
(daba una bof', slap)
(a la, the)
(verde, green)
(bruja, witch)
(bruja verde, green witch)
X (la bruja verde ,the green witch)

Probabilities for Phrase Pairs

- For any phrase pair *(f,e)* extracted from the training data, can calculate:

$$t(f|e) = \frac{\text{count(f, e)}}{\text{count(e)}}$$

- For example:

$$t(\text{daba una bofetada}|\text{slap}) = \frac{\text{count(daba una bofetada, slap)}}{\text{count(slap)}}$$

- Probabilistic model?

Example Phrase Translation Table

An example from Koehn, EACL 2006 tutorial. (Note that we have $t(e|f)$ not $t(f|e)$ in this example.)

- Phrase Translations for *den Vorschlag*

English	t(e|f)	**English**	t(e|f)
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based model
- MT evaluation

Phrase-Based Systems: A Sketch

$\boxed{\text{Today}}$

$\boxed{\text{Heute}}$ werden wir uber die Wiedereroffnung des Mont-Blanc-Tunnels diskutieren

$$\text{Score} = \underbrace{\log q(\text{Today} \mid *, *)}_{\text{Language model}}$$

$$+ \underbrace{\log t(\text{Heute} \mid \text{Today})}_{\text{Phrase model}}$$

$$+ \underbrace{\eta \times 0}_{\text{Distortion model}}$$

Phrase-Based Systems: A Sketch

Today $\boxed{\text{we shall be}}$

Heute $\boxed{\text{werden wir}}$ uber die Wiedereroffnung
des Mont-Blanc-Tunnels diskutieren

$$\text{Score} \quad = \quad \underbrace{\log q(\text{we}|\text{*, Today}) + \log q(\text{shall}|\text{Today, we}) + \log q(\text{be}|\text{we, shall})}_{\text{Language model}}$$

$$+ \quad \underbrace{\log t(\text{werden wir} \mid \text{we shall be})}_{\text{Phrase model}}$$

$$+ \quad \underbrace{\eta \times 0}_{\text{Distortion model}}$$

Phrase-Based Systems: A Sketch

Today we shall be $\boxed{\text{debating}}$
Heute werden wir uber die Wiedereroffnung
des Mont-Blanc-Tunnels $\boxed{\text{diskutieren}}$

$$
\begin{aligned}
\text{Score} \quad = & \quad \underbrace{\log q(\text{debating}|\text{shall, be})}_{\text{Language model}} \\
+ & \quad \underbrace{\log t(\text{diskutieren} \mid \text{debating})}_{\text{Phrase model}} \\
+ & \quad \underbrace{\eta \times 6}_{\text{Distortion model}}
\end{aligned}
$$

Phrase-Based Systems: A Sketch

Today we shall be debating $\boxed{\text{the reopening}}$

Heute werden wir $\boxed{\text{uber die Wiedereroffnung}}$
des Mont-Blanc-Tunnels diskutieren

Phrase-Based Systems: A Sketch

Today we shall be debating the reopening
$\boxed{\text{of the Mont Blanc tunnel}}$

Heute werden wir uber die Wiedereroffnung
$\boxed{\text{des Mont-Blanc-Tunnels}}$ diskutieren

Phrase-Based Systems: A Sketch

Today we shall be debating the reopening
$\boxed{\text{of the Mont Blanc tunnel}}$

Heute werden wir uber die Wiedereroffnung
$\boxed{\text{des Mont-Blanc-Tunnels}}$ diskutieren

Key problem?

Each choice
- Language model score
- Phrase score
- Distortion score

Search the space of choices

Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based model
- MT evaluation

Phrase-based Translation

An example sentence:

wir müssen auch diese kritik ernst nehmen

A phrase-based lexicon contains phrase entries (f, e) where f is a sequence of one or more foreign words, e is a sequence of one or more English words. Example phrase entries that are relevant to our example:

(wir müssen, we must)

(wir müssen auch, we must also)

(ernst, seriously)

Each phrase (f, e) has a score $g(f, e)$. E.g.,

$$g(f, e) = \log \left(\frac{\text{Count}(f, e)}{\text{Count(e)}} \right)$$

Definitions

- A phrase-based model consists of:
  1. A phrase-based lexicon, consisting of entries (f, e) such as

     $$(\text{wir müssen, we must})$$

     Each lexical entry has a score $g(f, e)$, e.g.,

     $$g(\text{wir müssen, we must}) = \log\left(\frac{\text{Count(wir müssen, we must)}}{\text{Count(we must)}}\right)$$

  2. A trigram language model, with parameters $q(w|u, v)$. E.g., $q(\text{also}|\text{we, must})$.
  3. A "distortion parameter" η (typically negative).

Definitions

An example sentence:

wir müssen auch diese kritik ernst nehmen

- For a particular input (source-language) sentence $x_1 \dots x_n$, a phrase is a tuple (s, t, e), signifying that the subsequence $x_s \dots x_t$ in the source language sentence can be translated as the target-language string e, using an entry from the phrase-based lexicon. E.g., $(1, 2, \text{we must})$
- $\mathcal{P}$ is the set of all phrases for a sentence.
- For any phrase p, $s(p)$, $t(p)$ and $e(p)$ are its three components. $g(p)$ is the score for a phrase.

Definitions

- A derivation y is a finite sequence of phrases, $p_1, p_2, \ldots p_L$, where each p_j for $j \in \{1 \ldots L\}$ is a member of $\mathcal{P}$.
- The length L can be any positive integer value.
- For any derivation y we use $e(y)$ to refer to the underlying translation defined by y. E.g.,

  $y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

  and

  $e(y) =$ we must also take this criticism seriously

Valid Derivations

- For an input sentence $x = x_1 \ldots x_n$, we use $\mathcal{Y}(x)$ to refer to the set of valid derivations for x.
- $\mathcal{Y}(x)$ is the set of all finite length sequences of phrases $p_1 p_2 \ldots p_L$ such that:
  - Each p_k for $k \in \{1 \ldots L\}$ is a member of the set of phrases $\mathcal{P}$ for $x_1 \ldots x_n$.
  - Each word in x is translated exactly once.
  - For all $k \in \{1 \ldots (L-1)\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where $d \geq 0$ is a parameter of the model. In addition, we must have $|1 - s(p_1)| \leq d$

Examples

Distortion limit = 4

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

$y =$ (1, 2, we must), (7, 7, take), (3, 3, also), (4, 5, this criticism), (6, 6, seriously)

Examples

Distortion limit = 4

wir müssen auch diese kritik ernst nehmen

V $y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

X $y =$ (1, 3, we must also), (1, 2, we must), (4, 5, this criticism), (6, 6, seriously)

X $y =$ (1, 2, we must), (7, 7, take), (3, 3, also), (4, 5, this criticism), (6, 6, seriously)

Valid Derivations

- For an input sentence $x = x_1 \ldots x_n$, we use $\mathcal{Y}(x)$ to refer to the set of valid derivations for x.
- $\mathcal{Y}(x)$ is the set of all finite length sequences of phrases $p_1 p_2 \ldots p_L$ such that:
  - Each p_k for $k \in \{1 \ldots L\}$ is a member of the set of phrases $\mathcal{P}$ for $x_1 \ldots x_n$.
  - Each word in x is translated exactly once.
  - For all $k \in \{1 \ldots (L-1)\}$, $|t(p_k) + 1 - s(p_{k+1})| \leq d$ where $d \geq 0$ is a parameter of the model. In addition, we must have $|1 - s(p_1)| \leq d$

How many valid derivation exist?

Scoring Derivations

The optimal translation under the model for a source-language sentence x will be

$$\arg\max_{y \in \mathcal{Y}(x)} f(y)$$

In phrase-based systems, the score for any derivation y is calculated as follows:

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

where the parameter η is the distortion penalty (typically negative). (We define $t(p_0) = 0$).

$h(e(y))$ is the trigram language model score. $g(p_k)$ is the phrase-based score for p_k.

Example

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

Example

$$h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$$\log p(\text{we} \mid *, *) + \log p(\text{must} \mid \text{we}, *) + \log p(\text{also} \mid \text{must}, \text{we}) +$$
$$\log p(\text{take} \mid \text{also}, \text{must}) \cdots + \log p(\text{seriously} \mid \text{criticism}, \text{this}) +$$
$$g(1, 3, \text{we must also}) + g(7, 7, \text{take}) + g(4, 5, \text{this criticism}) + g(6, 6, \text{seriously}) +$$
$$\eta|0 + 1 - 1| + \eta|3 + 1 - 7| + \eta|7 + 1 - 4| + \eta|5 + 1 - 6|$$

Decoding Algorithm: Definitions

- A state is a tuple

$$(e_1, e_2, b, r, \alpha)$$

where e_1, e_2 are English words, b is a bit-string of length n, r is an integer specifying the end-point of the last phrase in the state, and α is the score for the state.

- The initial state is

$$q_0 = (*, *, 0^n, 0, 0)$$

where 0^n is bit-string of length n, with n zeroes.

State Length: $len(q)$

- Given a state q, $len(q)$ is the number of words translated
  - The number of 1's in the bitmask b

States and the Search Space

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$$(*, *, 0000000, 0, 0)$$

States and the Search Space

wir müssen auch diese kritik ernst nehmen

$y =$ (1, 3, we must also), (7, 7, take), (4, 5, this criticism), (6, 6, seriously)

$$(*, *, 0000000, 0, 0) \rightarrow (\text{must}, \text{also}, 1110000, 3, ?) \rightarrow$$
$$(\text{also}, \text{take}, 1110001, 7, ?) \rightarrow (\text{this}, \text{criticism}, 1111101, 5, ?) \rightarrow$$
$$(\text{criticism}, \text{seriously}, 1111111, 6, ?)$$

Transitions

- We have $ph(q)$ for any state q, which returns set of phrases that are allowed to follow state $q = (e_1, e_2, b, r, \alpha)$.
- For a phrase p to be a member of $ph(q)$, it must satisfy the following conditions:
  - p must not overlap with the bit-string b. I.e., we need $b_i = 0$ for $i \in \{s(p) \dots t(p)\}$.
  - The distortion limit must not be violated. More formally, we must have $|r + 1 - s(p)| \leq d$ where d is the distortion limit.

Transition Function: Example

wir müssen auch diese kritik ernst nehmen

$(\textsf{must, also}, 1110000, 3, -2.5)$

X $(3, 3, \text{also})$

X $(1, 2, \text{we must})$

V $(6, 6, \text{seriously})$

V $(4, 5, \text{this criticism})$

V $(5, 6, \text{criticism seriously})$

V $(5, 5, \text{review})$

Transition Function: Example

wir müssen auch diese kritik ernst nehmen

(must, also, $1110000, 3, -2.5$)

$$(6, 6, \text{seriously})$$
$$(4, 5, \text{this criticism})$$
$$(5, 6, \text{criticism seriously})$$
$$(5, 5, \text{review})$$

In addition, we define $next(q, p)$ to be the state formed by combining state q with phrase p.

The *next* function

Formally, if $q = (e_1, e_2, b, r, \alpha)$, and $p = (s, t, \epsilon_1 \ldots \epsilon_M)$, then next$(q, p)$ is the state $q' = (e'_1, e'_2, b', r', \alpha')$ defined as follows:

- First, for convenience, define $\epsilon_{-1} = e_1$, and $\epsilon_0 = e_2$.
- Define $e'_1 = \epsilon_{M-1}$, $e'_2 = \epsilon_M$.
- Define $b'_i = 1$ for $i \in \{s \ldots t\}$. Define $b'_i = b_i$ for $i \notin \{s \ldots t\}$
- Define $r' = t$
- Define

$$\alpha' = \alpha + g(p) + \sum_{i=1}^{M} \log q(\epsilon_i | \epsilon_{i-2}, \epsilon_{i-1}) + \eta \times |r + 1 - s|$$

$$\text{next}((\text{must}, \text{also}, 1110000, 3, ?), (7, 7, \text{take})) = (\text{also}, \text{take}, 1110001, 7, ?)$$

The Equality Function

- The function

$$\mathsf{eq}(q, q')$$

returns true or false.

- Assuming $q = (e_1, e_2, b, r, \alpha)$, and $q' = (e'_1, e'_2, b', r', \alpha')$, $eq(q, q')$ is true if and only if $e_1 = e'_1$, $e_2 = e'_2$, $b = b'$ and $r = r'$.

The Decoding Algorithm

- Inputs: sentence $x_1 \dots x_n$. Phrase-based model $(\mathcal{L}, h, d, \eta)$. The phrase-based model defines the functions $ph(q)$ and $\text{next}(q, p)$.
- Initialization: set $Q_0 = \{q_0\}$, $Q_i = \emptyset$ for $i = 1 \dots n$.
- For $i = 0 \dots n - 1$
  - For each state $q \in \text{beam}(Q_i)$, for each phrase $p \in ph(q)$:
    (1) $q' = \text{next}(q, p)$
    (2) $\text{Add}(Q_i, q', q, p)$ where $i = \text{len}(q')$
- Return: highest scoring state in Q_n. Backpointers can be used to find the underlying sequence of phrases (and the translation).

Definition of Add (Q,q’,q,p)

- If there is some $q'' \in Q$ such that $eq(q'', q') =$ True:
  - If $\alpha(q') > \alpha(q'')$
    - $Q = \{q'\} \cup Q \setminus \{q''\}$
    - set $bp(q') = (q, p)$
  - Else return
- Else
  - $Q = Q \cup \{q'\}$
  - set $bp(q') = (q, p)$

Definition of beam(Q)

Define

$$\alpha^* = \arg\max_{q \in Q} \alpha(q)$$

i.e., α^* is the highest score for any state in Q.

Define $\beta \geq 0$ to be the *beam-width* parameter
Then

$$\text{beam}(Q) = \{q \in Q : \alpha(q) \geq \alpha^* - \beta\}$$

The Decoding Algorithm

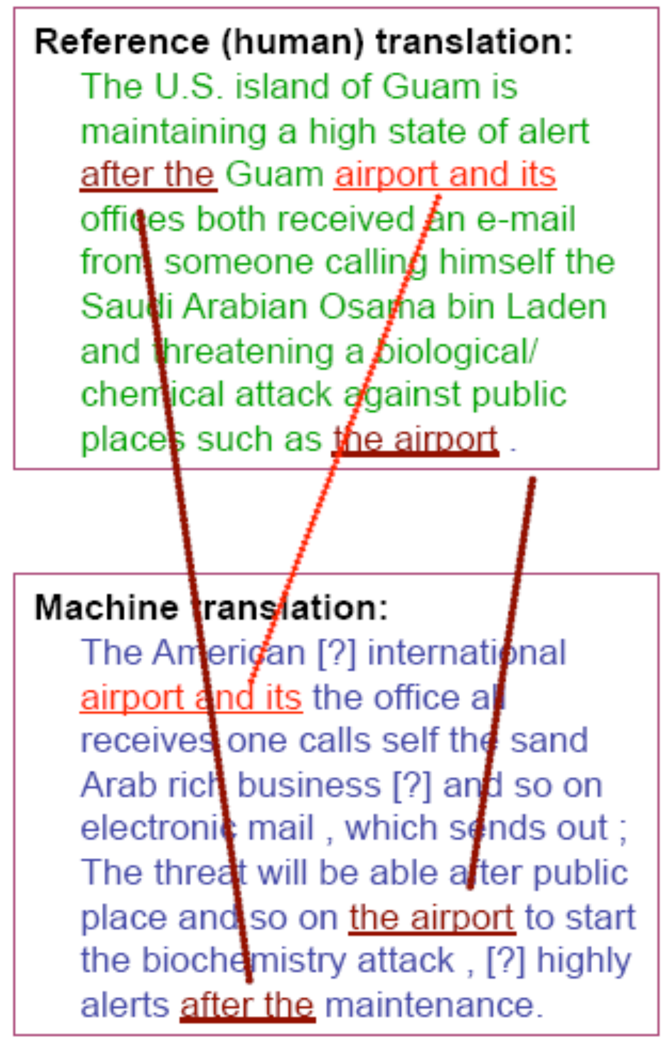
- Inputs: sentence $x_1 \ldots x_n$. Phrase-based model $(\mathcal{L}, h, d, \eta)$. The phrase-based model defines the functions $ph(q)$ and $\text{next}(q, p)$.
- Initialization: set $Q_0 = \{q_0\}$, $Q_i = \emptyset$ for $i = 1 \ldots n$.
- For $i = 0 \ldots n - 1$
  - For each state $q \in \text{beam}(Q_i)$, for each phrase $p \in ph(q)$:
    (1) $q' = \text{next}(q, p)$
    (2) $\text{Add}(Q_i, q', q, p)$ where $i = \text{len}(q')$
- Return: highest scoring state in Q_n. Backpointers can be used to find the underlying sequence of phrases (and the translation).

Overview

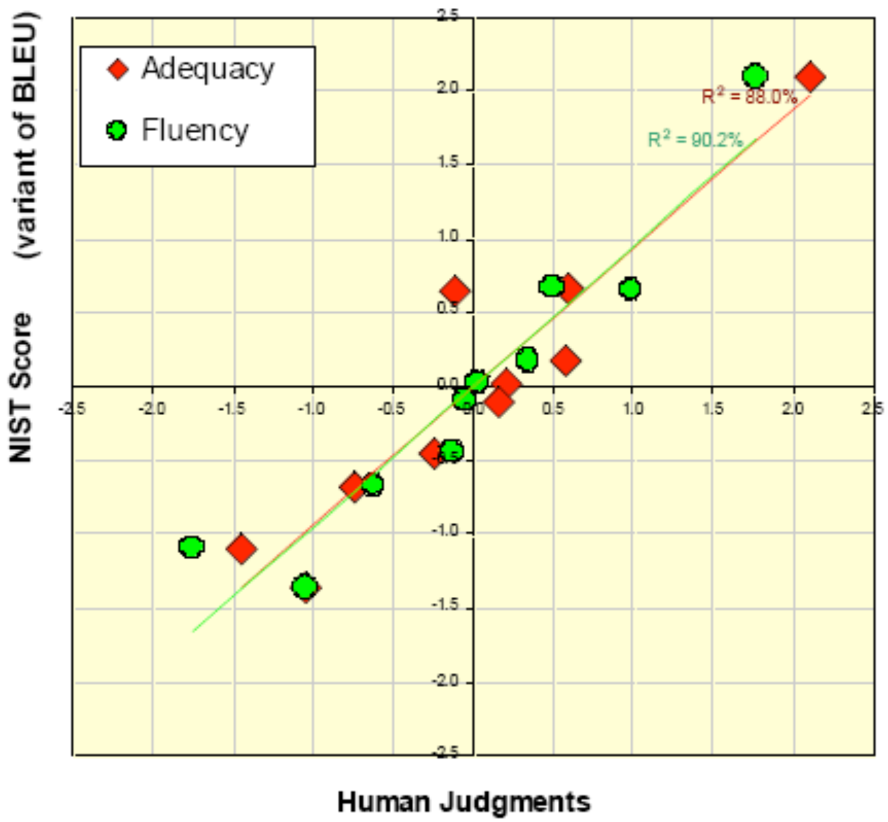
- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based model
- MT evaluation

Automatic Evaluation

- Human evaluations: subject measures, fluency/adequacy
- Automatic measures: n-gram match to references
  - NIST measure: n-gram recall (worked poorly)
  - BLEU: n-gram precision (no one really likes it, but everyone uses it)
- BLEU:
  - P1 = unigram precision
  - P2, P3, P4 = bi-, tri-, 4-gram precision
  - Weighted geometric mean of P1-4
  - Brevity penalty (why?)
  - Somewhat hard to game…



Correlation with Human Evaluataion



slide from G. Doddington (NIST)