Machine Translation

Natalie Parde, Ph.D.

Department of Computer Science

University of Illinois at Chicago

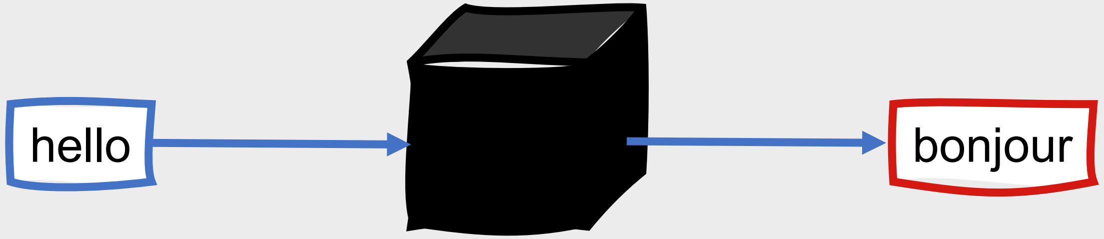
CS 421: Natural Language Processing

Fall 2019

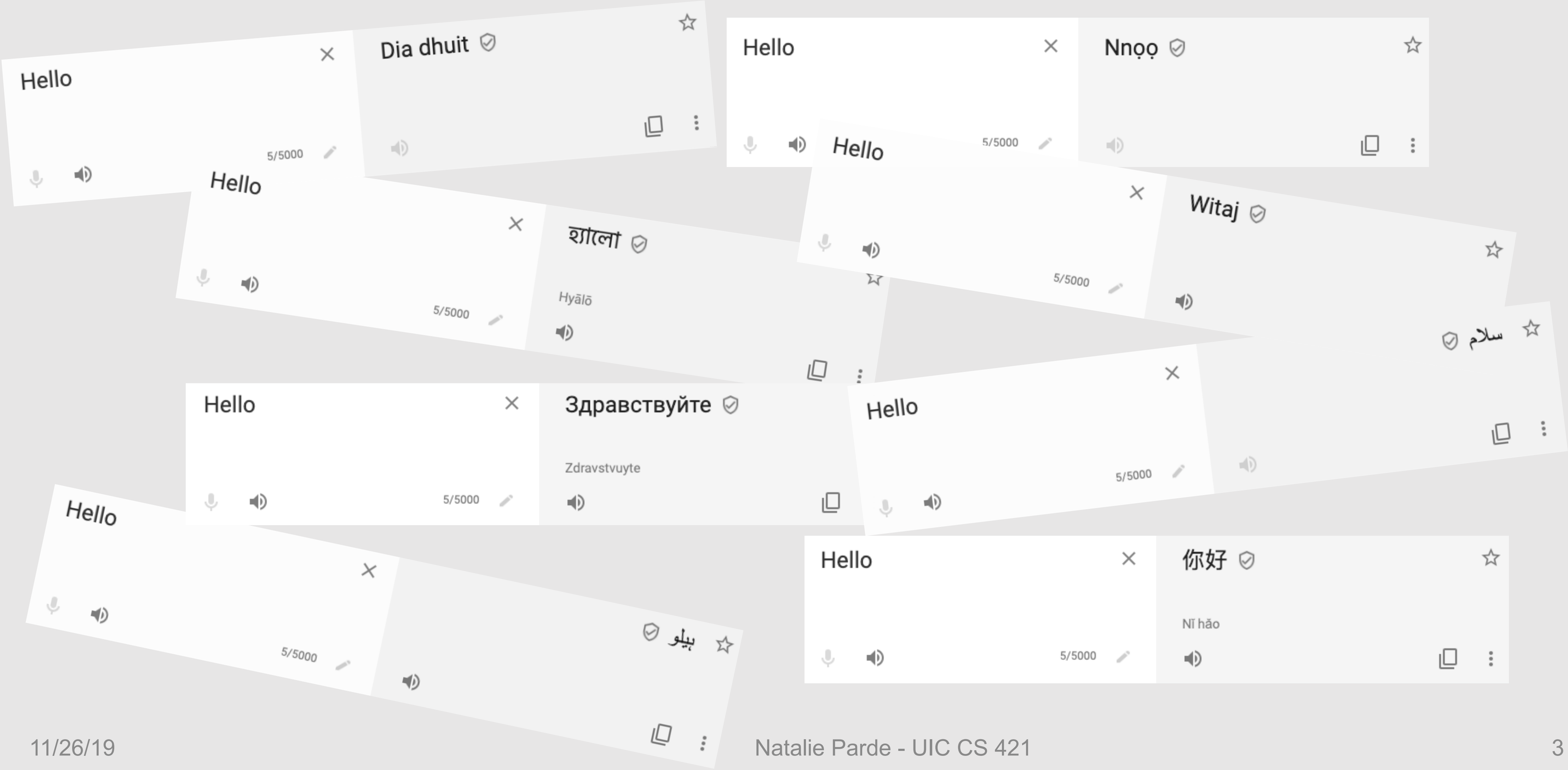
Many slides adapted from Jurafsky and Martin (https://web.stanford.edu/~jurafsky/slp3/).

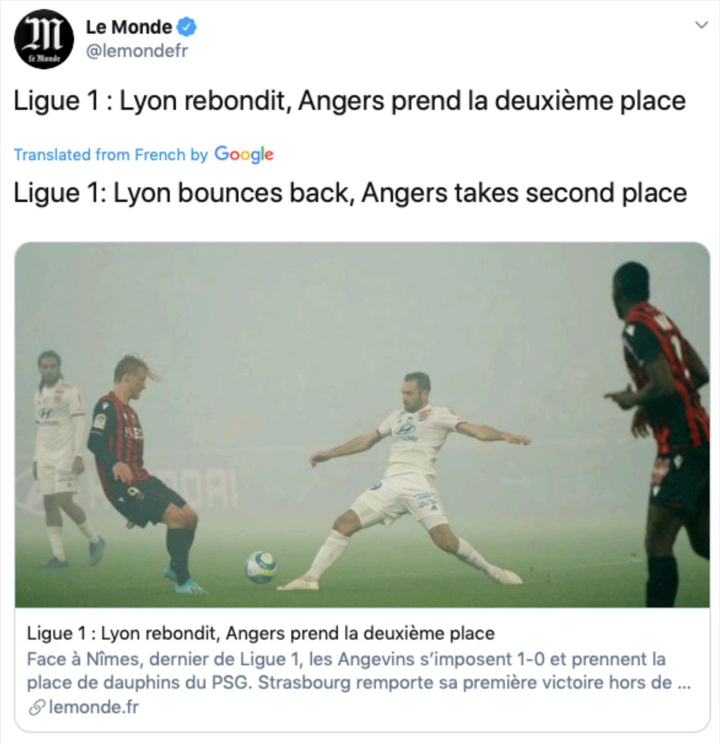
What is machine translation?

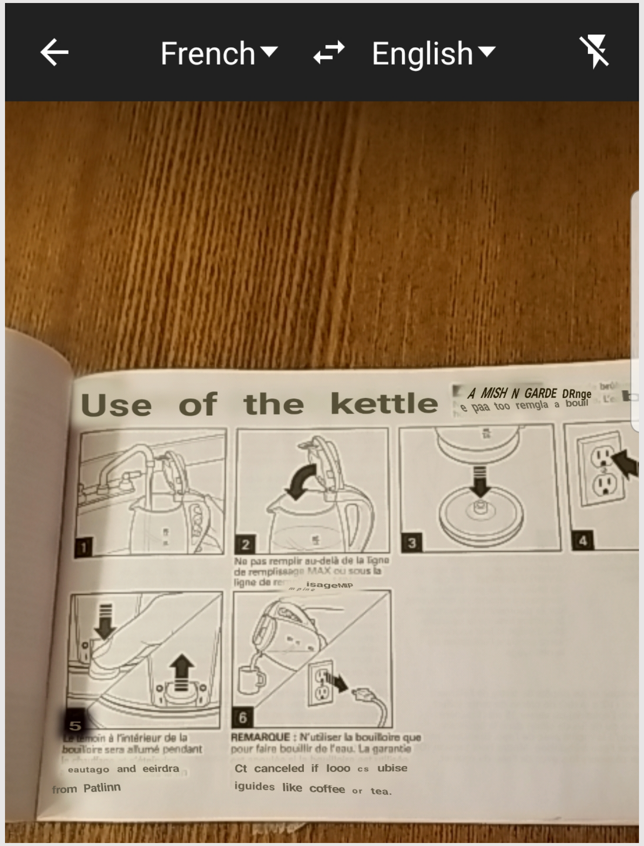
- The process of automatically converting a text from one language to another



Machine Translation in Action







Machine translation is increasingly ubiquitous, and useful in a wide variety of contexts!

Machine translation is also difficult, for a variety of reasons.

Structural and lexical differences between languages

Differences in word order

Morphological differences

Stylistic and cultural differences

Sample Translated Passage

AGAIN LISTEN-TO WINDOW OUTSIDE BAMBOO *TIP* PLANTAIN *LEAF OF* ON-TOP RAIN SOUND SIGH DRIP

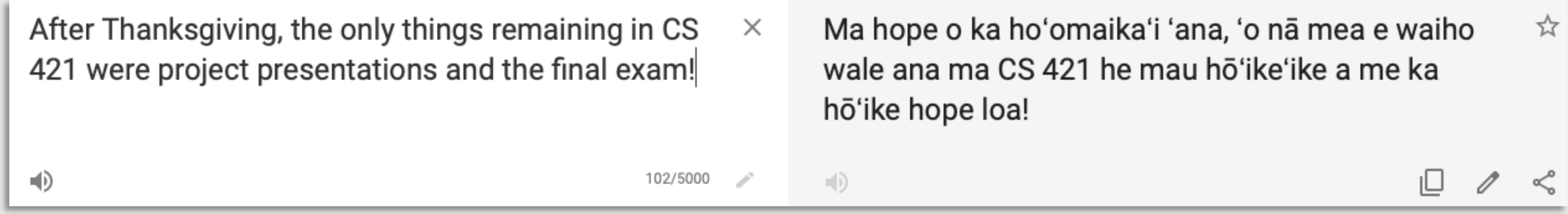
Then she listened to *the* insistent rustle of *the* rain on the bamboos *and* plantains outside *her* window.

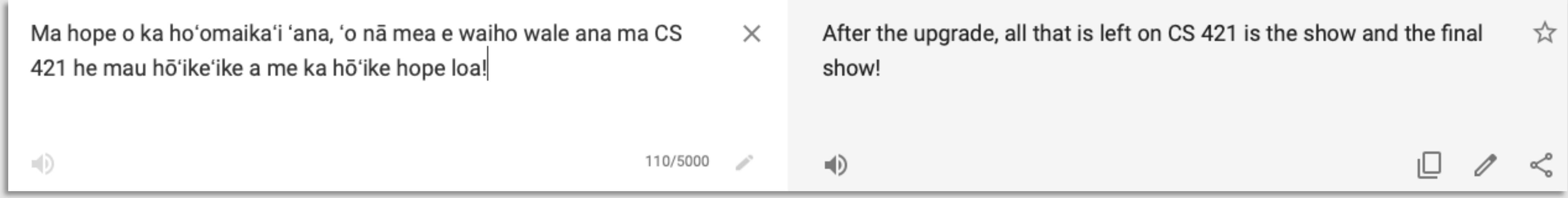
- *Dream of the Red Chamber*, Cao Xueqin

Creating high-quality translations requires a deep understanding of both the source and target language.

- It is particularly difficult to translate creative text!
- Current machine translation methods tend to excel in scenarios in which:
  - A rough translation is adequate
  - A human post-editor is used
  - The task is limited to a small sublanguage domain (e.g., weather forecasting)

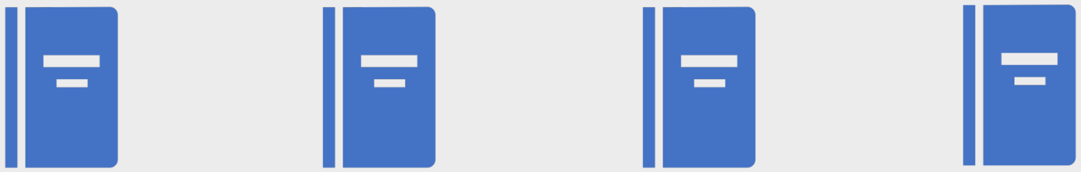
Otherwise, results may be more confusing than helpful!

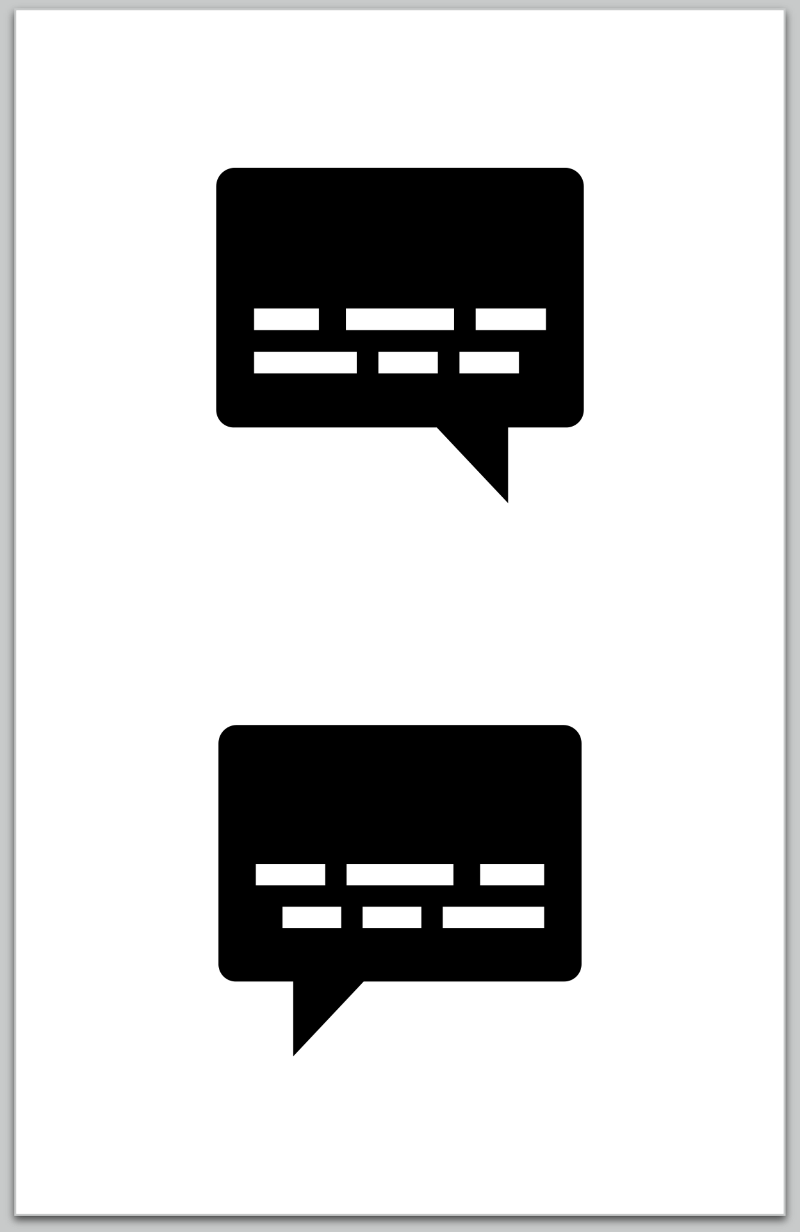




Computer-Aided Human Translation

- Even poor translations are useful for some purposes!
- **Computer-Aided Human Translation:** Computers provide draft translations, which are then fixed in a post-editing phase by a human translator
- Effective for:
  - High volume jobs
  - Jobs requiring quick turnaround





Cross-Linguistic Similarities and Differences

- **Typology:** The study of systematic cross-linguistic similarities and differences
  - Although some aspects of language are universal, others tend to differ
  - Differences between languages often have systematic structure

Morphological Differences

Number of morphemes per word

- **Isolating languages:** Each word generally has one morpheme
- **Polysynthetic languages:** Each word may have many morphemes

Degree to which morphemes can be segmented

- **Agglutinative languages:** Morphemes have well-defined boundaries
- **Fusion languages:** Morphemes may be conflated with one another

Syntactic Differences

- Primary difference between languages: Word order
  - **SVO languages:** Verb tends to come between the subject and object
  - **SOV languages:** Verb tends to come at the end of basic clauses
  - **VSO languages:** Verb tends to come at the beginning of basic clauses
- Languages with similar basic word order also tend to share other similarities
  - SVO languages generally have prepositions
  - SOV languages generally have postpositions

Differences in Argument Structure and Linking

Head-Marking languages: Tend to mark the relation between the head and its dependent on the head

Dependent-Marking languages: Tend to mark the relation on the dependent

the man**'s** house

az ember ház**a**

the man house-his

Differences in Argument Structure and Linking

Verb-framed languages: Generally mark the direction of motion on the verb, leaving its satellites (particles, prepositional phrases, and adverbial phrases) to mark the manner of motion

Satellite-framed languages: Generally mark the direction of motion on the satellite, leaving the verb to mark the manner of motion

The bottle floated out.

La botella salió flotando.

The bottle exited floating.

Differences in Permissible Omissions

- Languages differ in terms of what components can be omitted from a sentence
- **Pro-Drop languages:** Can omit pronouns when talking about certain referents
- Some pro-drop languages permit more pronoun omission than others
  - **Referentially dense** and **sparse** languages
- Converting text from pro-drop languages (e.g., Japanese) to non-pro-drop languages (e.g., English) requires that all missing pronoun locations are identified and their appropriate **anaphors** recovered

Other Differences

Differences in noun-adjective order

- Blue house → Maison bleue

Differences in homonymy and polysemy

Differences in grammatical constraints

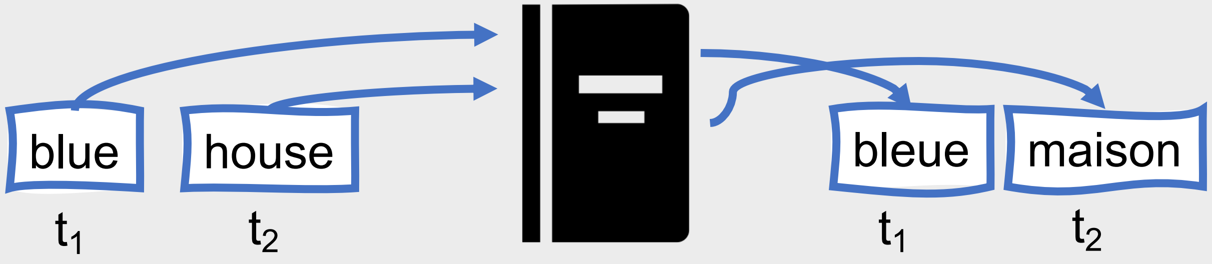
- Some languages require gender for nouns
- Some languages require gender for pronouns

Lexical gaps

- No word or phrase in the target language can express the meaning of a word in the source language

Classical Machine Translation

- **Direct translation**
  - Take a large bilingual dictionary
  - Proceed through the source text word by word
  - Translate each word according to the dictionary



Direct Translation

No intermediate structures

Simple reordering rules may be applied

- Moving adjectives so that they are after nouns when translating from English to French

Dictionary entries may be relatively complex

- Tiny, rule-based programs for translating a word to the target language

Direct Translation

Pros:

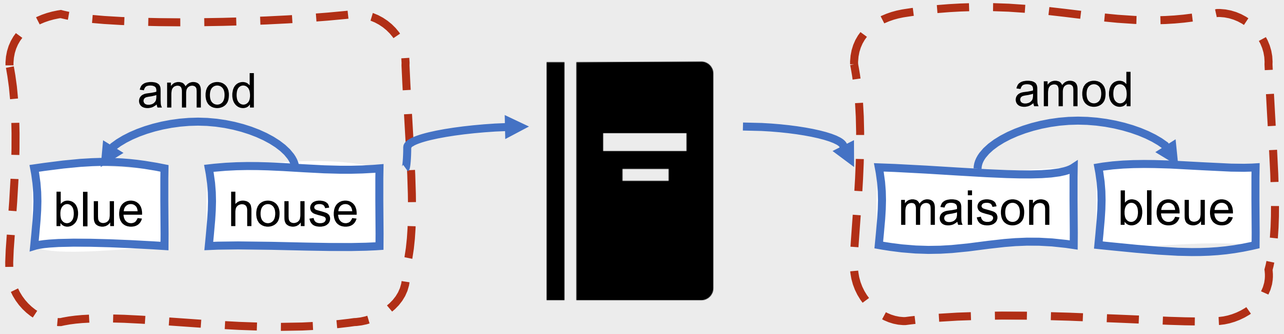
- Simple
- Easy to implement

Cons:

- Cannot reliably handle long-distance reorderings
- Cannot handle reorderings involving phrases or larger structures
- Too focused on individual words

Classical Machine Translation

- **Transfer approaches**
  - Parse the input text
  - Apply rules to transform the source language parse structure into a target language parse structure



Transfer Approaches

Three phases:

- Analysis
- Transfer
- Generation

Transfer Approach Phases: Analysis

Morphological analysis

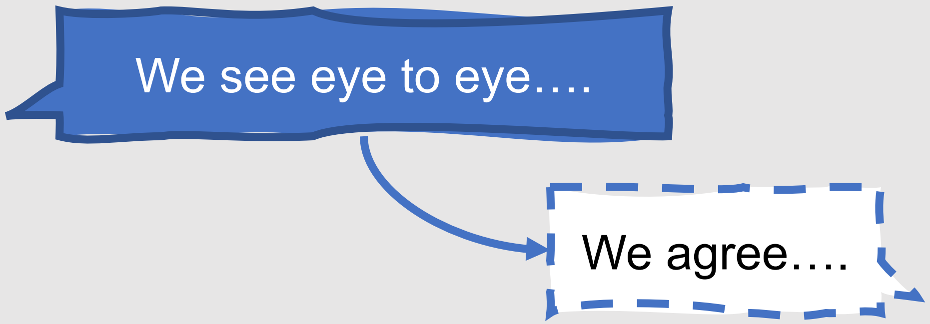
Part-of-speech tagging

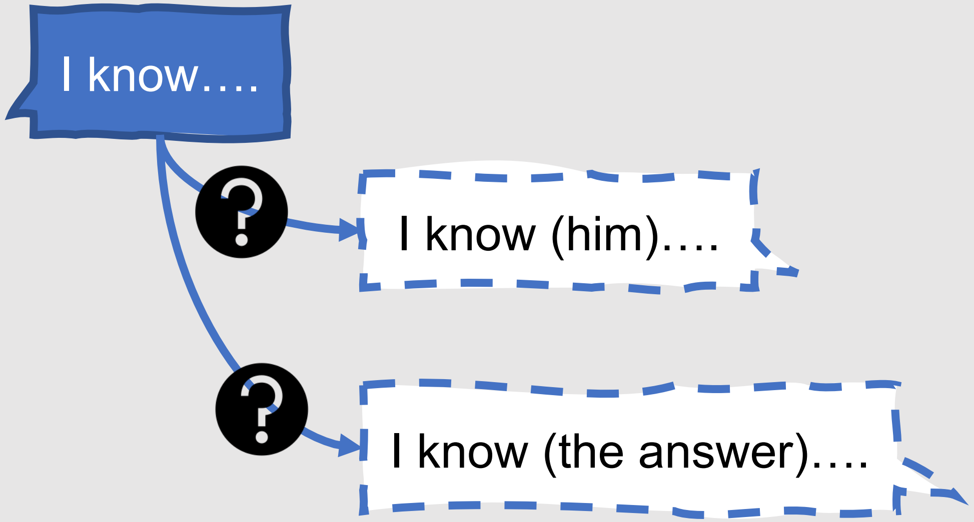
Constituency parsing

Dependency Parsing

Transfer Approach Phrases: Transfer

- Translation of idioms
- Word sense disambiguation
- Preposition assignment





Transfer Approach Phases: Generation

- Lexical translation via a bilingual dictionary
- Word reorderings
- Morphological generation

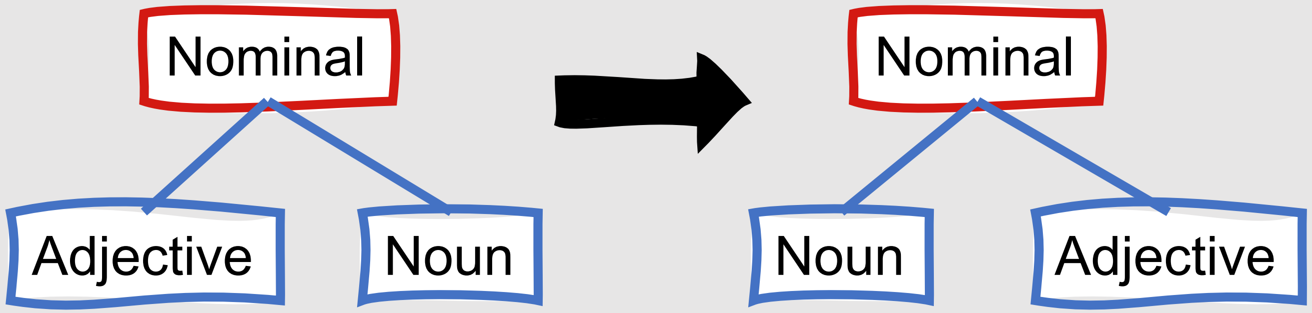
Transfer Approaches

Two subcategories of transformations:

- Syntactic transfer
- Lexical transfer

Syntactic Transfer

- Modifies the source parse tree to resemble the target parse tree
- For some languages, may also include **thematic structures**
  - Directional or locative prepositional phrases vs. recipient prepositional phrases



Lexical Transfer

- Generally based on a bilingual dictionary
  - As with direct translation, dictionary entries can be complex to accommodate many possible translations

Transfer Approaches

Pros:

- Can handle more complex language phenomena than direct translation

Cons:

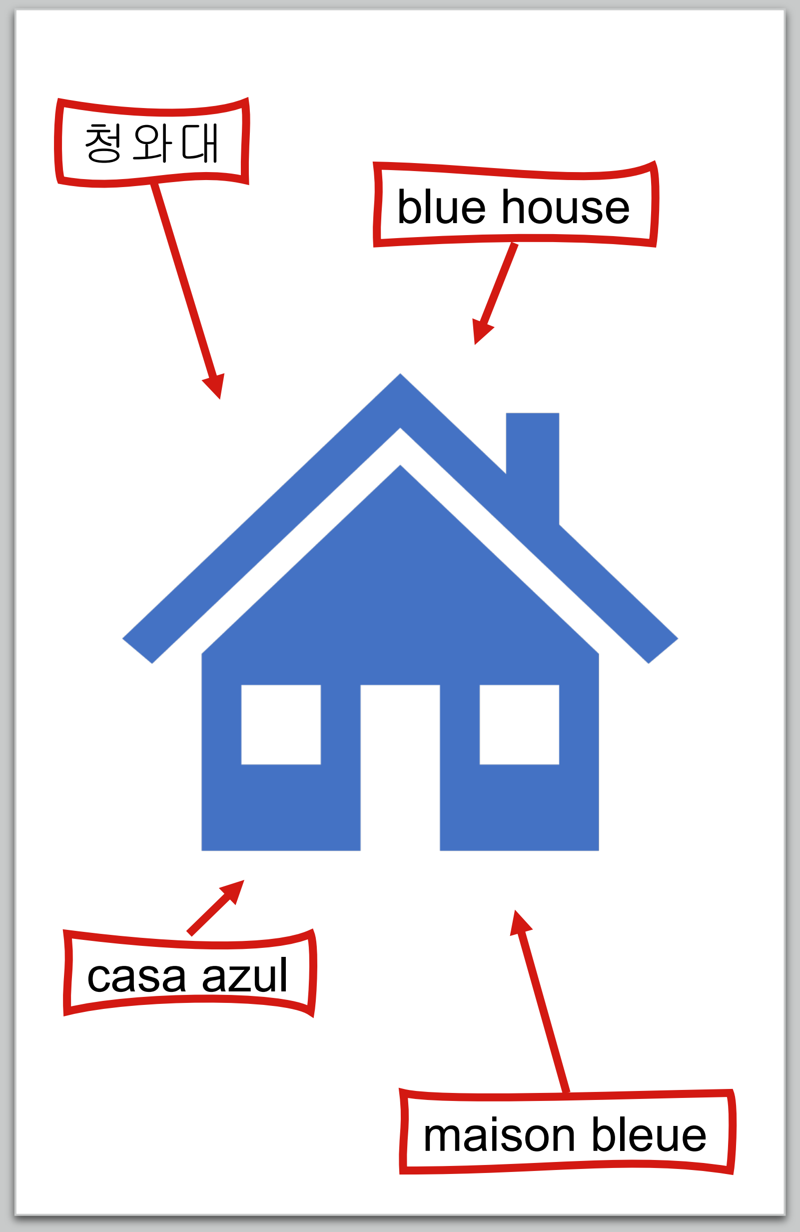
- Still not sufficient for many cases!

Classical Machine Translation

- **Interlingua approaches**
  - Convert the source language text into an abstract meaning representation
  - Generate the target language text based on the abstract meaning representation

Interlingua Approaches

- Goal: Represent all sentences that mean the same thing in the same way, regardless of language
- What kind of representation scheme should be used?
  - Classical approaches:
    - First-order logic
    - Semantic primitives
    - Event-based representation
  - More recently, neural models learn vector representations for this purpose



Interlingua Approaches

- Require more analysis work than transfer approaches
  - Semantic analysis
  - Sentiment analysis
- No need for syntactic or lexical transformations

Interlingua Approaches

Pros:

- Direct mapping between meaning representation and lexical realization
- No need for transformation rules

Cons:

- Extra (often unnecessary) work
  - Classical approaches require an exhaustive analysis and formalization of the semantics of the domain

Statistical Machine Translation

- Models automatically learn to map from the source language to the target language
  - Doesn’t require intermediate transformation rules
  - Doesn’t require an explicitly defined internal meaning representation

Why is this useful?

It is often impossible for a sentence in the target language to be an exact translation of a sentence in the source language

- Culture-specific concepts
- Figurative language

Statistical approaches strive to find the best possible fit, given the circumstances

Statistical Machine Translation

- Goal: Produce an output that maximizes some function representing translation **faithfulness** and **fluency**
- One possible approach: **Bayesian noisy channel model**
  - Assume a possible target language translation t_i and a source language sentence s
  - Select the translation t' from the set of all possible translations $t_i \in T$ that maximizes the probability $P(t_i|s)$

Bayesian Noisy Channel Model

- To find P(t_i|s), we can use Bayes rule:
  - $t' = \underset{t_i \in T}{\operatorname{argmax}} P(t_i \mid s)$
  - $t' = \underset{t_i \in T}{\operatorname{argmax}} \frac{P(s|t_i)P(t_i)}{P(s)}$
- We can ignore the denominator ($P(s)$) since it will remain the same for all possible translations
- Thus:
  - $t' = \underset{t_i \in T}{\operatorname{argmax}} P(s|t_i)P(t_i)$

This means that we need to consider two separate components.

$$t' = \underset{t_i \in T}{\operatorname{argmax}} \underbrace{P(s|t_i)}_{\text{translation model}} \underbrace{P(t_i)}_{\text{language model}}$$

- The language model is just like the language models used for other NLP tasks
- One common type of translation model is the **phrase-based translation model**

The Phrase-Based Translation Model

- Computes the probability that a given translation t_i generates the original sentence s based on its **constituent phrases**
- Intuition: Phrases, as well as single words, are fundamental units of translation
  - Often entire phrases need to be translated and moved as a unit

Stages of Phrase-Based Translation

01

Group the words from the source sentence into phrases

02

Translate each source phrase into a target language phrase

03

(Optionally) reorder the target language phrases

Probability in Phrase-Based Translation Models

- Relies on two probabilities:
  - **Translation probability**
    - Probability of generating a source language phrase from a target language phrase
  - **Distortion probability**
    - Probability of two consecutive target language phrases being separated in the source language by a word span of a particular length
- $P(t|s) = \prod_{i=1}^{I} \phi(\overline{t_i}, \overline{s_i}) d(a_i - b_{i-1})$
  - I is the total number of target phrases
  - a_i is the start position of the phrase generated by s_i
  - b_{i-1} is the end position of the phrase generated by s_{i-1}

Example: Probability in Phrase-Based Translation Models

Position	1	2	3	4	5
English	Usman	did not	slap	the	green witch
Spanish	Usman	no	dió una bofetada	a la	bruja verde

$$P(t|s) = \prod_{i=1}^{I} \phi(\bar{t_i}, \bar{s_i}) d(a_i - b_{i-1})$$

Example: Probability in Phrase-Based Translation Models

Position	1	2	3	4	5
English	Usman	did not	slap	the	green witch
Spanish	Usman	no	dió una bofetada	a la	bruja verde

$$\boxed{P(t|s) = \prod_{i=1}^{I} \phi(\overline{t_i}, \overline{s_i}) d(a_i - b_{i-1})}$$

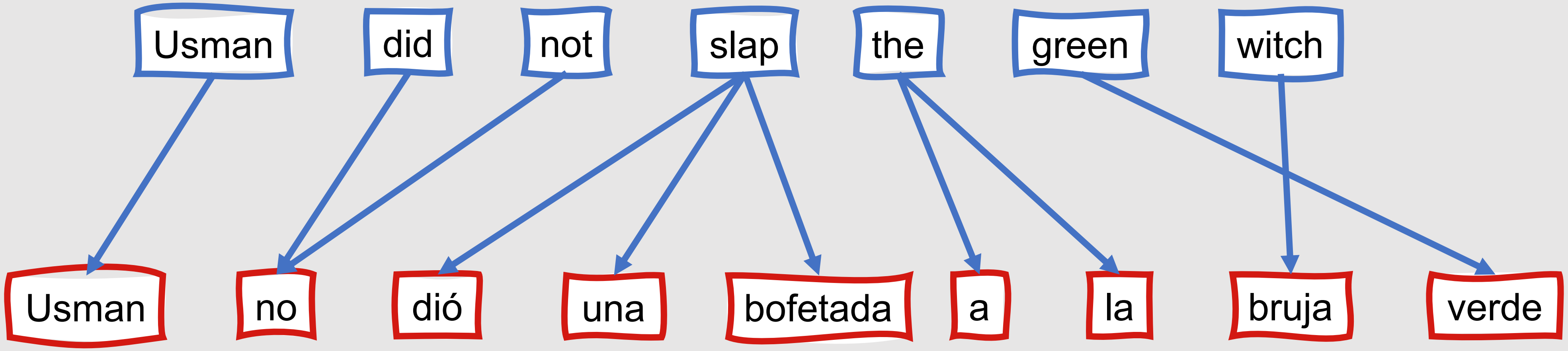
P(t | s) = P(Usman | Usman) * d(1-0) * P(no | did not) * d(2-1) * P(dió una bofetada | slap) * d(3-2) * P(a la | the) * d(4-3) * P(bruja verde | green witch) * d(5-4)

- We need to train two sets of parameters:
  - $\phi(\bar{t_i}, \bar{s_i})$
  - $d(a_i - b_{i-1})$
- We learn these based on large bilingual training sets in which we know which phrase in a source sentence is translated to which phrase in a target sentence
- These mappings are called **phrase alignments**
- Since large, phrase-aligned training sets are uncommon, we can also learn parameters using **word alignments**

How do we learn the probabilities for this model?

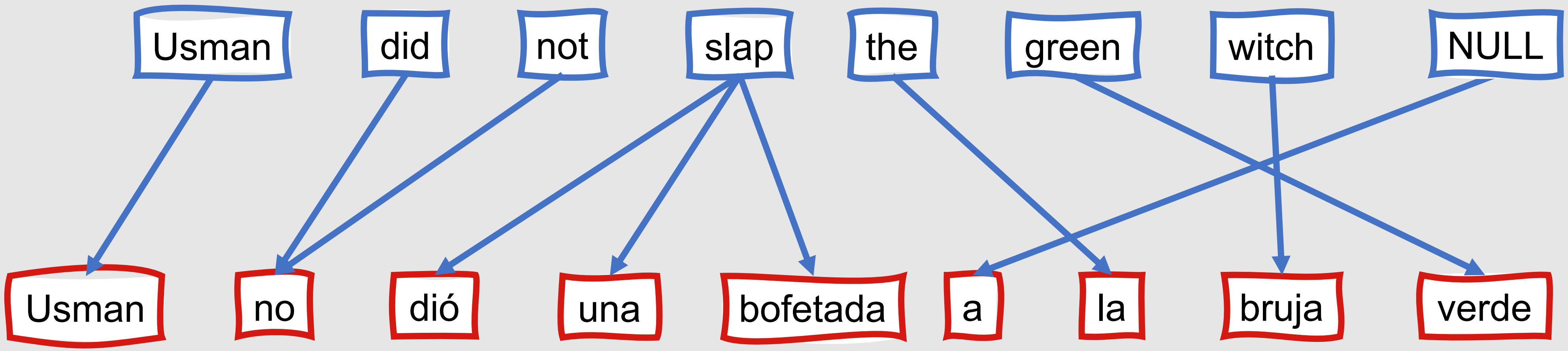
Alignment in Machine Translation

- Mappings between one word or phrase to another



Alignment in Machine Translation

- Different alignment models tend to apply different constraints
  - Each word in language x can be translated to exactly one word in language y
    - Not necessarily implying that each word in language y can be translated to exactly one word in language x!
  - Spurious words can be mapped to NULL

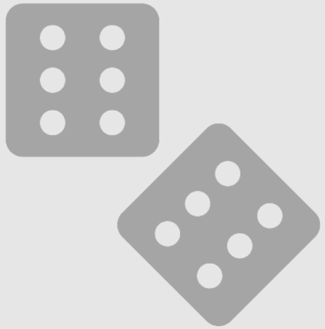


Training Alignment Models

- Generally trained with large, parallel corpora
- Common samples used:
  - Legal text and proceedings from countries with multiple official languages
  - Literary translations
  - Religious texts

Training Alignment Models

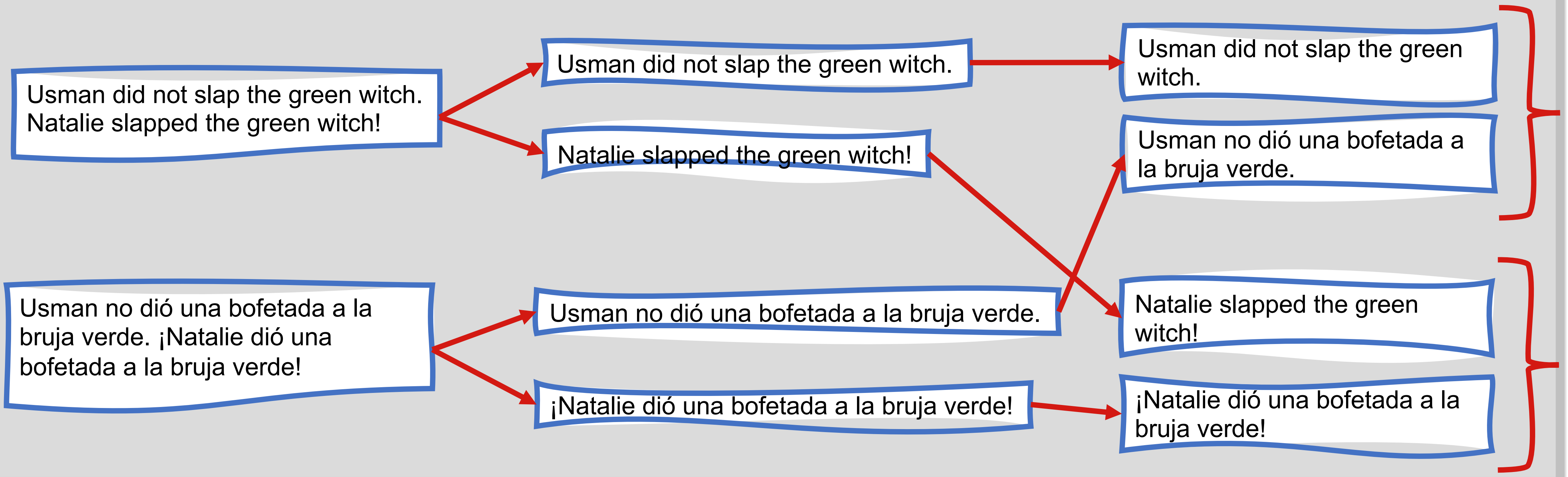
Sentence segmentation and alignment



Probability estimation

Sentence Segmentation and Alignment

- Simple approaches align sentences based on word and character length
- More sophisticated methods make use of word alignment methods



Probability Estimation

- Traditionally done using the **expectation-maximization** algorithm
  - Estimate parameters
  - Compute alignments from those estimates
  - Use the alignments to re-estimate the parameters
  - Repeat

Symmetrizing Alignments for Phrase-Based Machine Translation

- Once we have word alignments, we can extract aligned pairs of phrases
- One way to do this:
  - Take the **intersection of a source-to-target and target-to-source alignment** for a given sentence
    - This results in a set of high-precision aligned words
  - Take **the union of the two alignments**
    - This results in many less accurately aligned words
  - **Incrementally add alignments from the union to the intersection** to produce a minimal intersective alignment
  - From that alignment, **extract all phrase pairs for which all words are aligned only with each other** and not to any external words

Symmetrizing Alignments for Phrase-Based Machine Translation

Spanish to English

	Usman	no	dió	una	bofetada	a	la	bruja	verde
Usman	■								
did		■							
not		■							
slap					■				
the							■		
green									■
witch								■	

English to Spanish

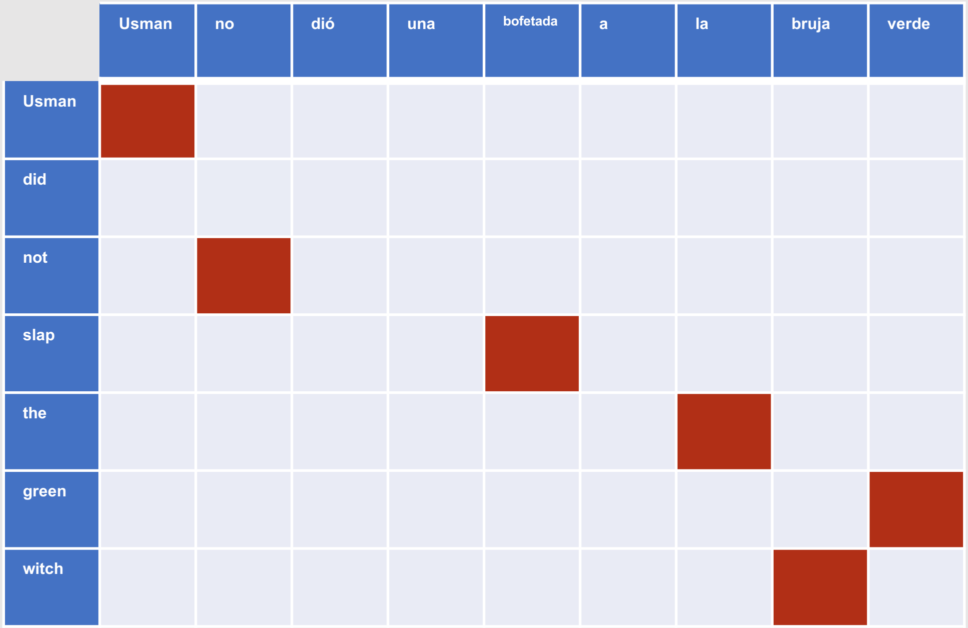
	Usman	no	dió	una	bofetada	a	la	bruja	verde
Usman	■								
did						■			
not		■							
slap			■	■	■				
the							■		
green									■
witch								■	

Symmetrizing Alignments for Phrase-Based Machine Translation

Spanish to English

English to Spanish

Intersection



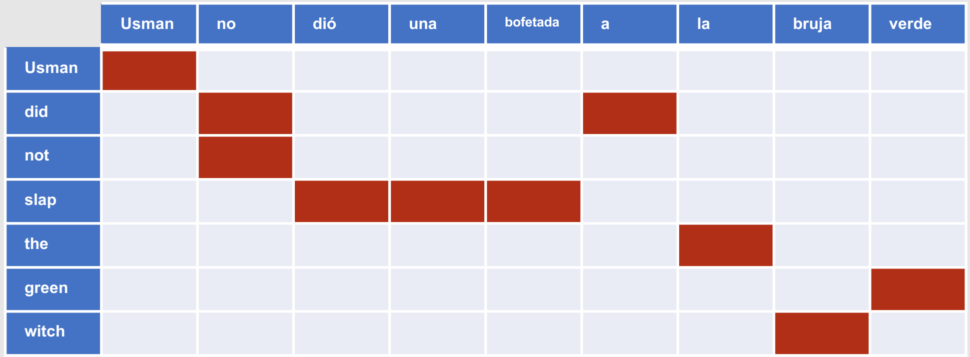
Symmetrizing Alignments for Phrase-Based Machine Translation

Spanish to English

Intersection

English to Spanish

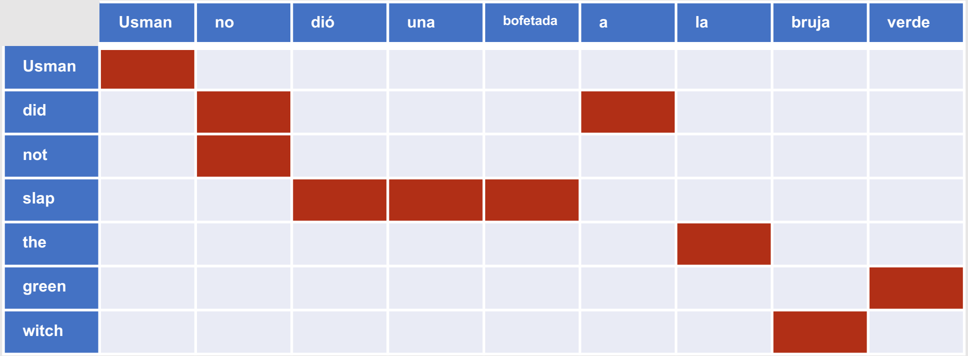
Union



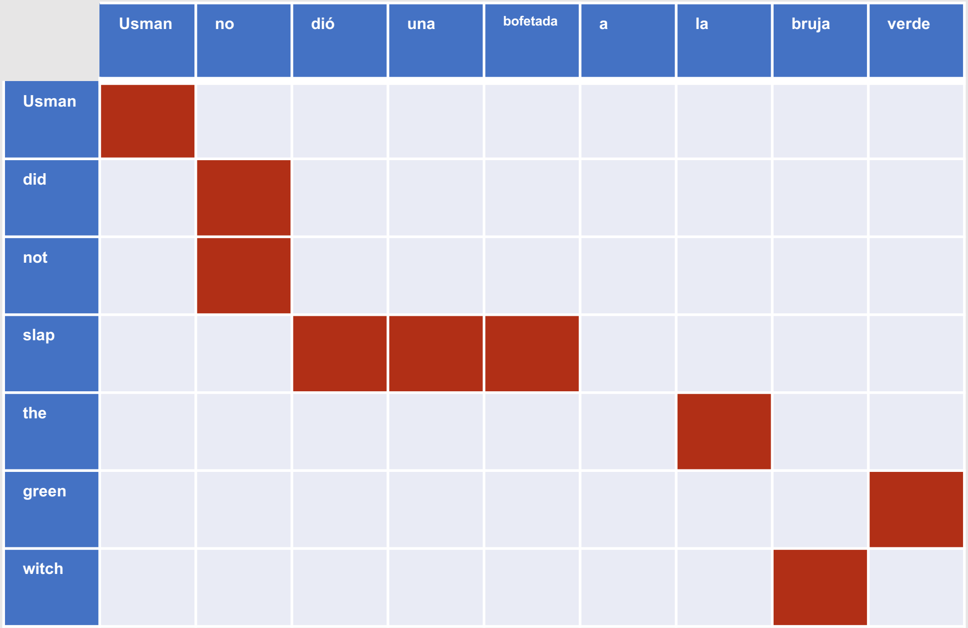
Symmetrizing Alignments for Phrase-Based Machine Translation

Intersection

Union



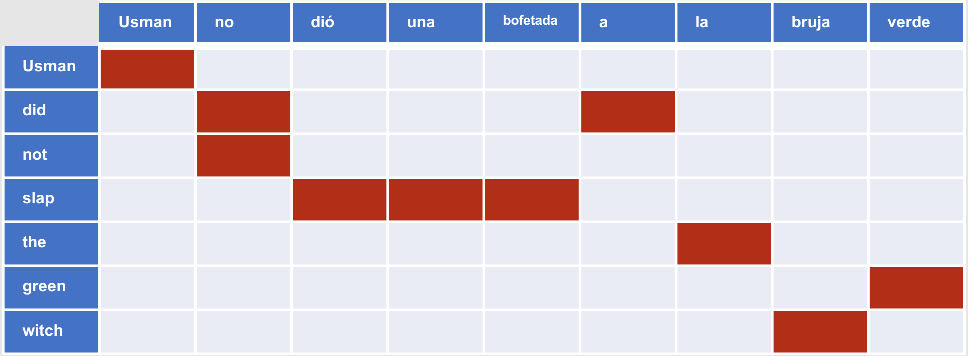
Potential Minimal Intersective Alignment



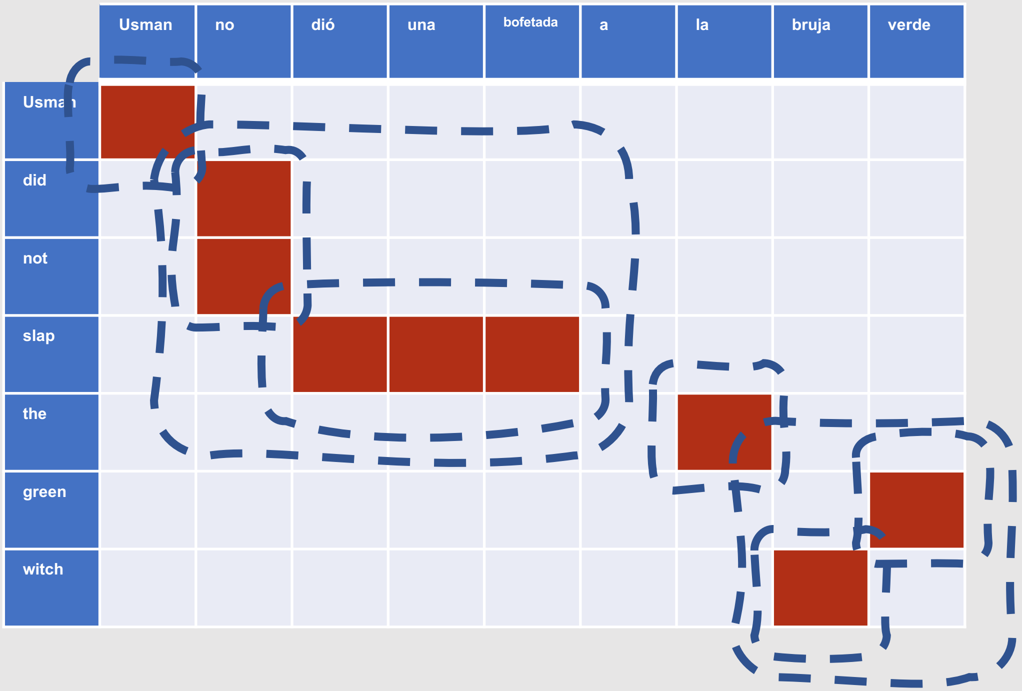
Symmetrizing Alignments for Phrase-Based Machine Translation

Intersection

Union



Potential Minimal Intersective Alignment



Decoding for Phrase-Based Machine Translation

- Aligned phrases can be stored in a **phrase-translation** table
- **Decoding algorithms** can then search through this table to find the overall translation that maximizes the phrase translation probabilities
- Since it is impractical to search the entire state space of possible translations, many decoders apply **beam search pruning**
  - At every iteration, keep the most promising states and prune unlikely states (those outside the "search beam")

So far….

Classical machine translation

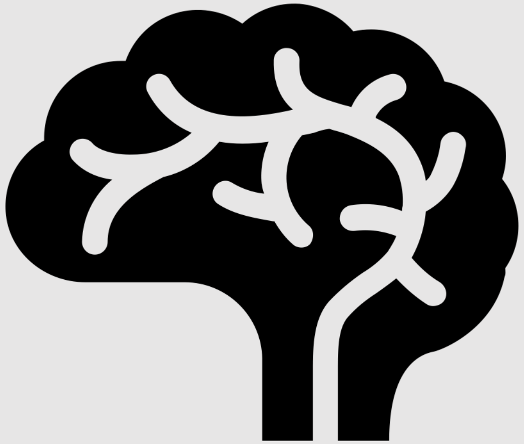
Rule-based approaches utilizing dictionaries and formal representations

Statistical machine translation

Probabilistic approaches based on word and phrase alignment

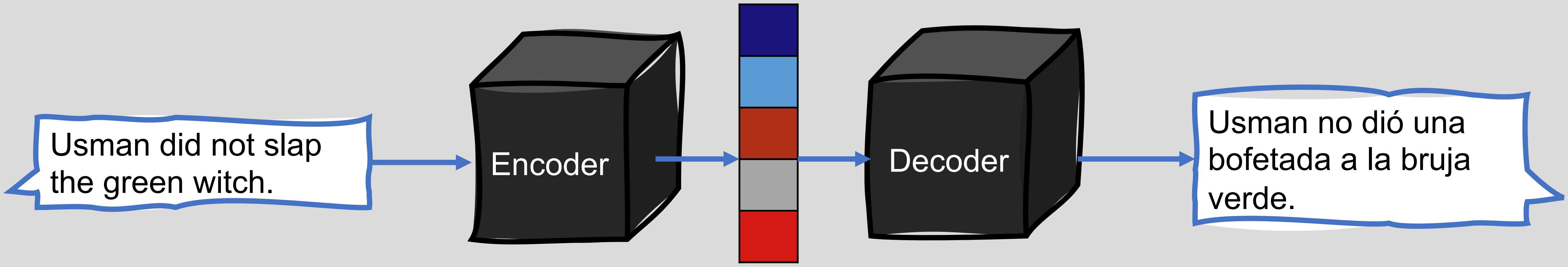
Recently….

- **Neural machine translation**
  - Neural network approaches that learn mappings to and from internal representations



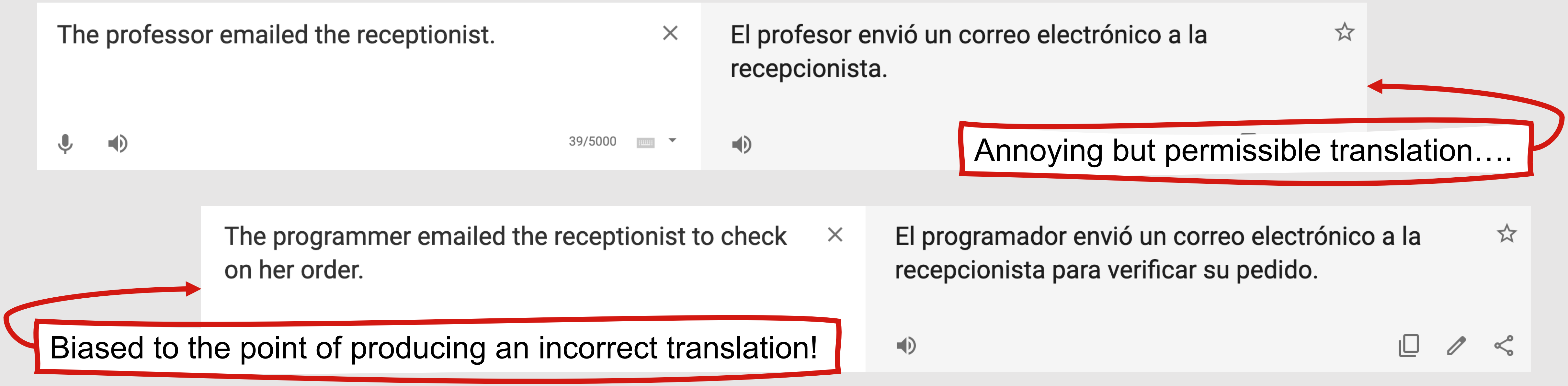
Neural Machine Translation

- Key advantages:
  - Can be learned directly from parallel source and target corpora
  - End-to-end (no need for intricate pipelines)
- Often built using encoder-decoder models



Neural Machine Translation

- A few disadvantages:
  - Can be sensitive to subtle changes in input
  - Can be subject to human biases, similar to other data-driven approaches



How do we evaluate machine translation models?

- Translation quality tends to be very subjective!
- Two common approaches:
  - **Human ratings**
  - **BLEU scores**

Evaluating Machine Translation Using Human Ratings

- Typically evaluated along multiple dimensions
- Tend to check for both **fluency** and **fidelity**
- **Fluency:**
  - Clarity
  - Naturalness
  - Style
- **Fidelity:**
  - Adequacy
  - Informativeness

Evaluating Machine Translation Using Human Ratings

- How to get quantitative measures of fluency?
  - Ask humans to rate different aspects of fluency along a scale
  - Measure how long it takes humans to read a segment of text
  - Ask humans to guess the identity of the missing word
    - “After such a late night working on my project, it was hard to wake up this ______!”

Evaluating Machine Translation Using Human Ratings

- How to get quantitative measures of fidelity?
  - Ask bilingual raters to rate how much information was preserved in the translation
  - Ask monolingual raters to do the same, given access to a gold standard reference translation
  - Ask humans to answer multiple-choice questions about content present in a translation

Another set of human evaluation metrics considers post-editing.

- Ask a human to **post-edit** or "fix" a translation
- Compute the number of edits required to correct the output to an acceptable level
  - Can be measured via number of word changes, number of keystrokes, amount of time taken, etc.

Evaluating Using BLEU Scores

- Intuition: A good machine translation output is one that is very similar to a human translation
- Thus, compute a weighted average of the number of n-gram overlaps with human translations
- **Precision-based metric**
  - What percentage of words in the candidate translation also occur in the gold standard translation(s)?

How is BLEU computed?

- Count the maximum number of times each n-gram is used in any single reference translation, c_{max}(*n-gram*)
- Count the number of times each n-gram is used in the candidate translation
- Clip that amount so that the highest it can be is c_{max}(*n-gram*)
- Compute precision for each word in the candidate translation based on that clipped amount
  - $p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$
- Take the geometric mean of the modified n-gram precisions for unigrams, bigrams, trigrams, and 4-grams

- Otherwise, extremely short translations (e.g., "the") could receive perfect scores!
- The penalty is based on two values:
  - The effective reference length, *r*, for the corpus
    - The sum of the lengths of the best matches for each candidate sentence
  - The total length of the candidate translation corpus, l_c
- Formally, the penalty is set to:
  - $BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$

BLEU also adds a penalty for translation brevity.

Computing BLEU

- The full BLEU score for a set of translations is then:
  - $BLEU = BP \ * \ \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$

Example: Computing BLEU

Usman no dió una bofetada a la bruja verde. Source Sentence

Usman didn’t slap the green witch. Reference Translation

Usman did not give a slap to the green witch. Candidate Translation

Example: Computing BLEU

Usman no dió una bofetada a la bruja verde. — Source Sentence

Usman didn't slap the green witch. — Reference Translation

Usman did not give a slap to the green witch. — Candidate Translation

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP \ * \ \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Example: Computing BLEU

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP * \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Unigram	Unigram Frequency (Candidate)	Unigram Frequency (Reference)
Usman	1	1
did	1	0
not	1	0
give	1	0
a	1	0
slap	1	1
to	1	0
the	1	1
green	1	1
witch	1	1
.	1	1

Example: Computing BLEU

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(c(n-gram), c_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} c(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP * \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Unigram	Unigram Frequency (Candidate)	Unigram Frequency (Reference)
Usman	1	1
did	1	0
not	1	0
give	1	0
a	1	0
slap	1	1
to	1	0
the	1	1
green	1	1
witch	1	1
.	1	1

$$p_1 = \frac{1+0+0+0+0+1+0+1+1+1+1}{1+1+1+1+1+1+1+1+1+1+1} = \frac{6}{11}$$

Example: Computing BLEU

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$$p_n = \frac{\sum_{c\in\{Candidates\}}\sum_{n-gram\in C}\min(\mathrm{c}(n-gram),\mathrm{c}_{\max}(n-gram))}{\sum_{c\in\{Candidates\}}\sum_{n-gram\in C}\mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \le r \end{cases}$$

$$BLEU = BP\ *\ \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n\right)$$

Bigram	Bigram Frequency (Candidate)	Bigram Frequency (Reference)
Usman did	1	0
did not	1	0
not give	1	0
give a	1	0
a slap	1	0
slap to	1	0
to the	1	0
the green	1	1
green witch	1	1
witch .	1	1

$$p_1 = \frac{1+0+0+0+0+1+0+1+1+1+1}{1+1+1+1+1+1+1+1+1+1+1} = \frac{6}{11}$$

$$p_2 = \frac{0+0+0+0+0+0+0+1+1+1}{1+1+1+1+1+1+1+1+1+1} = \frac{3}{10}$$

Example: Computing BLEU

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP * \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

Trigram	Trigram Frequency (Candidate)	Trigram Frequency (Reference)
Usman did not	1	0
did not give	1	0
not give a	1	0
give a slap	1	0
a slap to	1	0
slap to the	1	0
to the green	1	0
the green witch	1	1
green witch .	1	1

$$p_1 = \frac{6}{11} \qquad p_2 = \frac{3}{10}$$

$$p_3 = \frac{0+0+0+0+0+0+0+1+1}{1+1+1+1+1+1+1+1+1} = \frac{2}{9}$$

Example: Computing BLEU

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

4-gram	4-gram Frequency (Candidate)	4-gram Frequency (Reference)
Usman did not give	1	0
did not give a	1	0
not give a slap	1	0
give a slap to	1	0
a slap to the	1	0
slap to the green	1	0
to the green witch	1	0
the green witch .	1	1

$$BLEU = BP \ * \ \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

$$p_1 = \frac{6}{11} \qquad p_2 = \frac{3}{10} \qquad p_3 = \frac{2}{9}$$

$$p_4 = \frac{0+0+0+0+0+0+0+1}{1+1+1+1+1+1+1+1} = \frac{1}{8}$$

Example: Computing BLEU

$l_c = 11$

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$r = 7$

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP * \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

$$p_1 = \frac{6}{11} \quad p_2 = \frac{3}{10} \quad p_3 = \frac{2}{9} \quad p_4 = \frac{1}{8}$$

$$BP = 1$$

Example: Computing BLEU

$l_c = 11$

Usman didn't slap the green witch.

Usman did not give a slap to the green witch.

$r = 7$

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \min(\mathrm{c}(n-gram), \mathrm{c}_{\max}(n-gram))}{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} \mathrm{c}(n-gram)}$$

$$BP = \begin{cases} 1 & if\ l_c > r \\ e^{(1-\frac{r}{l_c})} & if\ l_c \leq r \end{cases}$$

$$BLEU = BP * \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n\right)$$

$$p_1 = \frac{6}{11} \qquad p_2 = \frac{3}{10} \qquad p_3 = \frac{2}{9} \qquad p_4 = \frac{1}{8}$$

$$BP = 1$$

$$BLEU = 1 * \exp\left(\frac{1}{4}\sum_{n-1}^{4} \log p_n\right) = 1 * \exp\left(\frac{1}{4} * (\log .55 + \log .3 + \log .22 + \log .125)\right) = 1 * \exp(-.59) = 0.55$$

Limitations of BLEU

- Word or phrase order is of minimal importance
  - When computing unigram precision, a word can exist anywhere in the translation!
- Does not consider word similarity
- Relatively low correlation with human ratings
- Nonetheless, BLEU is reasonable to use in cases when a quick, automated metric is needed to assess translation performance

Summary: Machine Translation

- **Machine translation** is the process of automatically converting a text from one language to another
- Many approaches to machine translation exist
  - **Classical** machine translation
  - **Statistical** machine translation
  - **Neural** machine translation
- Machine translation is typically evaluated using metrics designed to consider both **fluency** and **fidelity**
- Computing **BLEU scores** is a common automated way to evaluate machine translation approaches