CS 4650/7650:
Natural Language Processing

Machine Translation I

Diyi Yang

Slides credit to Yulia Tsvetkov (CMU)

Machine Translation

Tower of Babel

Google Translate

Text | Documents

DETECT LANGUAGE | ENGLISH | SPANISH | FRENCH

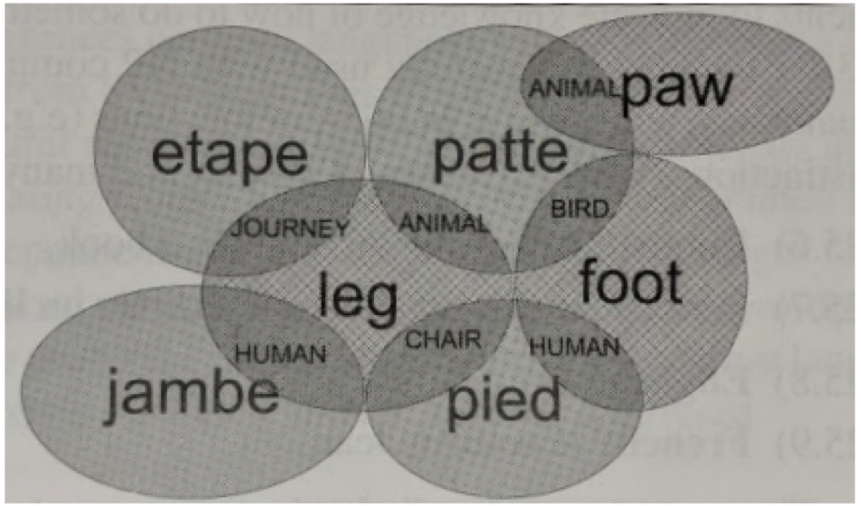
ENGLISH | SPANISH | ARABIC

Search languages

Detect language	Danish	Hmong	Lithuanian	Romanian	Telugu
Afrikaans	Dutch	Hungarian	Luxembourgish	Russian	Thai
Albanian	English	Icelandic	Macedonian	Samoan	Turkish
Amharic	Esperanto	Igbo	Malagasy	Scots Gaelic	Turkmen
Arabic	Estonian	Indonesian	Malay	Serbian	Ukrainian
Armenian	Filipino	Irish	Malayalam	Sesotho	Urdu
Azerbaijani	Finnish	Italian	Maltese	Shona	Uyghur
Basque	French	Japanese	Maori	Sindhi	Uzbek
Belarusian	Frisian	Javanese	Marathi	Sinhala	Vietnamese
Bengali	Galician	Kannada	Mongolian	Slovak	Welsh
Bosnian	Georgian	Kazakh	Myanmar (Burmese)	Slovenian	Xhosa
Bulgarian	German	Khmer	Nepali	Somali	Yiddish
Catalan	Greek	Kinyarwanda	Norwegian	Spanish	Yoruba
Cebuano	Gujarati	Korean	Odia (Oriya)	Sundanese	Zulu

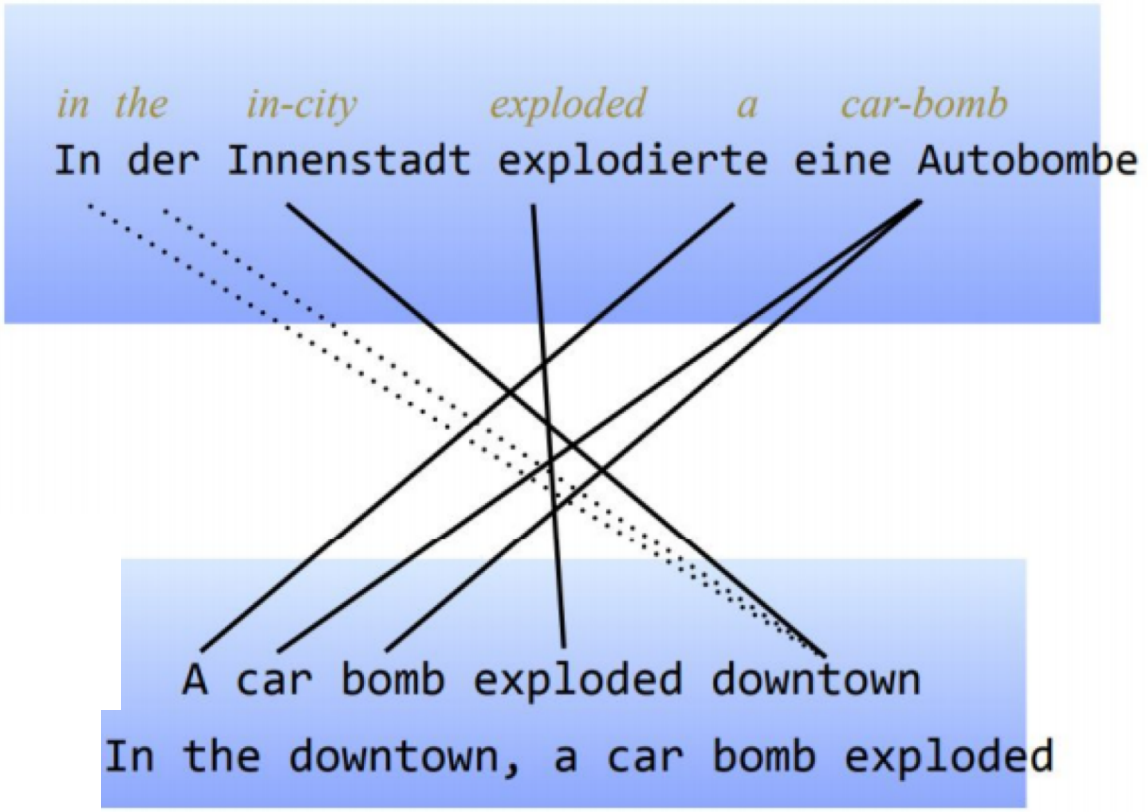
Dictionaries

- English: leg, foot, paw
- French: jambe, pied, patte, etape



Challenges

- Ambiguities
  - Words
  - Morphology
  - Syntax
  - Semantics
  - Pragmatics
- Gaps in data
  - Availability of corpus
  - Commonsense knowledge
- Understanding of context, connotation, social norms, etc



Research Problems

- How can we formalize the process of learning to translate from examples?
- How can we formalize the process of finding translations for new inputs?
- If our model produces many outputs, how do we find the best one?
- If we have a gold standard translation, how can we tell if our output is good or bad?

Two Views Of MT

MT as Code Breaking

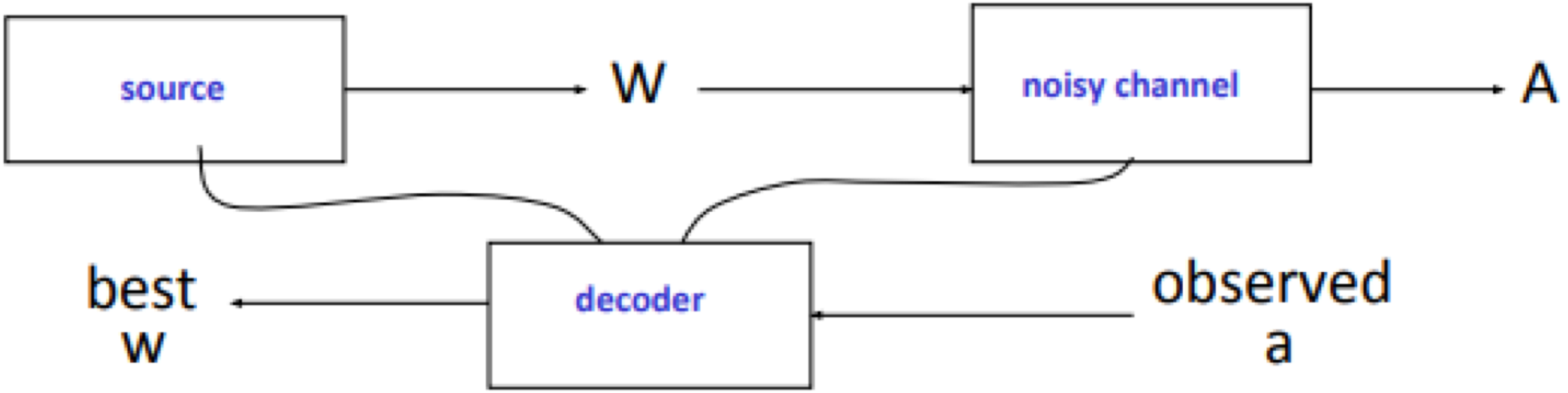
One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: '***This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.***'

Warren Weaver to Norbert Wiener, March, 1947

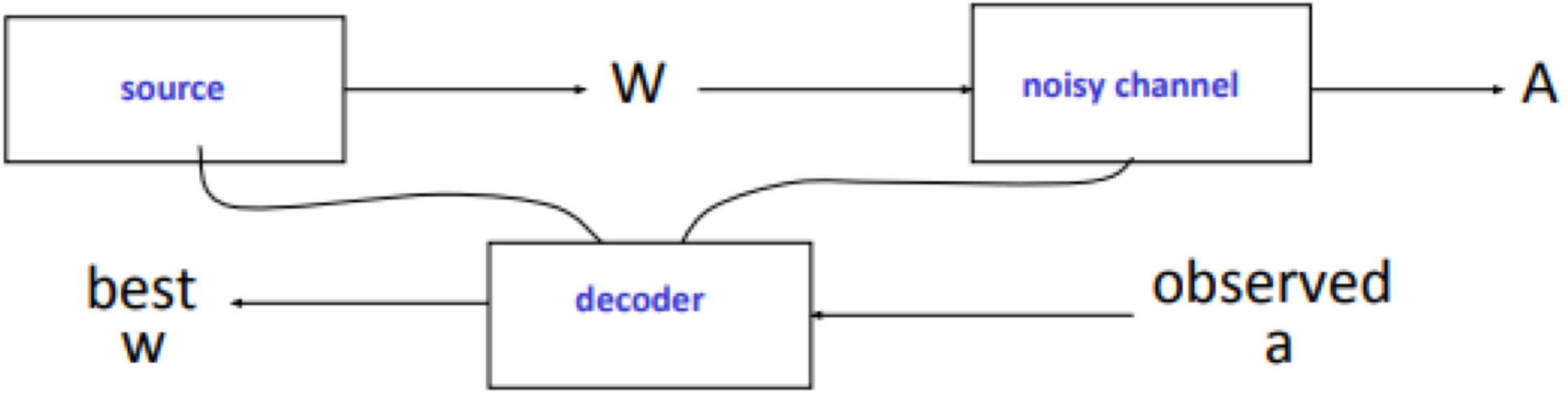
The Noisy-Channel Model

Claude Shannon. "A Mathematical Theory of Communication" 1948.

The Noisy-Channel Model



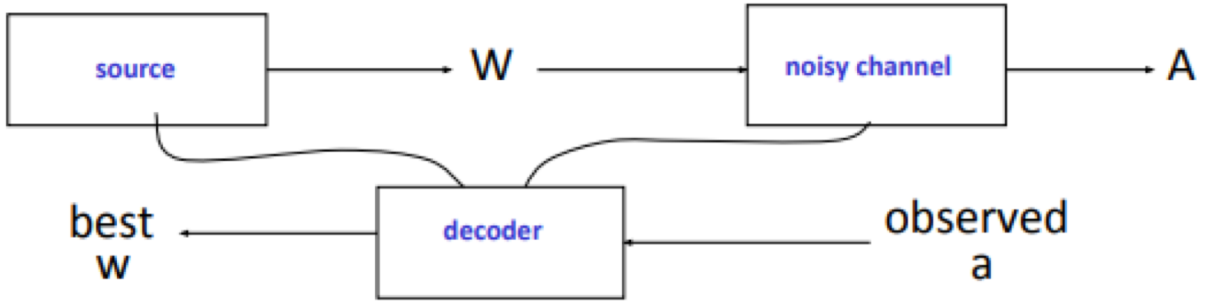
The Noisy-Channel Model



We want to predict a sentence given acoustics:

$$w^* = \arg\max_{w} P(w|a)$$

The Noisy-Channel Model



$$w^* = \arg\max_{w} P(w|a)$$

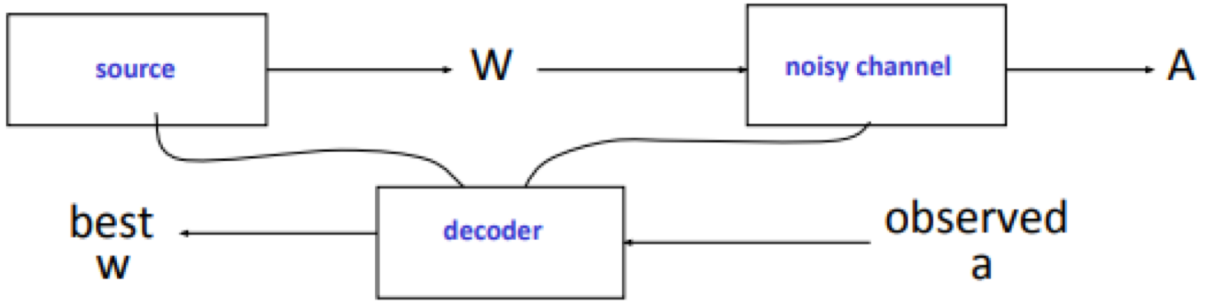
$$= \arg\max_{w} \boldsymbol{P(a|w)}\ \boldsymbol{P(w)} \ / \ P(a)$$

$$= \arg\max_{w} \boldsymbol{P(a|w)}\ \boldsymbol{P(w)}$$

Channel model

Source model

The Noisy-Channel Model



$$w^* = \arg\max_{w} P(w|a)$$

$$= \arg\max_{w} \boldsymbol{P(a|w)}\ \boldsymbol{P(w)} \ / \ P(a)$$

$$= \arg\max_{w} \boldsymbol{P(a|w)}\ \boldsymbol{P(w)}$$

Likelihood
Acoustic model (HMMs)
Translation model

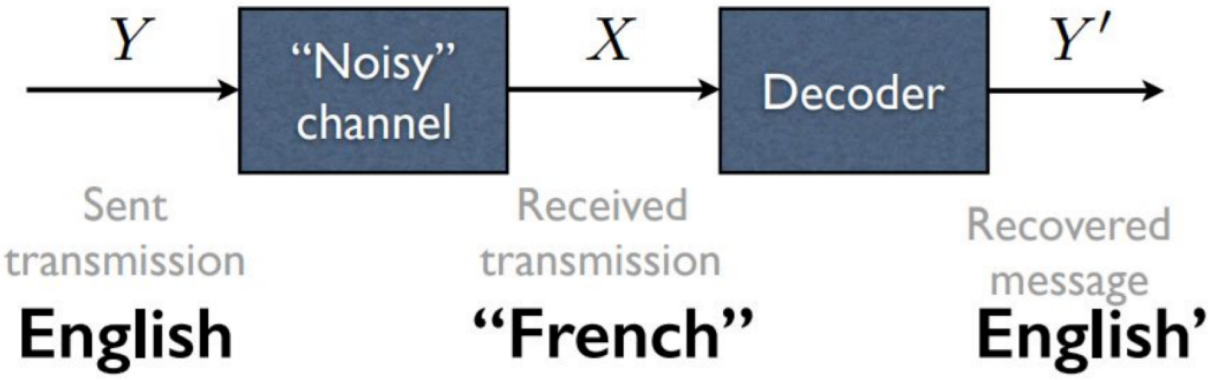
Prior
Language model: Distributions over sequence of words

The Noisy-Channel Model

$$\hat{e} = \arg\max_{e} p_{\varphi}(e) \times p_{\theta}(f \mid e)$$

Language model

Translation model



MT as Direct Modeling

$$\hat{e} = \arg\max_{e} p_\lambda(e \mid f)$$

target

source

- One model does everything
- Trained to reproduce a corpus of translations

Two Views of MT

- Code breaking (aka the noisy channel, Bayes rule)
  - I know the target language
  - I have example translations texts (example enciphered data)
- Direct modeling (aka pattern matching)
  - I have really good learning algorithms and a bunch of example inputs (source language sentences) and outputs (target language translations)

Which is Better?

- **Noisy channel** - $p_{\varphi}(e) \times p_{\theta}(f \mid e)$
  - Easy to use monolingual target language data
  - Search happens under a product of two models (individual models can be simple, product can be powerful)
- **Direct Model** - $p_{\lambda}(e \mid f)$
  - Directly model the process you care about
  - Model must be very powerful

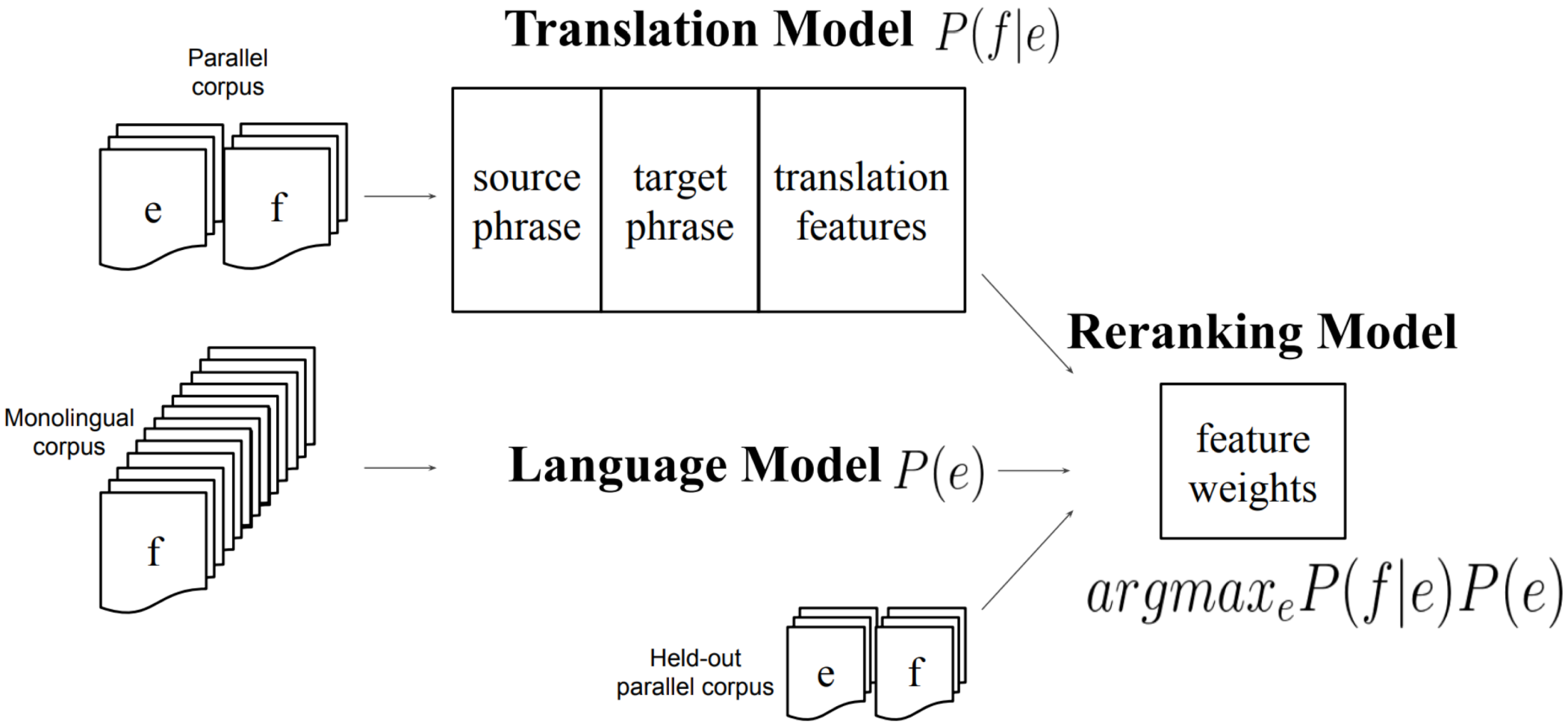
Where are we in 2020?

- Direct modeling is where most of the action is
  - Neural networks are very good at generalizing and conceptually very simple
  - Inference in “product of two models” is hard
- Noisy channel ideas are incredibly important and still play a big role in how we think about translation

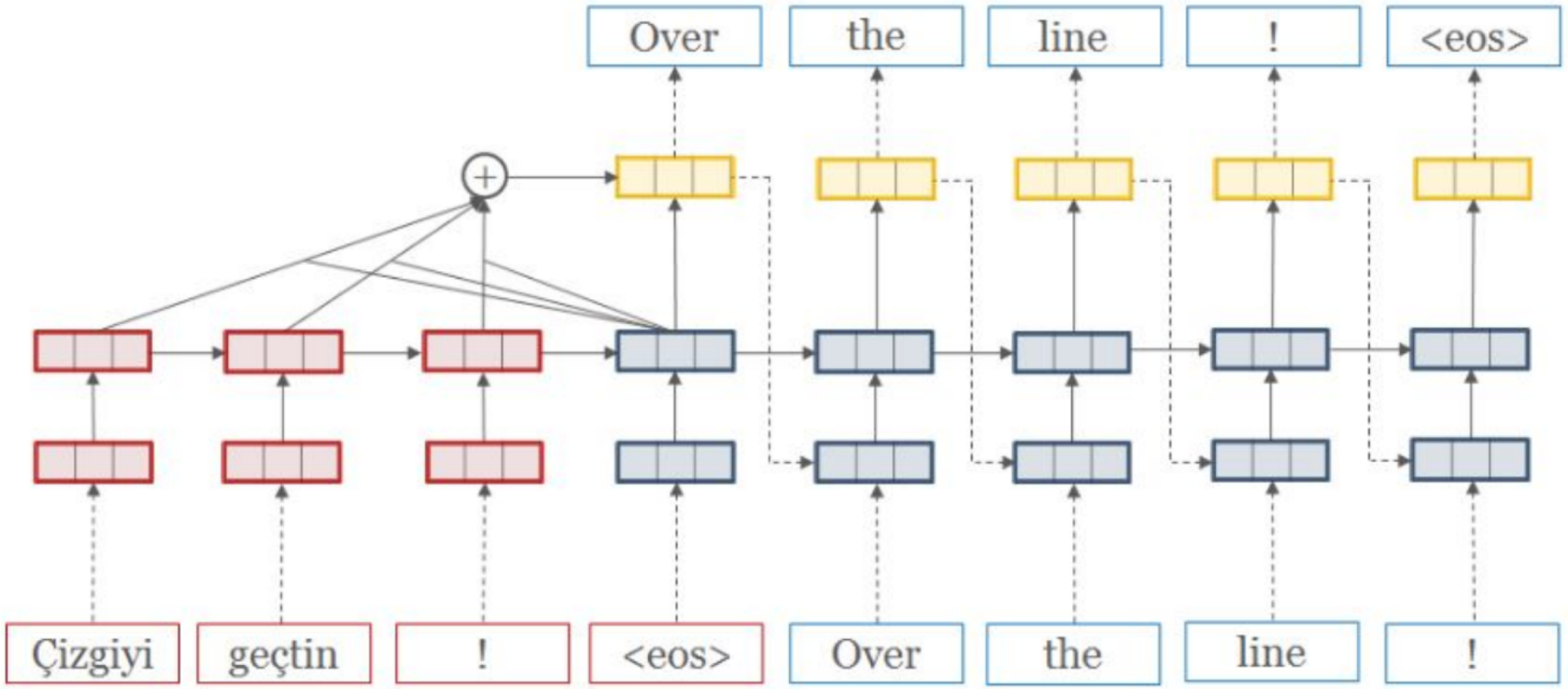
Two Views of MT

- Noisy channel $\hat{e} = \arg\max_{e} p_{\varphi}(e) \times p_{\theta}(f \mid e)$
- Direct $\hat{e} = \arg\max_{e} p_{\lambda}(e \mid f)$

Noisy Channel: Phrase-Based MT



Neural MT: Conditional Language Modeling



http://opennmt.net/

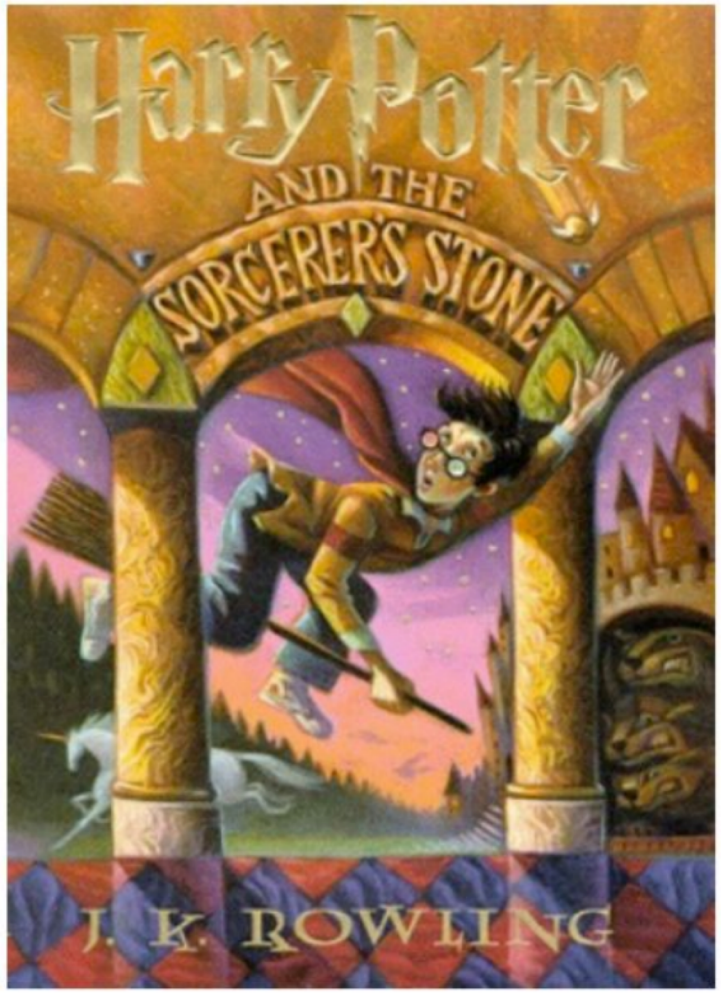
A Common Problem

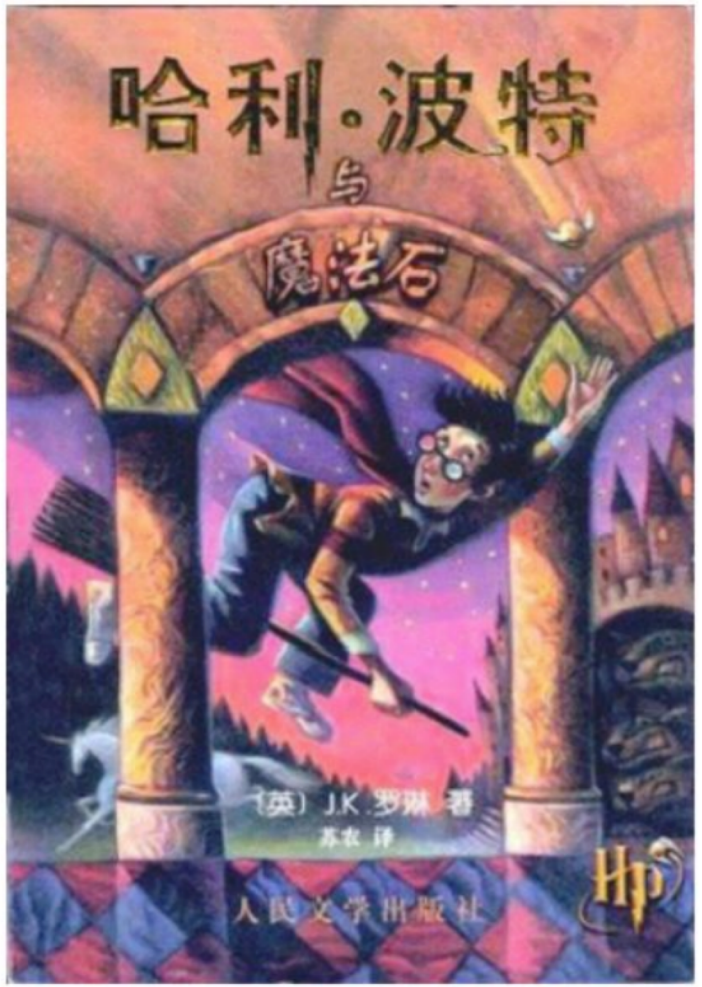
- Noisy channel $\hat{e} = \arg\max_{e} p_{\varphi}(e) \times \boldsymbol{p_{\theta}(f \mid e)}$
- Direct $\hat{e} = \arg\max_{e} \boldsymbol{p_{\lambda}(e \mid f)}$

- Both models must assign probabilities to how a sentence in one language translates into a sentence in another language

Learning From Data

Parallel Corpora





Parallel Corpora

				Sm.	Lg.
			CLASSIC SOUPS		
清燉雞湯	57.		House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot)	1.50	2.75
雞飯湯	58.		Chicken Rice Soup	1.85	3.25
雞麵湯	59.		Chicken Noodle Soup	1.85	3.25
廣東雲吞	60.		Cantonese Wonton Soup	1.50	2.75
蕃茄蛋湯	61.		Tomato Clear Egg Drop Soup	1.65	2.95
雲吞湯	62.		Regular Wonton Soup	1.10	2.10
酸辣湯	63.	❧	Hot & Sour Soup	1.10	2.10
蛋花湯	64.		Egg Drop Soup	1.10	2.10
雲蛋湯	65.		Egg Drop Wonton Mix	1.10	2.10
豆腐菜湯	66.		Tofu Vegetable Soup	NA	3.50
雞玉米湯	67.		Chicken Corn Cream Soup	NA	3.50
蟹肉玉米湯	68.		Crab Meat Corn Cream Soup	NA	3.50
海鮮湯	69.		Seafood Soup	NA	3.50

Parallel Corpora (mining parallel data from microblogs Ling et al., 2013)

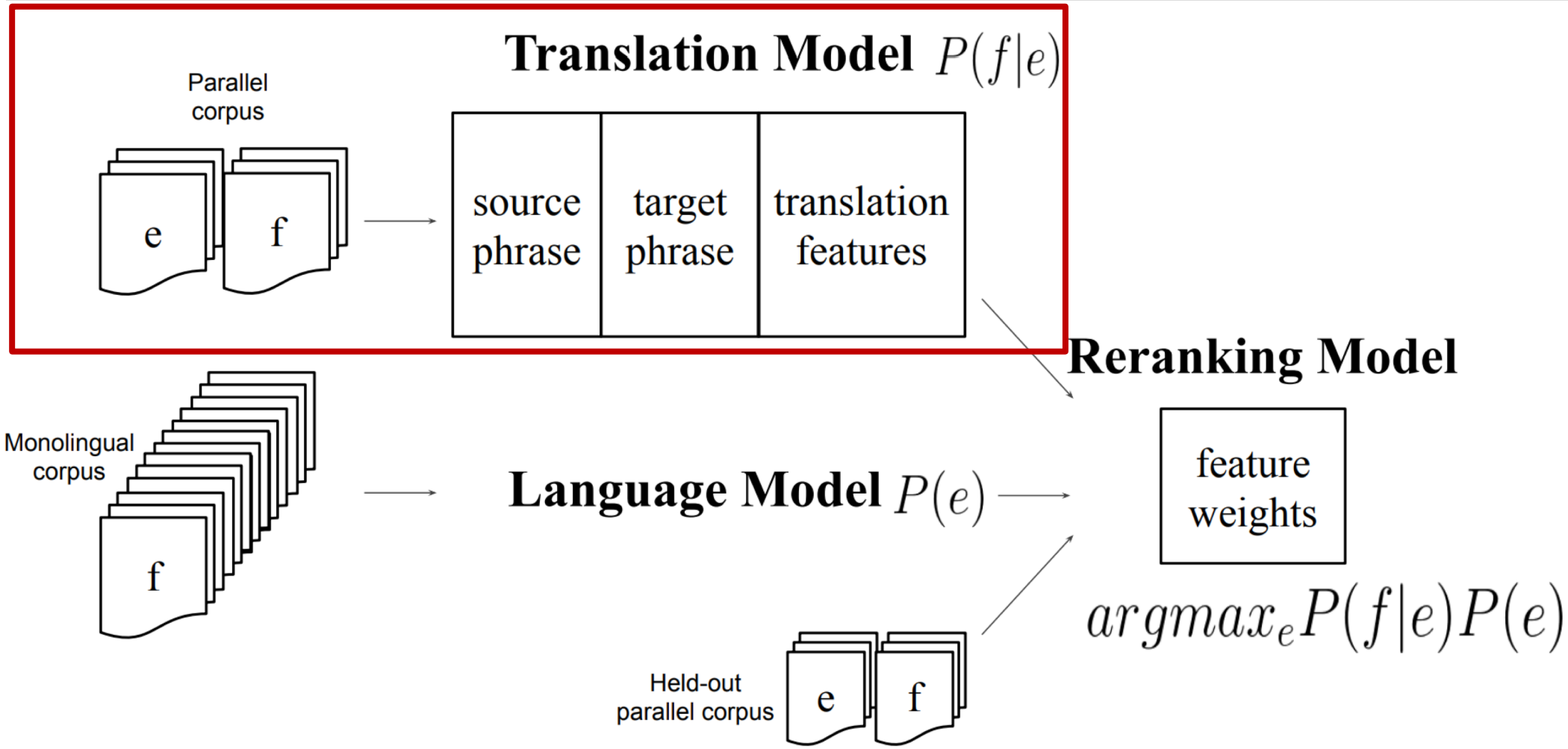
	ENGLISH	MANDARIN
1	i **wanna** live in a wes anderson world	我想要生活在Wes Anderson的世界里
2	Chicken soup, corn never truly digests. **TMI**.	鸡汤吧，玉米神马的从来没有真正消化过.恶心
3	To DanielVeuleman **yea iknw imma** work on that	对DanielVeuleman说，是的我知道，我正在向那方面努力
4	**msg 4** Warren G his **cday** is today 1 **yr** older.	发信息给Warren G，今天是他的生日，又老了一岁了。
5	Where **the hell** have you been all these years?	这些年你**TMD**到哪去了
	ENGLISH	ARABIC
6	It's **gonna** be a warm week!	الاسبو ع الياي حر
7	onni this gift only **4 u**	أوني هذة الهدية فقط لك
8	sunset in aqaba **:)**	غروب الشمس في العقبة:)
9	RT @MARYAMALKHAWAJA: there is a call for widespread protests in #bahrain **tmrw**	هناك نداء لظاهرات في عدة مناطق غدا

Table 2: Examples of English-Mandarin and English-Arabic sentence pairs. The English-Mandarin sentences were extracted from Sina Weibo and the English-Arabic sentences were extracted from Twitter. Some messages have been shorted to fit into the table. Some interesting aspects of these sentence pairs are marked in bold.

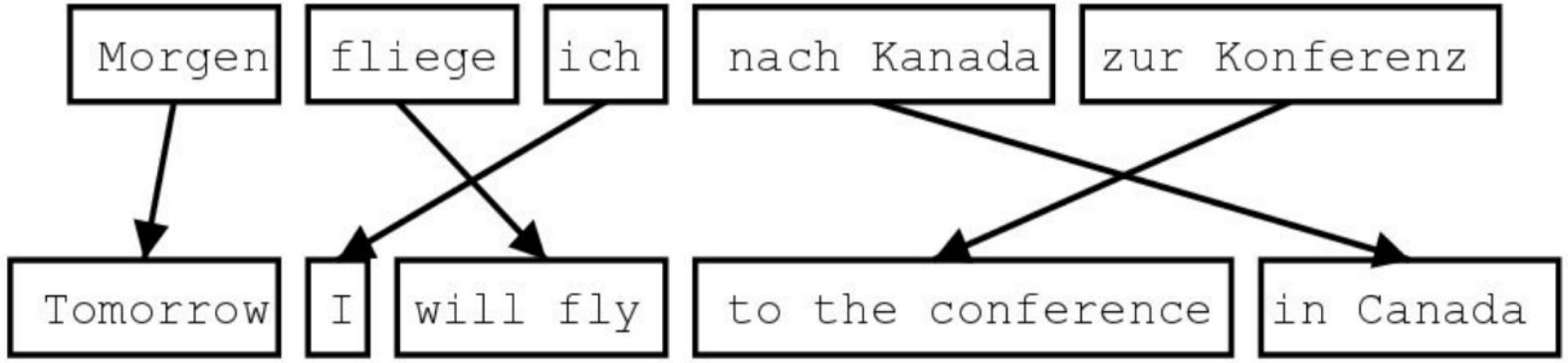
Discussions

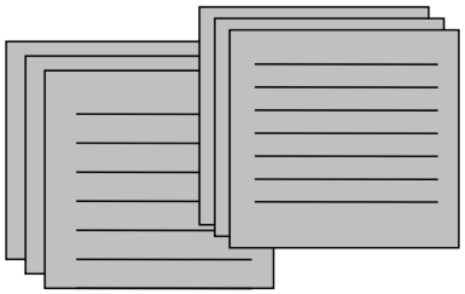
- There is a lot more monolingual data in the world than translated data
- Easy to get about 1 trillion words of English by crawling the web
- With some work, you can get 1 billion translated words of English-French
  - What about Japanese-Turkish?

Phrase-Based MT

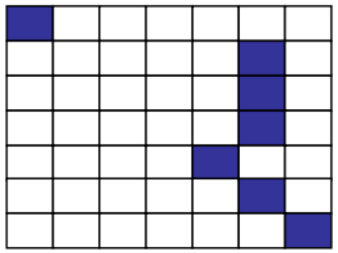


Construction of t-table





Sentence-aligned corpus



Word alignments

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table (translation model)

Word Alignment Models

Lexical Translation

- How do we translate a word? Look it up in the dictionary

  Haus – house, building, home, household, shell

- Multiple translations
  - Some more frequent than others
  - Different word senses, different registers, different functions
  - *House, home* are common
- *Shell* is specialized (the Haus of a snail is a shell)

How Common is Each?

- Look at a parallel corpus (German text along with English translation)

Translation of Haus	Count
house	8000
building	1600
home	200
household	150
shell	50

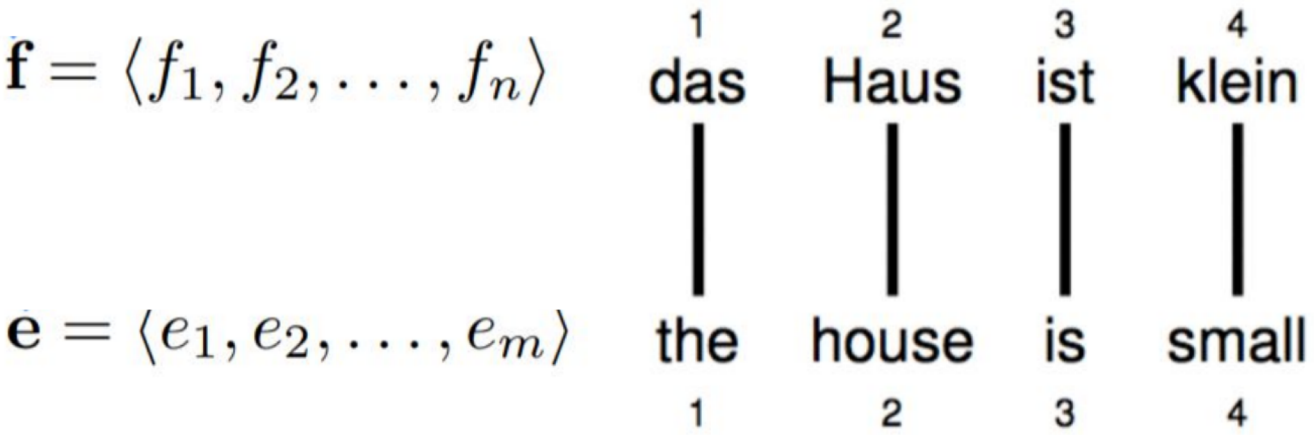
Estimate Translation Probabilities

- Maximum likelihood estimation

$$\hat{p}_{\text{MLE}}(e \mid \texttt{Haus}) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

Word Alignment

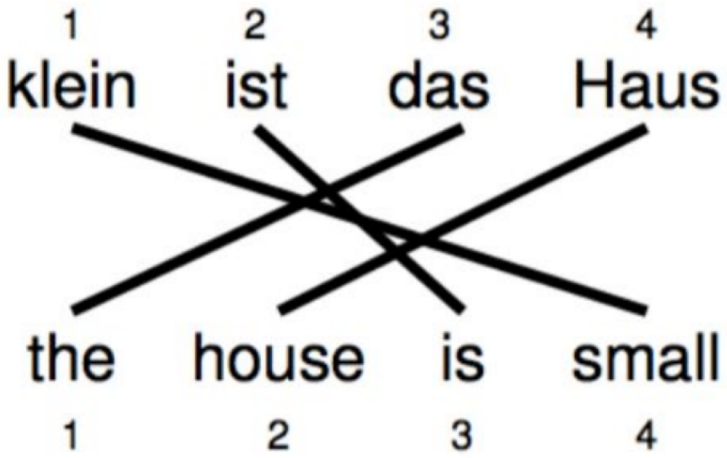
- Alignment can be visualized by drawing links between two sentences, and they are represented as vectors of positions



$$\mathbf{a} = (1, 2, 3, 4)^\top$$

Reordering

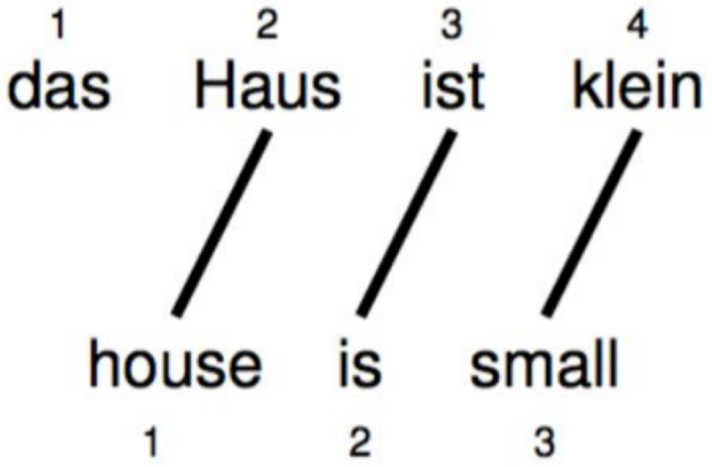
- Words may be reordered during translation



$$\mathbf{a} = (3, 4, 2, 1)^\top$$

Word Dropping

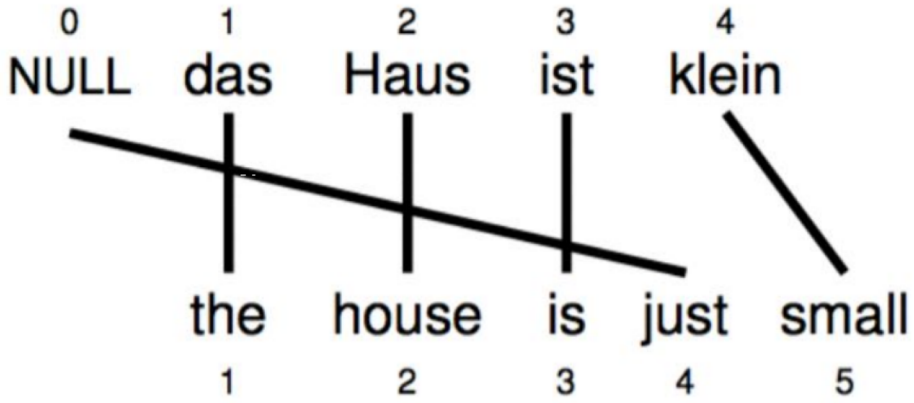
- A source word may not be translated at all



$$\mathbf{a} = (2, 3, 4)^\top$$

Word Insertion

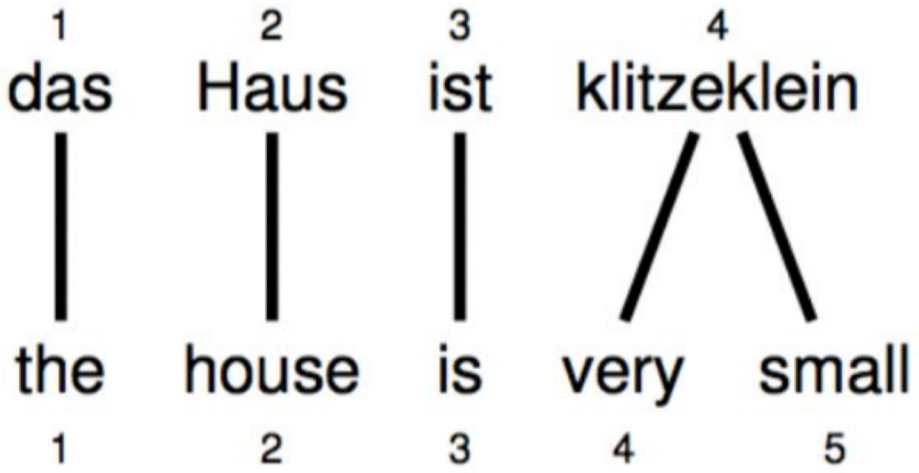
- Words may be inserted during translation
  - English just does not have an equivalent
  - But it must be explained – we typically assume every source sentence contains a NULL token



$$\mathbf{a} = (1, 2, 3, 0, 4)^\top$$

One-to-many Translation

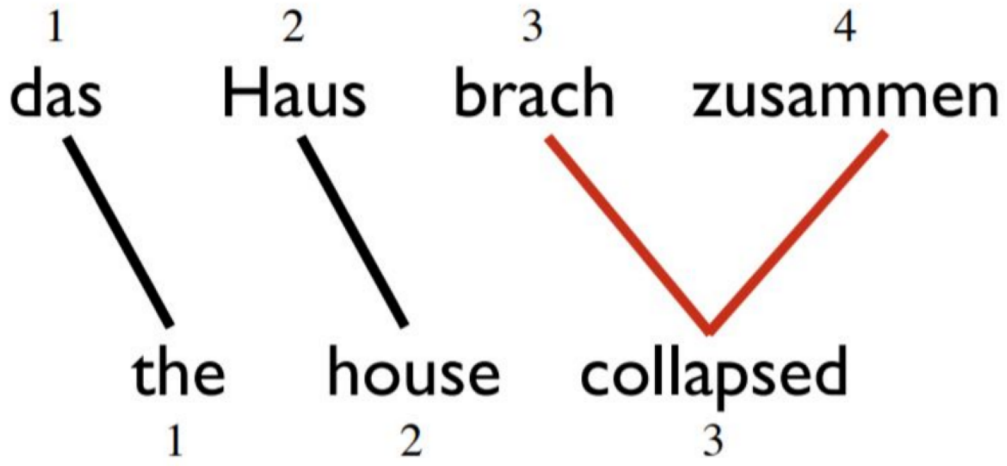
- A source word may translate into more than one target word



$$\mathbf{a} = (1, 2, 3, 4, 4)^\top$$

Many-to-one Translation

- More than one source word may not translate as a unit in lexical translation



$\mathbf{a} = ???$ $\mathbf{a} = (1, 2, (3, 4)^\top)^\top$?

Computing Word Alignments

- Word alignments are the basis for most translation algorithms
- Given two sentences F and E, find a good alignment
- But a word-alignment algorithm can also be part of a mini-translation model itself
- One the most basic alignment models is also a simplistic translation model

IBM Model 1

- Generative model: break up translation process into smaller steps
- Simplest possible lexical translation model
- Additional assumptions
  - All alignment decisions are independent
  - The alignment distribution for each a_i is uniform over all source words and NULL

Lexical Translation

- Goal: a model $p(\boldsymbol{e}|\boldsymbol{f}, m)$
  - Where **e** and **f** are complete English and Foreign sentences
  - $\boldsymbol{e} = < e_1, e_2, \ldots, e_m >$
  - $\boldsymbol{f} =< f_1, f_2, \ldots, f_n >$

Lexical Translation

- Goal: a model $p(\boldsymbol{e}|\boldsymbol{f}, m)$
  - Where **e** and **f** are complete English and Foreign sentences
- Lexical translation makes the following assumptions
  - Each word e_i in e is generated from exactly one word in f
  - Thus, we have an alignment a_i that indicates which word e_i "came from", specifically it came from f_{a_i}
  - Given the alignments **a**, translation decisions are conditionally independent of each other and depend only on the aligned source word f_{a_i}

Lexical Translation

- Putting our assumptions together, we have:

$$p(\boldsymbol{e}|\boldsymbol{f},m) = \sum_{\boldsymbol{a}\in[0,n]^m} p(\boldsymbol{a}|\boldsymbol{f},m) \times \prod_{i=1}^{m} p(e_i|f_{a_i})$$

Alignment × Translation | Alignment

IBM Model 1: P(E|F)

- Translation probability
  - For a foreign sentence $f = (f_1, \ldots, f_{l_f})$ of length l_f
  - To an English sentence $e = (e_1, \ldots, e_{l_e})$ of length l_e
  - With an alignment of each English word e_j to a foreign word f_i according to the alignment function $a: j \rightarrow i$

$$p(e, a \mid f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j \mid f_{a(j)})$$

  - Parameter ϵ is a normalization constant

Computing P(E|F) in IBM Model 1

$$p(e, a \mid f) = \underbrace{\frac{\epsilon}{(l_f + 1)^{l_e}}}_{p(a|f)} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

- A normalization factor, since there are $(l_f + 1)^{l_e}$ possible alignments
- Parameter ϵ is a normalization constant
- The probability of an alignment given the foreign sentence

Computing P(E|F) in IBM Model 1

$p(a|f)$ $p(e|f,a)$

$$p(e, a \mid f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

$$p(\boldsymbol{e}|\boldsymbol{f}) = \sum_{\boldsymbol{a}} p(\boldsymbol{e}, \boldsymbol{a} \mid \boldsymbol{f}) = \sum_{\boldsymbol{a}} p(\boldsymbol{a}|\boldsymbol{f}) \times \prod_{j=1}^{l_e} p(e_j | f_{a_j})$$

Example

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

$$p(e,a|f) = \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein})$$
$$= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4$$
$$= 0.0028\epsilon$$

Estimate Translation Probabilities

- Maximum likelihood estimation

$$\hat{p}_{\text{MLE}}(e \mid \texttt{Haus}) = \begin{cases} 0.8 & \text{if } e = \text{house}, \\ 0.16 & \text{if } e = \text{building}, \\ 0.02 & \text{if } e = \text{home}, \\ 0.015 & \text{if } e = \text{household}, \\ 0.005 & \text{if } e = \text{shell}. \end{cases}$$

Estimate Alignments Given t-table

- If we have translation probabilities

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

- The goal is to find the most probable alignment given a parameterized model

Estimating the Alignment

$$p(e, a \mid f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} p(\boldsymbol{e}, \boldsymbol{a} \mid \boldsymbol{f})$$

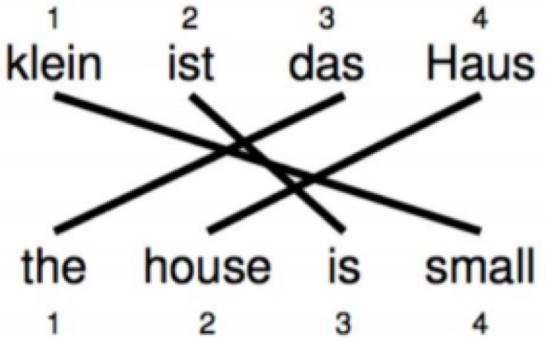
$$= \arg\max_{\boldsymbol{a}} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

$$= \arg\max_{\boldsymbol{a}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

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

$$a_i^* = \arg\max_{a_i=0}^{n} t(e_i | f_{a_i})$$

Learning Lexical Translation Models



- We'd like to estimate the lexical translation probabilities $t(e|f)$ from a parallel corpus but we do not have the alignments
- **Chick and egg problem**
  - If we had the alignments, we could estimate the parameters of our generative model (MLE)
  - If we had the parameters, we could estimate the alignments

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

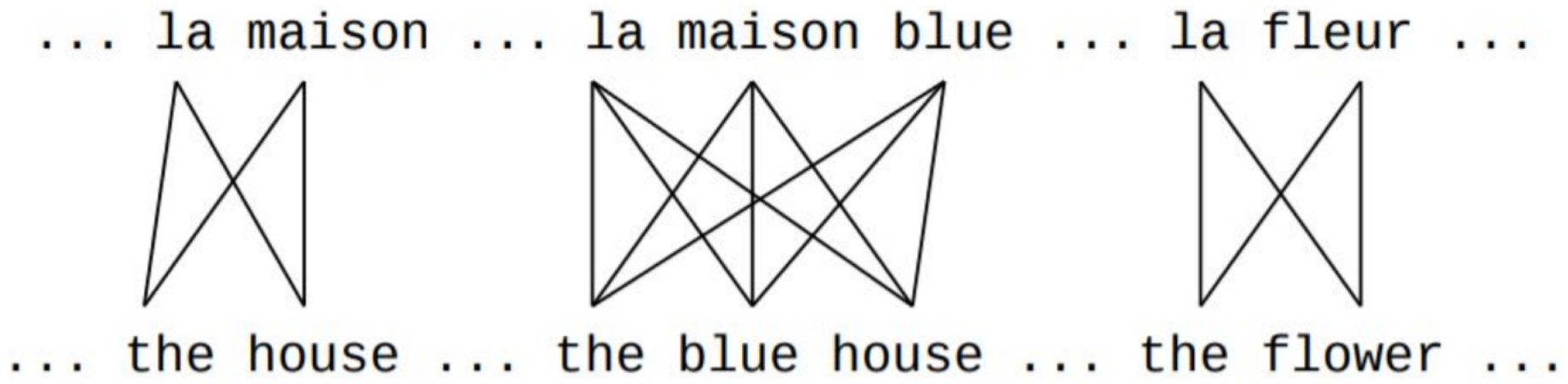
EM Algorithm

- Incomplete data
  - If we had complete data, we could estimate the model
  - If we had the model, we could fill in the gaps in the data

- **Expectation Maximization (EM)** in a nutshell
  1. Initialize model parameters (e.g., uniform, random)
  2. Assign probabilities to the missing data
  3. Estimate model parameters from completed data
  4. Iterate steps 2-3 until convergence

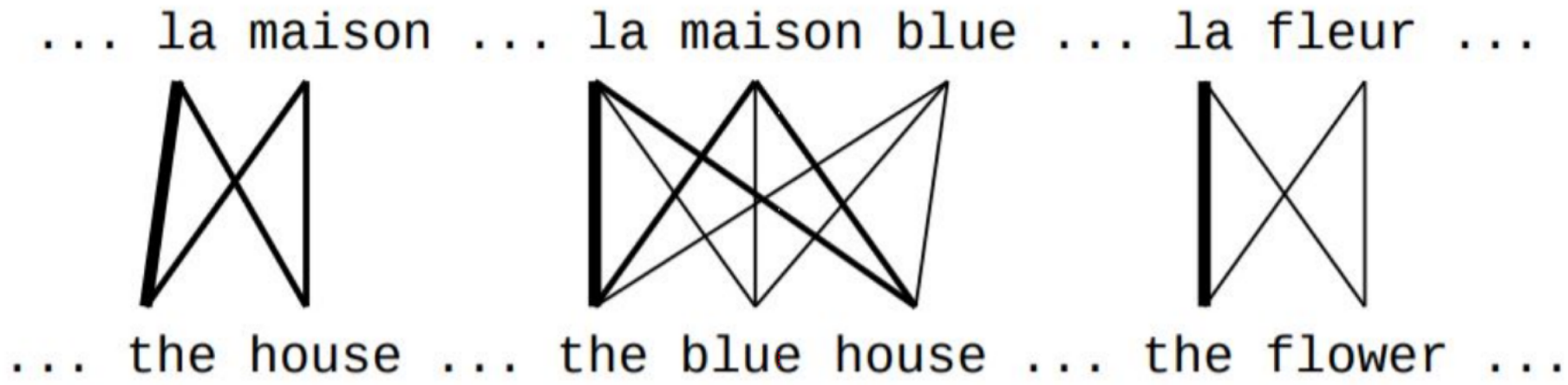
EM Algorithm

Kevin Knight's example



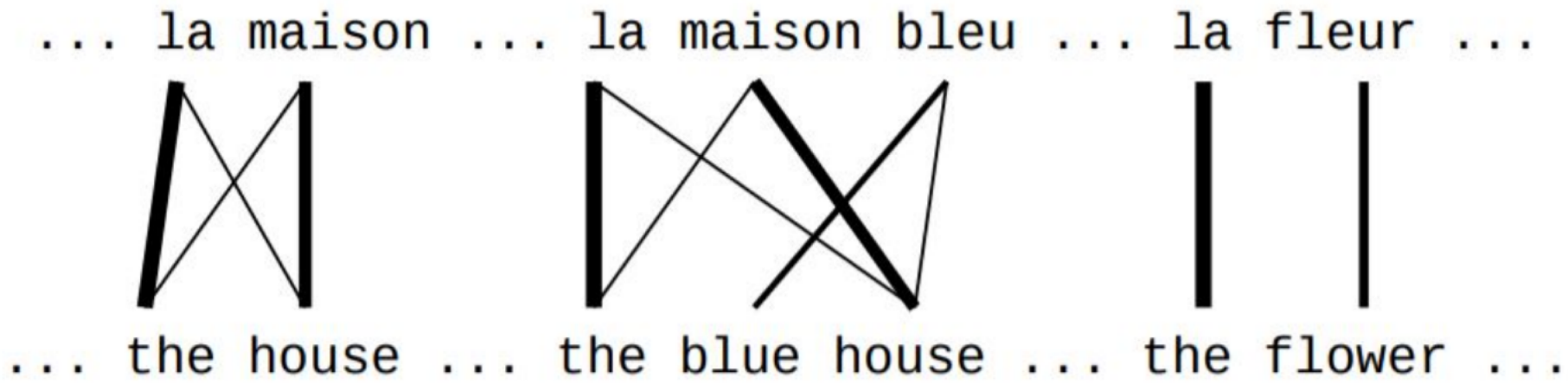
- Initial step: all word alignments equally likely
- Model learns that: e.g., *la* is often aligned with *the*

EM Algorithm



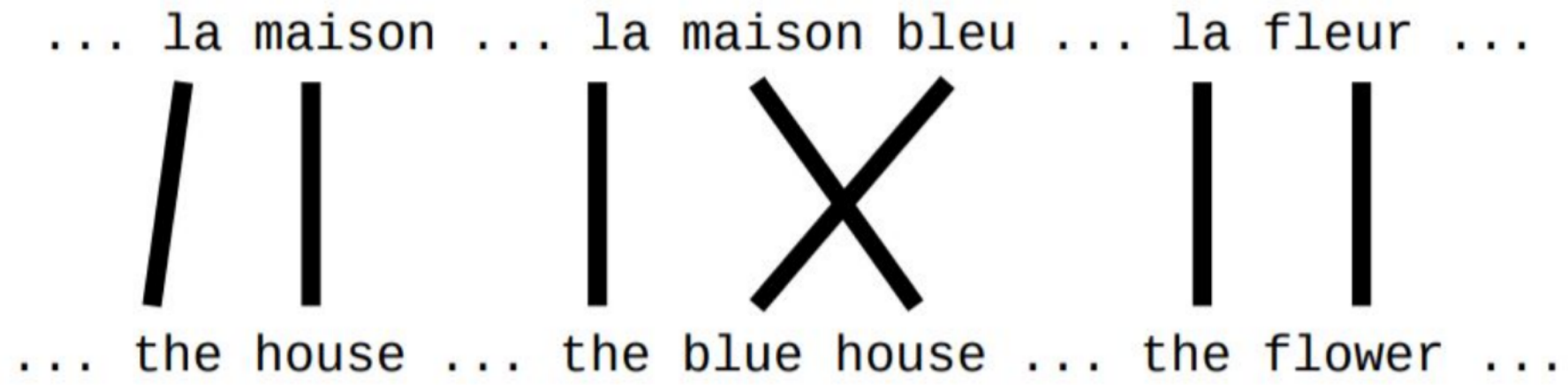
- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

EM Algorithm



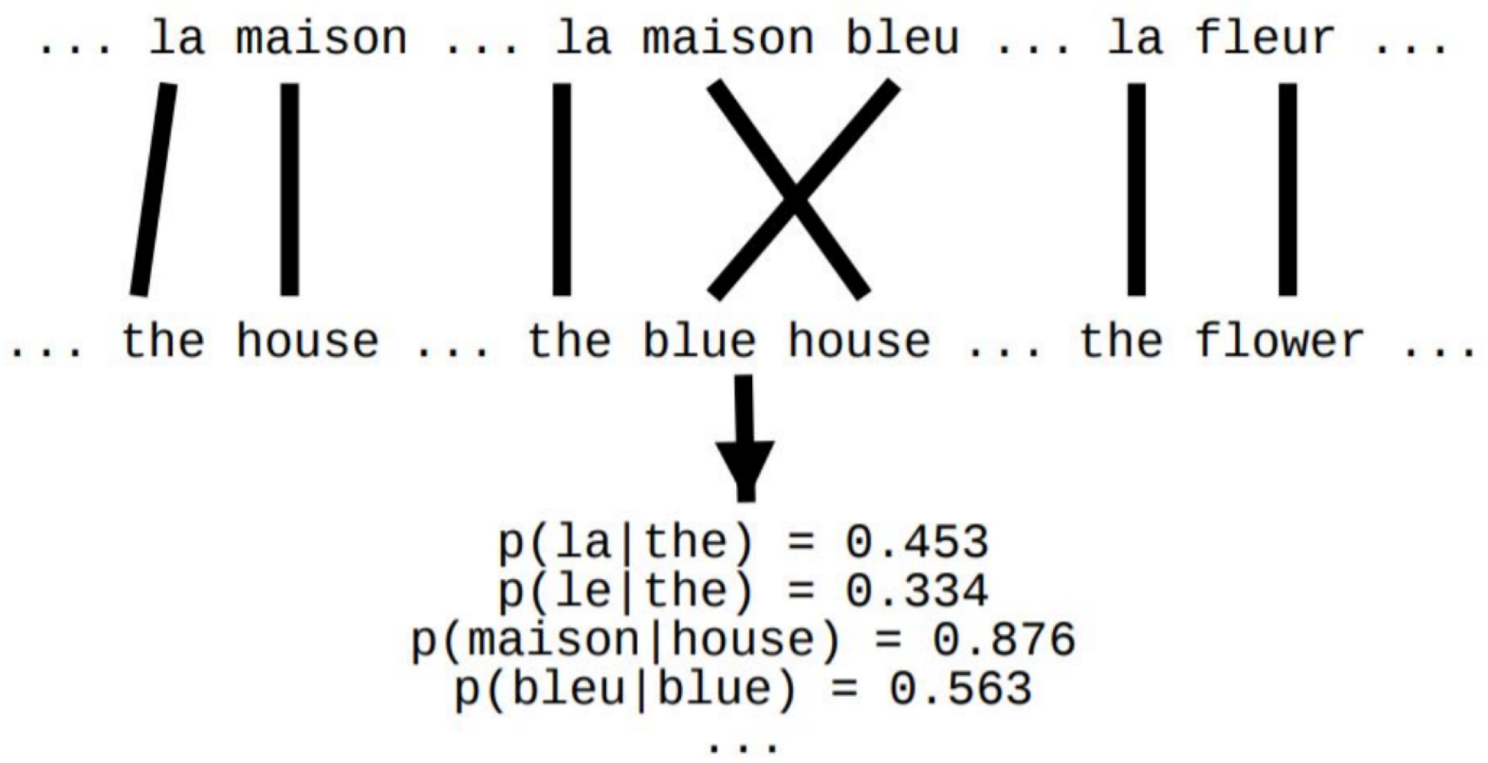
- After another iteration
- It becomes apparent that alignments, e.g., between *fleur* and *flower* are more likely

EM Algorithm



- Convergence
- Inherent hidden structure revealed by EM !

EM Algorithm



- **Parameter estimation from the aligned corpus**

The EM Algorithm for Word Alignment

- Initialize the mode, typically with uniform distributions
- Repeat
  - **E Step**: use the current model to compute the probability of all possible alignments of the training data
  - **M Step**: use these alignment probability estimates to re-estimate values for all of the parameters
- Until convergence (i.e., parameters no longer change)

IBM Model 1 and EM

t-table **Probabilities**

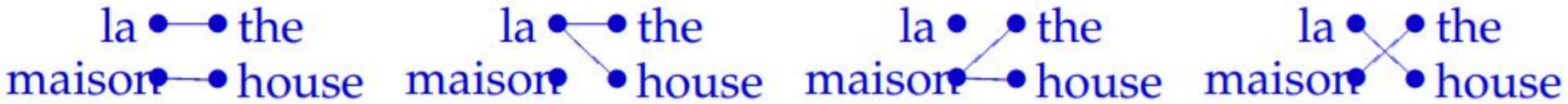
$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

IBM Model 1 and EM

t-table **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \qquad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \qquad p(\text{house}|\text{maison}) = 0.8$$

Alignments



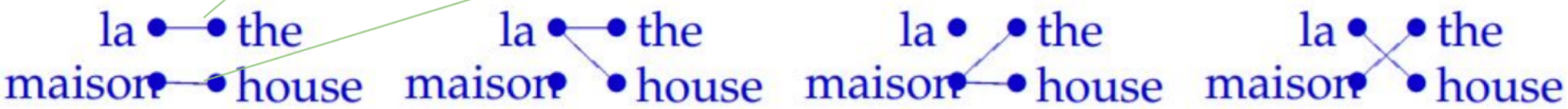
IBM Model 1 and EM

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $p(\text{house}|\text{la}) = 0.05$

$p(\text{the}|\text{maison}) = 0.1$ $p(\text{house}|\text{maison}) = 0.8$

Alignments



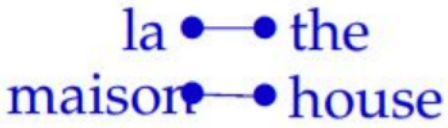
$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

IBM Model 1 and EM

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $p(\text{house}|\text{la}) = 0.05$
$p(\text{the}|\text{maison}) = 0.1$ $p(\text{house}|\text{maison}) = 0.8$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$

$p(\mathbf{e}, a|\mathbf{f}) = 0.035$

$p(\mathbf{e}, a|\mathbf{f}) = 0.08$

$p(\mathbf{e}, a|\mathbf{f}) = 0.005$

Chain rule: $p(a\ |\boldsymbol{e}, \boldsymbol{f}) = \frac{p(\boldsymbol{e}, a|\ \boldsymbol{f})}{p(\boldsymbol{e}|\boldsymbol{f})}$

IBM Model 1 and EM: Expectation Step

$$p(a\,|\boldsymbol{e},\boldsymbol{f}) = \frac{p(\boldsymbol{e},a|\,\boldsymbol{f})}{p(\boldsymbol{e}|\boldsymbol{f})}$$

$$= \frac{\frac{\epsilon}{(l_f+1)^{l_e}}\prod_{j=1}^{l_e} t(e_j|f_{a_j})}{\frac{\epsilon}{(l_f+1)^{l_e}}\prod_{j=1}^{l_e}\sum_{i=0}^{l_f} t(e_j|f_{a_j})}$$

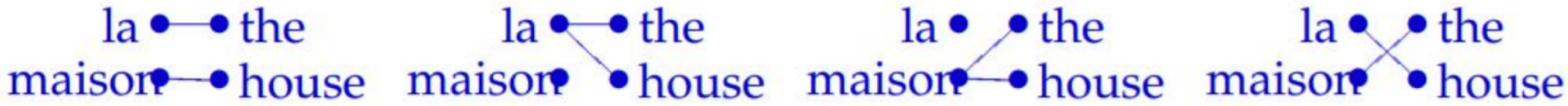
$$= \prod_{j=1}^{l_e}\frac{t(e_j|f_{a_j})}{\sum_{i=0}^{l_f} t(e_j|f_{a_j})}$$

IBM Model 1 and EM

t-table **Probabilities**

$p(\text{the}|\text{la}) = 0.7$ $p(\text{house}|\text{la}) = 0.05$
$p(\text{the}|\text{maison}) = 0.1$ $p(\text{house}|\text{maison}) = 0.8$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $p(a|\mathbf{e}, \mathbf{f}) = 0.007$

IBM Model 1 and EM: Maximization Step

- Now we have to collect counts
- Evidence from a sentence pair (**e,f**) that word e is a translation of word f:

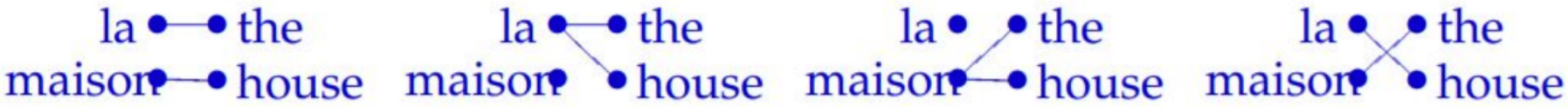
$$c(e|f) = \sum_{a} p(a|\boldsymbol{e},\boldsymbol{f}) \sum_{j=1}^{l_e} \delta\left(e, e_j\right)\delta(f, f_{a_j})$$

IBM Model 1 and EM

t-table **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8$$

Alignments



$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

E-step $p(a|\mathbf{e}, \mathbf{f}) = 0.824$ $p(a|\mathbf{e}, \mathbf{f}) = 0.052$ $p(a|\mathbf{e}, \mathbf{f}) = 0.118$ $p(a|\mathbf{e}, \mathbf{f}) = 0.007$

M-step **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \quad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \quad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

IBM Model 1 and EM

t-table **Probabilities**

$$p(\text{the}|\text{la}) = 0.7 \quad p(\text{house}|\text{la}) = 0.05$$
$$p(\text{the}|\text{maison}) = 0.1 \quad p(\text{house}|\text{maison}) = 0.8$$

E-step **Alignments**

$$p(a|\mathbf{e},\mathbf{f}) = 0.824 \quad p(a|\mathbf{e},\mathbf{f}) = 0.052 \quad p(a|\mathbf{e},\mathbf{f}) = 0.118 \quad p(a|\mathbf{e},\mathbf{f}) = 0.007$$

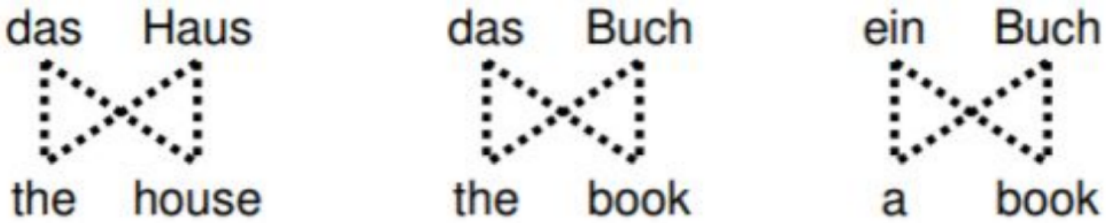
M-step **Counts**

$$c(\text{the}|\text{la}) = 0.824 + 0.052 \quad c(\text{house}|\text{la}) = 0.052 + 0.007$$
$$c(\text{the}|\text{maison}) = 0.118 + 0.007 \quad c(\text{house}|\text{maison}) = 0.824 + 0.118$$

Update t-table:

$$p(\text{the}|\text{la}) = c(\text{the}|\text{la})/c(\text{la})$$

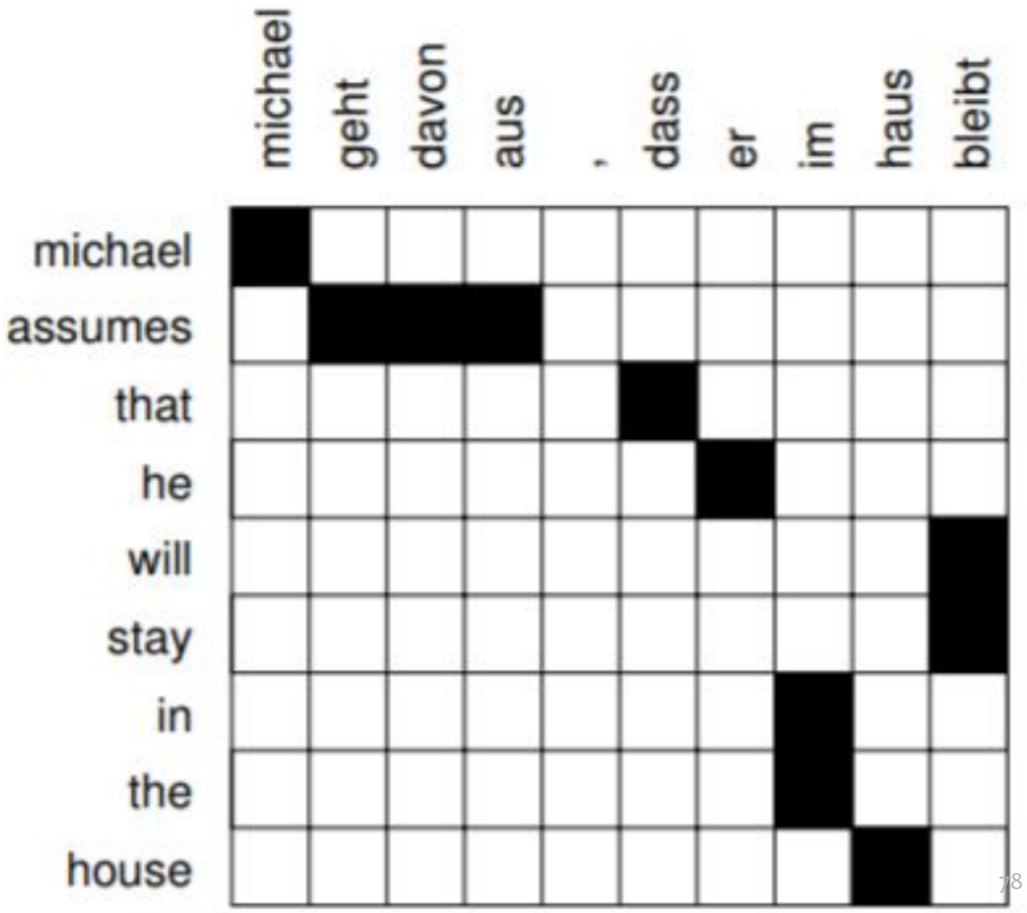
Convergence



e	f	initial	1st it.	2nd it.	3rd it.	...	final
the	das	0.25	0.5	0.6364	0.7479	...	1
book	das	0.25	0.25	0.1818	0.1208	...	0
house	das	0.25	0.25	0.1818	0.1313	...	0
the	buch	0.25	0.25	0.1818	0.1208	...	0
book	buch	0.25	0.5	0.6364	0.7479	...	1
a	buch	0.25	0.25	0.1818	0.1313	...	0
book	ein	0.25	0.5	0.4286	0.3466	...	0
a	ein	0.25	0.5	0.5714	0.6534	...	1
the	haus	0.25	0.5	0.4286	0.3466	...	0
house	haus	0.25	0.5	0.5714	0.6534	...	1

Word Alignment:

- Given a sentence pair, which words correspond to each other?



Higher IBM Models

IBM Model 1	Lexical translation
IBM Model 2	Adds absolute reordering model
IBM Model 3	Adds fertility model
IBM Model 4	Relative reordering model
IBM Model 5	Fixes deficiency

- Only IBM Model 1 has global maximum
  - Training of a higher IBM model builds upon previous model
- Computationally biggest change in Model 3

Word Alignment and IBM Models

- IBM models create a many-to-one mapping
  - Words are aligned using an alignment function
  - A function may return the same value for different input (one-to-many mapping)
  - A function cannot return multiple values for one input (no many-to-one mapping)
- Real world alignments have many-to-many mappings

Evaluation Metrics

- Manual evaluation is most accurate, but expensive
- Automated evaluation metrics:
  - Compare system hypothesis with reference translations
  - BiLingual Evaluation Understudy (BLEU) (Papineni et al., 2002):
    - Modified n-gram precision

$$p_n = \frac{\text{number of } n\text{-grams appearing in both reference and hypothesis translations}}{\text{number of } n\text{-grams appearing in the hypothesis translation}}$$

BLEU

$$\text{BLEU} = \exp \frac{1}{N} \sum_{n=1}^{N} \log p_n$$

- Two modifications:
  - To avoid $\log 0$, all precisions are smoothed
  - Each n-gram in reference can be used at most once
    - Ex. **Hypothesis**: *to to to to to* vs **Reference**: *to be or not to be* should not get a unigram precision of 1
- Precision-based metrics favor short translations
  - Solution: Multiply score with a brevity penalty (BP) for translations shorter than reference, $e^{1-r/h}$

BLEU Scores

	Translation	p_1	p_2	p_3	p_4	BP	BLEU
Reference	*Vinay likes programming in Python*						
Sys1	*To Vinay it like to program Python*	$\frac{2}{7}$	0	0	0	1	.21
Sys2	*Vinay likes Python*	$\frac{3}{3}$	$\frac{1}{2}$	0	0	.51	.33
Sys3	*Vinay likes programming in his pajamas*	$\frac{4}{6}$	$\frac{3}{5}$	$\frac{2}{4}$	$\frac{1}{3}$	1	.76

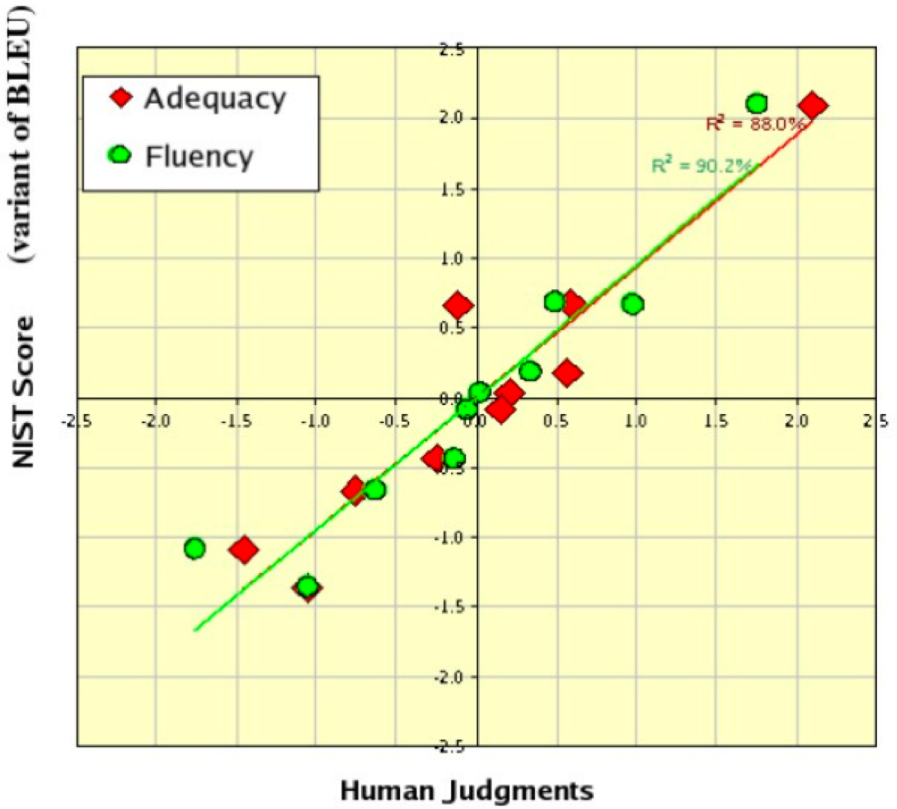
Sample BLEU scores for various system outputs

- Alternatives have been proposed:
  - METEOR: weighted F-measure
  - Translation Error Rate (TER): Edit distance between hypothesis and reference

Other Issues?

BLEU

- Correlates somewhat well with human judgments



(G. Doddington, NIST)

Problems with Lexical Translation

- Complexity – exponential in sentence length
- Weak reordering – the output is not fluent
- Many local decisions – error propagation