1 Recap from previous lectures

We can induce alignments using EM algorithm on parallel data without alignment (and do it for different models/generative stories).

Technically, what we find are only *probabilities* of particular alignments, not the alignment links themselves. The standard practice is to commit to the **most probable alignment** for every sentences pair.

1.1 Word alignments with IBM Models

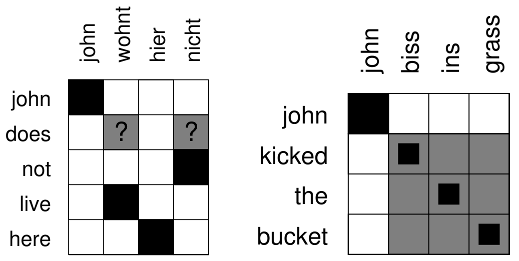
All generative stories decided on alignments for every output word according to the respective model. Remember, that this makes every output word be aligned to at most one foreign word (called also **many-to-one**[1]).

Although, IBM Models create a many-to-one mapping, the real-world alignment functions can be one-to-many or many-to-many mappings.

Examples:

1. function words

2. idioms



	john	wohnt	hier	nicht
john	■			
does		?		?
not				■
live		■		
here			■	

	john	biss	ins	grass
john	■			
kicked		■	▨	▨
the		▨	■	▨
bucket		▨	▨	■

What to align 'does' to?:

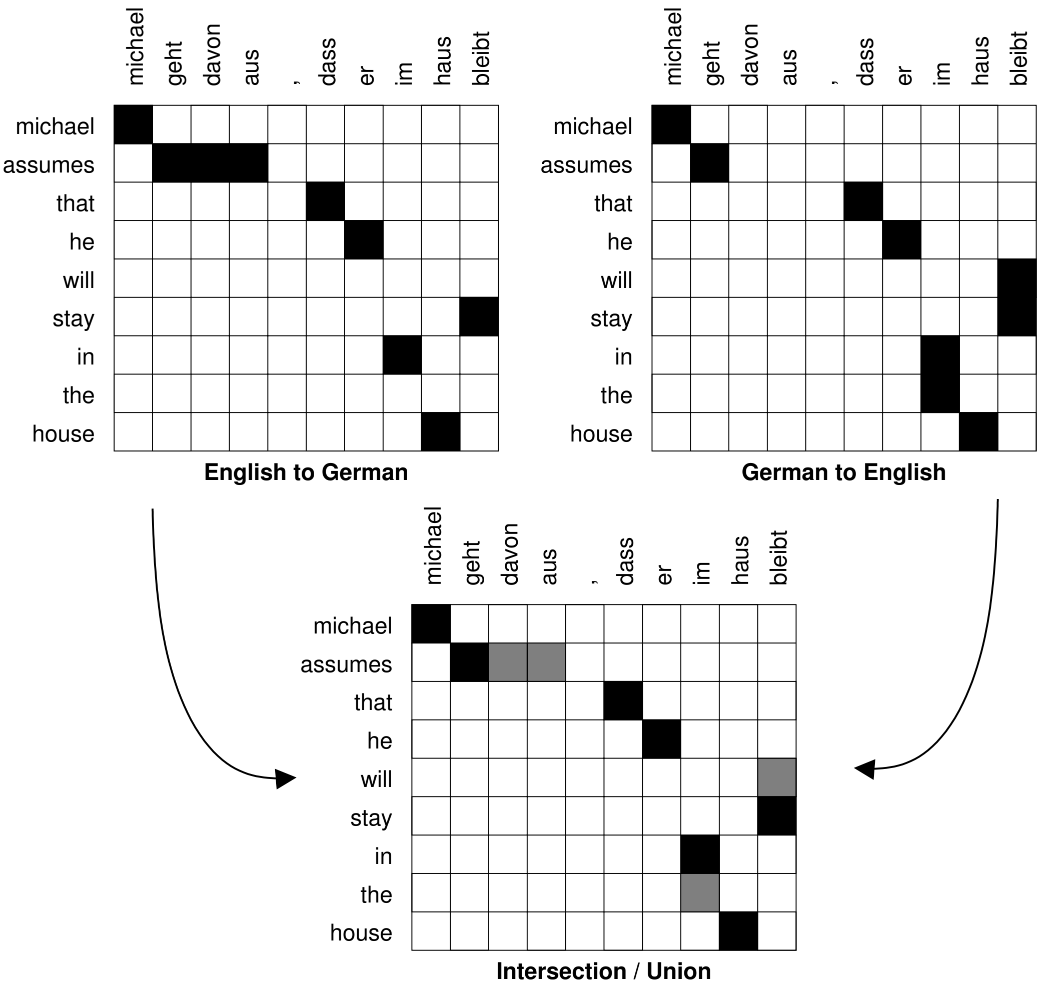
1. does → NULL (because there is no direct equivalent in German)

2. does → wohnt (because carries the number and the tense information)

3. does → nicht (because it is only necessary in negation)

[1] in the expression "[1]-to-[2]" alignments", [1] and [2] refer to, resp., output and input sentences.

Solution:
Straight forward idea is to use alignment in **both** directions, i.e. do the so-called **symmetrization** of the alignments (src - trg, trg - src).



English to German

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										
stay										■
in								■		
the										
house									■	

German to English

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■								
that						■				
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

Intersection / Union

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	▨	▨						
that						■				
he							■			
will										▨
stay										■
in								■		
the								▨		
house									■	

If we simply form the intersection of the alignments, this will promote very precise alignments, but we will miss some of them. That's why we do an intermediate thing and add alignment points from the union (*growing* the intersection).

The **grow-diag-final** symmetrization heuristic adds neighboring alignment points from the union and unaligned points to the intersection:

Neighborhoods:

	*	*	*
	*	$\boxed{\text{o}}$	*
	*	*	*

grow-diag-final(e2f,f2e)

```
neighboring = {(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)}
alignment $A$ = intersect(e2f,f2e);
grow-diag($A$);
final($A$);
```

grow-diag()

```
while new points added do
    for all English word $e \in [1...e_n]$, foreign word $f \in [1...f_n]$, $\boxed{(e,f) \in A}$ do
        for all neighboring alignment points ($e_{\text{new}}$, $f_{\text{new}}$) do
            if ($e_{\text{new}}$ unaligned OR $f_{\text{new}}$ unaligned) AND $(e_{\text{new}}, f_{\text{new}}) \in$ union(e2f,f2e) then
                add $(e_{\text{new}}, f_{\text{new}})$ to $A$
            end if
        end for
    end for
end while
```

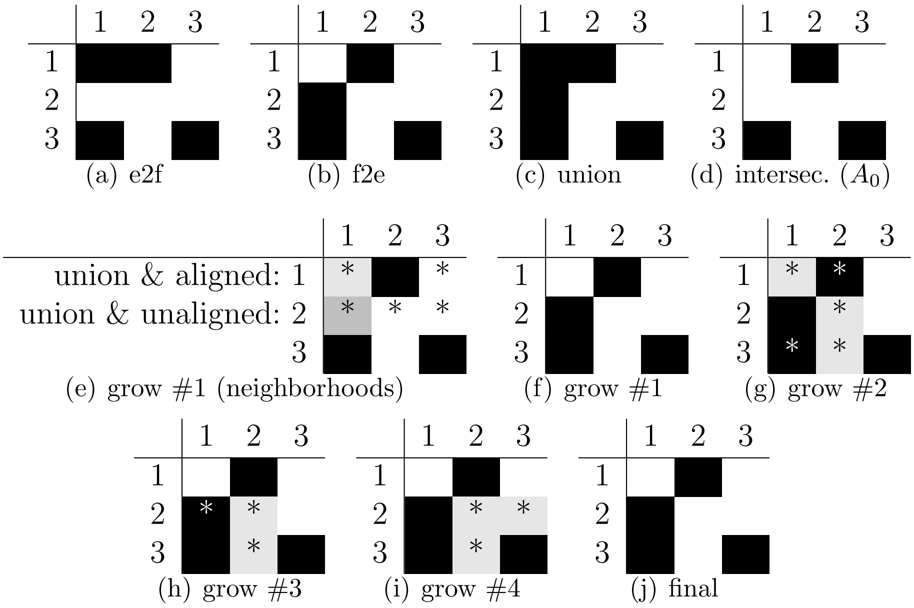
final()

```
for all English word $e_{\text{new}} \in [1...e_n]$, foreign word $f_{\text{new}} \in [1...f_n]$ do
    if ($e_{\text{new}}$ unaligned OR $f_{\text{new}}$ unaligned) AND $(e_{\text{new}}, f_{\text{new}}) \in$ union(e2f,f2e) then
        add $(e_{\text{new}}, f_{\text{new}})$ to $A$
    end if
end for
```

In brief, the grow-diag-final heuristics adds alignment points from the union only if the corresponding words were not already aligned elsewhere in the matrix.

A more restrictive variant is **grow-diag-final-and** where the OR in the **final()** is replaced with AND.

Example (assuming one-to-many alignments are also possible):
e2f = $\{1 \to 1,\ 1 \to 2,\ 3 \to 3,\ 3 \to 1\}$
f2e = $\{2 \to 1,\ 1 \to 2,\ 3 \to 3,\ 1 \to 3\}$



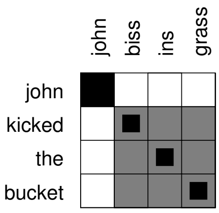
(a) e2f (b) f2e (c) union (d) intersec. (A_0)

(e) grow #1 (neighborhoods) (f) grow #1 (g) grow #2

(h) grow #3 (i) grow #4 (j) final

1.1.1 Measuring Alignment Quality

We can manually align a corpus with sure (S) and possible (P) alignment points (**note**: we assume $S \subseteq P$).



In the alignment between the two idiomatic expressions, all alignment points between each of the words are labeled as *possible* alignment points.

Alignment error rate:

$$\mathrm{AER}(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|S| + |A|} \qquad (1)$$

Intuition: precision $Prec = \frac{|A \cap P|}{|A|}$, recall $Recal = \frac{|A \cap S|}{|S|}$

2 Phrase-based SMT

2.1 Basics

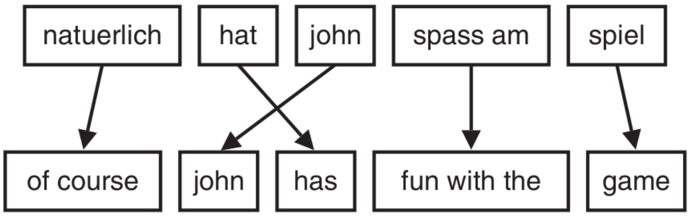
Word-based models translate words as atomic units. Phrase-based models translate *phrases* as atomic units.
While words are convenient to work with they are not the best atomic units:

1. probabilistic models impose limitations on possible word alignments that can be resolved by considering bigger chunks

2. translating groups of words helps resolve linguistic ambiguities in alignment (like in the “spass am”→“fun with the” example)

3. useful to memorize long correctly formed (i.e., words in the correct order) chunks; potentially whole sentences can be memorized

4. allows some model simplifications (now may not model fertility, etc.)



Roughly the translation process goes like this (note that this is no longer a proper generative story):

- input is segmented into phrases (not necessarily linguistically motivated)

- translated one-to-one into phrases in English

- possibly reordered.

Be aware that ”phrases” are **not the same as linguistic phrases**, but multiword expressions (including single words or non-sense pieces of sentences).

Example: "Spaß am" → "fun with the"

single-word translation:	**phrase-based translation:**
spaß → fun	The context of the prior word "spaß"
am → on the (0.4)	chooses the perfect translation for "am".
→ at the (0.4)	
⋮	
→ with the (0.0001)	

2.1.1 Advantages of Phrase-based SMT

- The local context can disambiguate translation options. (e.g. "spaß am" → "fun with the")
- Many-to-many translations can handle non-compositional phrases and idioms. (e.g. "beißt ins gras" → "kicks the bucket")
- Longer phrases can incorporate the correct word order. (e.g. "heim gehen" → "go home")

Real Example "den Vorschlag" (note a number of non-linguistic phrases):

English	$\phi(\bar{e}\|\bar{f})$	**English**	$\phi(\bar{e}\|\bar{f})$
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	...	...

2.2 Learning a Phrase Translation Table

So far we only argued that translating in phrases can be advantageous and showed a heuristic method to correct (grow) alignments found by probabilistic models. Now we discuss the algorithm for extraction phrases from (corrected) alignments.

4 stages:

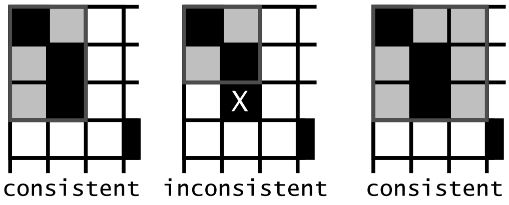
- IBM Models for word alignment.
- Symmetrization of alignment.
- Phrase extraction from symmetrized alignment table.
- Estimation of phrase translation probabilities.

Phrase-pair $(\bar{e}, \bar{f})$ is **consistent with alignment** A iff

$$\begin{aligned} & \forall e_i \in \bar{e} : \text{IF } (e_i, f_j) \in A, \text{ THEN } f_j \in \bar{f} \\ & \text{AND } \forall f_j \in \bar{f} : \text{IF } (e_i, f_j) \in A, \text{ THEN } e_i \in \bar{e} \\ \text{(non-emptiness condition)} \qquad & \text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} \text{ S.T. } (e_i, f_j) \in A \end{aligned}$$

Phrase pair $(\bar{e}, \bar{f})$ is consistent with alignment A if all words $f_1, \ldots, f_n \in \bar{f}$, that have alignment points in A, have these points with words $e_1, \ldots, e_n \in \bar{e}$ and vice versa.

Example:



2.2.1 ”Phrase extraction algorithm”

- Loop over English phrases and find minimal foreign phrases that build a consistent phrase pair.
- Bordering unaligned words may be included in consistent phrase.
- If only unaligned English words are found, no phrase is built.

Example:

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

michael	–	michael
michael assumes	–	michael geht davon aus
michael assumes	–	michael geht davon aus ,
michael assumes that	–	michael geht davon aus , dass
michael assumes that he	–	michael geht davon aus , dass er
michael assumes that he will stay in the house	–	michael geht davon aus , dass er im haus bleibt
assumes	–	geht davon aus
assumes	–	geht davon aus ,
assumes that	–	geht davon aus , dass
assumes that he	–	geht davon aus, dass er
assumes that he will stay in the house	–	geht davon aus , dass er im haus bleibt
that	–	dass
that	–	, dass
that he	–	dass er
that he	–	, dass er
that he will stay in the house	–	dass er im haus bleibt
that he will stay in the house	–	, dass er im haus bleibt
he	–	er
he will stay in the house	–	er im haus bleibt
will stay	–	bleibt
will stay in the house	–	im haus bleibt
in the	–	im
in the house	–	im haus
house	–	haus

2.2.2 Estimating Phrase Translation Probabilities

(Maximum Likelihood) Estimation by relative frequency counting:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e}, \bar{f})}{\sum_{f_i} \text{count}(\bar{e}, \bar{f}_i)} \tag{2}$$

2.2.3 Working with huge phrase tables

Phrase tables are much larger than a parallel corpus, even if we set a limit on the phrase length (e.g. max. 7 words). Therefore we need to find other solutions, for instance to read from disk instead of from memory for training. We also have to use smart data structures (e.g. suffix arrays for quick lookup in decoding).