Statistical Machine Translation
Part III – Many-to-Many Alignments

Alexander Fraser

CIS, LMU München

2016.11.08 SMT and NMT

- Last time, we discussed Model 1 and Expectation Maximization
- Today we will discuss getting useful alignments for translation and a translation model

IBM Model 1

- *Generative model*: break up translation process into smaller steps
  - **IBM Model 1** only uses *lexical translation*

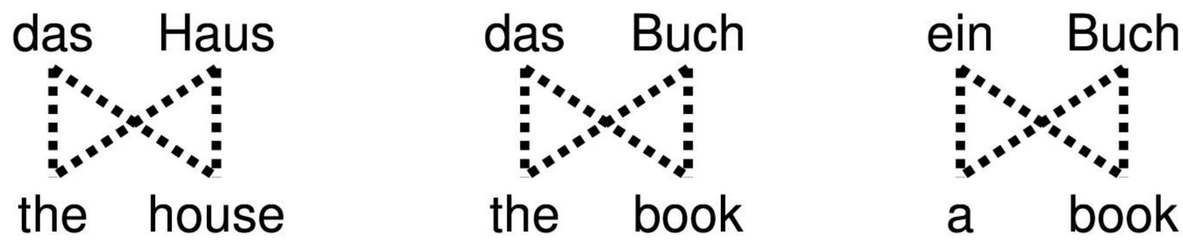
- Translation probability
  - for a foreign sentence $\mathbf{f} = (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence $\mathbf{e} = (e_1, ..., e_{l_e})$ of length l_e
  - with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow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

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

  - parameter ϵ is a *normalization constant*

Slide from Koehn 2008

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

Slide from Koehn 2009

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 on previous model
- Compuationally biggest change in Model 3
  - trick to simplify estimation does not work anymore
  - $\rightarrow$ exhaustive count collection becomes computationally too expensive
  - sampling over high probability alignments is used instead

HMM Model

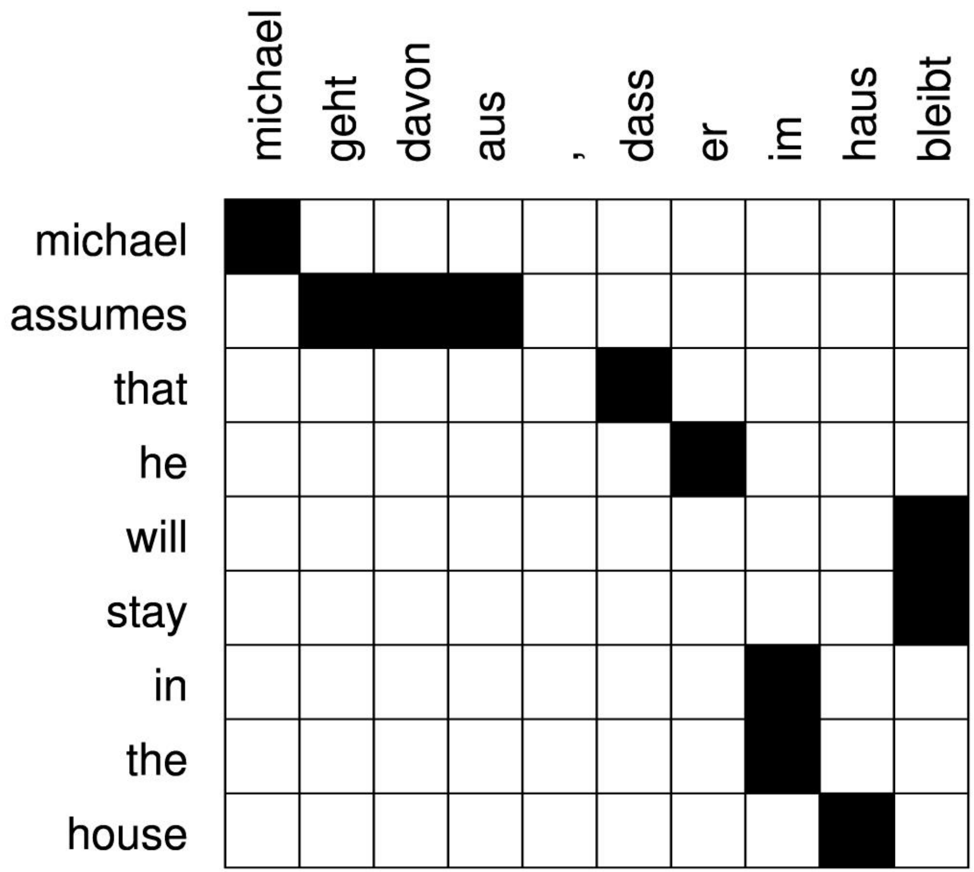
- Model 4 requires local search (making small changes to an initial alignment and rescoring)
- Another popular model is the HMM model, which is similar to Model 2 except that it uses relative alignment positions (like Model 4)
- Popular because it supports inference via the forward-backward algorithm

Overcoming 1-to-N

- We'll now discuss overcoming the poor assumption behind alignment functions

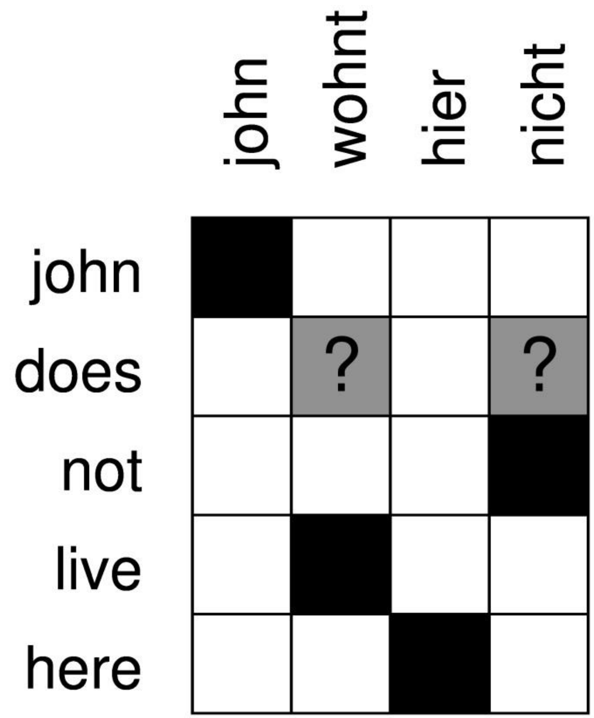
Word Alignment

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



Slide from Koehn 2009

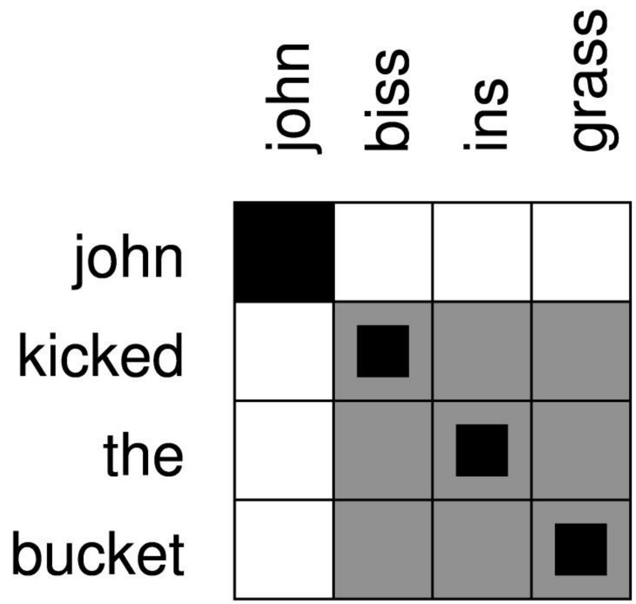
Word Alignment?



Is the English word does aligned to the German wohnt (verb) or nicht (negation) or neither?

Slide from Koehn 2009

Word Alignment?

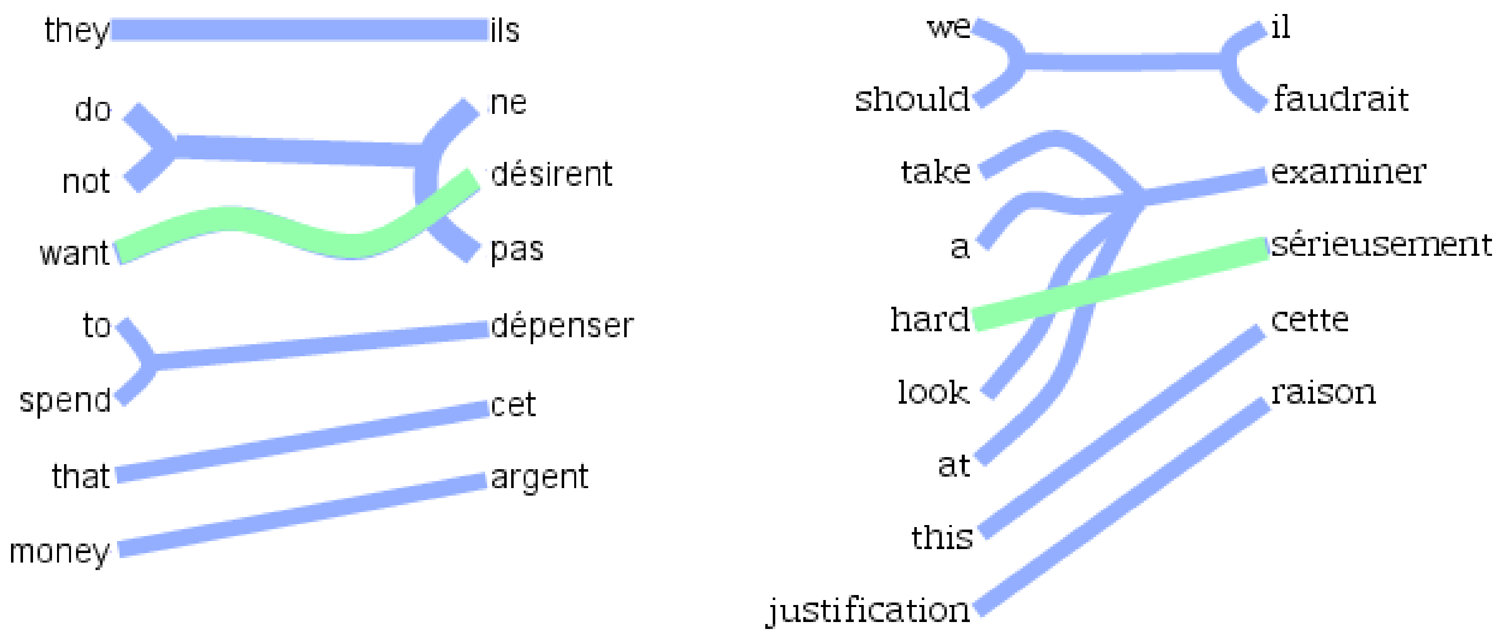


How do the idioms kicked the bucket and biss ins grass match up?
Outside this exceptional context, bucket is never a good translation for grass

Word Alignment with IBM Models

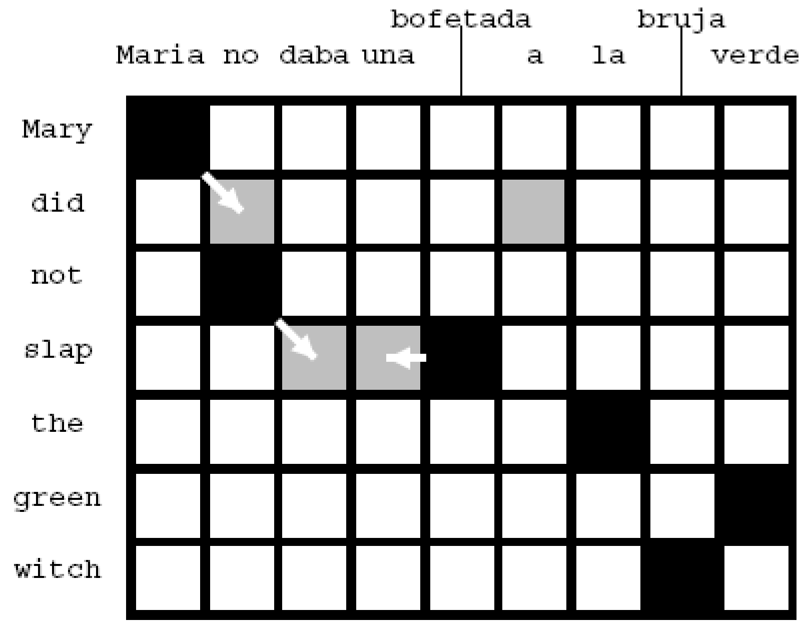
- IBM Models create a **many-to-one** mapping
  - words are aligned using an alignment function
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (no many-to-one mapping)
- Real word alignments have **many-to-many** mappings

IBM Models: 1-to-N Assumption



- 1-to-N assumption
  - Multi-word “cepts” (words in one language translated as a unit) only allowed on target side. Source side limited to single word “cepts”.
  - Forced to create M-to-N alignments using heuristics

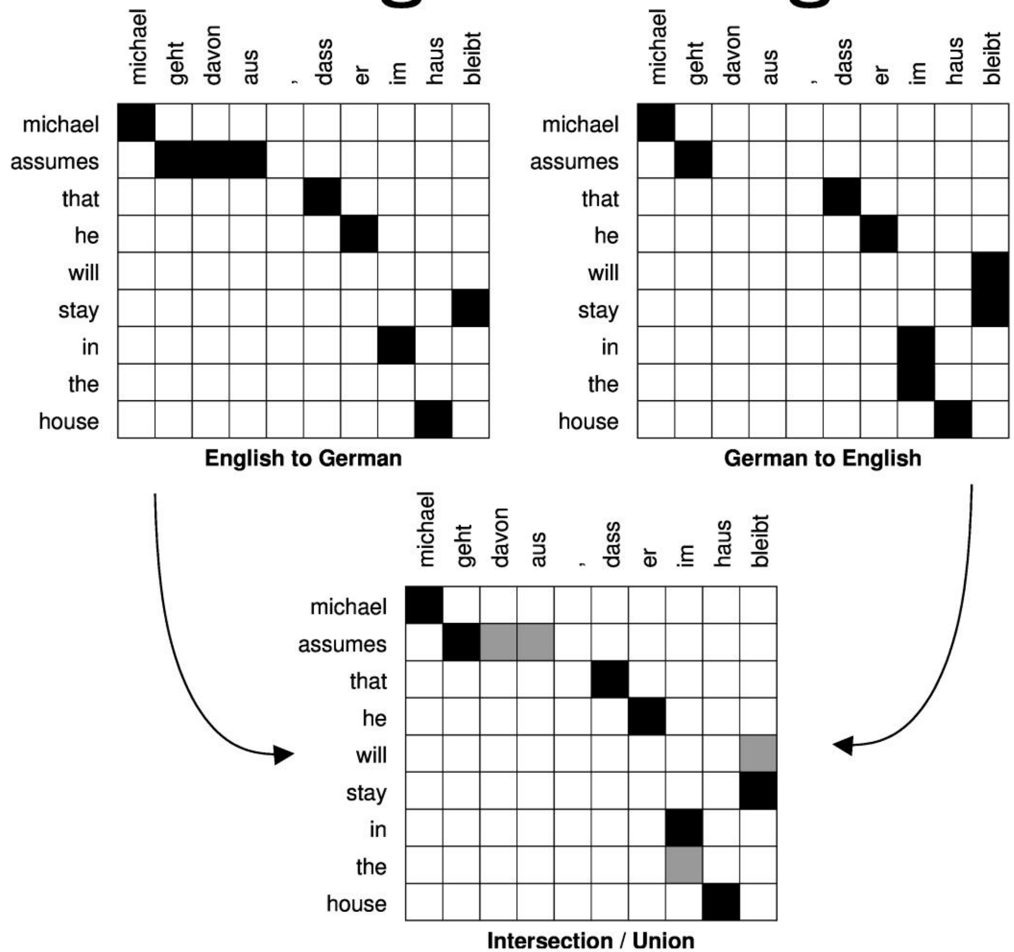
Symmetrizing word alignments



- *Grow* additional alignment points [Och and Ney, CompLing2003]

Slide from Koehn 2008

Symmetrizing Word Alignments



- Intersection of GIZA++ bidirectional alignments
- Grow additional alignment points [Och and Ney, CompLing2003]

Slide from Koehn 2009

Growing heuristic

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(); final(e2f); final(f2e);

grow-diag()

while new points added **do**
  for all English word $e \in [1...e_n]$, foreign word $f \in [1...f_n]$, $(e, f) \in A$ **do**
    for all neighboring alignment points $(e_{\text{new}}, f_{\text{new}})$ **do**
      if (e_{new} unaligned OR f_{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_{new} unaligned OR f_{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

Discussion

- Most state of the art SMT systems are built as I presented
- Use IBM Models to generate both:
  - one-to-many alignment
  - many-to-one alignment
- Combine these two alignments using symmetrization heuristic
  - output is a many-to-many alignment
  - used for building decoder
- Moses toolkit for implementation: www.statmt.org
  - Uses Och and Ney GIZA++ tool for Model 1, HMM, Model 4

- However, there is newer work on alignment that is interesting!

Where we have been

- We defined the overall problem and talked about evaluation
- We have now covered word alignment
  - IBM Model 1, true Expectation Maximization
  - Briefly mentioned: IBM Model 4, approximate Expectation Maximization
  - Symmetrization Heuristics (such as Grow)
    - Applied to two Viterbi alignments (typically from Model 4)
    - Results in final word alignment

Where we are going

- We will discuss the "traditional" phrase-based model (which noone actually uses, but gives a good intuition)
- Then we will define a high performance translation model (next slide set)
- Finally, we will show how to solve the search problem for this model (= decoding)

Outline

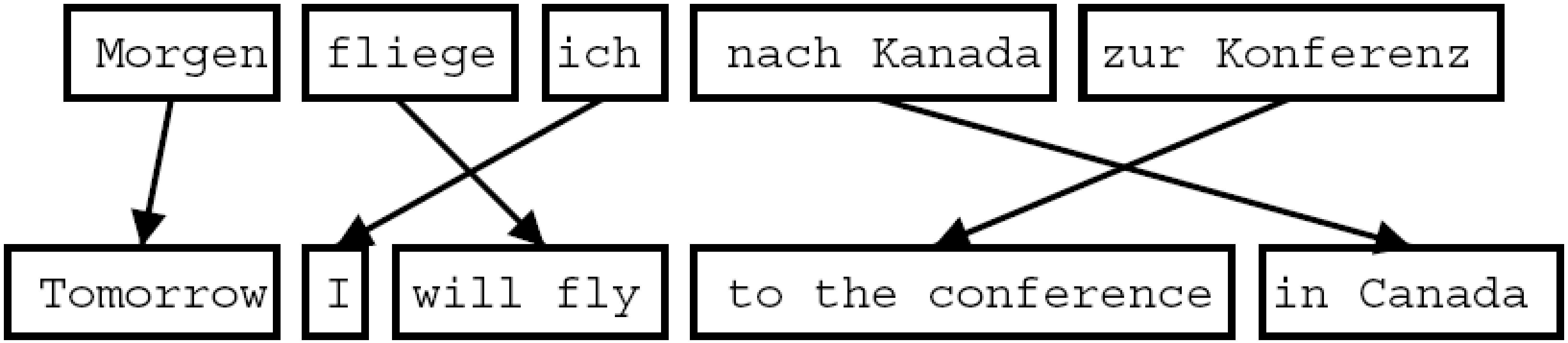
- Phrase-based translation
  - Model
  - Estimating parameters
- Decoding

- We could use IBM Model 4 in the direction p(f|e), together with a language model, p(e), to translate

$$\underset{e}{\operatorname{argmax}}\ P(e \mid f) = \underset{e}{\operatorname{argmax}}\ P(f \mid e)\ P(e)$$

- However, decoding using Model 4 doesn't work well in practice
  - One strong reason is the bad 1-to-N assumption
  - Another problem would be defining the search algorithm
    - If we add additional operations to allow the English words to vary, this will be very expensive
  - Despite these problems, Model 4 decoding was briefly state of the art
- We will now define a better model…

Phrase-based translation

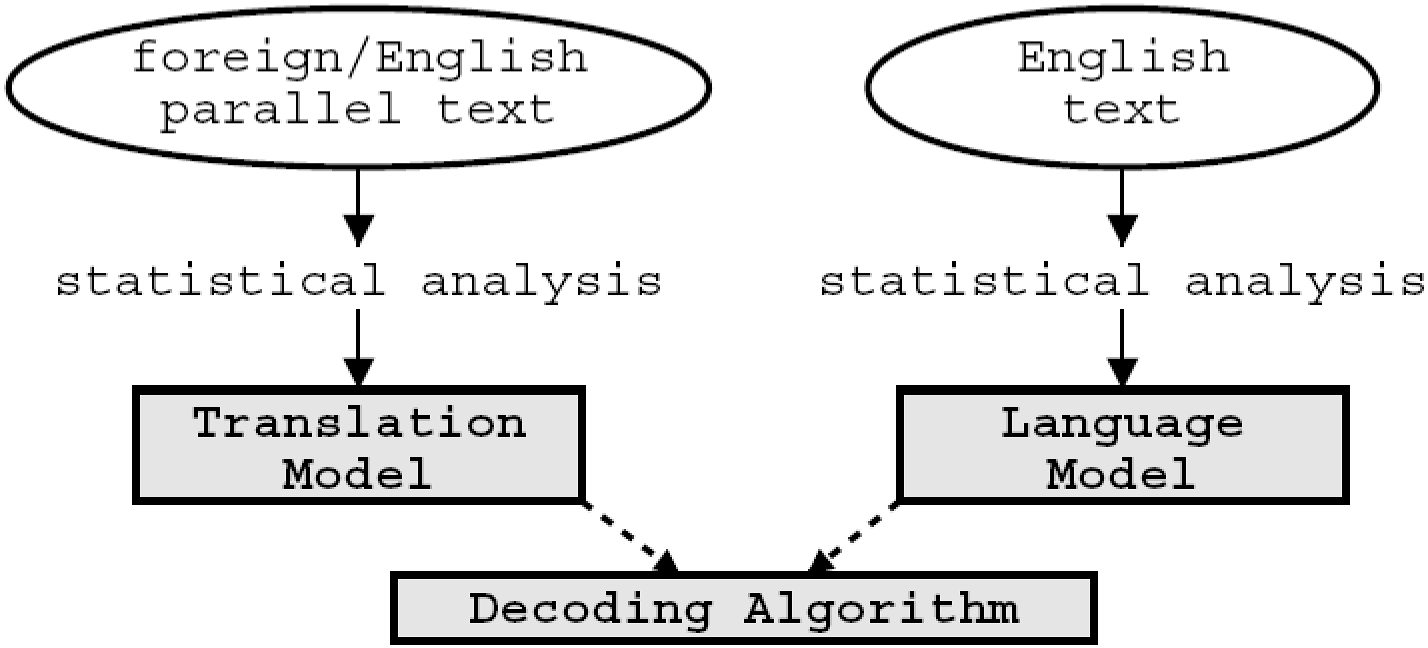


- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

Slide from Koehn 2008

Statistical Machine Translation

- Components: Translation model, language model, decoder



Slide from Koehn 2008

Language Model

- Often a trigram language model is used for p(e)
  - P(the man went home) = p(the | START) p(man | START the) p(went | the man) p(home | man went)
- Language models work well for comparing the grammaticality of strings of the same length
  - However, when comparing short strings with long strings they favor short strings
  - For this reason, an important component of the language model is the length bonus
    - This is a constant > 1 multiplied for each English word in the hypothesis
    - It makes longer strings competitive with shorter strings

Phrase-based translation model

- Major components of phrase-based model
  - **phrase translation model** $\phi(\mathbf{f}|\mathbf{e})$
  - **reordering model** d
  - **language model** $p_{\text{LM}}(\mathbf{e})$
- Bayes rule

$$\begin{aligned} \text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) &= \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) p(\mathbf{e}) \\ &= \text{argmax}_{\mathbf{e}} \phi(\mathbf{f}|\mathbf{e}) p_{\text{LM}}(\mathbf{e}) \omega^{\text{length}(\mathbf{e})} \end{aligned}$$

- Sentence $\mathbf{f}$ is decomposed into I phrases $\bar{f}_1^I = \bar{f}_1, ..., \bar{f}_I$
- Decomposition of $\phi(\mathbf{f}|\mathbf{e})$

$$\phi(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

Modified from Koehn 2008

Advantages of phrase-based translation

- *Many-to-many* translation can handle non-compositional phrases
- Use of *local context* in translation
- The more data, the *longer phrases* can be learned

Slide from Koehn 2008

Phrase translation table

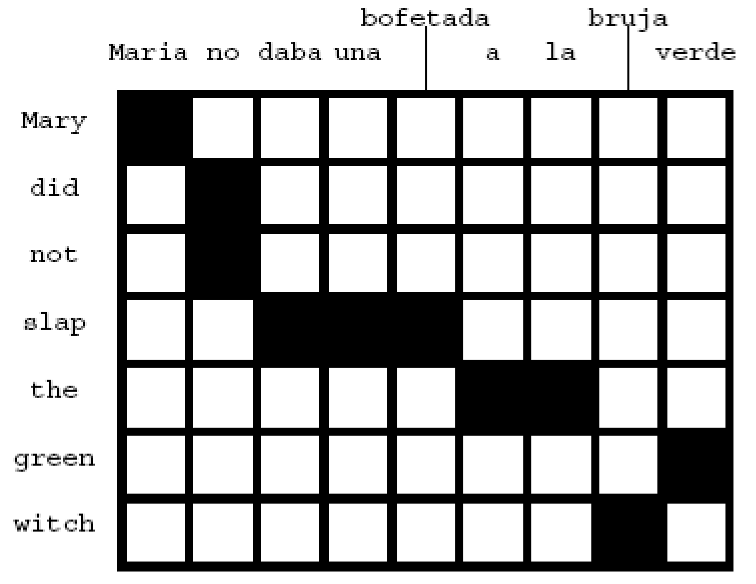
- Phrase translations for *den Vorschlag*

English	$\phi(e\|f)$	English	$\phi(e\|f)$
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	...	...

Slide from Koehn 2008

How to learn the phrase translation table?

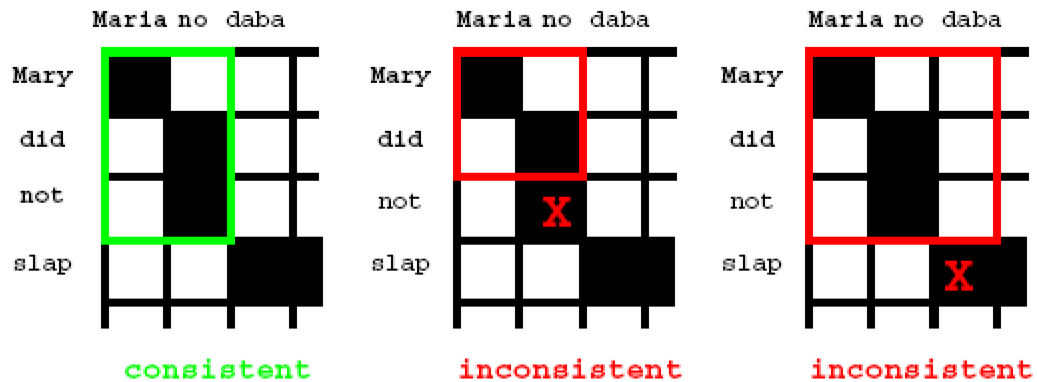
- Start with the *word alignment*:



- Collect all phrase pairs that are **consistent** with the word alignment

Slide from Koehn 2008

Consistent with word alignment



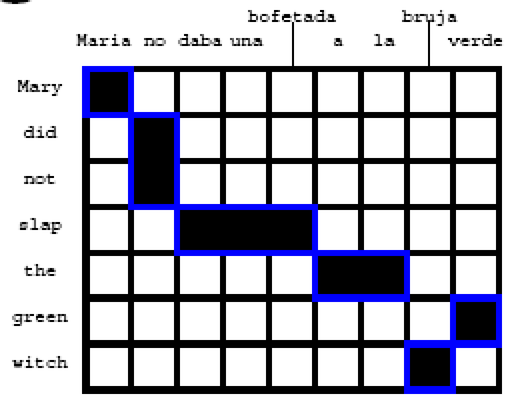
- **Consistent with the word alignment** :=

  phrase alignment has to *contain all alignment points* for all covered words

$$(\overline{e}, \overline{f}) \in BP \Leftrightarrow \quad \forall e_i \in \overline{e} : (e_i, f_j) \in A \rightarrow f_j \in \overline{f}$$
$$\text{AND} \quad \forall f_j \in \overline{f} : (e_i, f_j) \in A \rightarrow e_i \in \overline{e}$$

Slide from Koehn 2008

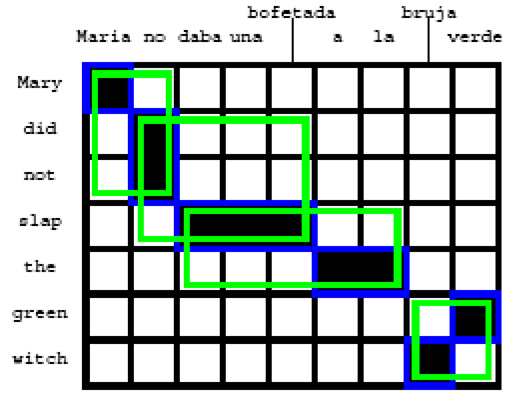
Word alignment induced phrases



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

Slide from Koehn 2008

Word alignment induced phrases

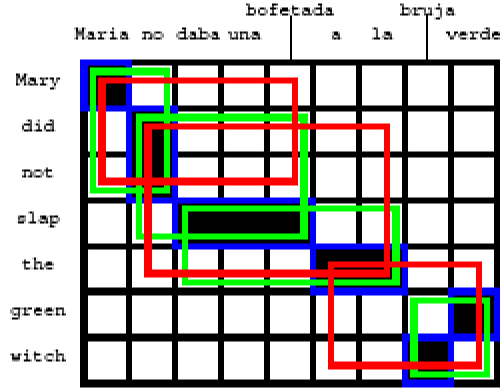


(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

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

Slide from Koehn 2008

Word alignment induced phrases



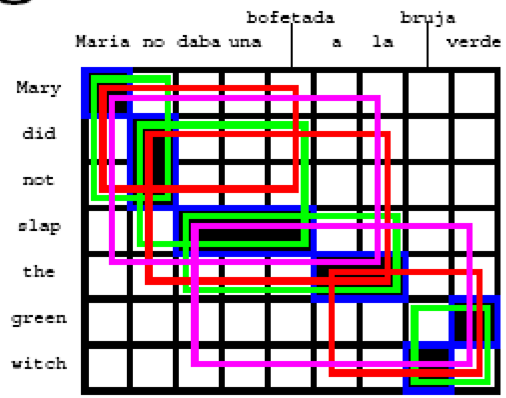
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),

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

(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

Slide from Koehn 2008

Word alignment induced phrases



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

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),

(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),

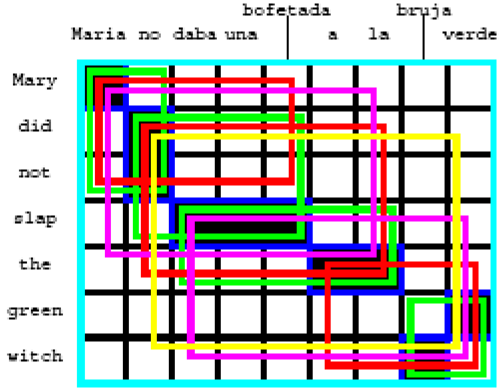
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),

(Maria no daba una bofetada a la, Mary did not slap the),

(daba una bofetada a la bruja verde, slap the green witch)

Slide from Koehn 2008

Word alignment induced phrases (5)



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

Slide from Koehn 2008

Probability distribution of phrase pairs

- We need a **probability distribution** $\phi(\bar{f}|\bar{e})$ over the collected phrase pairs

$\Rightarrow$ Possible *choices*

- *relative frequency* of collected phrases: $\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{f},\bar{e})}{\sum_{\bar{f}} \text{count}(\bar{f},\bar{e})}$
- or, conversely $\phi(\bar{e}|\bar{f})$
- use *lexical translation probabilities*

Slide from Koehn 2008

Reordering

- *Monotone* translation
  - do not allow any reordering
  - $\rightarrow$ worse translations
- *Limiting* reordering (to movement over max. number of words) helps
- *Distance-based* reordering cost
  - moving a foreign phrase over n words: cost z^n
- *Lexicalized* reordering model

Slide from Koehn 2008