Machine Translation:
Overview & Word Alignment

Introduction to Natural Language Processing
Computer Science 585—Fall 2009
University of Massachusetts Amherst

David Smith
With slides from Charles Schafer and Philipp Koehn

Translation and NLP

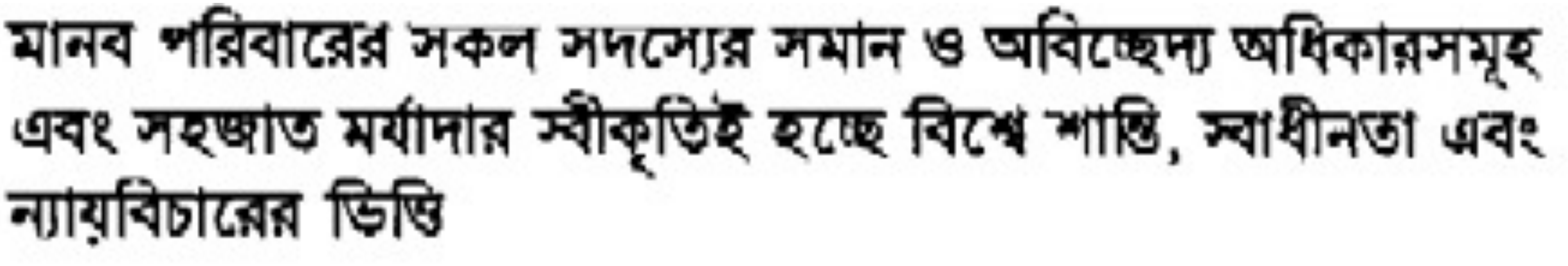
- Translation is one of the oldest language tasks tried on a computer
  - Just look at that archaic name: "Machine Translation"!
- Translation involves many linguistic systems
- "Apollo program" dual-use argument:
  - Translation models of alignment and transfer are useful in question answering, paraphrase, information retrieval, etc.

Overview

- What problems does MT address? What does it (currently) not address?
- Models: What makes a good translation?
- Alignment: Learning dictionaries from parallel text
- Next: non-parallel text, translation decoding and training

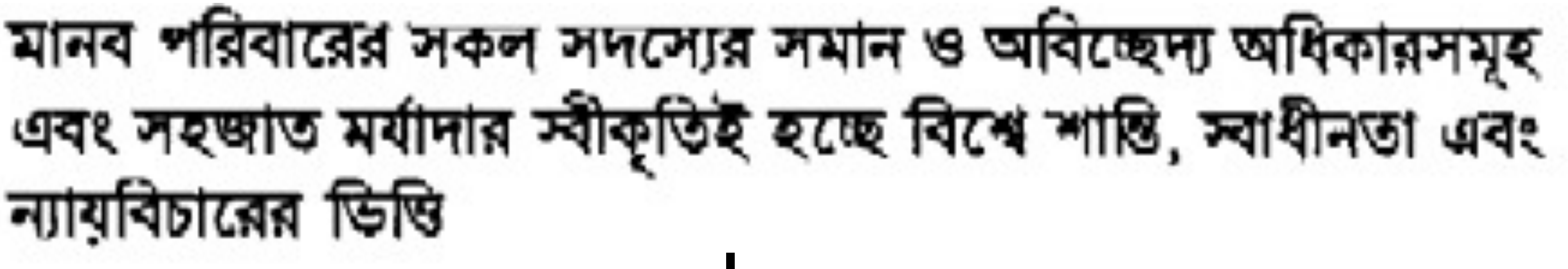
The Translation Problem
and
Translation Data

The Translation Problem



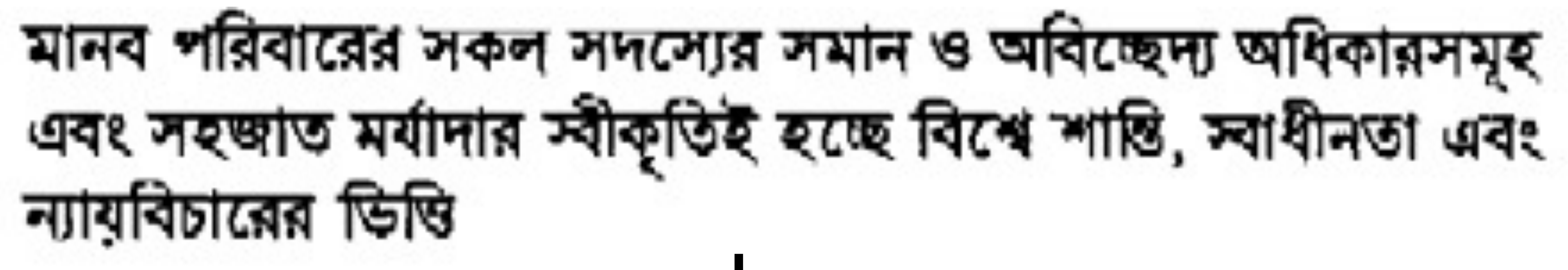
মানব পরিবারের সকল সদস্যের সমান ও অবিচ্ছেদ্য অধিকারসমূহ এবং সহজাত মর্যাদার স্বীকৃতিই হচ্ছে বিশ্বে শান্তি, স্বাধীনতা এবং ন্যায়বিচারের ভিত্তি

The Translation Problem



মানব পরিবারের সকল সদস্যের সমান ও অবিচ্ছেদ্য অধিকারসমূহ এবং সহজাত মর্যাদার স্বীকৃতিই হচ্ছে বিশ্বে শান্তি, স্বাধীনতা এবং ন্যায়বিচারের ভিত্তি

The Translation Problem



মানব পরিবারের সকল সদস্যের সমান ও অবিচ্ছেদ্য অধিকারসমূহ
এবং সহজাত মর্যাদার স্বীকৃতিই হচ্ছে বিশ্বে শান্তি, স্বাধীনতা এবং
ন্যায়বিচারের ভিত্তি

Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world

Why Machine Translation?

* Cheap, universal access to world's online information regardless of original language. (That's the goal)

Why Statistical (or at least Empirical) Machine Translation?

* We want to translate real-world documents. Thus, we should model real-world documents.

* A nice property: design the system once, and extend to new languages automatically by training on existing data.

F(training data, model) -> parameterized MT system

Ideas that cut across empirical language processing problems and methods

Real-world: don't be (too) prescriptive. Be able to process (translate/summarize/identify/paraphrase) relevant bits of human language as they are, not as they "should be". For instance, genre is important: translating French blogs into English is different from translating French novels into English.

Model: a fully described procedure, generally having variable parameters, that performs some interesting task (for example, translation).

Training data: a set of observed data instances which can be used to find good parameters for a model via a training procedure.

Training procedure: a method that takes observed data and refines the parameters of a model, such that the model is improved according to some objective function.

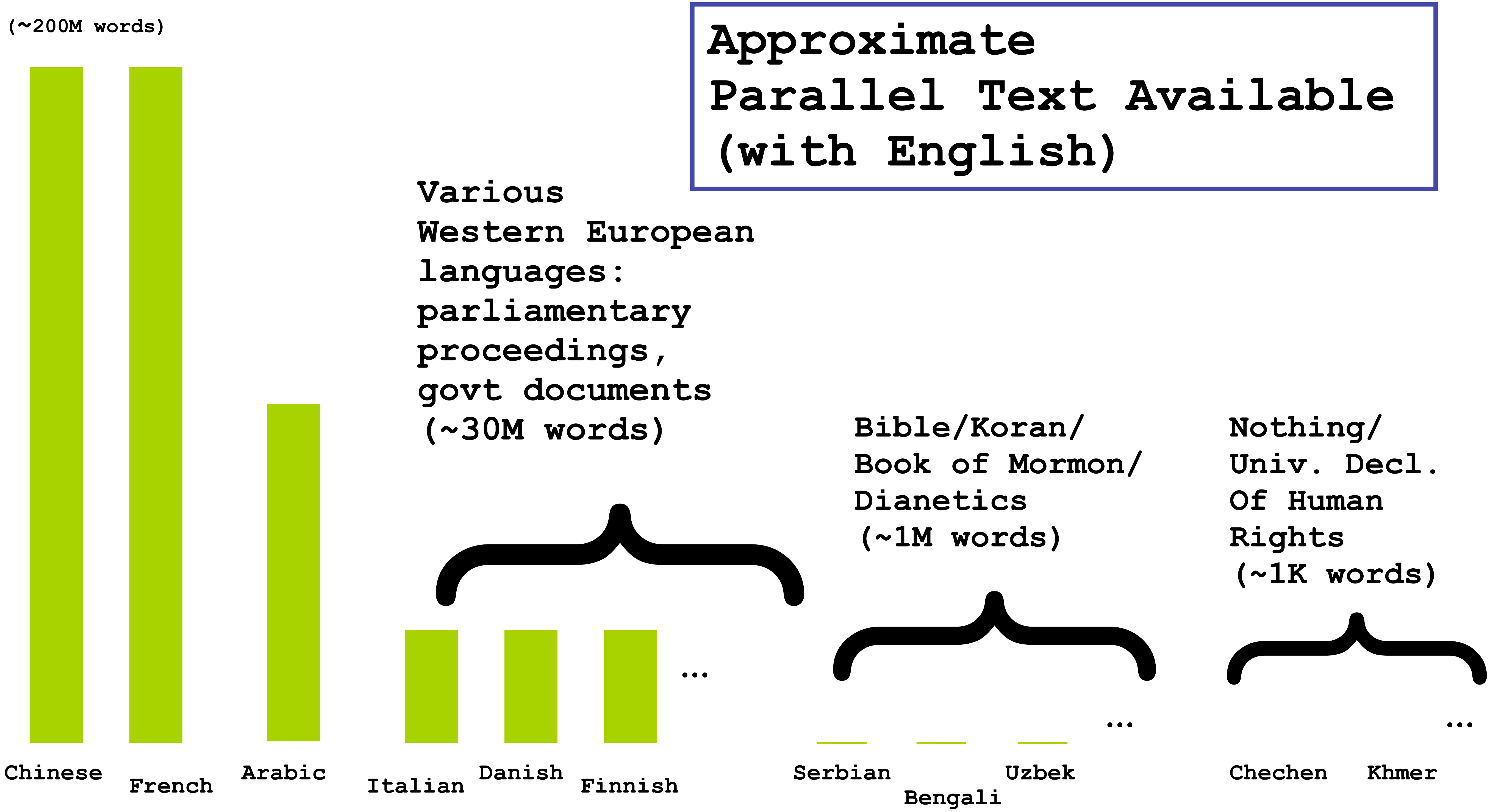
Resource Availability

Most of this lecture

Most statistical machine translation (SMT) research has focused on a few “high-resource” languages(European, Chinese, Japanese, Arabic).

Some other work: translation for the rest of the world’s languages found on the web.

Most statistical machine translation research has focused on a few high-resource languages (European, Chinese, Japanese, Arabic).



Approximate Parallel Text Available (with English)

Various Western European languages: parliamentary proceedings, govt documents (~30M words)

Bible/Koran/ Book of Mormon/ Dianetics (~1M words)

Nothing/ Univ. Decl. Of Human Rights (~1K words)

Resource Availability

Most statistical machine translation (SMT) research has focused on a few “high-resource” languages(European, Chinese, Japanese, Arabic).

Some other work: translation for the rest of the world’s languages found on the web.

Romanian Catalan Serbian Slovenian Macedonian Uzbek Turkmen Kyrgyz Uighur Pashto Tajikh Dari Kurdish Azeri Bengali Punjabi Gujarati Nepali Urdu Marathi Konkani Oriya Telugu Malayalam Kannada Cebuano

We’ll discuss this briefly

The Translation Problem

Document translation? Sentence translation? Word translation?

What to translate? The most common use case is probably document translation.

Most MT work focuses on sentence translation.

What does sentence translation ignore?

- Discourse properties/structure.
- Inter-sentence coreference.

Sentence Translation

- SMT has generally ignored extra-sentence structure (good future work direction for the community).

- Instead, we've concentrated on translating individual sentences as well as possible. This is a very hard problem in itself.

- Word translation (knowing the possible English translations of a French word) is not, by itself, sufficient for building readable/useful automatic document translations – though it is an important component in end-to-end SMT systems.

Sentence translation using only a word translation dictionary is called "glossing" or "gisting".

Word Translation (learning from minimal resources)

We'll come back to this later…

and address learning the word translation component (dictionary) of MT systems without using parallel text.

(For languages having little parallel text, this is the best we can do right now)

Sentence Translation

- Training resource: parallel text (bitext).

- Parallel text (with English) on the order of 20M-200M words (roughly, 1M-10M sentences) is available for a number of languages.

- Parallel text is expensive to generate: human translators are expensive ($0.05-$0.25 per word). Millions of words training data needed for high quality SMT results. So we take what is available. This is often of less than optimal genre (laws, parliamentary proceedings, religious texts).

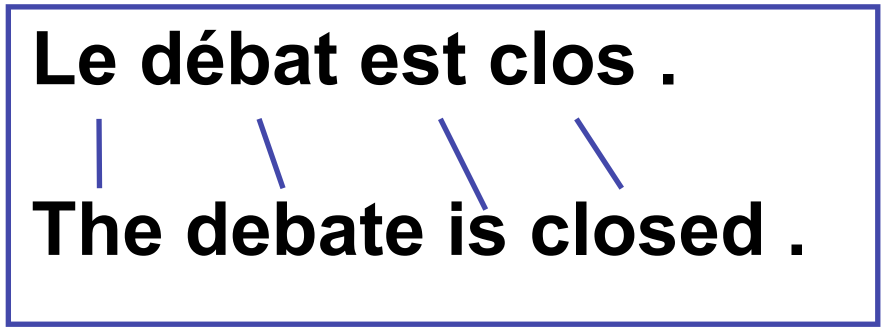
Sentence Translation: examples of more and less literal translations in bitext

French, English from Bitext	Closely Literal English Translation
Le débat est clos . **The debate is closed .**	*The debate is closed.*
Accepteriez - vous ce principe ? **Would you accept that principle ?**	*Accept-you that principle?*
Merci , chère collègue . **Thank you , Mrs Marinucci .**	*Thank you, dear colleague.*
Avez - vous donc une autre proposition ? **Can you explain ?**	*Have you therefore another proposal?*

(from French-English European Parliament proceedings)

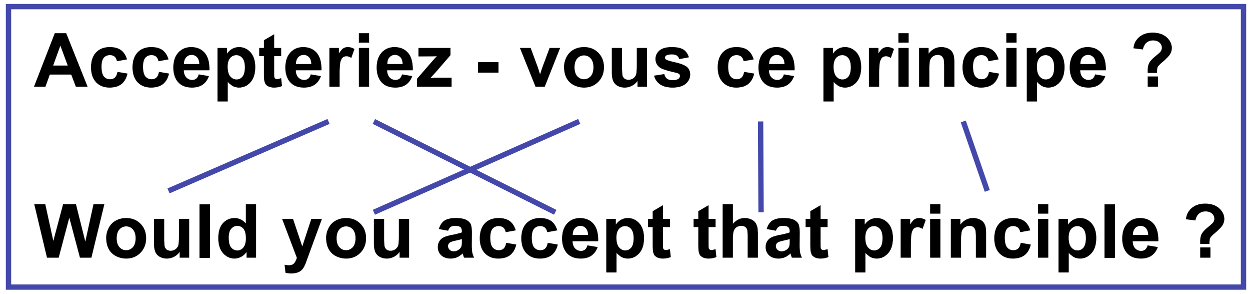
Sentence Translation: examples of more and less literal translations in bitext

Word alignments illustrated.
Well-defined for more literal translations.



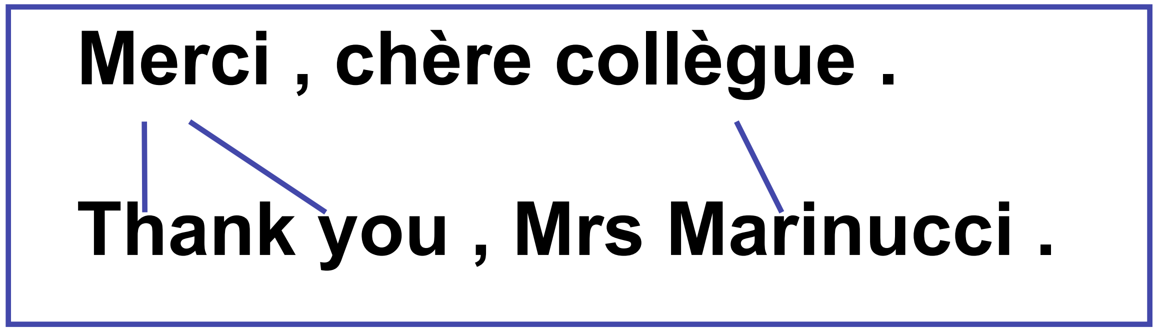
Le débat est clos .

The debate is closed .



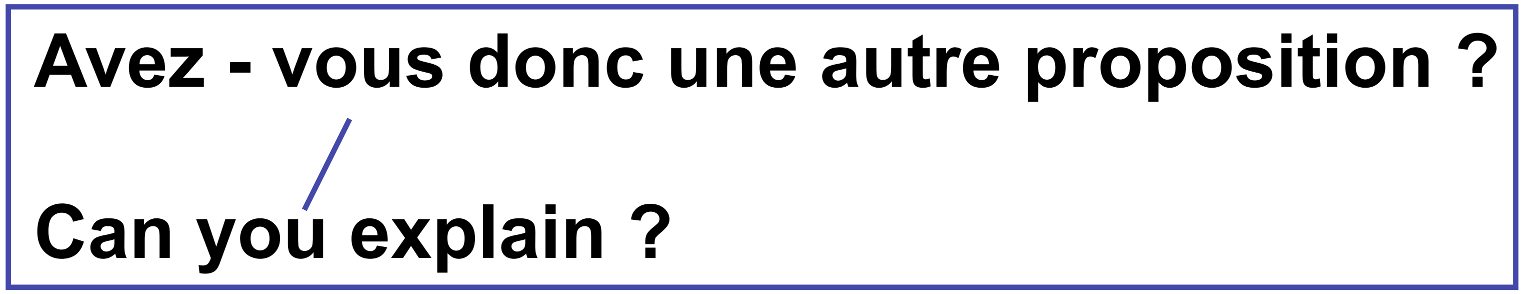
Accepteriez - vous ce principe ?

Would you accept that principle ?



Merci , chère collègue .

Thank you , Mrs Marinucci .



Avez - vous donc une autre proposition ?

Can you explain ?

Translation and Alignment

- As mentioned, translations are expensive to commission and generally SMT research relies on already existing translations

- These typically come in the form of aligned documents.

- A sentence alignment, using pre-existing document boundaries, is performed automatically. Low-scoring or non-one-to-one sentence alignments are discarded. The resulting aligned sentences constitute the training bitext.

- For many modern SMT systems, induction of word alignments between aligned sentences, using algorithms based on the IBM word-based translation models, is one of the first stages of processing. Such induced word alignments are generally treated as part of the observed data and are used to extract aligned phrases or subtrees.

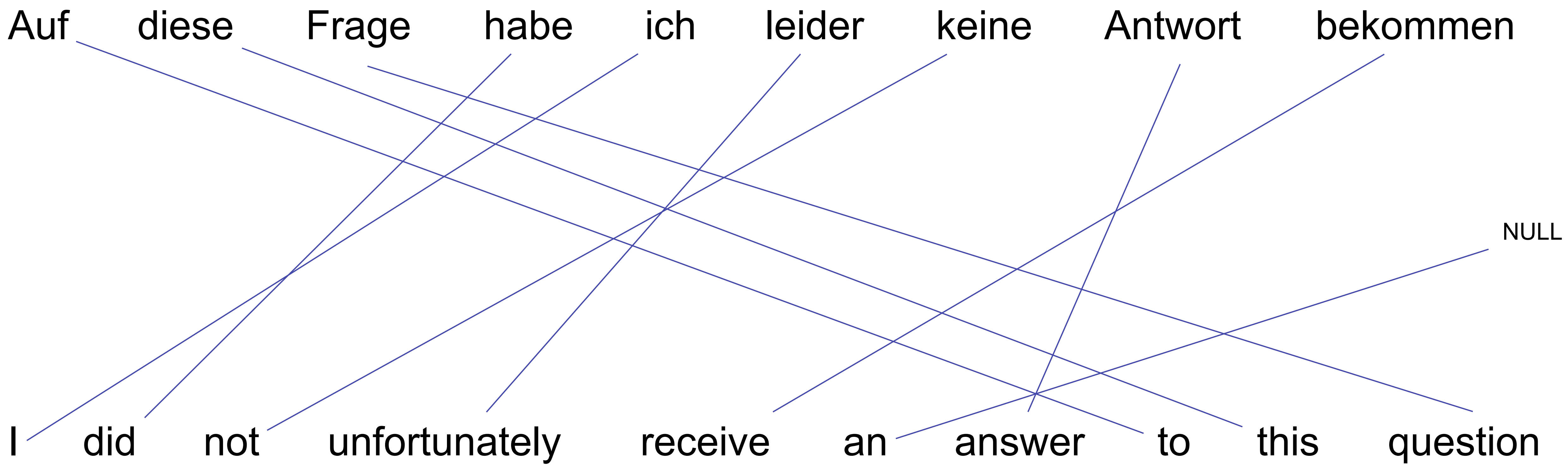
Modeling

What Makes a Good Translation?

Modeling

- Translation models
  - "Adequacy"
  - Assign better scores to accurate (and complete) translations
- Language models
  - "Fluency"
  - Assign better scores to natural target language text

Word Translation Models

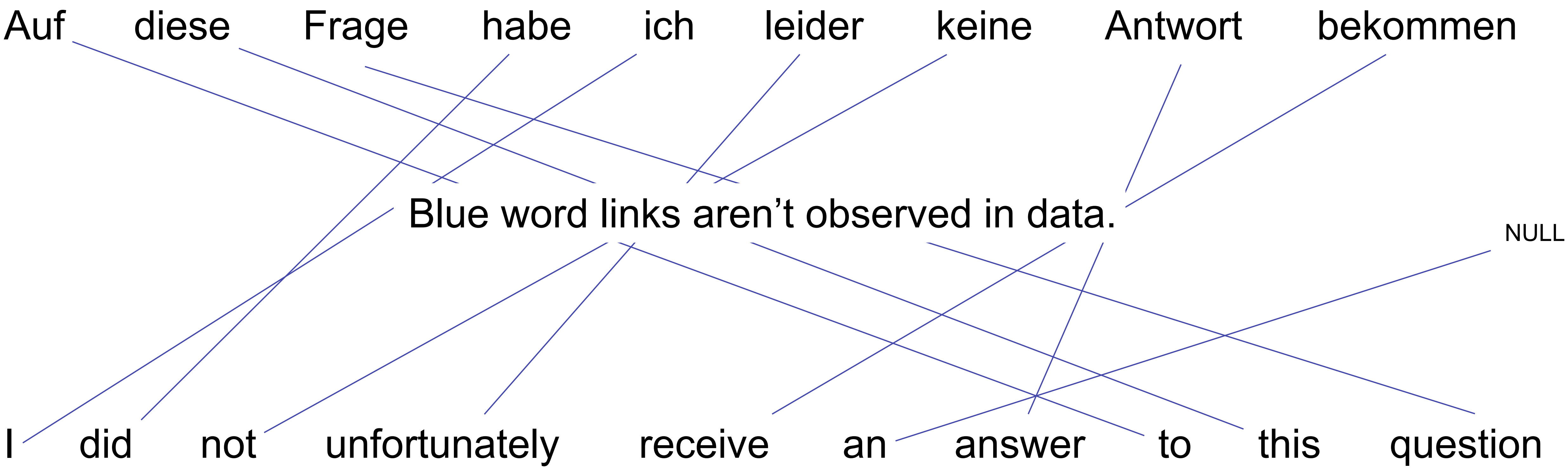


Auf diese Frage habe ich leider keine Antwort bekommen

NULL

I did not unfortunately receive an answer to this question

Word Translation Models



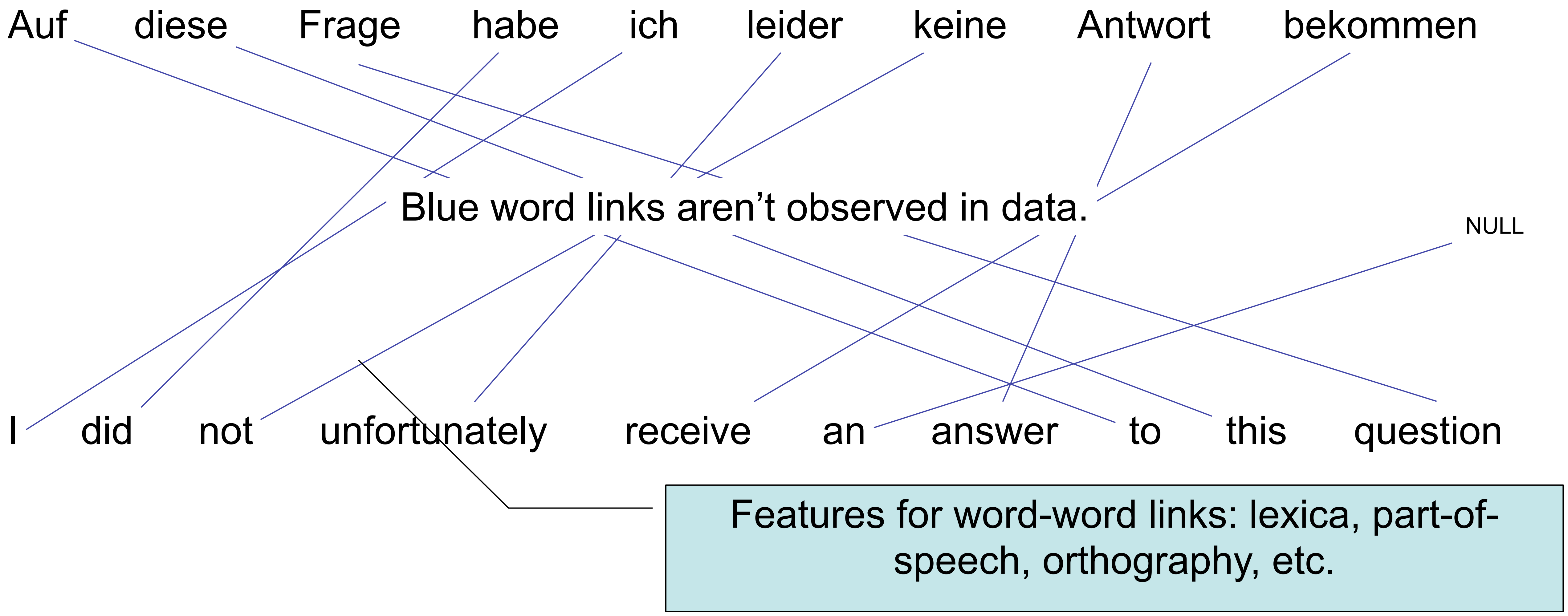
Auf diese Frage habe ich leider keine Antwort bekommen

Blue word links aren’t observed in data.

NULL

I did not unfortunately receive an answer to this question

Word Translation Models



Auf diese Frage habe ich leider keine Antwort bekommen

Blue word links aren’t observed in data.

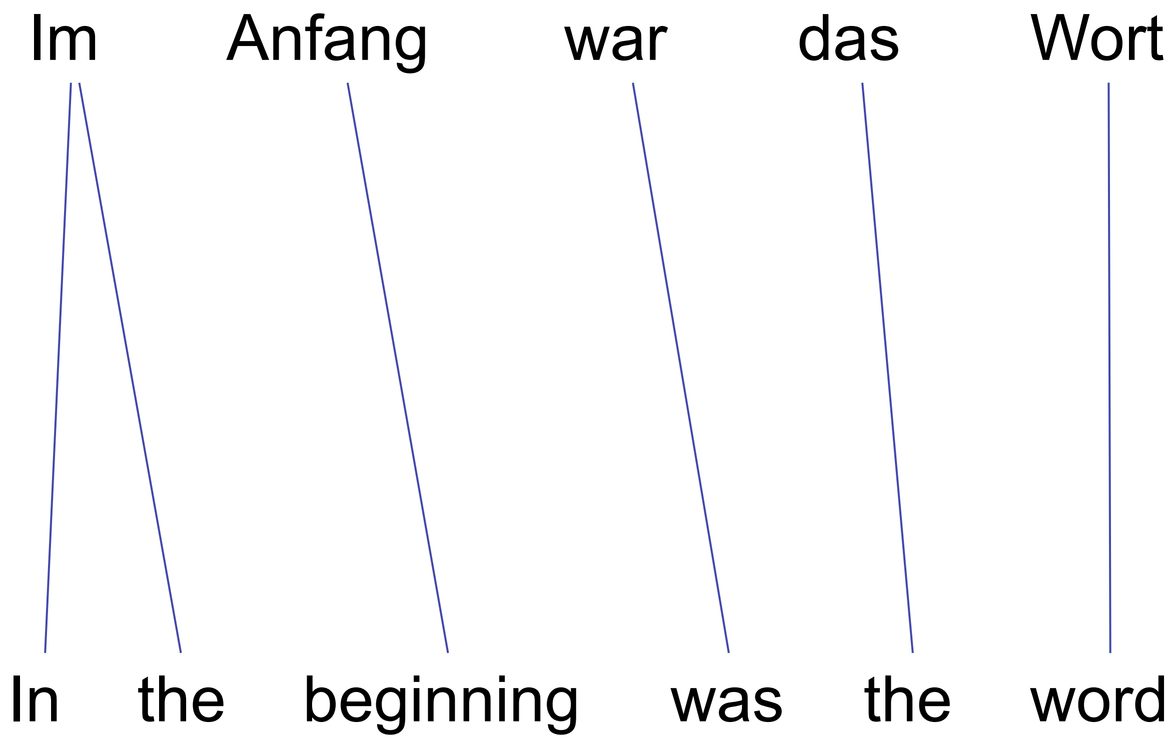
NULL

I did not unfortunately receive an answer to this question

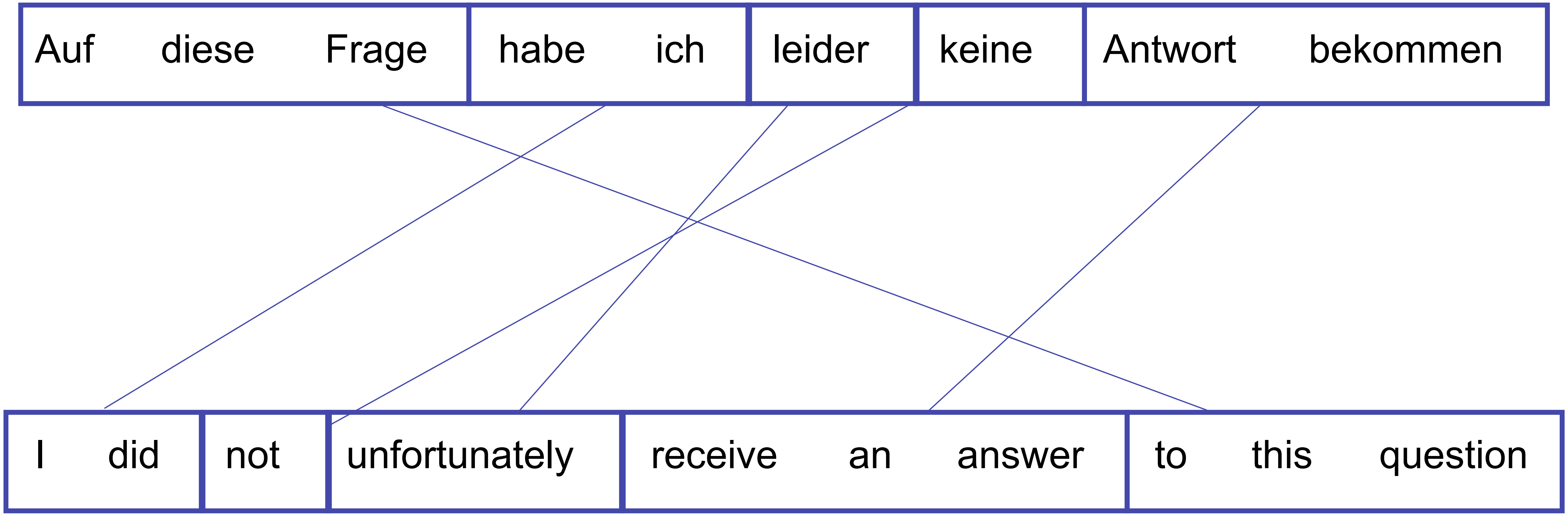
Features for word-word links: lexica, part-of-speech, orthography, etc.

Word Translation Models

- Usually directed: each word in the target generated by one word in the source
- Many-many and null-many links allowed
- Classic IBM models of Brown et al.
- Used now mostly for word alignment, not translation



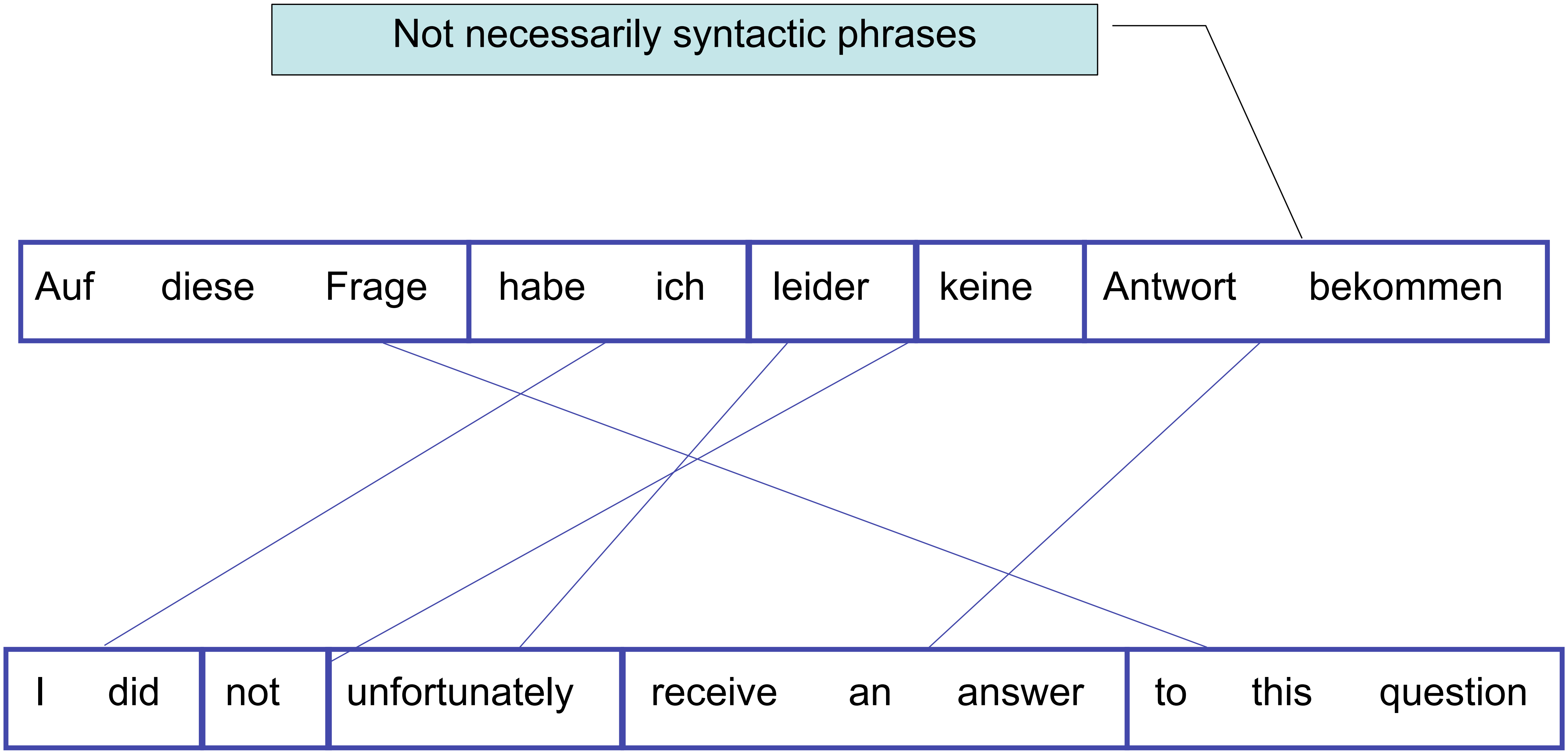
Phrase Translation Models



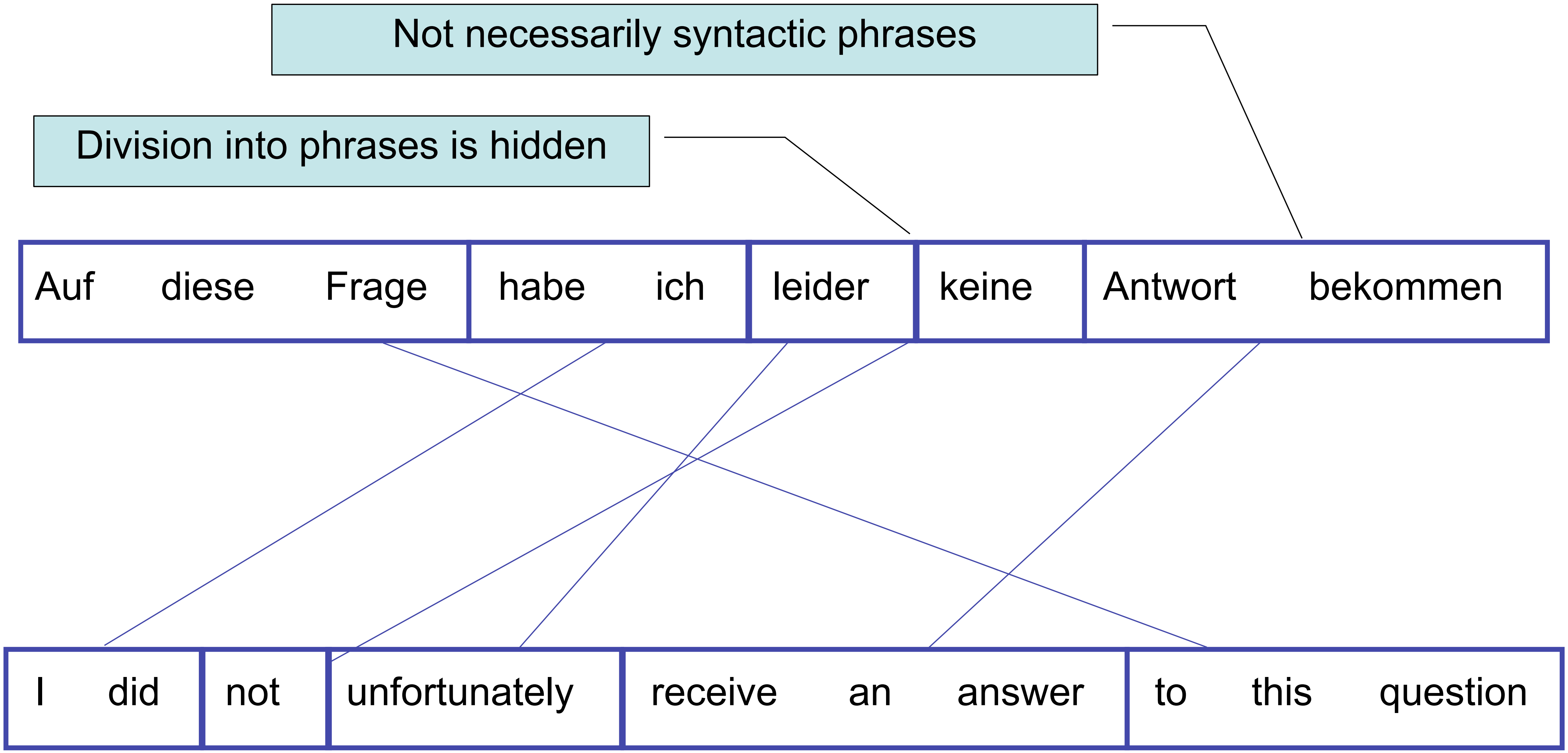
Auf diese Frage habe ich leider keine Antwort bekommen

I did not unfortunately receive an answer to this question

Phrase Translation Models



Phrase Translation Models



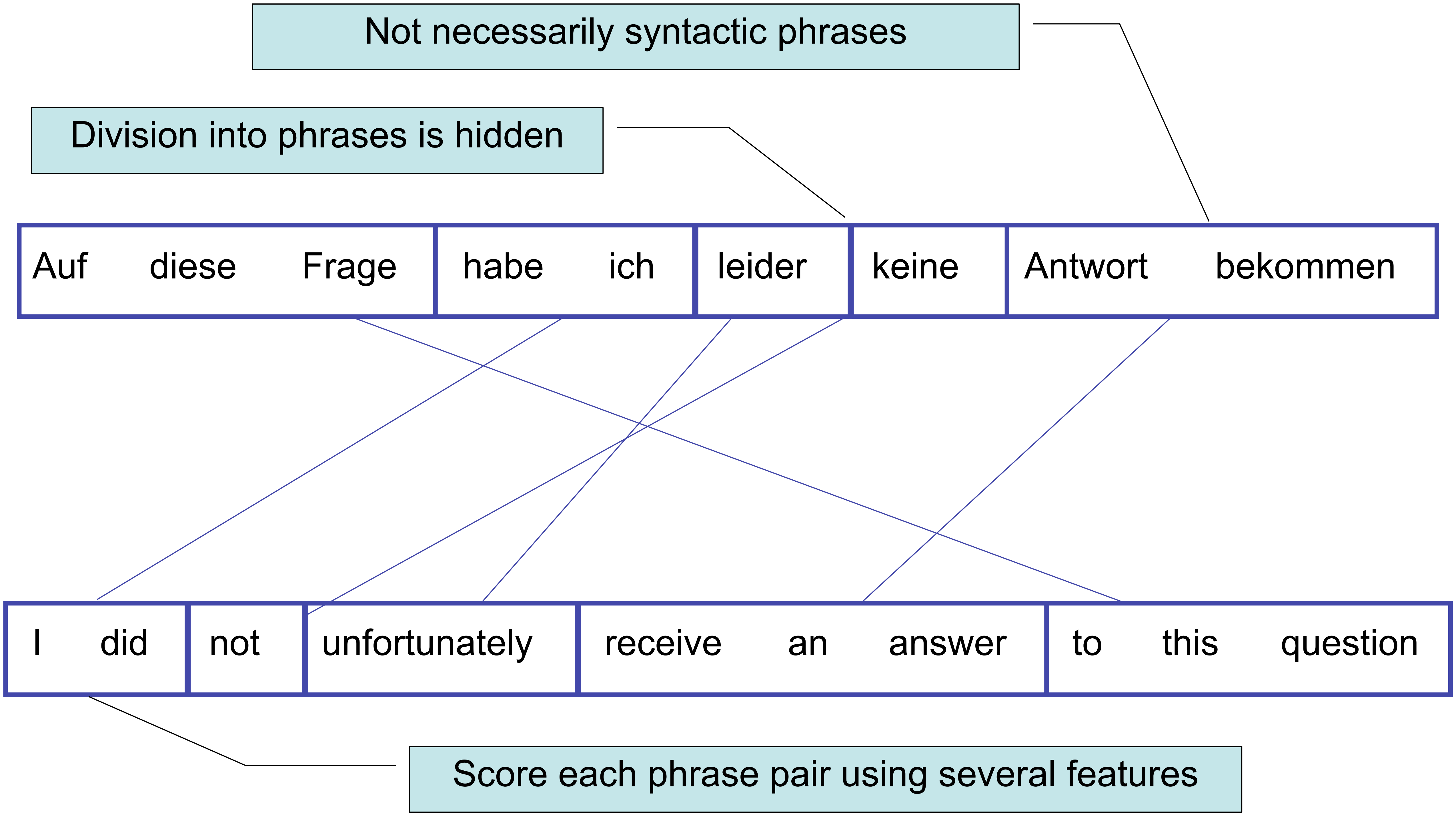
Not necessarily syntactic phrases

Division into phrases is hidden

Auf diese Frage | habe ich | leider | keine | Antwort bekommen

I did | not | unfortunately | receive an answer | to this question

Phrase Translation Models



Not necessarily syntactic phrases

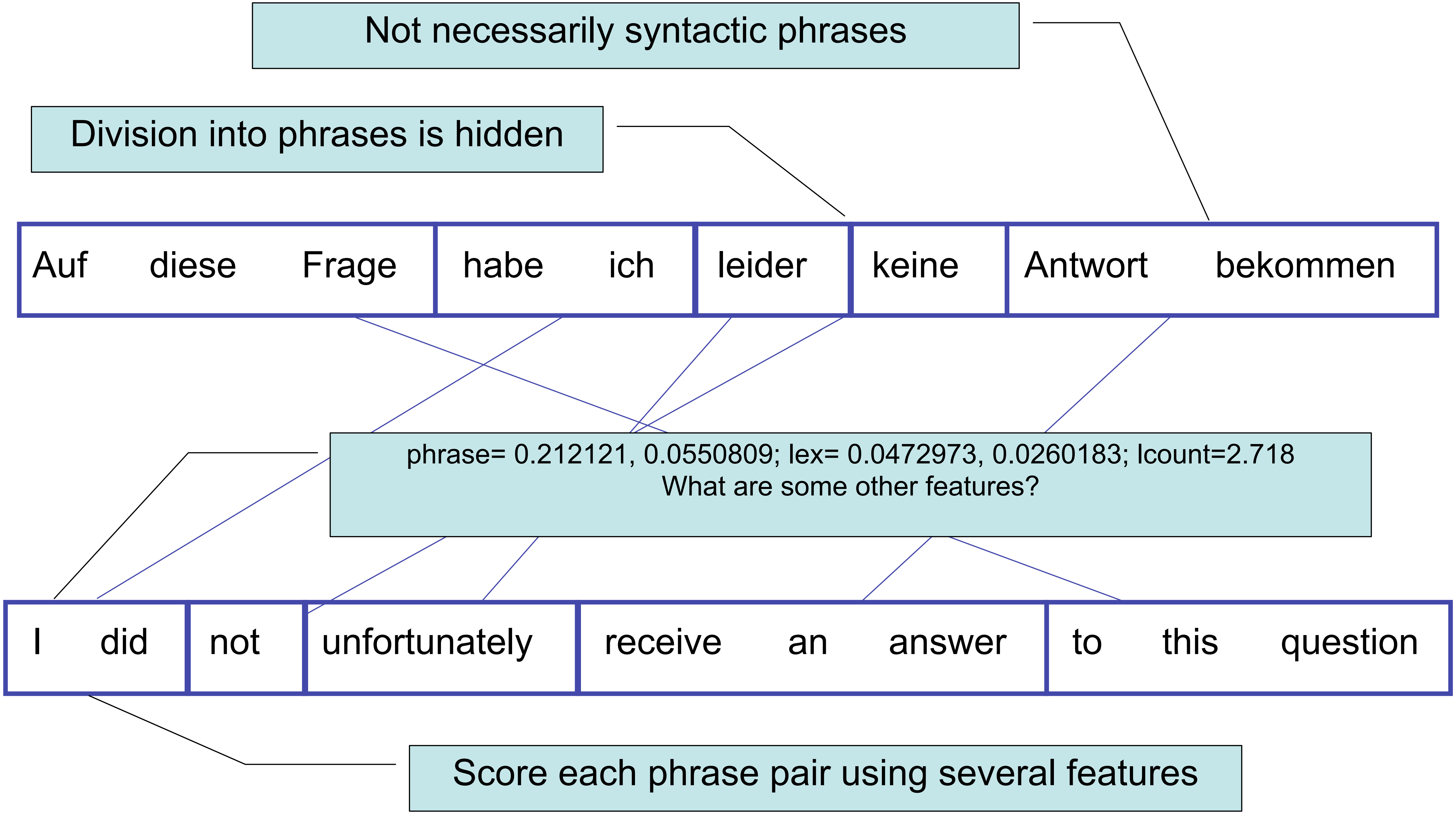
Division into phrases is hidden

Auf diese Frage | habe ich | leider | keine | Antwort bekommen

I did | not | unfortunately | receive an answer | to this question

Score each phrase pair using several features

Phrase Translation Models



Not necessarily syntactic phrases

Division into phrases is hidden

Auf diese Frage | habe ich | leider | keine | Antwort bekommen

phrase= 0.212121, 0.0550809; lex= 0.0472973, 0.0260183; lcount=2.718
What are some other features?

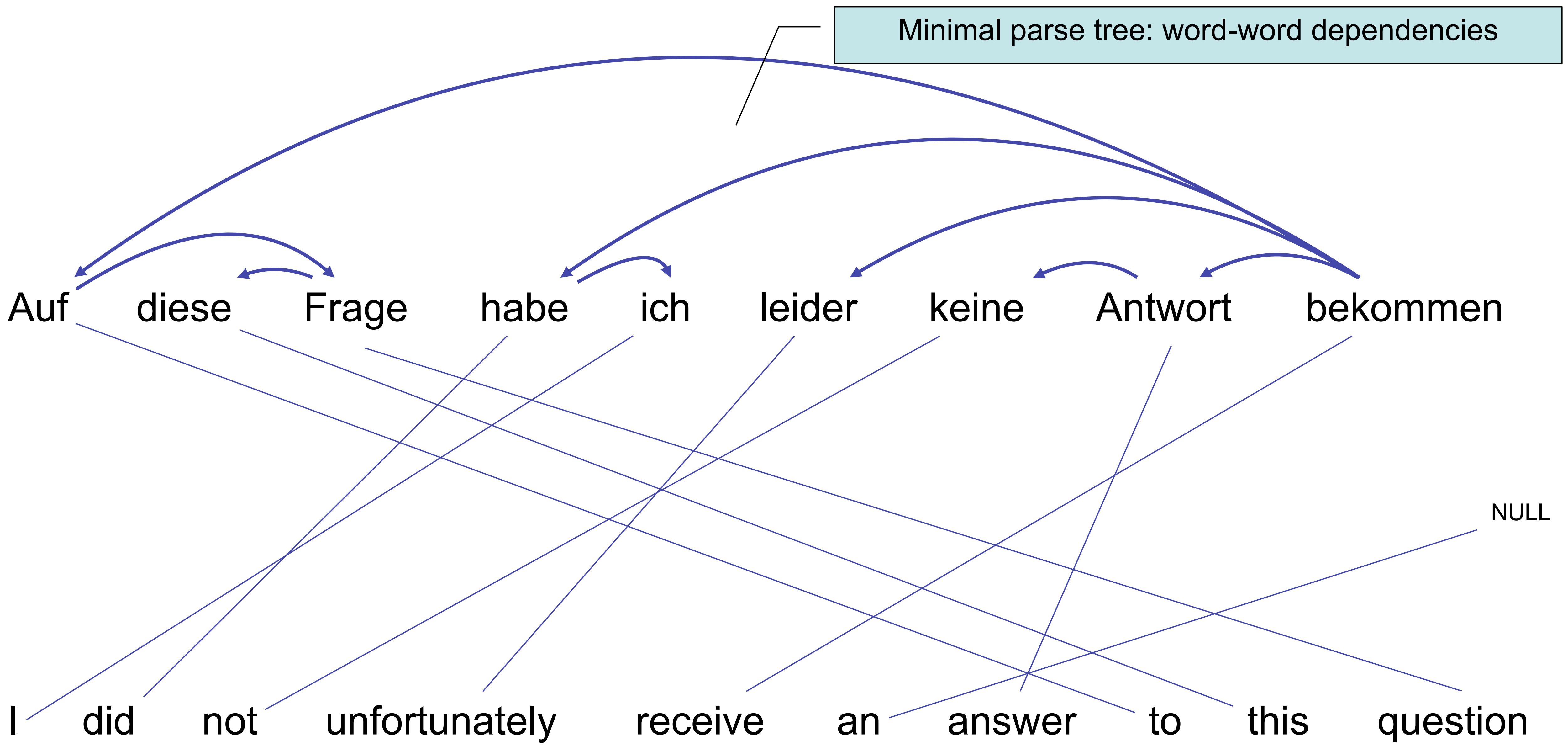
I did | not | unfortunately | receive an answer | to this question

Score each phrase pair using several features

Phrase Translation Models

- Capture translations **in context**
  - *en Amerique*: **to** America
  - *en anglais*: **in** English
- State-of-the-art for several years
- Each source/target phrase pair is scored by several weighted features.
- The weighted sum of model features is the whole translation’s score.
- Phrases don’t overlap (cf. language models) but have “reordering” features.

Single-Tree Translation Models

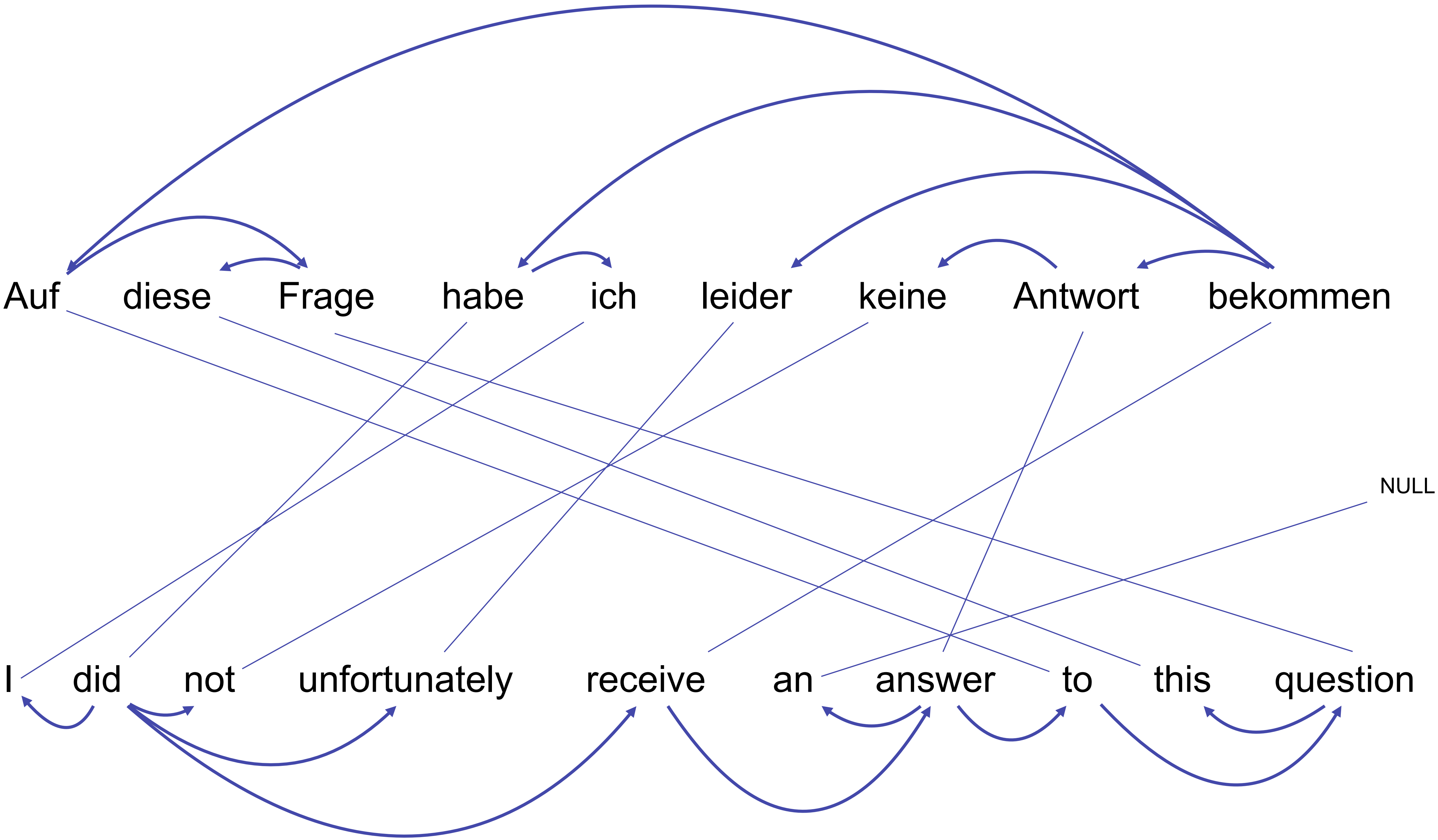

Minimal parse tree: word-word dependencies
Auf diese Frage habe ich leider keine Antwort bekommen
NULL
I did not unfortunately receive an answer to this question

Parse trees with deeper structure have also been used.

Single-Tree Translation Models

- Either source or target has a hidden tree/parse structure
  - Also known as “tree-to-string” or “tree-transducer” models
- The side with the tree generates words/phrases in tree, not string, order.
- Nodes in the tree also generate words/phrases on the other side.
- English side is often parsed, whether it’s source or target, since English parsing is more advanced.

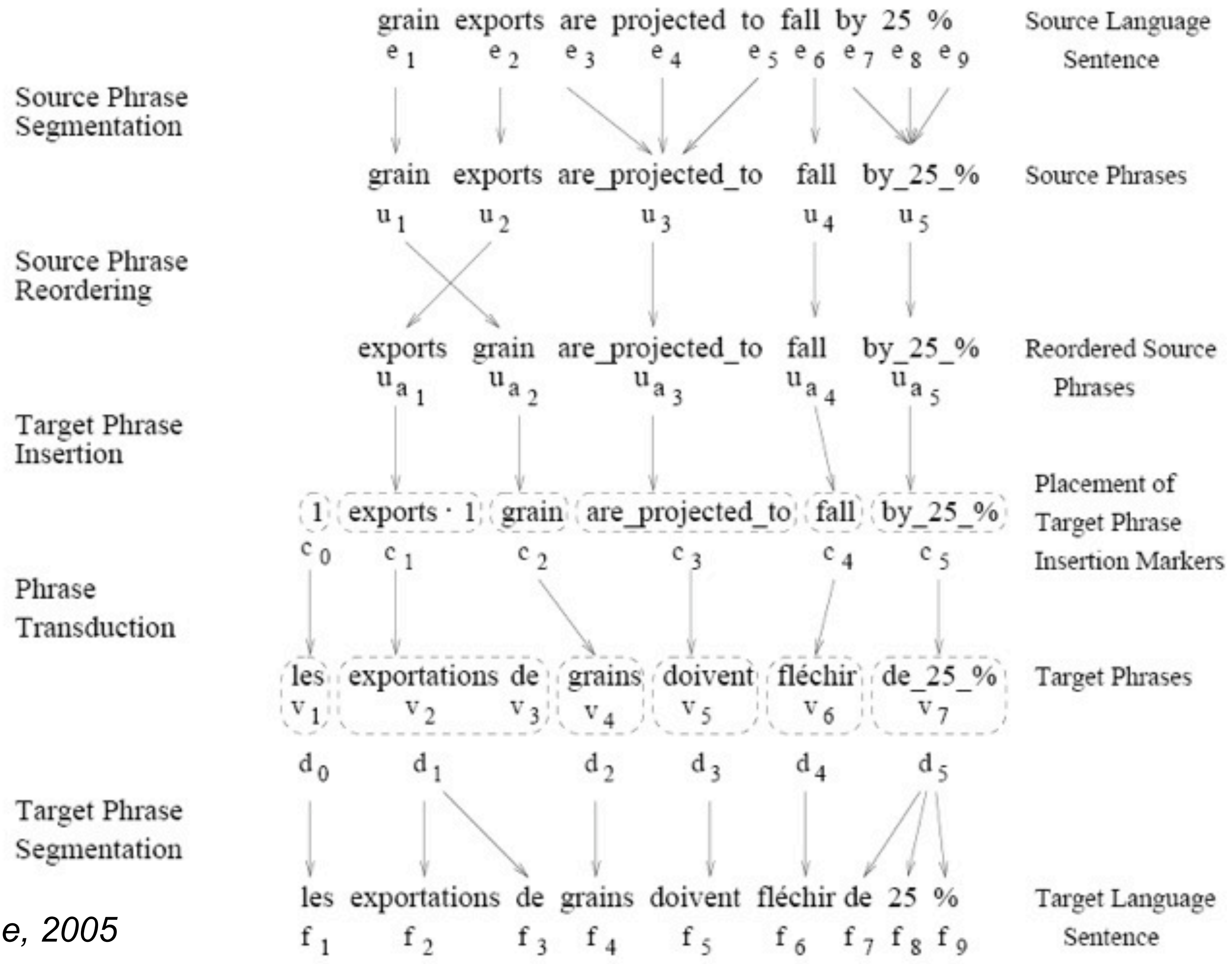
Tree-Tree Translation Models



Tree-Tree Translation Models

- Both sides have hidden tree structure
  - Can be represented with a “synchronous” grammar
- Some models assume isomorphic trees, where parent-child relations are preserved; others do not.
- Trees can be fixed in advance by monolingual parsers or induced from data (e.g. Hiero).
- Cheap trees: project from one side to the other

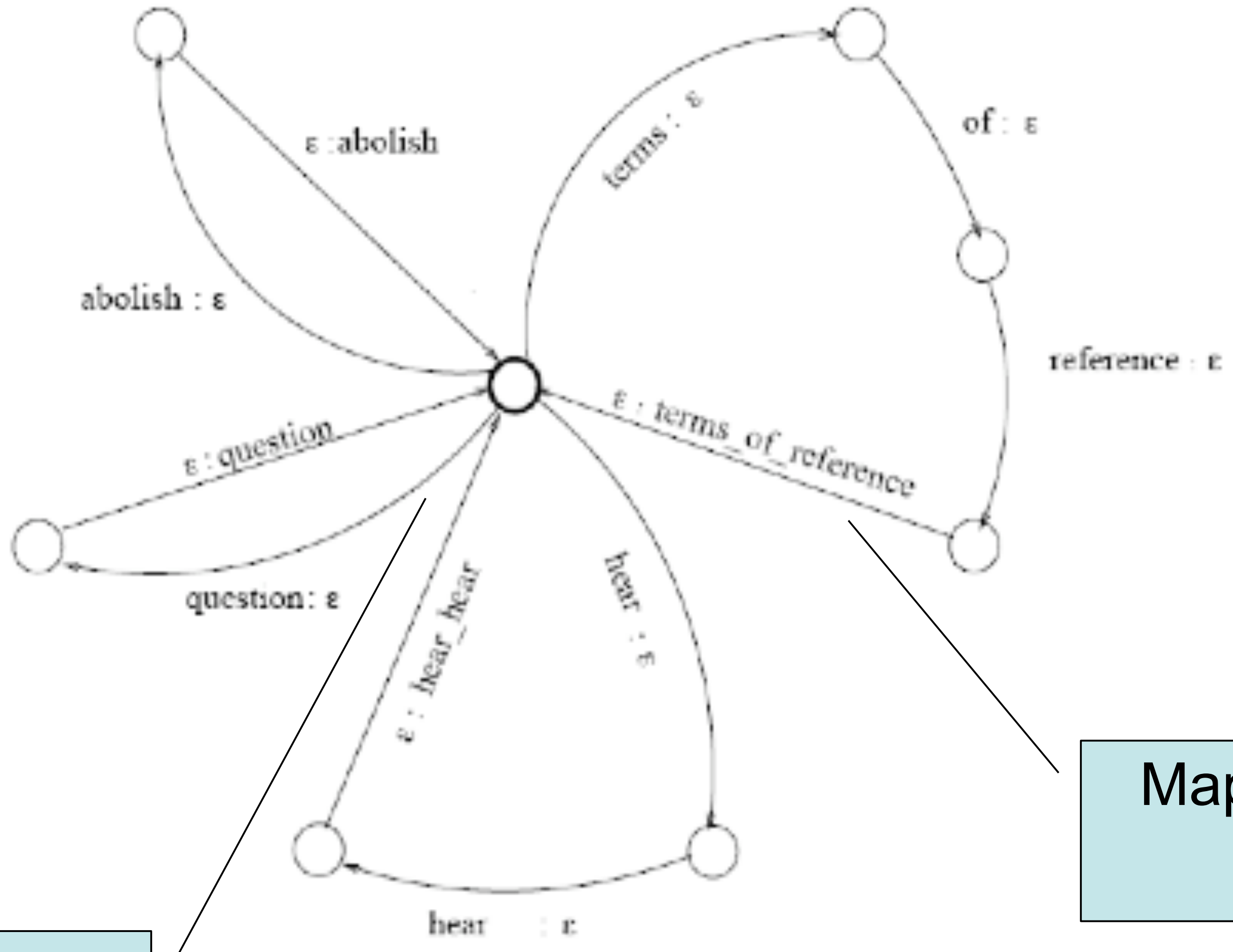
Finite State Models


grain exports are projected to fall by 25 %
e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8 e_9
Source Language Sentence
Source Phrase Segmentation
grain exports are_projected_to fall by_25_%
u_1 u_2 u_3 u_4 u_5
Source Phrases
Source Phrase Reordering
exports grain are_projected_to fall by_25_%
u_{a_1} u_{a_2} u_{a_3} u_{a_4} u_{a_5}
Reordered Source Phrases
Target Phrase Insertion
1 exports · 1 grain are_projected_to fall by_25_%
c_0 c_1 c_2 c_3 c_4 c_5
Placement of Target Phrase Insertion Markers
Phrase Transduction
les exportations de grains doivent fléchir de_25_%
v_1 v_2 v_3 v_4 v_5 v_6 v_7
Target Phrases
d_0 d_1 d_2 d_3 d_4 d_5
Target Phrase Segmentation
les exportations de grains doivent fléchir de 25 %
f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9
Target Language Sentence

Kumar, Deng & Byrne, 2005

Finite State Models

First transducer in the pipeline



Here a unigram model of phrases

Map distinct words to phrases

Kumar, Deng & Byrne, 2005

Finite State Models

- Natural composition with other finite state processes, e.g. Chinese word segmentation
- Standard algorithms and widely available tools (e.g. AT&T fsm toolkit)
- Limit reordering to finite offset
- Often impractical to compose all finite state machines offline

Learning Word Translations from Parallel Text

The “IBM Models”

Lexical translation

- How to translate a word → look up in dictionary

  Haus — *house, building, home, household, shell.*

- *Multiple translations*
  - some more frequent than others
  - for instance: *house*, and *building* most common
  - special cases: *Haus* of a *snail* is its *shell*

- Note: During all the lectures, we will translate from a foreign language into English

Collect statistics

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

Translation of *Haus*	**Count**
house	8,000
building	1,600
home	200
household	150
shell	50

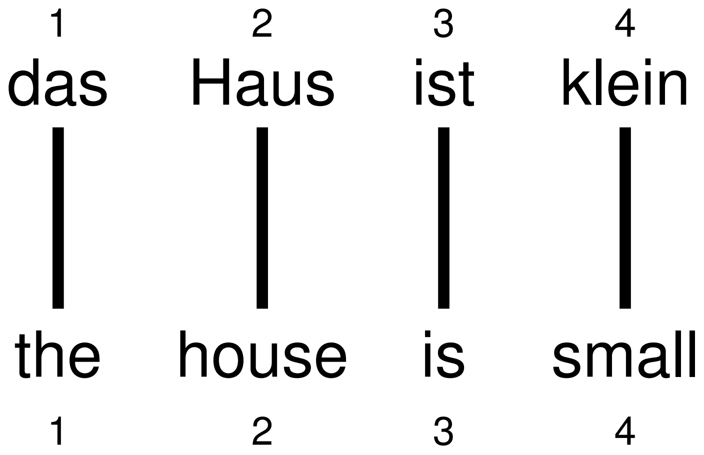
Estimate translation probabilities

- *Maximum likelihood estimation*

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \textit{house}, \\ 0.16 & \text{if } e = \textit{building}, \\ 0.02 & \text{if } e = \textit{home}, \\ 0.015 & \text{if } e = \textit{household}, \\ 0.005 & \text{if } e = \textit{shell}. \end{cases}$$

Alignment

- In a parallel text (or when we translate), we **align** words in one language with the words in the other



- Word *positions* are numbered 1–4

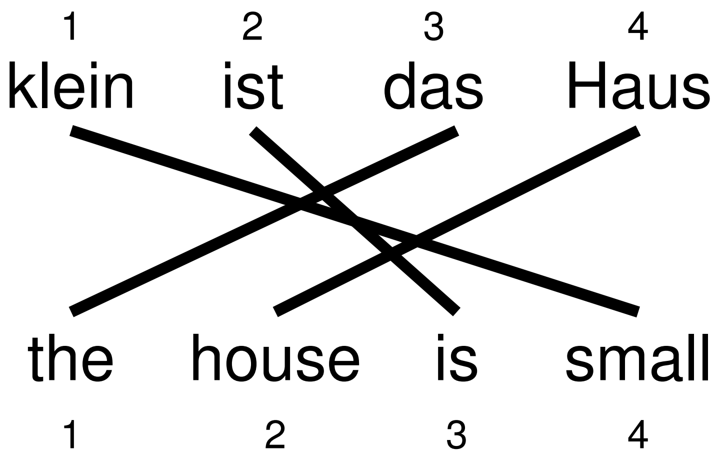
Alignment function

- Formalizing *alignment* with an **alignment function**
- Mapping an English target word at position i to a German source word at position j with a function $a : i \rightarrow j$
- Example

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

Reordering

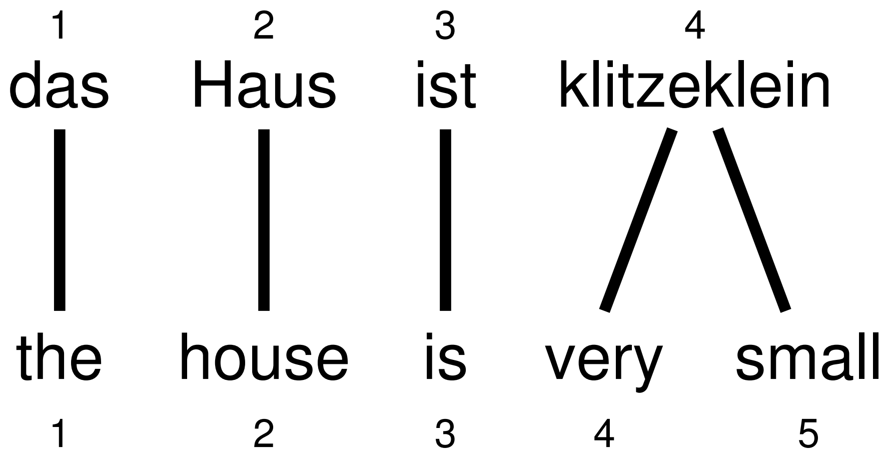
- Words may be **reordered** during translation



$$a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$

One-to-many translation

- A source word may translate into **multiple** target words



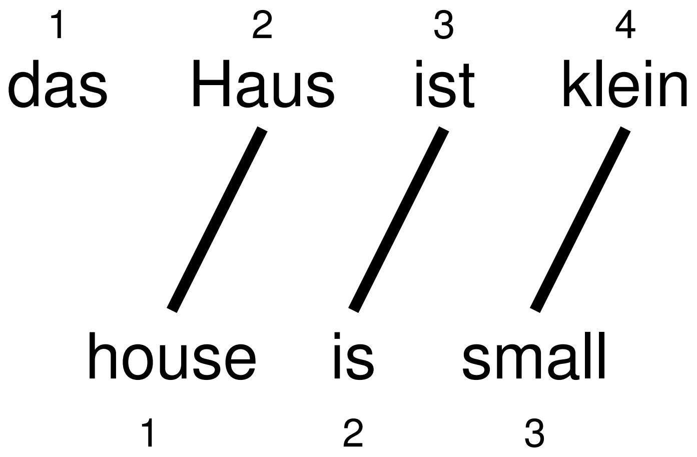
1 das 2 Haus 3 ist 4 klitzeklein

the 1 house 2 is 3 very 4 small 5

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

Dropping words

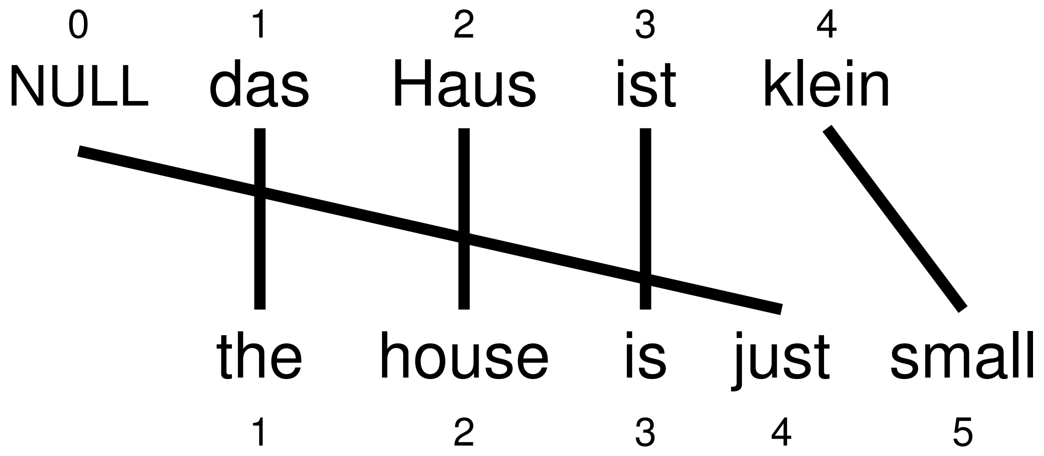
- Words may be **dropped** when translated
  - The German article *das* is dropped



$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

Inserting words

- Words may be **added** during translation
  - The English *just* does not have an equivalent in German
  - We still need to map it to something: special NULL token



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

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 **f** $= (f_1, ..., f_{l_f})$ of length l_f
  - to an English sentence **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*

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

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

Learning lexical translation models

- We would like to *estimate* the lexical translation probabilities $t(e|f)$ from a parallel corpus

- ... but we do not have the alignments

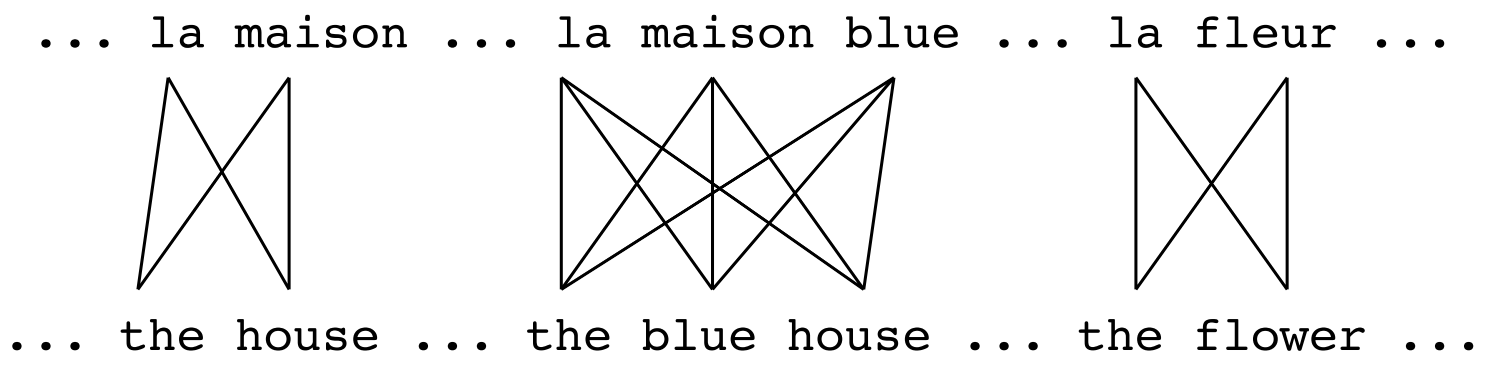
- **Chicken and egg problem**
  - if we had the *alignments*,
    $\longrightarrow$ we could estimate the *parameters* of our generative model
  - if we had the *parameters*,
    $\longrightarrow$ we could estimate the *alignments*

EM algorithm

- **Incomplete data**
  - if we had *complete data*, would could estimate *model*
  - if we had *model*, we could fill in the *gaps in the data*

- **Expectation Maximization (EM)** in a nutshell
  - initialize model parameters (e.g. uniform)
  - assign probabilities to the missing data
  - estimate model parameters from completed data
  - iterate

EM algorithm

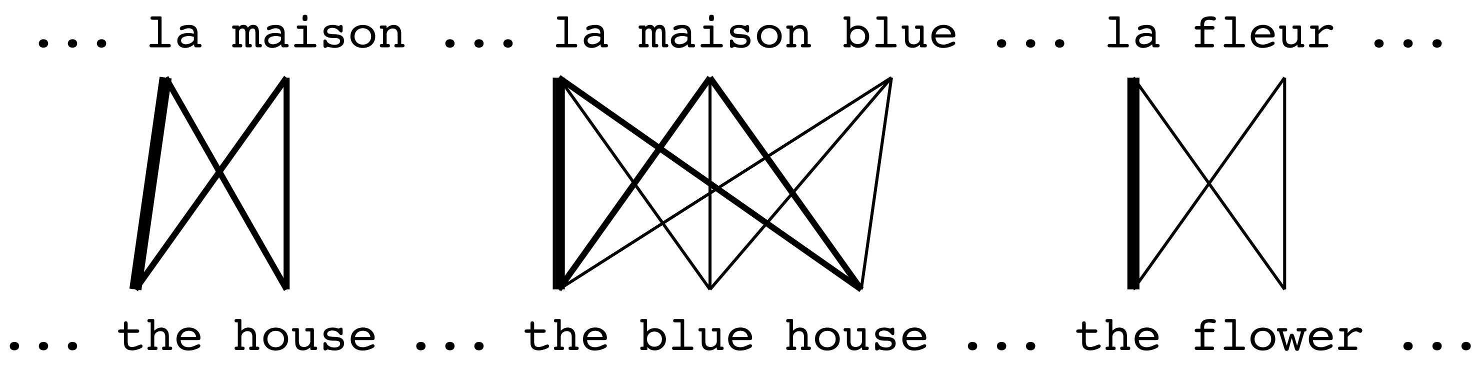


... la maison ... la maison blue ... la fleur ...

... the house ... the blue house ... the flower ...

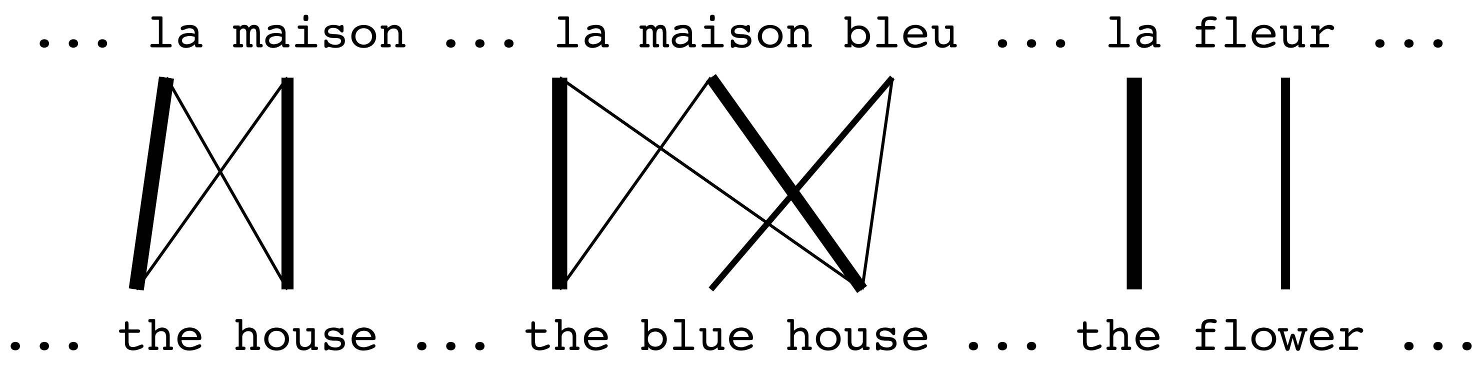
- Initial step: all alignments equally likely
- Model learns that, e.g., *la* is often aligned with *the*

EM algorithm


... la maison ... la maison blue ... la fleur ...
... the house ... the blue house ... the flower ...

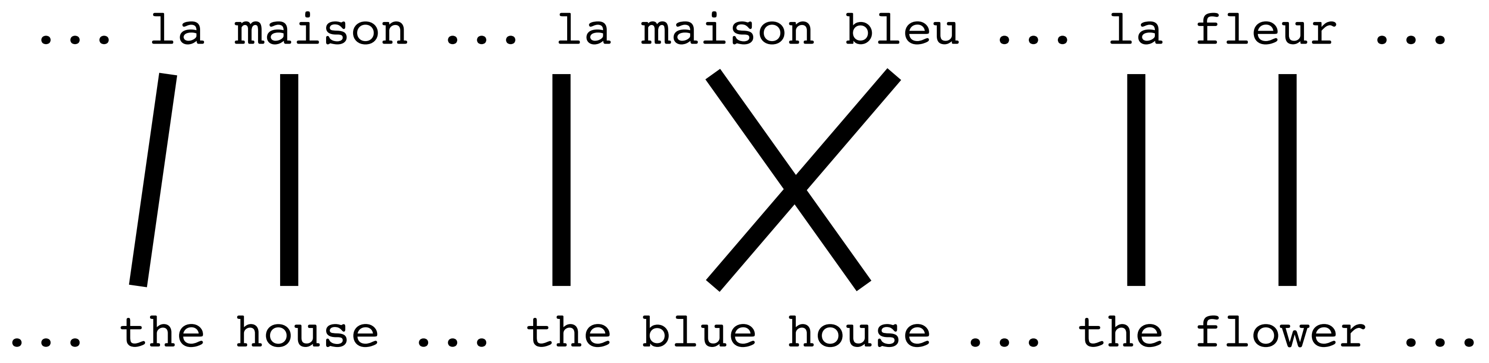
- After one iteration
- Alignments, e.g., between *la* and *the* are more likely

EM algorithm


... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...

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

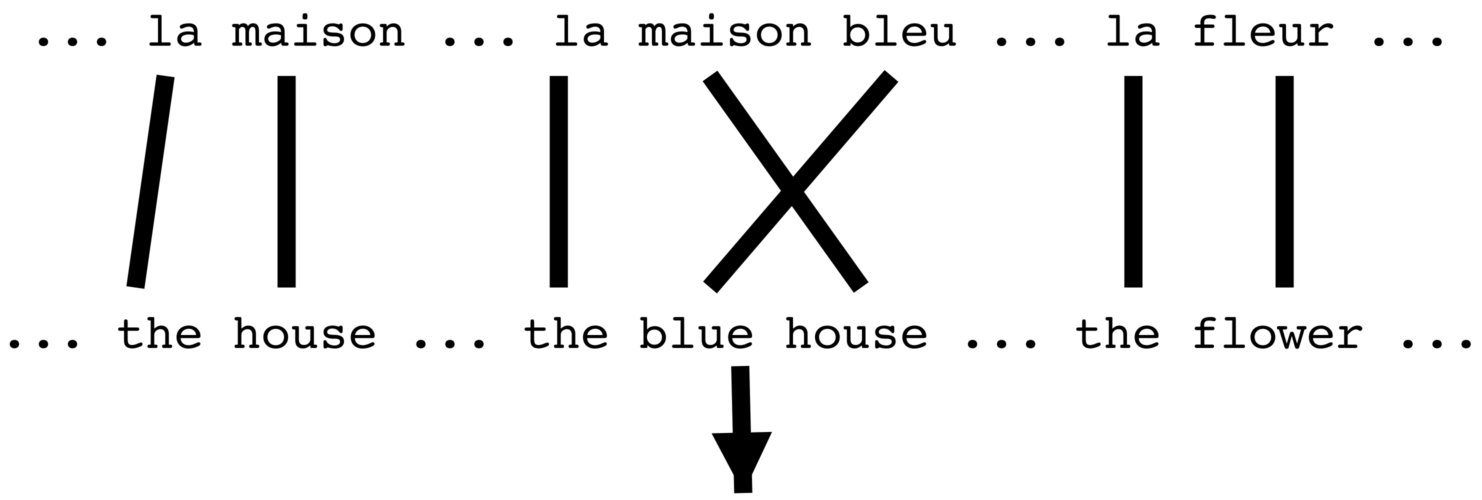
EM algorithm



```
... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...
```

- Convergence
- Inherent hidden structure revealed by EM

EM algorithm



... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...

$$p(\mathtt{la}|\mathtt{the}) = 0.453$$
$$p(\mathtt{le}|\mathtt{the}) = 0.334$$
$$p(\mathtt{maison}|\mathtt{house}) = 0.876$$
$$p(\mathtt{bleu}|\mathtt{blue}) = 0.563$$
$$\ldots$$

- Parameter estimation from the aligned corpus

IBM Model 1 and EM

- EM Algorithm consists of two steps
- **Expectation-Step**: Apply model to the data
  - parts of the model are hidden (here: alignments)
  - using the model, assign probabilities to possible values
- **Maximization-Step**: Estimate model from data
  - take assign values as fact
  - collect counts (weighted by probabilities)
  - estimate model from counts
- Iterate these steps until **convergence**

IBM Model 1 and EM

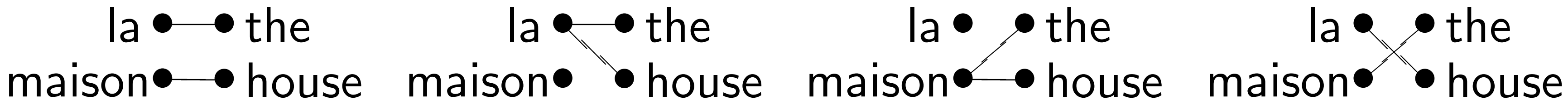
- We need to be able to compute:
  - Expectation-Step: probability of alignments
  - Maximization-Step: count collection

IBM Model 1 and EM

- **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**

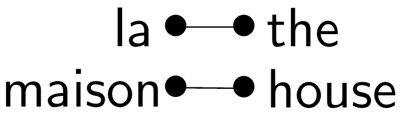


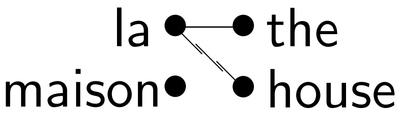
IBM Model 1 and EM

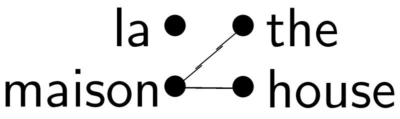
- **Probabilities**

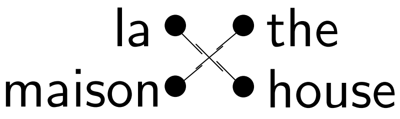
$$\begin{array}{cc} p(\text{the}|\text{la}) = 0.7 & p(\text{house}|\text{la}) = 0.05 \\ p(\text{the}|\text{maison}) = 0.1 & p(\text{house}|\text{maison}) = 0.8 \end{array}$$

- **Alignments**









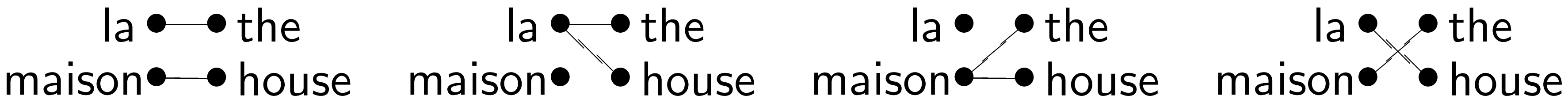
$p(\mathbf{e}, a|\mathbf{f}) = 0.56$ $p(\mathbf{e}, a|\mathbf{f}) = 0.035$ $p(\mathbf{e}, a|\mathbf{f}) = 0.08$ $p(\mathbf{e}, a|\mathbf{f}) = 0.005$

IBM Model 1 and EM

- **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**



la the maison house

la the maison house

la the maison house

la the maison house

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

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

- **Counts**

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

IBM Model 1 and EM: Expectation Step

- We need to compute $p(a|\mathbf{e},\mathbf{f})$

- Applying the *chain rule*:

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

- We already have the formula for $p(\mathbf{e},\mathbf{a}|\mathbf{f})$ (definition of Model 1)

IBM Model 1 and EM: Expectation Step

- We need to compute $p(\mathbf{e}|\mathbf{f})$

$$
\begin{aligned}
p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e},a|\mathbf{f}) \\
&= \sum_{a(1)=0}^{l_f} ... \sum_{a(l_e)=0}^{l_f} p(\mathbf{e},a|\mathbf{f}) \\
&= \sum_{a(1)=0}^{l_f} ... \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})
\end{aligned}
$$

IBM Model 1 and EM: Expectation Step

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} ... \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \\ &= \frac{\epsilon}{(l_f+1)^{l_e}} \sum_{a(1)=0}^{l_f} ... \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \\ &= \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) \end{aligned}$$

- Note the trick in the last line
  - removes the need for an *exponential* number of products

→ this makes IBM Model 1 estimation **tractable**

IBM Model 1 and EM: Expectation Step

- Combine what we have:

$$
\begin{aligned}
p(\mathbf{a}|\mathbf{e},\mathbf{f}) &= p(\mathbf{e},\mathbf{a}|\mathbf{f})/p(\mathbf{e}|\mathbf{f}) \\
&= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)} \\
&= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)}
\end{aligned}
$$

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:

$$c(e|f;\mathbf{e},\mathbf{f}) = \sum_{a} p(a|\mathbf{e},\mathbf{f}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)})$$

- With the same simplication as before:

$$c(e|f;\mathbf{e},\mathbf{f}) = \frac{t(e|f)}{\sum_{j=1}^{l_e} t(e|f_{a(j)})} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

IBM Model 1 and EM: Maximization Step

- After collecting these counts over a corpus, we can estimate the model:

$$t(e|f;\mathbf{e},\mathbf{f}) = \frac{\sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}{\sum_f \sum_{(\mathbf{e},\mathbf{f})} c(e|f;\mathbf{e},\mathbf{f}))}$$

IBM Model 1 and EM: Pseudocode

```
initialize t(e|f) uniformly
do
  set count(e|f) to 0 for all e,f
  set total(f) to 0 for all f
  for all sentence pairs (e_s,f_s)
    for all words e in e_s
      total_s = 0
      for all words f in f_s
        total_s += t(e|f)
    for all words e in e_s
      for all words f in f_s
        count(e|f) += t(e|f) / total_s
        total(f)   += t(e|f) / total_s
  for all f in domain( total(.) )
    for all e in domain( count(.|f) )
      t(e|f) = count(e|f) / total(f)
until convergence
```

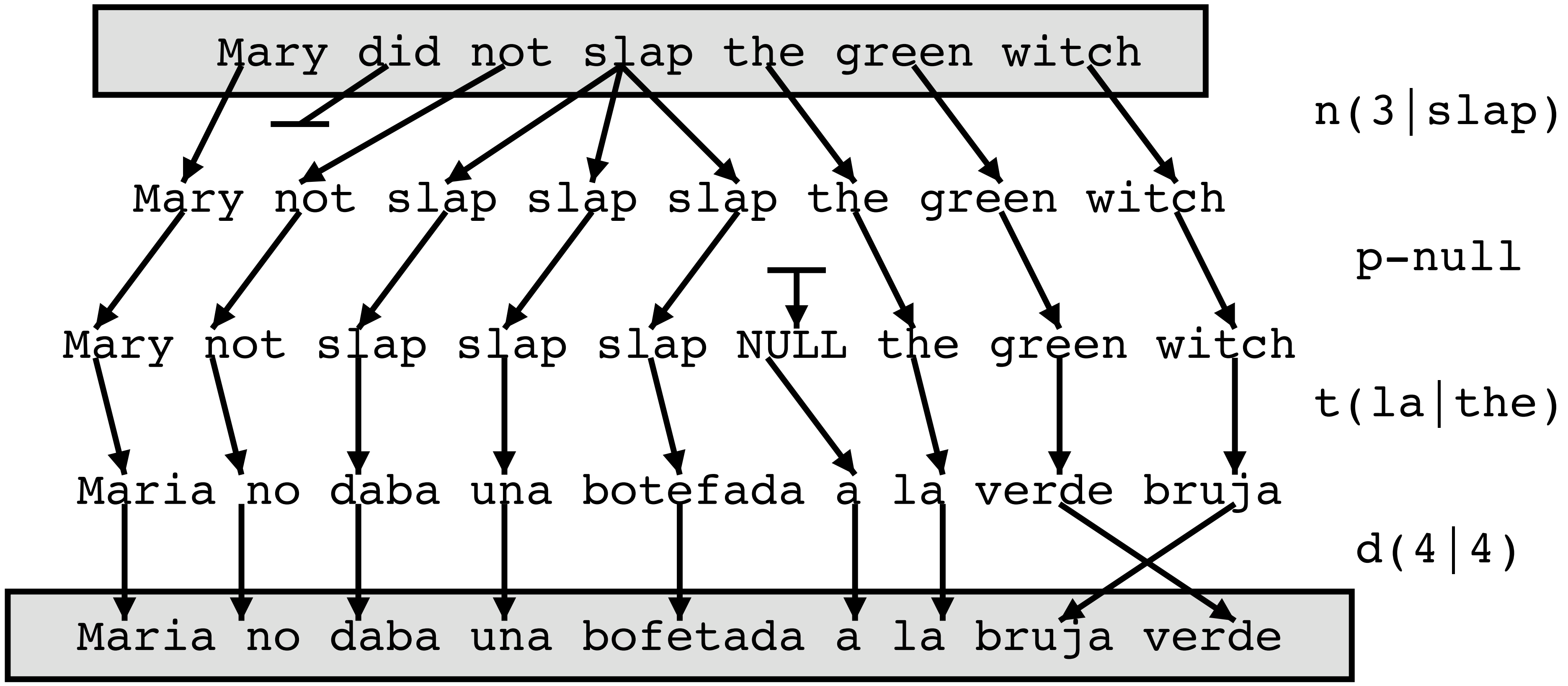
Higher IBM Models

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
  - → *exhaustive* count collection becomes computationally too expensive
  - **sampling** over high probability alignments is used instead

IBM Model 4



Mary did not slap the green witch

n(3|slap)

Mary not slap slap slap the green witch

p-null

Mary not slap slap slap NULL the green witch

t(la|the)

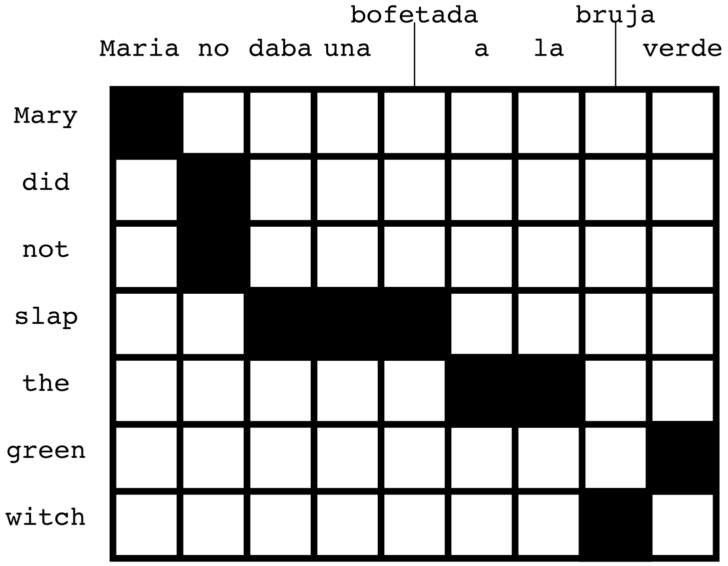
Maria no daba una botefada a la verde bruja

d(4|4)

Maria no daba una bofetada a la bruja verde

Word alignment

- Notion of **word alignment** valuable
- Shared task at NAACL 2003 and ACL 2005 workshops

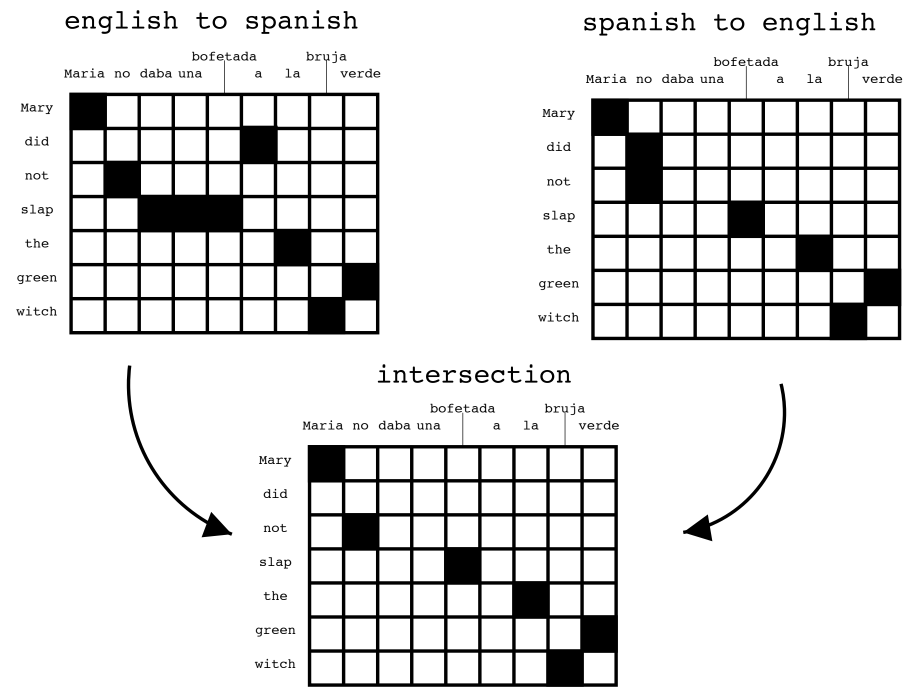


	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap			■	■	■				
the						■	■		
green									■
witch								■	

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)
- But we need *many-to-many* mappings

Symmetrizing word alignments



english to spanish

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

spanish to english

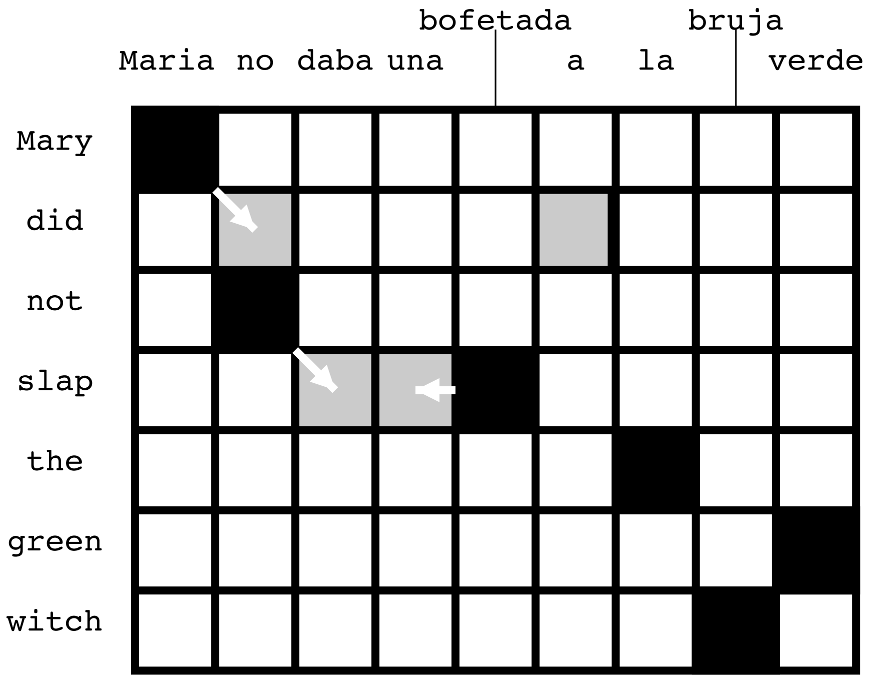
	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap					■				
the							■		
green									■
witch								■	

intersection

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

- *Intersection* of GIZA++ bidirectional alignments

Symmetrizing word alignments



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

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 = intersect(e2f,f2e);
  GROW-DIAG(); FINAL(e2f); FINAL(f2e);

GROW-DIAG():
  iterate until no new points added
    for english word e = 0 ... en
      for foreign word f = 0 ... fn
        if ( e aligned with f )
          for each neighboring point ( e-new, f-new ):
            if ( ( e-new not aligned and f-new not aligned ) and
                 ( e-new, f-new ) in union( e2f, f2e ) )
              add alignment point ( e-new, f-new )
FINAL(a):
  for english word e-new = 0 ... en
    for foreign word f-new = 0 ... fn
      if ( ( e-new not aligned or f-new not aligned ) and
           ( e-new, f-new ) in alignment a )
        add alignment point ( e-new, f-new )
```