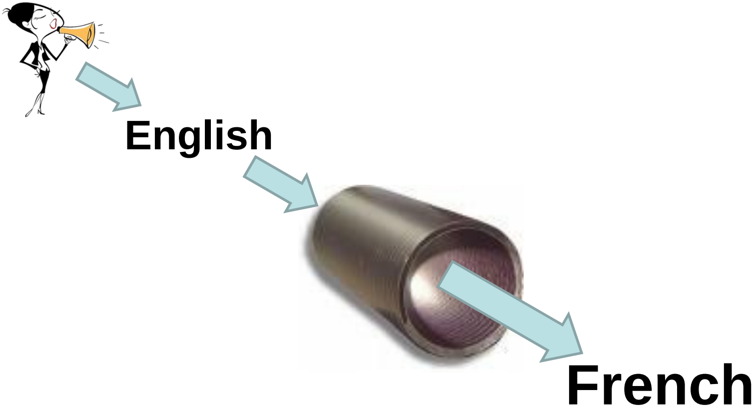
Noisy Channel Framework

$$p(e) \rightarrow e \rightarrow p(f|e) \rightarrow f$$



Noisy Channel Framework

$$\boxed{p(e)} \rightarrow e \rightarrow \boxed{p(f|e)} \rightarrow f$$

$$p(f) = p(e)p(f|e)$$

English

French

Noisy Channel Framework

Applying Bayes' Rule, we have:

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

Thus:

$$\hat{e} = \operatorname*{argmax}_{e} p(e|f) = \operatorname*{argmax}_{e} p(e)p(f|e)$$

Noisy Channel Framework

$$\hat{e} = \operatorname*{argmax}_{e} p(e)p(f|e)$$

Noisy Channel Framework

$$\hat{e} = \underbrace{\operatorname*{argmax}_{e}}_{\text{Decoder}} \underbrace{p(e)}_{\text{Language Model}} \underbrace{p(f|e)}_{\text{Translation Model}}$$

Noisy Channel Framework

- The *translation model* models how likely it is that *f* is a translations of *e* – **adequacy**.
- The *language model* models how likely it is that *e* is an acceptable sentence – **fluency**.
- The *decoder* searches for the most likely *e*.

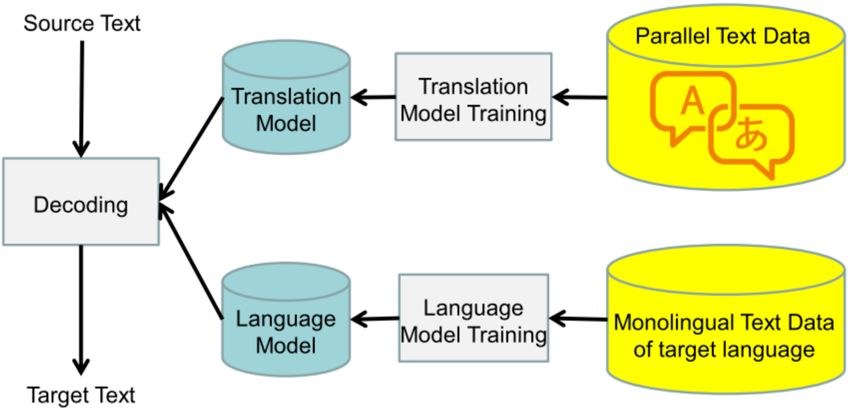
We have introduced language models in previouse lectures, here we will mainly focus on translation models and decoding algorithms

Noisy Channel Framework

- The *translation model* models how likely it is that f is a translations of e – **adequacy.**
- The *language model* models how likely it is that e is an acceptable sentence – **fluency.**
- The *decoder* searches for the most likely e.

We have introduced language models in previouse lectures, here we will mainly focus on translation models and decoding algorithms

SMT Workflow



Content

Categories of translation models

Various translation models have been proposed, which belong to different categories, according to the language units on which they are built up:

- Word-based models
  - IBM models 1-5
  - HMM models
- Phrase-based models
- Syntax-based models
  - Tree-to-string models
  - String-to-tree models
  - Tree-to-tree models
  - Dependency-based models

IBM Models

IBM researchers proposed 5 models with increased complexity:

- IBM Model 1: only consider lexical translation probabilities
- IBM Model 2: add a absolute reordering model
- IBM Model 3: add a fertility model
- IBM Model 4: add a relative reordering model
- IBM Model 5:

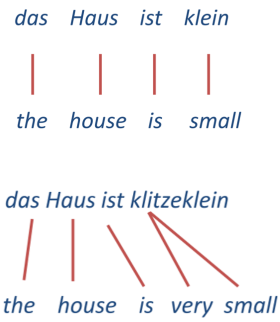
Lexical translation probabilities

English	Chinese	Prob.
a	一	0.2
a	一个	0.4
a	个	0.2
a	一只	0.1
a	一本	0.05
a	...	...

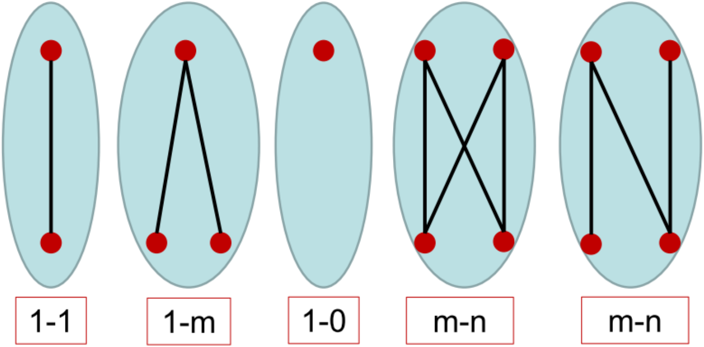
English	Chinese	Prob.
book	书	0.7
book	预定	0.2
book	...	...
take	拿	0.4
take	带走	0.3
take	...	...

Word alignment

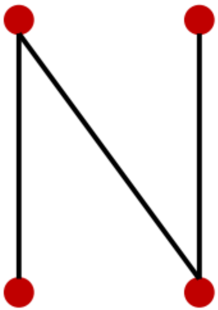
- To estimate the word translation probabilities, we need alignment between words in the parallel sentences



Word alignment patterns

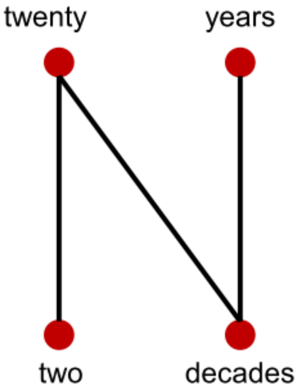


Word alignment patterns



Can you image a word alignment pattern like this?

Word alignment patterns



Can you image a word alignment pattern like this?

Learning lexical translation models

- We would like to estimate the lexical translation probabilities from a parallel corpus...
- but we do not have the alignments:
  - If we had the alignments, we could estimate the lexical translation probabilities.
  - If we had the probabilities, we could estimate the alignments.

Learning lexical translation models

- We would like to estimate the lexical translation probabilities from a parallel corpus...
- but we do not have the alignments:
  - If we had the alignments, we could estimate the lexical translation probabilities.
  - If we had the probabilities, we could estimate the alignments.

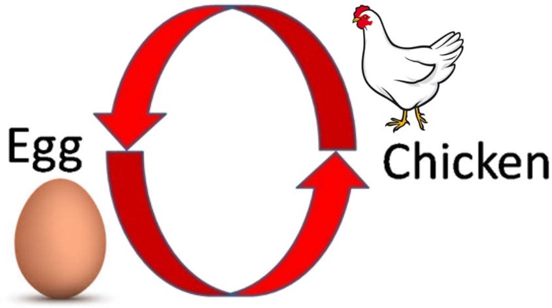
Learning lexical translation models

- We would like to estimate the lexical translation probabilities from a parallel corpus...
- but we do not have the alignments:
  - If we had the alignments, we could estimate the lexical translation probabilities.
  - If we had the probabilities, we could estimate the alignments.

Learning lexical translation models

- We would like to estimate the lexical translation probabilities from a parallel corpus...
- but we do not have the alignments:
  - If we had the alignments, we could estimate the lexical translation probabilities.
  - If we had the probabilities, we could estimate the alignments.

A Paradox



EM Algorithm

- Incomplete data
  - If we had complete data, we could estimate model.
  - If we had the model, we could fill in the gaps in the data.
- Solution: Expectation Maximization (EM) Algorithm
  - Initialize model parameters. (e.g. uniform)
  - Assign probabilities to the missing data. (E-step)
  - Astimate model parameters from completed data. (M-step)
  - Iterate E-step and M-step until the model converges.

How does EM algorithm work?

EM Algorithm consists of two steps:

1. Expectation-Step: Apply model to the data
   - parts of the data are hidden (here: alignments)
   - using the model, assign probabilities of the hidden data to possible values (alignments)
2. Maximization-Step: Estimate new model from data
   - take assigned values as fact
   - collect counts (weighted by probabilities)
   - estimate new model from counts

Iterate the E-step and the M-step until convergence

Example

Consider a parallel corpus containing just two pairs:

blue house	house
maison bleu	maison

How many possible alignments in the first pair?

How many in the second pair?

Example

Consider a parallel corpus containing just two pairs:

blue house	house
maison bleu	maison

How many possible alignments in the first pair?

How many in the second pair?

We will simplify the example by ruling out many-to-one or zero-to-one alignments.

Example

Consider a parallel corpus containing just two pairs:

blue house	house
maison bleu	maison

How many possible alignments in the first pair? 2

How many in the second pair? 1

We will simplify the example by ruling out many-to-one or zero-to-one alignments.

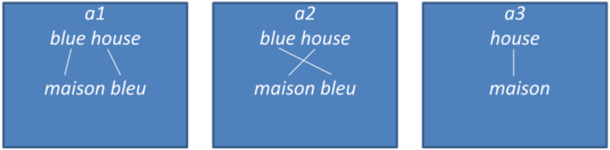
Step 1 (Initialisation)

Set parameter values uniformly.

- t(bleu|house) = 1/2
- t(maison|house) = 1/2
- t(bleu|blue) = 1/2
- t(maison|blue) = 1/2

Step 2 (Expectation)

Compute the probability of all alignments.



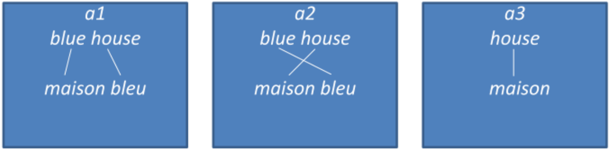
p(a1, maison bleu|blue house) = t(maison|blue) * t(bleu|house) = ½ * ½ = ¼

p(a2, maison bleu|blue house) = t(maison|house) * t(bleu|blue) = ½ * ½ = ¼

p(a3, maison| house) = t(maison|house) = ½

Step 3 (Expectation)

Normalise for all alignments.



p(a1|maison bleu,blue house) = 1/4 ÷ 2/4 = 1/2

p(a2|maison bleu,blue house) = 1/4 ÷2/4 = 1/2

p(a3|maison, house) = 1/2 ÷ 1/2 = 1

Step 4 (Maximisation)

Collect fractional counts

- tc(bleu|house) = 1/2
- tc(maison|house) = 1/2 + 1 = 3/2
- tc(bleu|blue) = 1/2
- tc(maison|blue) = 1/2

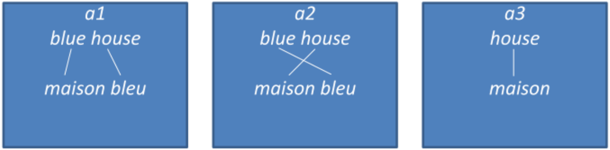
Step 5 (Maximisation)

Normalise fractional counts to yield revised parameter values

- t(bleu|house) = 1/2 ÷ 2 = 1/4
- t(maison|house) = 3/2 ÷ 2 = 3/4
- t(bleu|blue) = 1/2 ÷ 1 = 1/2
- t(maison|blue) = 1/2 ÷ 1 = 1/2

Repeat Step 2 (Expectation)

Compute the probability of all alignments.



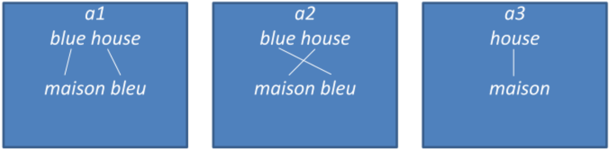
p(a1, maison bleu|blue house) = t(maison|blue) * t(bleu|house) = 1/2 * 1/4 = 1/8

p(a2, maison bleu|blue house) = t(maison|house) * t(bleu|blue) = 3/4 * 1/2 = 3/8

p(a3, maison| house) = t(maison|house) = 3/4

Repeat Step 3 (Expectation)

Normalise for all alignments.



p(a1|maison bleu,blue house) = 1/8 ÷ 4/8 = 1/4

p(a2|maison bleu,blue house) = 3/8 ÷ 4/8 = 3/4

p(a3|maison, house) = 3/4 ÷ 3/4 = 1

Repeat Step 4 (Maximisation)

Collect fractional counts

- tc(bleu|house) = 1/4
- tc(maison|house) = 3/4 + 1 = 7/4
- tc(bleu|blue) = 3/4
- tc(maison|blue) = 1/4

Repeat Step 5 (Maximisation)

Normalise fractional counts to yield revised parameter values

- t(bleu|house) = $1/4 \div 2 = 1/8$
- t(maison|house) = $7/4 \div 2 = 7/8$
- t(bleu|blue) = $3/4 \div 1 = 3/4$
- t(maison|blue) = $1/4 \div 1 = 1/4$

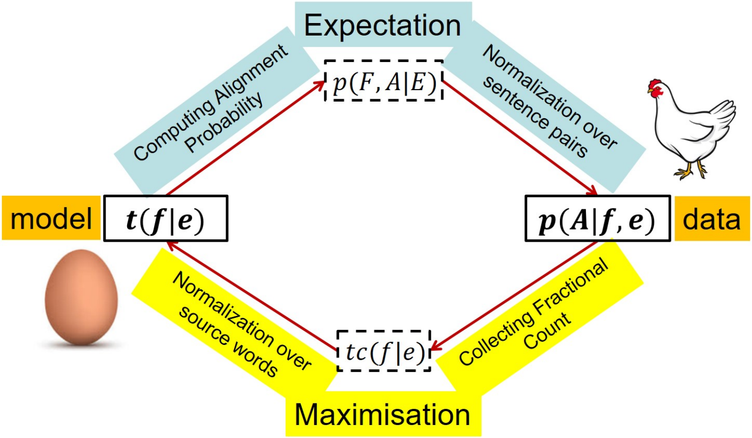
Convergence

Repeating steps 2, 3, 4 and 5 eventually yields:

- t(bleu|house) = 0.0001
- t(maison|house) = 0.9999
- t(bleu|blue) = 0.9999
- t(maison|blue) = 0.0001

It is proved that an EM algorithm is convergent.

EM Algorithm



Content

Shortcomings of word-based SMT

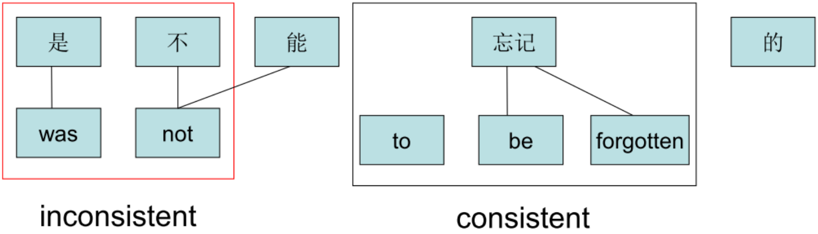
- Word-based translation models do not take into account contextual information for translation decisions
- They are not good at dealing with 1-to-many, many-to-1 and many-to-many translations.

Phrase-based Translation Models

- Phrase-based translate models are proposed to solve the problems for word-based models.
- Phrase-based models translate phrases as atomic units.
- A monolingual phrase can be any contiguous sequence of words in a sentence.
  - A phrase is not necessarily syntactically well-formed
  - A phrase is not necessarily semantically meaningful
- A bilingual phrase pair should be consistent with word alignment.

Bilingual Phrase Pairs

A bilingual phrase pair should be constistent with word alignment:



Bilingual Phrase Pairs

A real example taken from Europarl for the German phrase den Vorschlag:

English	Probability	English	Probability
the proposal	0.6277	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.025	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposals	0.0159	it	0.0068
the proposals	0.0159	....	

Learning a phrase translation table

- Task: learn the model from a parallel corpus
- Three stages:
  1. Word alignment: using IBM models or other method
  2. Extraction of phrase pairs
  3. Scoring phrase pairs

Bidirectional word alignment

- With IBM models, each target word can be aligned to at most one source word (patterns supported: 1-0,0-1,1-1,m-1).
- Therefore, it's not possible to end up with an alignment of one target word to many source words (patterns not supported: 1-m, m-m)
- To obtain a word alignment with all possible patterns, a symmetric word alignment algorithm should be adopted.

Bidirectional word alignment

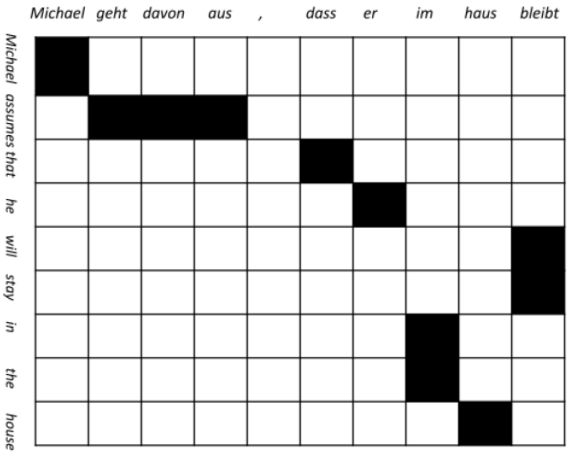
- A typical symmetric word alignment algorithm:
  - Word alignment using IBM Models in one direction.
  - Word alignment using IBM Models in the other direction.
  - Merge the above two alignment results with a certain criterion.

Consistent with word alignment

A phrase pair (e, f) is consistent with a bidirectional word alignment A if and only if:

- For all words $e_i \in e$, if exists an f_j: $(e_i, f_j) \in A$, then $f_j \in f$.
- For all word $f_j \in f$, if exists an e_i: $(e_i, f_j) \in A$, then $e_i \in e$.
- There exists an $e_i \in e$, and an $f_j \in f$: $(e_i, f_j) \in A$

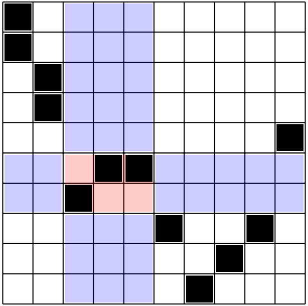
A matrix view of word alignment



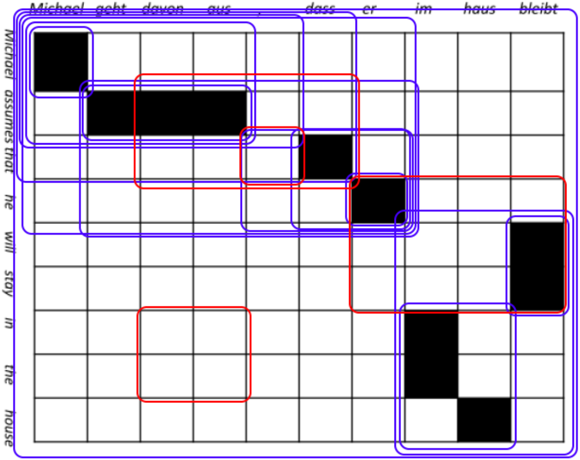
Consistent phrases in the matrix view

A consistent phrase pair defined by the red area should meet the following requirement:

- There should be one or more filled blocks in the red area.
- The blue areas should be all clear.



Phrase pair extraction



Blue box: consistent phrase pairs, Red box: inconsistent phrase pairs

Phrase pair extraction

Phrase pairs extracted from the above example:

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