# Homework 1

- Leaderboard
- Read through, submit the default output
- Time for questions on Tuesday

# Agenda

- Focus on Homework 1
  - Review IBM Models 1 & 2
  - Inference (compute best alignment from a corpus given model parameters)
  - Parameter estimation (find model parameters)
- Discuss HMM model and IBM models 3–5

# Models 1 & 2

$$p(\text{target} \mid \text{source}) = \sum_{\text{alignments}} p(\text{alignment}) \prod_{\text{target word}} p(\text{target word} \mid \text{source word})$$

Model 1: uniform
Model 2: absolute positioning

Word translation table

# Review: IBM Models 1 & 2

- A generative model is a data creation story
- IBM models: $p(f \mid e)$ — how f is generated from e
  - Assuming the data was generated by the model, which way did it most likely happen?
  - Always ask, *what are the parameters of the model?*
  - Each step of the story needs parameters

# Models 1 & 2

- Given: an English sentence **e**, parameters $\boldsymbol{q}$ and $\boldsymbol{t}$
- Choose a French length $\boldsymbol{m}$
- For each French word position $\boldsymbol{i} \in 1\ldots\boldsymbol{m}$
    - Choose a source word position $\mathbf{a_i} = \boldsymbol{q(j \mid i, l, m)}$
    - Choose a translation probability $\mathbf{t(f_i \mid e_{a_i})}$

# Model Parameters

| **t**(**f** \| **e**) | **q**(**j** \| **i, l, m**) |
|---|---|

Models 1 & 2

| f | e | p(f \| e) |
|---|---|---|
| le | the | 0.42 |
| la | the | 0.4 |
| programme | the | 0.001 |
| a | has | 0.78 |
| … | … | … |

Model 1

$$\frac{1}{l+1}$$

Model 2

| j | q(j \| 1, 6, 7) |
|---|---|
| 1 | 0.27 |
| 2 | 0.14 |
| … | … |
| 48 | 1E-75 |

# Task 1: Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently
    - For each target word, compute most likely alignment link

$$p(a_i = j \mid e, f) = q(j \mid i, l, m) t(f_i \mid e_{a_i})$$

    - Choose the one that maximizes this probability

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m) t(f_i \mid e_{a_i})$$

| | | | | | | |
|---|---|---|---|---|---|---|
| NULL | And | the | program | has | been | implemented |
| Le | programme | a | ete | mis | en | application |

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$

NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m) t(f_i \mid e_{a_i})$$

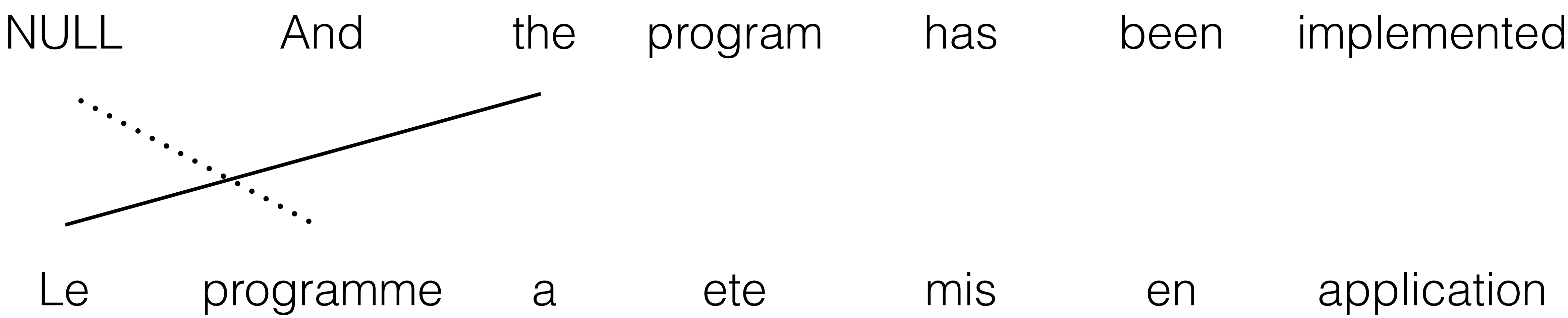
NULL And the program has been implemented

Le programme a ete mis en application

# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

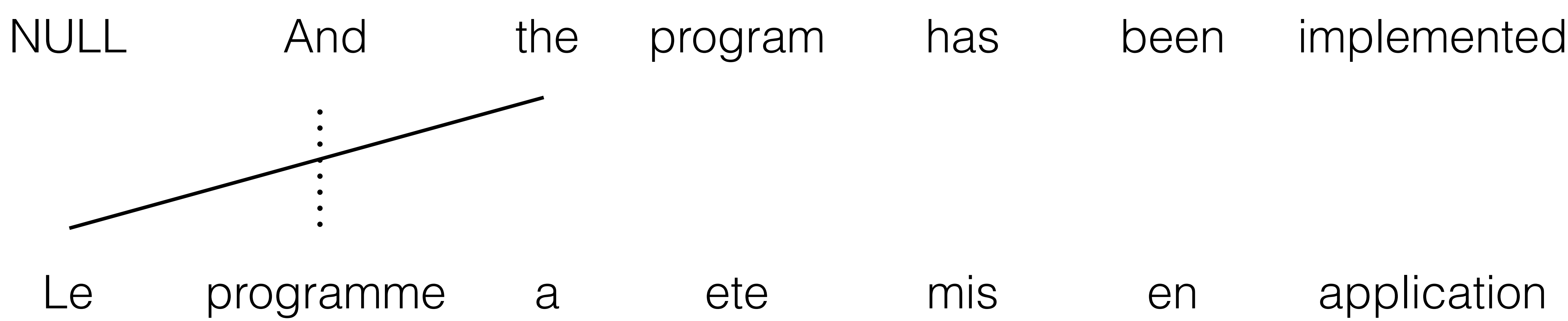
$$p(a_i = j \mid e, f) = q(j \mid i, l, m) t(f_i \mid e_{a_i})$$



# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

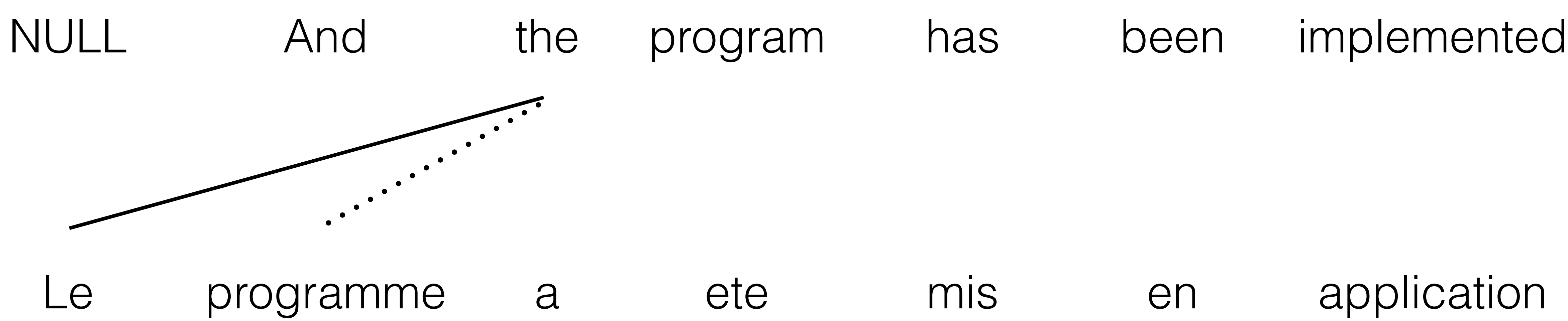
$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$



# Inference

- Input: a sentence pair (**e**,**f**) and a model (**t**,**q**)
- Models 1 & 2: each link is generated independently

$$p(a_i = j \mid e, f) = q(j \mid i, l, m)t(f_i \mid e_{a_i})$$



# Task 2: Parameter Estimation

- Guess parameters, compute expectations, adjust, repeat

```
initialize parameters t and q to something
repeat until convergence
    for every sentence
        for every target position j
            for every source position i
                count(f_j, e_i) += P(a_i = j | e_i, f_j)
                count(e_i) += P(a_i = j | e_i, f_j)
                count(j, i, l, m) += P(a_i = j | e_i, f_j)
                count(i, l, m) += P(a_i = j | e_i, f_j)
    t(f | e) = count(f, e) / count(e)
    q(j | i, l, m) = count(j, i, l, m) / count(i, l, m)
```

# Models 1 & 2

# Models 1 & 2

- Why are these algorithms so simple?

# Models 1 & 2

- Why are these algorithms so simple?
  - Each word and alignment link are generated separately; there are no dependencies between alignment links at all

# Models 1 & 2

- Why are these algorithms so simple?
  - Each word and alignment link are generated separately; there are no dependencies between alignment links at all

★ The cost of easy inference here is an overly simplistic model

# Pros and cons

- Some drawbacks of word based alignments
  - All reorderings have the same probability
  - Alignments are independent
  - No notion of multiword alignments
  - Alignments are asymmetric
  - No morphology
  - No syntax

# Pros and cons

- Some drawbacks of word based alignments
  - ~~All reorderings have the same probability~~ MODEL 2
  - Alignments are independent
  - No notion of multiword alignments
  - Alignments are asymmetric
  - No morphology
  - No syntax

# Building intuitions

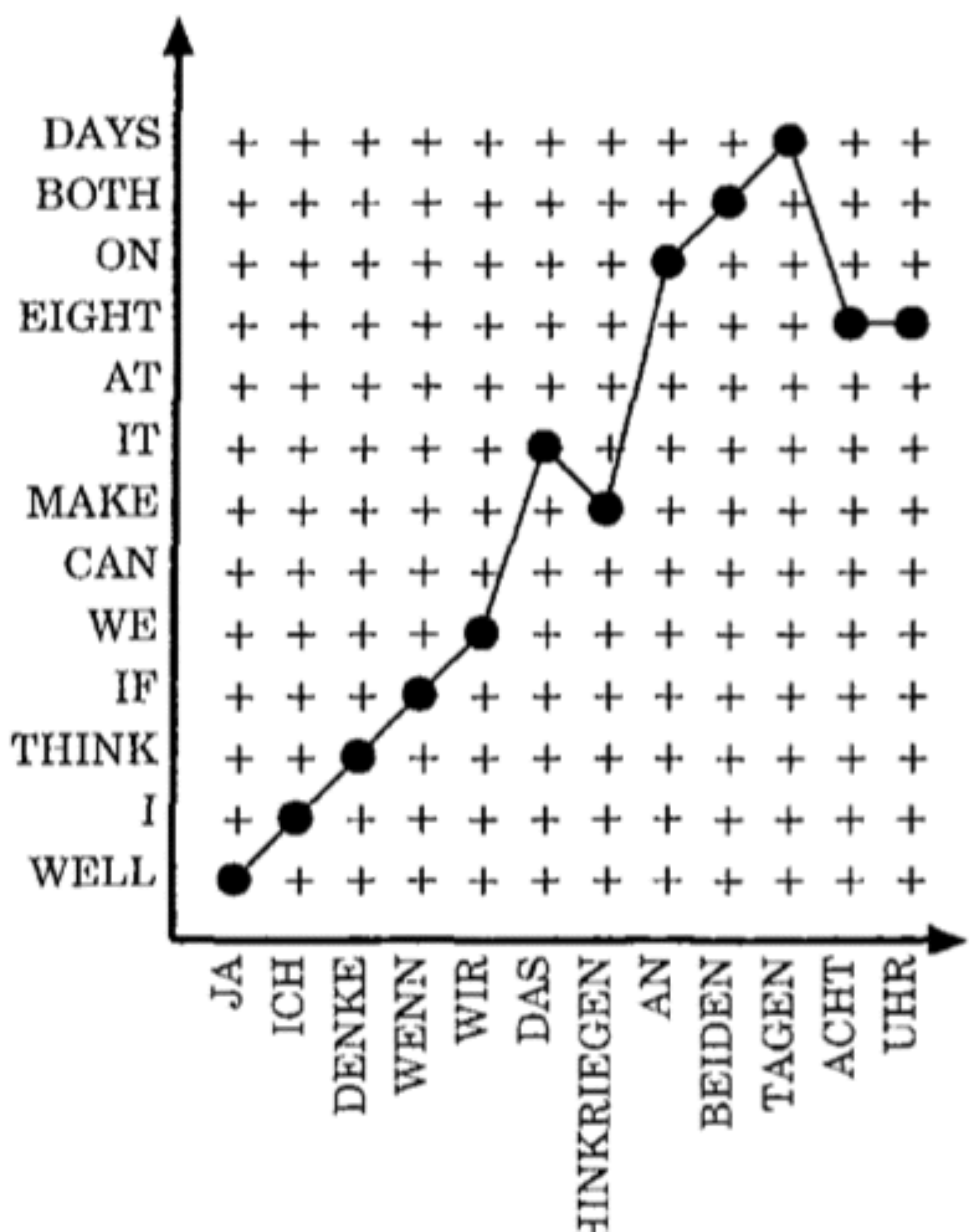
- Model 2 still generates all alignments independently
- Let's try to think of something we might change
    - Using an important (and oft-avoided) tool in the scientist's toolkit: looking at the data
    - We'll use Picaro, a tool for alignment visualization github.com/joshua-decoder/picaro

# Discuss

*5 minutes, with a neighbor or two*

- What patterns did you see in the alignments? (order them from simplest to most complex)
- Pick one pattern: how might you model it? What parameters would you need?

# Vogel, Ney, & Tillmann ('96)




*We now propose an HMM-based alignment model. The motivation is that typically we have a strong localization effect in aligning the words in parallel texts (for language pairs from Indoeuropean languages): the words are not distributed arbitrarily over the sentence positions, but tend to form clusters. Fig. 1 illustrates this effect for the language pair German–English.*

*Each word of the German sentence is assigned to a word of the English sentence. The alignments have a strong tendency to preserve the local neighborhood when going from the one language to the other language. In many cases, although not always, there is an even stronger restriction: the difference in the position index is smaller than 3.*

# HMM Model

(Hidden Markov Model)

$$p(a \mid e, m) = \prod_{i=1}^{m} q(a_i = j \mid i, l, m)$$

$$p(a \mid e, m) = \prod_{i=1}^{m} p(a_i \mid a_{i-1})$$

# HMM Model

(Hidden Markov Model)

- Model 2 used the **absolute positions** of words

$$p(a \mid e, m) = \prod_{i=1}^{m} q(a_i = j \mid i, l, m)$$

$$p(a \mid e, m) = \prod_{i=1}^{m} p(a_i \mid a_{i-1})$$

# HMM Model

(Hidden Markov Model)

- Model 2 used the **absolute positions** of words

$$p(a \mid e, m) = \prod_{i=1}^{m} q(a_i = j \mid i, l, m)$$

- A better idea: **relative positioning** using position *differences*

$$p(a \mid e, m) = \prod_{i=1}^{m} p(a_i \mid a_{i-1})$$

# HMM Model

(Hidden Markov Model)

- Model 2 used the **absolute positions** of words

$$p(a \mid e, m) = \prod_{i=1}^{m} q(a_i = j \mid i, l, m)$$

- A better idea: **relative positioning** using position *differences*

$$p(a \mid e, m) = \prod_{i=1}^{m} p(a_i \mid a_{i-1})$$

  - A “jump” probability

# HMM Model

- What are the parameters of this alignment model?

# HMM Model

- What are the parameters of this alignment model?
  - A simple table

| jump distance | prob |
|---|---|
| -3 | 0.03 |
| -2 | 0.05 |
| -1 | 0.12 |
| 0 | 0.2 |
| 1 | 0.3 |
| 2 | 0.09 |
| 3 | 0.08 |

# HMM Model

- What are the parameters of this alignment model?
  - A simple table
  - Other ways?

| jump distance | prob |
|---|---|
| -3 | 0.03 |
| -2 | 0.05 |
| -1 | 0.12 |
| 0 | 0.2 |
| 1 | 0.3 |
| 2 | 0.09 |
| 3 | 0.08 |

# HMM Model

- What are the parameters of this alignment model?
    - A simple table
    - Other ways?
- What else might we like to condition on?

| jump distance | prob |
|---|---|
| -3 | 0.03 |
| -2 | 0.05 |
| -1 | 0.12 |
| 0 | 0.2 |
| 1 | 0.3 |
| 2 | 0.09 |
| 3 | 0.08 |

# HMM Model

- What's different about inference with this model?
  - Alignment links are no longer (conditionally) independent!
  - Inference (and EM) now require something more complicated (dynamic programming)

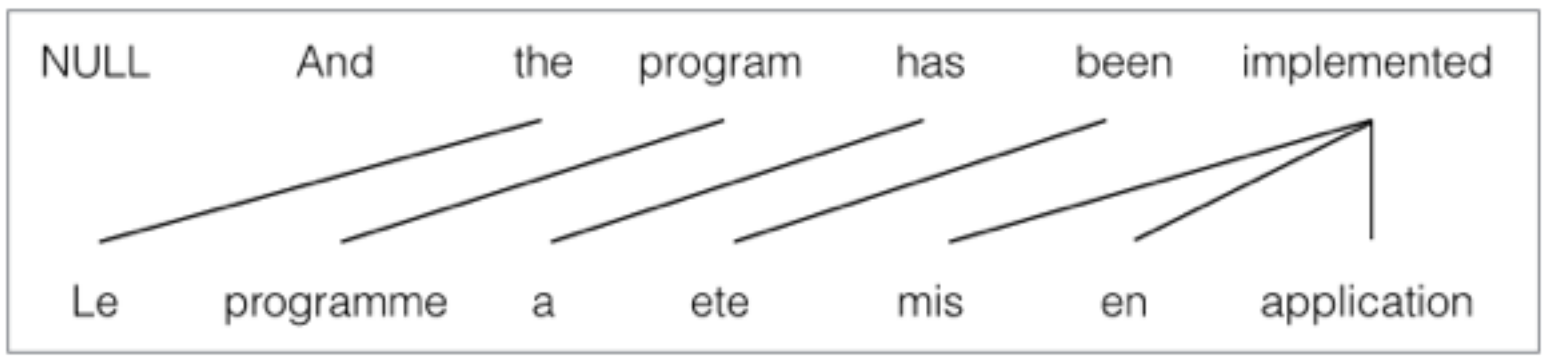
# Pros and cons

- Some drawbacks of word based alignments
  - ~~All reorderings have the same probability~~ MODEL 2
  - Alignments are independent
  - No notion of multiword alignments
  - Alignments are asymmetric
  - No morphology
  - No syntax

# Pros and cons

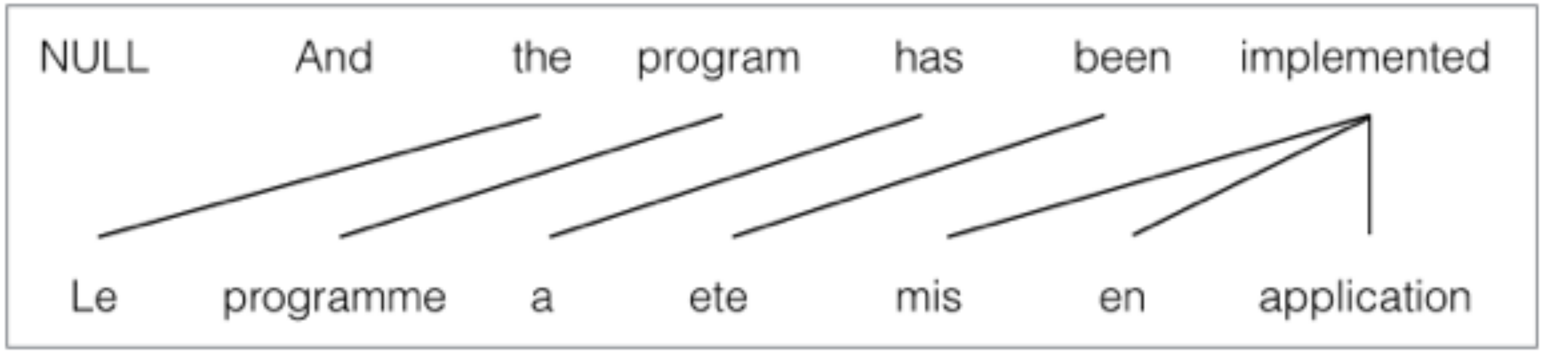
- Some drawbacks of word based alignments
  - ~~All reorderings have the same probability~~ MODEL 2
  - ~~Alignments are independent~~ HMM MODEL
  - No notion of multiword alignments
  - Alignments are asymmetric
  - No morphology
  - No syntax

# Model 3


NULL
And
the
program
has
been
implemented
Le
programme
a
ete
mis
en
application

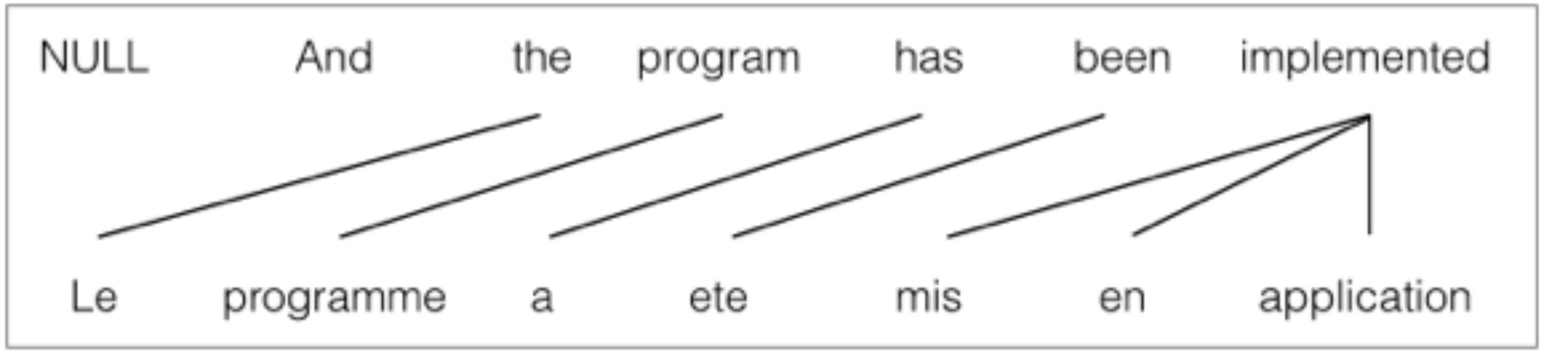
# Model 3

- **Fertility**: some words produce more translations



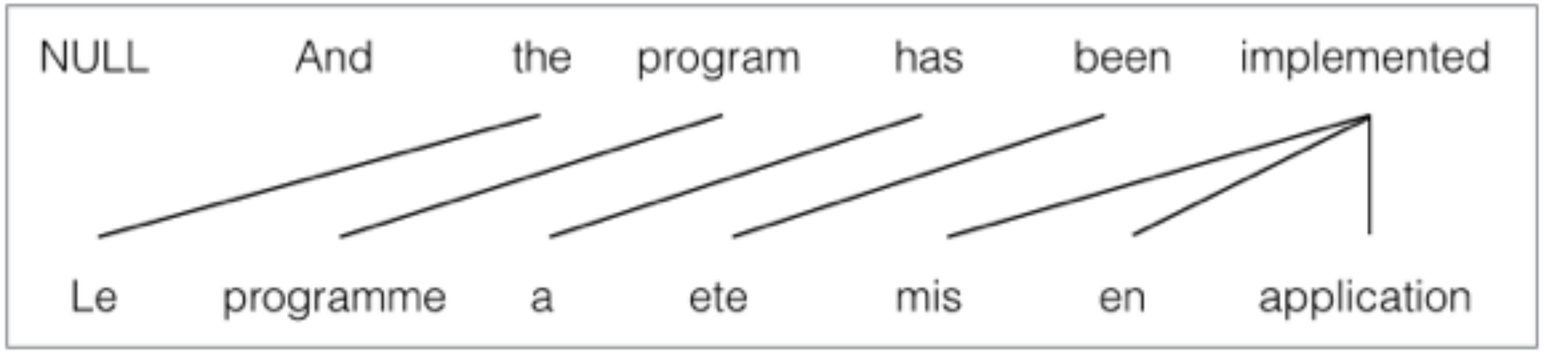
# Model 3

- **Fertility**: some words produce more translations



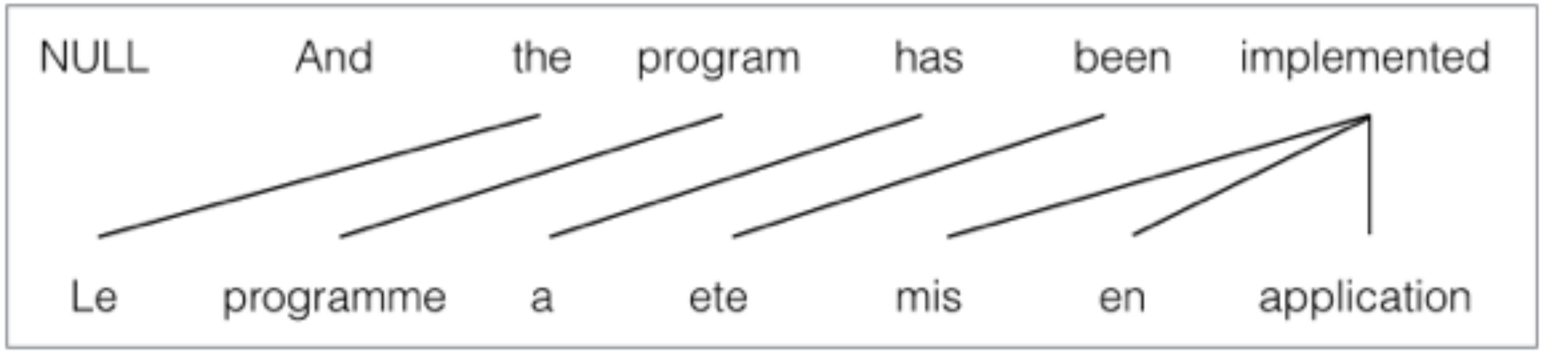
# Model 3

- **Fertility**: some words produce more translations



# Model 3

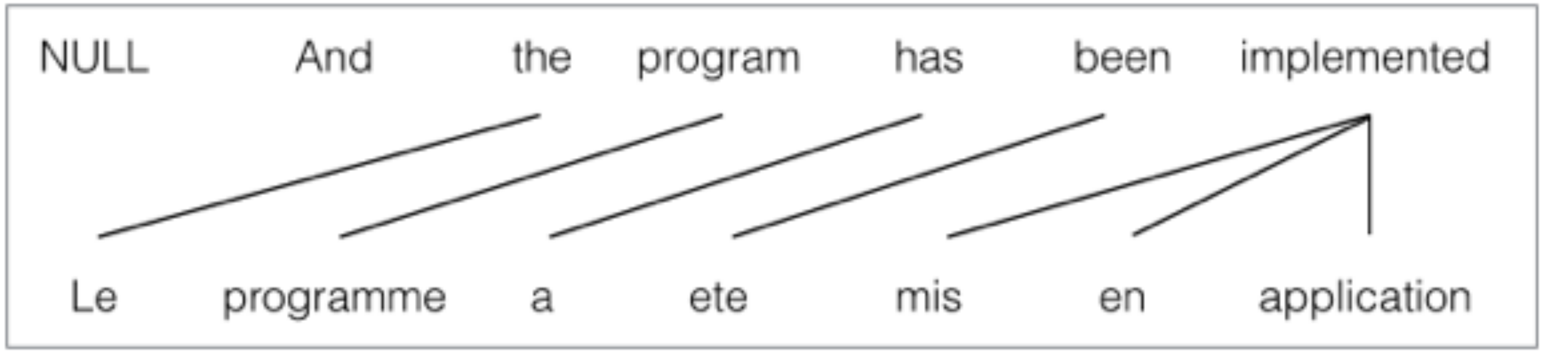
- **Fertility**: some words produce more translations




- *Allowed* in previous models, but not permitted / discouraged

# Model 3

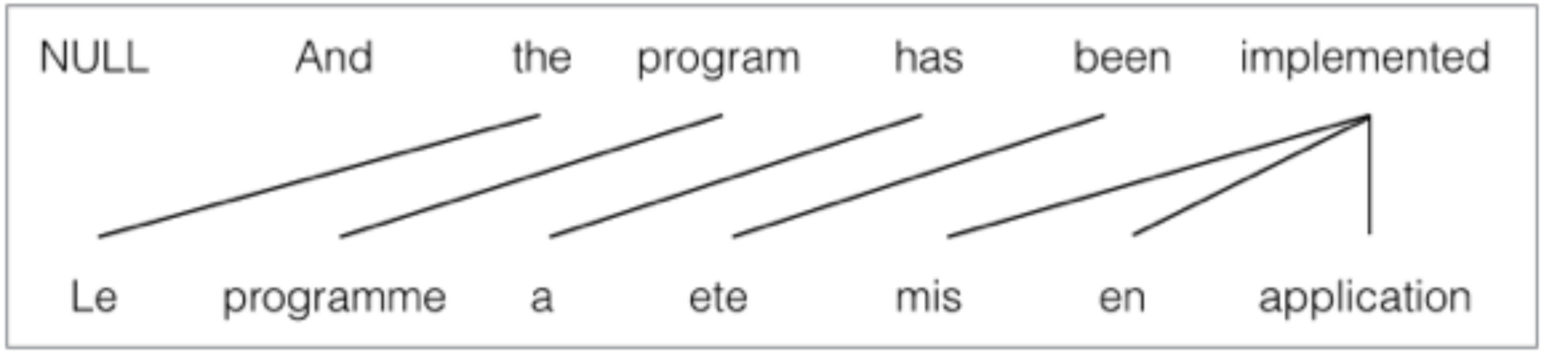
- **Fertility**: some words produce more translations




- *Allowed* in previous models, but not permitted / discouraged
- NULL fertility?

# Model 3

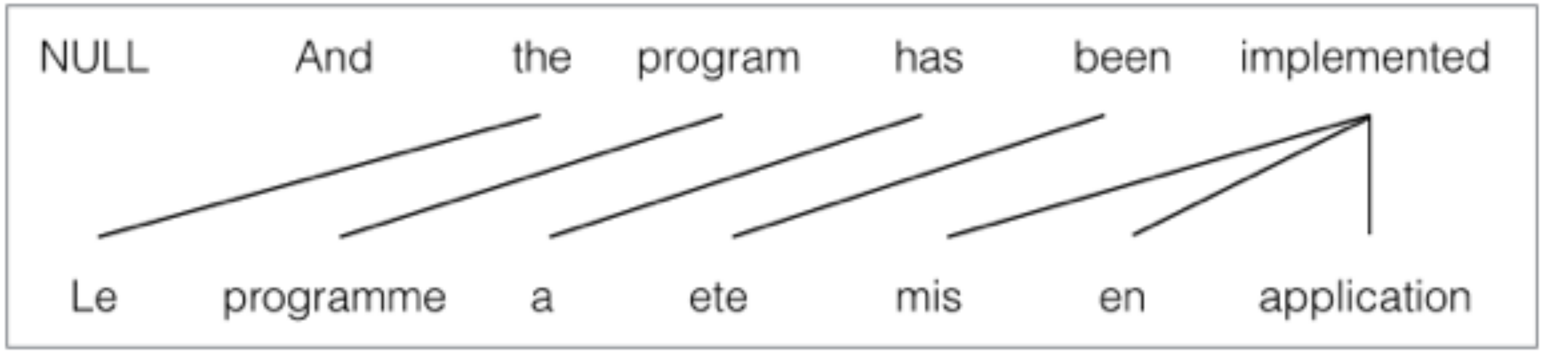
- **Fertility**: some words produce more translations




- *Allowed* in previous models, but not permitted / discouraged
- NULL fertility?
    - No, more related to sentence length

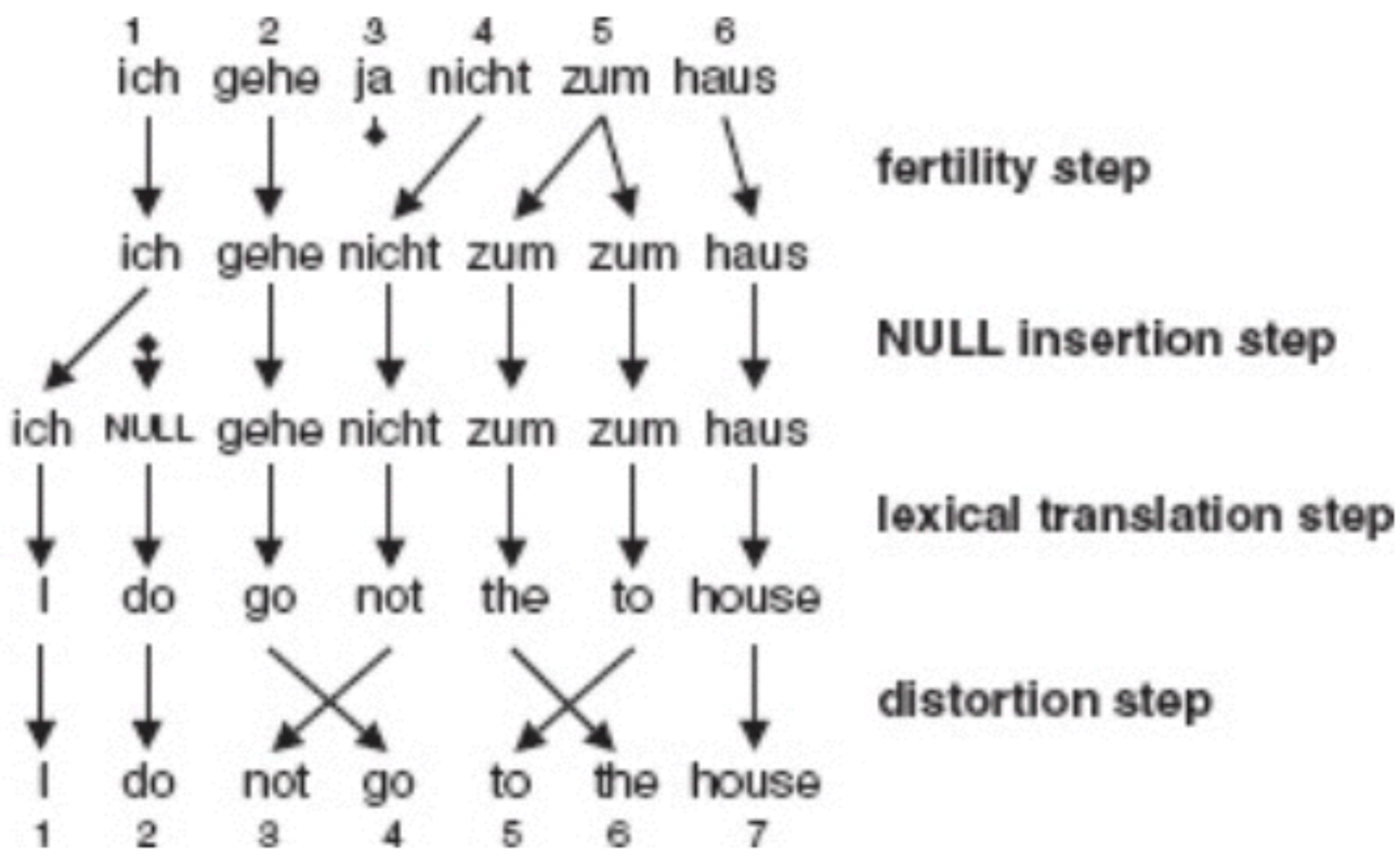
# Model 3

- **Fertility**: some words produce more translations




- *Allowed* in previous models, but not permitted / discouraged
- NULL fertility?
  - No, more related to sentence length
  - Instead, randomly insert NULL after each word with probability ***p***

# Model 3




$n(\phi \mid e)$

$p_1$

$t(f \mid e)$

$d(j \mid i, l, m)$

# Fertility

- The complete alignment p($\boldsymbol{a} \mid \boldsymbol{f}, \boldsymbol{m}$) no longer factorizes to independent alignment decisions
- Now have to resort to sampling
- Basic idea
  - Seed Model 3 parameters with best Model 2 alignment
  - Randomly make small changes, collect Model 3 counts every once in a while

# Pros and cons

- Some drawbacks of word based alignments
  - ~~All reorderings have the same probability~~ MODEL 2
  - Alignments are independent
  - No notion of multiword alignments
  - Alignments are asymmetric
  - No morphology
  - No syntax

# Pros and cons

- Some drawbacks of word based alignments
    - ~~All reorderings have the same probability~~ MODEL 2
    - ~~Alignments are independent~~ HMM MODEL
    - No notion of multiword alignments
    - Alignments are asymmetric
    - No morphology
    - No syntax

# Pros and cons

- Some drawbacks of word based alignments
    - ~~All reorderings have the same probability~~ **MODEL 2**
    - ~~Alignments are independent~~ **HMM MODEL**
    - ~~No notion of multiword alignments~~ **MODEL 3**
    - Alignments are asymmetric
    - No morphology
    - No syntax

# Higher IBM Models

- Increasingly model new phenomena at the cost of model complexity
  - Model 4: cepts and relative distortion
  - Model 5: solves deficiency of Model 4
- Inference is now accomplished with sampling

# Further notes

- Alignments are still asymmetric (why?)

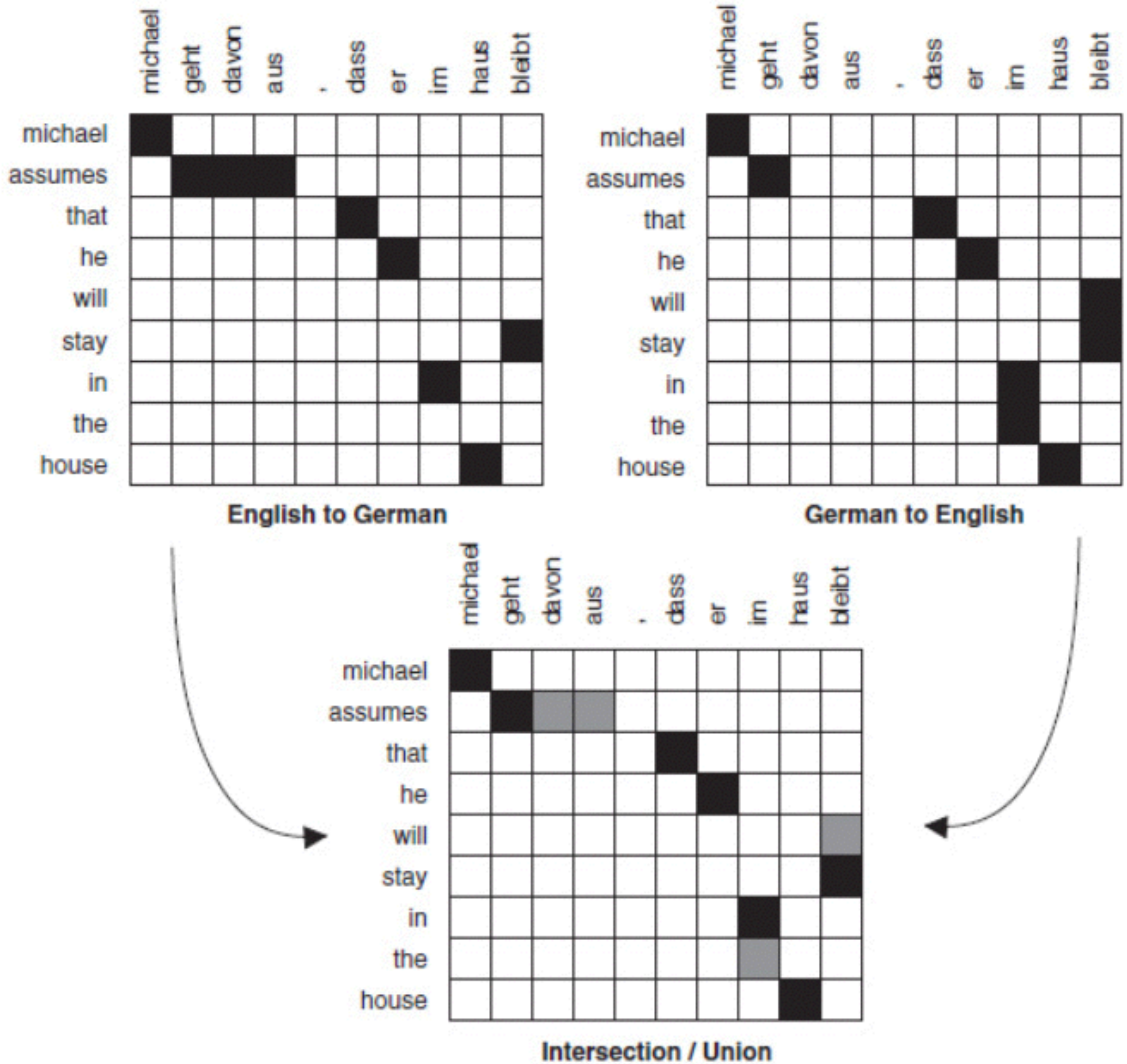
# Further notes

- Alignments are still asymmetric (why?)
  - All models explain each target word *f* with a link to a single source word *e*

# Further notes

- Alignments are still asymmetric (why?)
  - All models explain each target word *f* with a link to a single source word *e*
- Solution: build two models and combine them

p(German | English) + p(English | German)



# Further notes

- Alignments are still asymmetric (why?)
    - All models explain each target word *f* with a link to a single source word *e*
- Solution: build two models and combine them
- Used for **phrase-based translation** (next week)

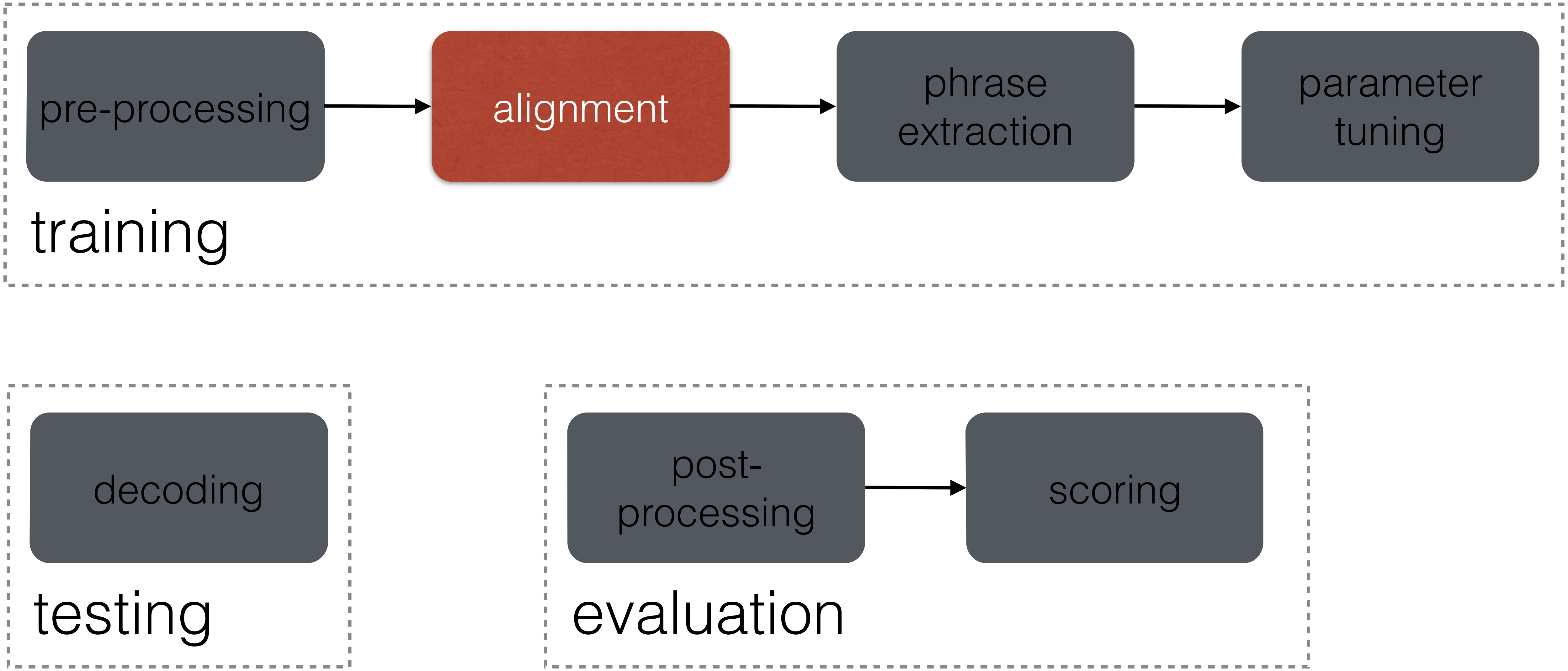
# Summary

- Lexical alignment: IBM Models 1–5
  - Model 1: word-based translation
  - Model 2: +non-uniform alignments
  - Model 3: +fertility
  - Model 4: +cepts and distortion
  - Model 5: –deficiency
- HMM alignment: relative positioning

# Key points

- General tradeoff between complexity of model and ease of inference
- Modeling ideas come from general knowledge and looking at the data
- Keep things concrete with a generative story and being explicit about how parameters are represented
- Simple models are useful for initializing more complicated ones

# Big Picture


pre-processing
alignment
phrase extraction
parameter tuning
training
decoding
testing
post-processing
scoring
evaluation