Machine Translation

CMSC 723 / LING 723 / INST 725

MARINE CARPUAT
marine@cs.umd.edu

Today: an introduction to machine translation

- The **noisy channel model** decomposes machine translation into
  - Word alignment
  - Language modeling

- How can we automatically **align words** within sentence pairs? We'll rely on:
  - **probabilistic modeling**
    - IBM1 and variants [Brown et al. 1990]
  - **unsupervised learning**
    - Expectation Maximization algorithm

MACHINE TRANSLATION AS A NOISY CHANNEL MODEL

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, put these words in order: **{ jjat, arrat, mat, bat, oloat, at-yurp }**

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturian was actually Spanish/English…

Translate: Clients do not sell pharmaceuticals in Europe.

1a. Garcia and associates . 1b. Garcia y asociados .	7a. the clients and the associates are enemies . 7b. los clients y los asociados son enemigos .
2a. Carlos Garcia has three associates . 2b. Carlos Garcia tiene tres asociados .	8a. the company has three groups . 8b. la empresa tiene tres grupos .
3a. his associates are not strong . 3b. sus asociados no son fuertes .	9a. its groups are in Europe . 9b. sus grupos estan en Europa .
4a. Garcia has a company also . 4b. Garcia tambien tiene una empresa .	10a. the modern groups sell strong pharmaceuticals 10b. los grupos modernos venden medicinas fuertes
5a. its clients are angry . 5b. sus clientes estan enfadados .	11a. the groups do not sell zenzanine . 11b. los grupos no venden zanzanina .
6a. the associates are also angry . 6b. los asociados tambien estan enfadados .	12a. the small groups are not modern . 12b. los grupos pequenos no son modernos .

Rosetta Stone

Egyptian hieroglyphs

Demotic

Greek

Warren Weaver (1947)

When I look at an article in Russian, I say to myself: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.

Weaver's intuition formalized as a Noisy Channel Model

- Translating a French sentence **f** is finding the English sentence **e** that maximizes **P(e|f)**
- The noisy channel model breaks down **P(e|f)** into two components

$$\hat{E} = \underset{E \in \text{English}}{\operatorname{argmax}} \overbrace{P(F|E)}^{\text{translation model}} \overbrace{P(E)}^{\text{language model}}$$

Translation Model & Word Alignments

- How can we define the translation model p(f|e) between a French sentence f and an English sentence e?

- Problem: there are many possible sentences!

- Solution: break sentences into words
  - model mappings between word position to represent translation
  - Just like in the Centauri/Arcturian example

PROBABILISTIC MODELS OF WORD ALIGNMENT

Defining a probabilistic model for word alignment

Probability lets us

1) Formulate a model of pairs of sentences
2) Learn an instance of the model from data
3) Use it to infer alignments of new inputs

Recall language modeling

Probability lets us

1) Formulate a model of a sentence

   e.g, bi-grams

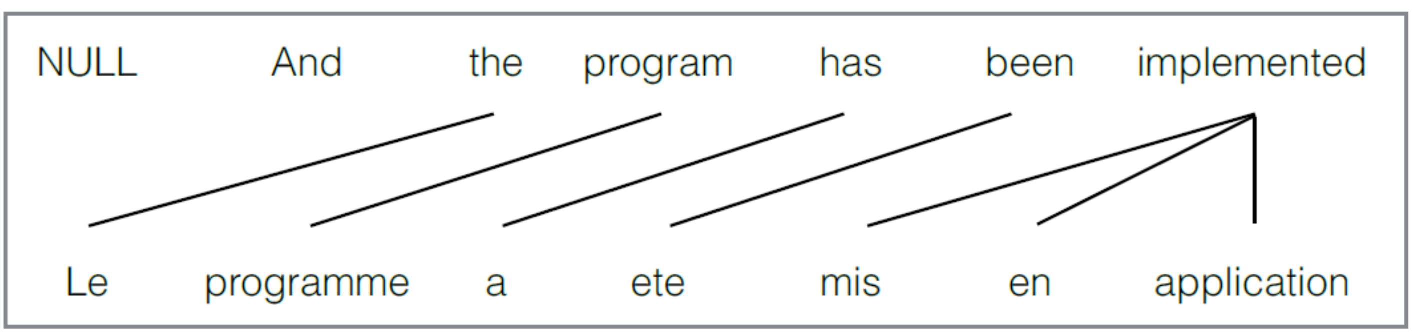
2) Learn an instance of the model from data

$$\hat{p}_{\mathrm{MLE}}(\texttt{call} \mid \texttt{friends}) = \frac{\mathrm{count}(\texttt{friends call})}{\mathrm{count}(\texttt{friends})}$$

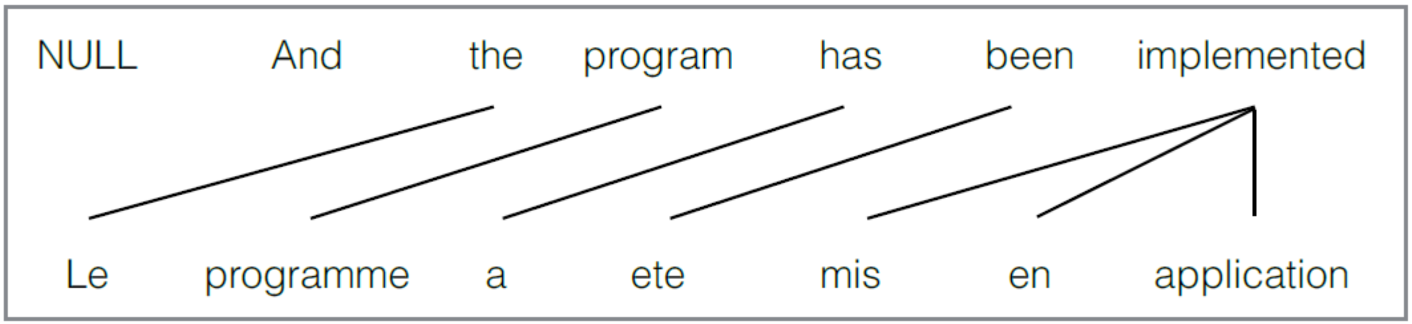
3) Use it to score new sentences

How can we model p(f|e)?

- We'll describe the word alignment models introduced in early 90s at IBM
- Assumption: each French word f is aligned to exactly one English word e
  - Including NULL

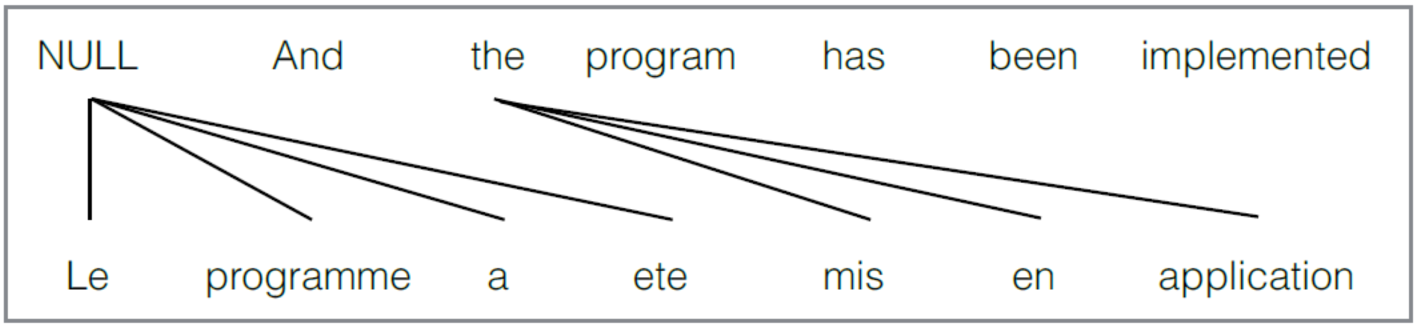


Word Alignment
Vector Representation



- Alignment vector a = [2,3,4,5,6,6,6]
  - length of a = length of sentence f
  - $a_i = j$ if French position i is aligned to English position j

Word Alignment
Vector Representation



- Alignment vector a = [0,0,0,0,2,2,2]

How many possible alignments?

- How many possible alignments for (f,e) where
  - f is French sentence with m words
  - e is an English sentence with l words

- For each of m French words, we choose an alignment link among (l+1) English words

- Answer: $(l+1)^m$

Formalizing the connection between word alignments & the translation model

$$p(f_1, f_2, \ldots, f_m \mid e_1, e_2, \ldots, e_l, m)$$
$$= \sum_{a \in A} p(f_1, \ldots, f_m, a_1, \ldots, a_m \mid e_1, \ldots, e_l, m)$$

- We define a conditional model
  - Projecting word translations
  - Through alignment links

IBM Model 1: generative story

- Input
  - an English sentence of length l
  - a length m

- For each French position i in 1..m
  - Pick an English source index j $q(j \mid i, l, m) = \frac{1}{l+1}$
  - Choose a translation $t(f_i \mid e_{a_i})$

IBM Model 1: generative story

- Input
  - an English sentence of length l
  - a length m

Alignment probabilities are UNIFORM

- For each French position i in 1..m
  - Pick an English source index j $q(j \mid i, l, m) = \dfrac{1}{l+1}$
  - Choose a translation $t(f_i \mid e_{a_i})$

Words are translated independently

IBM Model 1: Parameters

- t(f|e)
  - Word translation probability table
  - for all words in French & English vocab

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

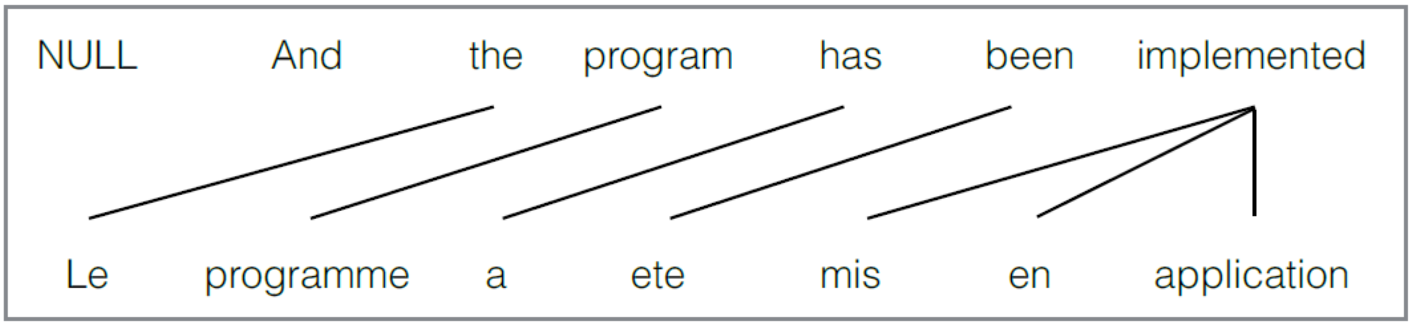
IBM Model 1: generative story

- Input
  - an English sentence of length l
  - a length m

- For each French position i in 1..m
  - Pick an English source index j $q(j \mid i, l, m) = \frac{1}{l+1}$
  - Choose a translation $t(f_i \mid e_{a_i})$

$$p(f_1 \dots f_m, a_1 \dots a_m | e_1 \dots e_l, m) = \prod_{i=1}^{m} q(a_i | i, l, m) t(f_i | e_{a_i})$$

IBM Model 1: Example



- Alignment vector a = [2,3,4,5,6,6,6]
- P(f,a|e)?

Improving on IBM Model 1: IBM Model 2

- Input
  - an English sentence of length l
  - a length m

Remove assumption that q is uniform

- For each French position i in 1..m
  - Pick an English source index j $q(j \mid i, l, m)$
  - Choose a translation $t(f_i \mid e_{a_i})$

IBM Model 2: Parameters

- q(j|i,l,m)
  - now a table
  - not uniform as in IBM1

- How many parameters are there?

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

Defining a probabilistic model for word alignment

Probability lets us

1) Formulate a model of pairs of sentences

   => IBM models 1 & 2

2) Learn an instance of the model from data
3) Use it to infer alignments of new inputs

2 Remaining Tasks

Inference

- Given
  - a sentence pair (e,f)
  - an alignment model with parameters $t(e|f)$ and $q(j|i,l,m)$
- What is the most probable alignment a?

Parameter Estimation

- Given
  - training data (lots of sentence pairs)
  - a model definition
- how do we learn the parameters $t(e|f)$ and $q(j|i,l,m)$?

Inference

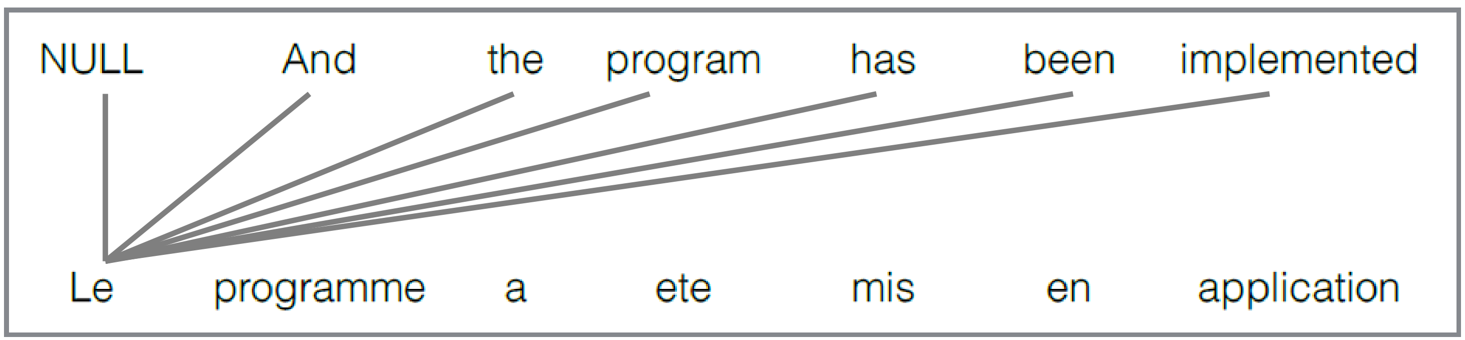
- Inputs
  - Model parameter tables for t and q
  - A sentence pair

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

- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

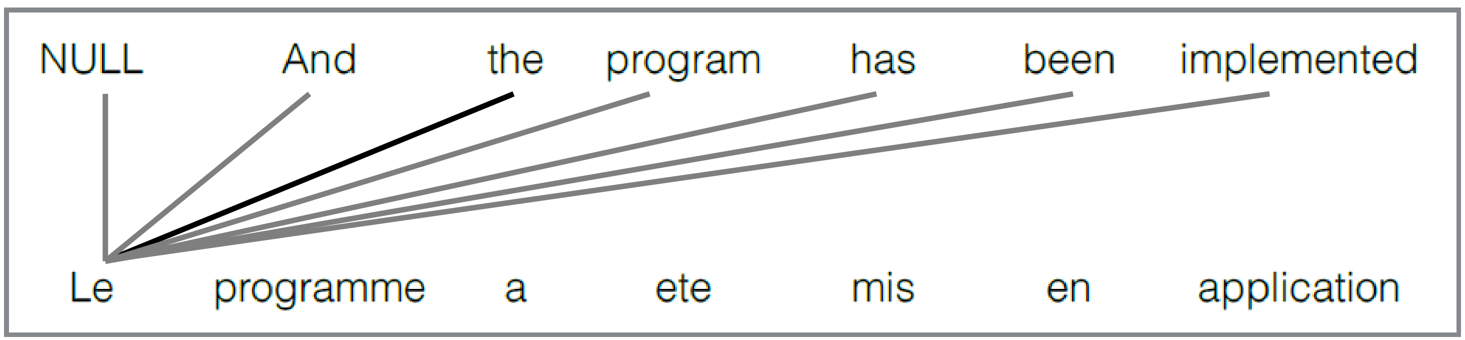
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

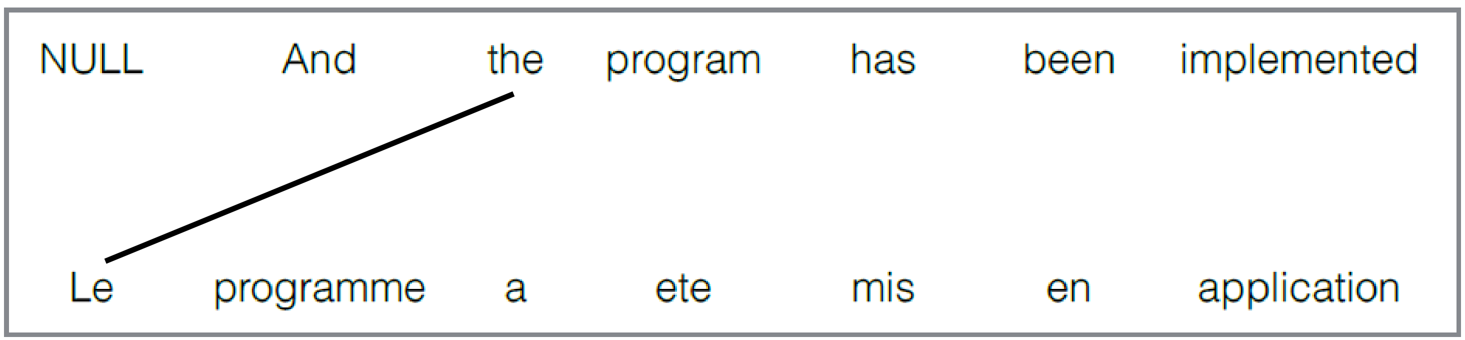
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

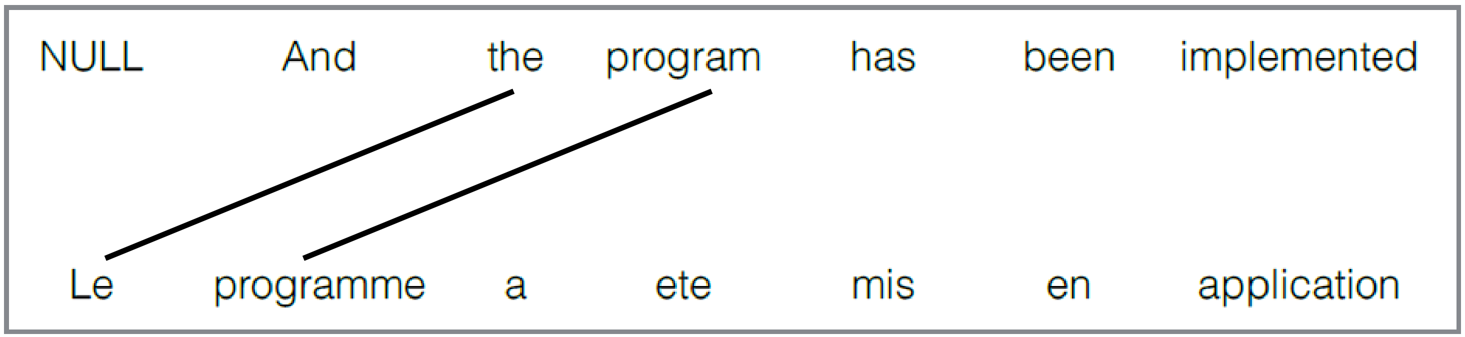
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

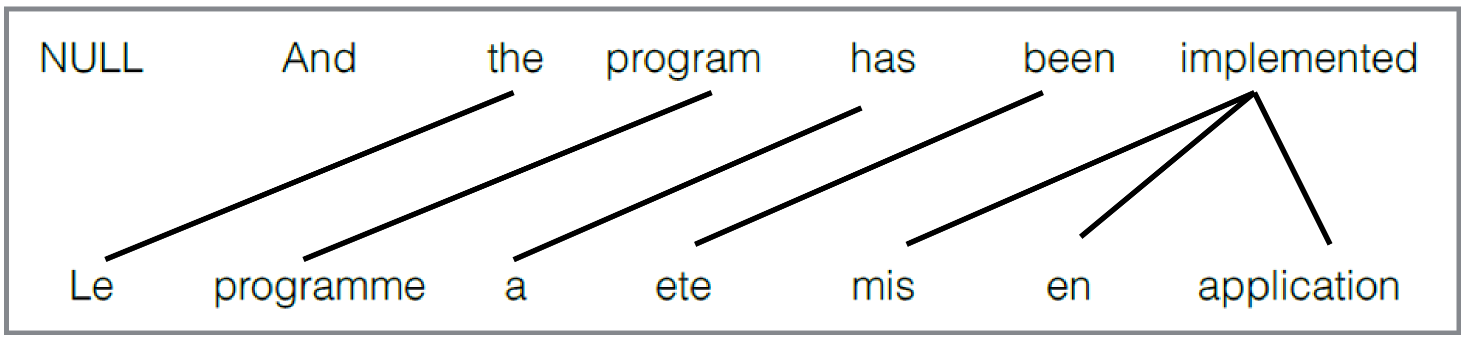
- Inputs
  - Model parameter tables for t and q
  - A sentence pair



- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Inference

- Inputs
  - Model parameter tables for t and q
  - A sentence pair

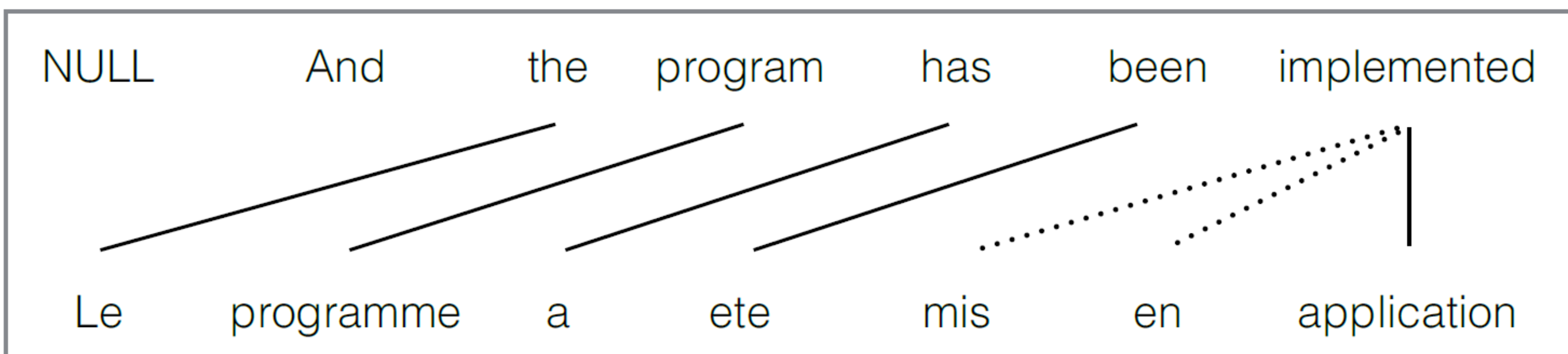


- How do we find the alignment a that maximizes P(e,a|f)?
  - Hint: recall independence assumptions!

Alignment Error Rates: How good is the prediction?

- Given: predicted alignments A, sure links S, and possible links P
- Precision: $\frac{|A \cap P|}{|A|}$ Recall: $\frac{|A \cap S|}{|S|}$
- AER(A|S,P) $= 1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$

Reference alignments, with **Possible** links and **Sure links**



1 Remaining Task

Inference

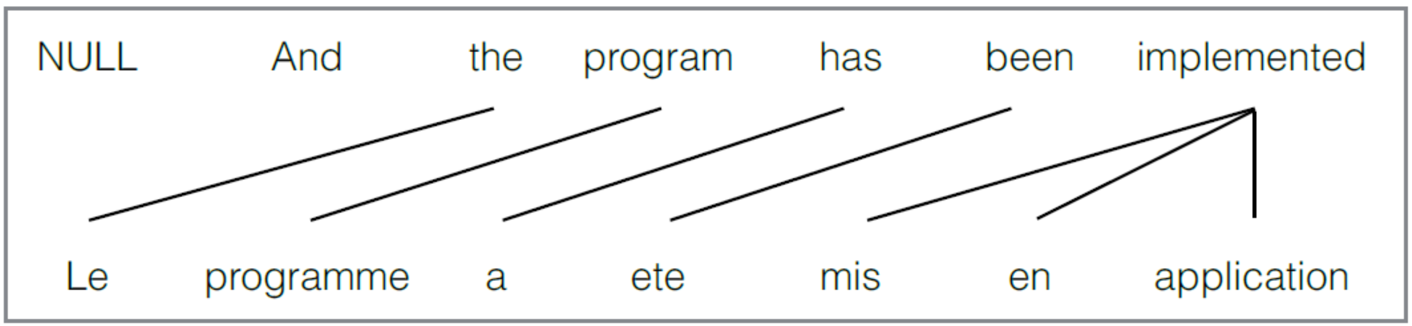
- Given a sentence pair (e,f), what is the most probable alignment a?

Parameter Estimation

- How do we learn the parameters $t(e|f)$ and $q(j|i,l,m)$ from data?

Parameter Estimation (warm-up)

- Inputs
  - Model definition (t and q)
  - A corpus of sentence pairs, with word alignment



- How do we build tables for t and q?
  - Use counts, just like for n-gram models!

Parameter Estimation (for real)

- Problem
  - Parallel corpus gives us **(e,f)** pairs only, **a is hidden**
- We know how to
  - estimate **t** and **q**, given **(e,a,f)**
  - compute **p(e,a|f)**, given t and q
- Solution: Expectation-Maximization algorithm (EM)
  - E-step: given hidden variable, estimate parameters
  - M-step: given parameters, update hidden variable

Parameter Estimation: hard EM

Algorithm 1 (hard EM)

```
initialize parameters t and q to something
repeat until convergence
    for every sentence
        for every target position j
            for every source position i
                if aligned(i, j)
                    count(f_j | e_i) += 1
                    count(e_i) += 1
                    count(j, i, l, m) += 1
                    count(i, l, m) += 1
    t(f | e) = count(f, e) / count(e)
    q(j | i, l, m) = count(j, i, l, m) / count(i, l, m)
```

Parameter Estimation: soft EM

Algorithm 1 (soft EM)

initialize parameters *t* and *q* to something
repeat until convergence
    for every sentence
        for every target position *j*
            for every source position *i*
                $\text{count}(f_j, e_i) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(e_i) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(j, i, l, m) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
                $\text{count}(i, l, m) \mathrel{+}= P(a_i = j \mid e_i, f_j)$
    $t(f \mid e) = \text{count}(f, e) / \text{count}(e)$
    $q(j \mid i, l, m) = \text{count}(j, i, l, m) / \text{count}(i, l, m)$

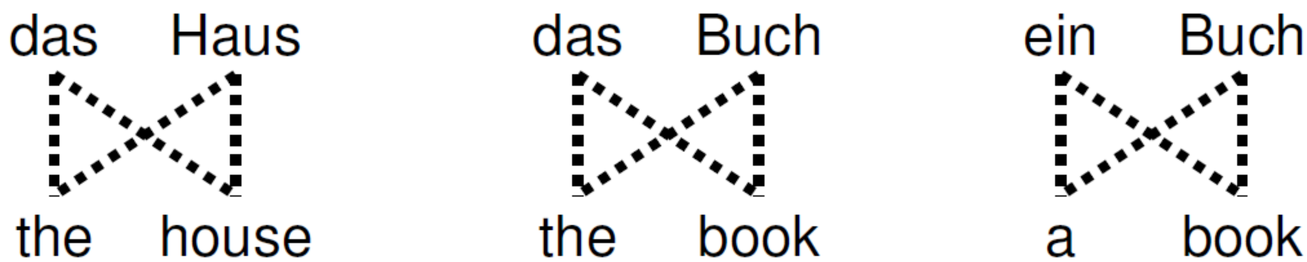
Use "Soft" values instead of binary counts

Parameter Estimation: soft EM

- Soft EM considers all possible alignment links
- Each alignment link now has a weight

$$P(a_i = j \mid e_i, f_j) = \frac{q(j \mid i, l, m) \cdot t(f_i \mid e_j)}{\sum_{j'=1}^{l} q(j' \mid i, l, m) \cdot t(f_i \mid e_{j'})}$$

Example: learning t table using EM for IBM1



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

We have now fully specified our probabilistic alignment model!

Probability lets us

1) Formulate a model of pairs of sentences

   => IBM models 1 & 2

2) Learn an instance of the model from data

   => using EM

3) Use it to infer alignments of new inputs

   => based on independent translation decisions

SUMMARY: INTRODUCTION TO MACHINE TRANSLATION

Summary: Noisy Channel Model for Machine Translation

- The **noisy channel model** decomposes machine translation into two independent subproblems
  - Word alignment
  - Language modeling

$$\hat{E} = \underset{E \in \text{English}}{\operatorname{argmax}} \overbrace{P(F|E)}^{\text{translation model}} \overbrace{P(E)}^{\text{language model}}$$

Summary: Word Alignment with IBM Models 1, 2

- Probabilistic models with **strong independence assumptions**
  - Results in linguistically naïve models
    - asymmetric, 1-to-many alignments
  - But allows efficient parameter estimation and inference

- Alignments are hidden variables
  - unlike words which are observed
  - require **unsupervised learning** (EM algorithm)