Lecture Overview

- Machine Translation is one of the big tasks in NLP. Tons of intellectual and technical progress has been driven by the desire to build better MT systems.
- We will focus on “the old way” of doing things today, i.e., Statistical Machine Translation (SMT). Later in the semester, we will talk about neural MT.
- A few key ideas from SMT that are worth knowing:
  - Noisy channel, IBM Model 1, EM algorithm. These are some of the ideas that pushed NLP from rule-based into ML-based models. They are also the kinds of ideas that will come up for other problems (not just NLP).
  - Language models. These are necessary for SMT, and so SMT led to early work on language modeling. Today, language models are all anyone can talk about. ;) We’ll be covering LMs for the next many lectures. Today we will just cover decoding algorithms.

MT is hard!

- Morphology
  - tuntussuqatarniksaitengqiggtuq (Yupik language, from Alaska)
  - tuntu -ssur -qatar -ni -ksaite -ngqiggte -uq
  - reindeer -hunt -FUTURE -say -NEG -again -3SG.IND
  - "He had not yet said again that he was going to hunt reindeer."

https://en.wikipedia.org/wiki/Polysynthetic_language

MT is hard!

- Syntax
  - SVO: German, English, French, Mandarin
  - SOV: Hindi, Japanese
  - VSO: Irish, Arabic, Biblical Hebrew
- Argument structure and marking
  - Possession:
    - The man’s house
    - Az ember haza: the man house-his (Hungarian)
  - Motion, manner
    - The bottle floated out
    - La botella salió flotando: The bottle exited floating (Spanish)

Noisy Channel Model for MT

$$P(tgt \mid src) = \frac{P(src \mid tgt)P(tgt)}{P(src)}$$

Noisy Channel Model for MT

$$P(tgt \mid src) \propto P(src \mid tgt) P(tgt)$$

Noisy Channel Model for MT

$$P(tgt \mid src) \propto P(src \mid tgt) P(tgt)$$

Translation Model

Noisy Channel Model for MT

$$P(tgt \mid src) \propto P(src \mid tgt) P(tgt)$$

Translation Model

Intuition: source language is a "corrupted" version of the target language

Noisy Channel Model for MT

$$P(tgt \mid src) \propto P(src \mid tgt) P(tgt)$$

Language Model

Noisy Channel Model for MT

$$P(tgt|src) \propto P(src|tgt)P(tgt)$$

Language Model

(We will talk more about this in the coming lectures)

Full Phrase-Based MT System

$$\text{argmax}_{tgt \in TGT} P(src \mid tgt) P(tgt)$$

Full Phrase-Based MT System

$$\text{argmax}_{tgt \in TGT} P(src \mid tgt) P(tgt)$$

1.Translation Model

Full Phrase-Based MT System

$$\text{argmax}_{tgt \in TGT} P(src \mid tgt) P(tgt)$$

1.Translation Model
2.Language Model

Full Phrase-Based MT System

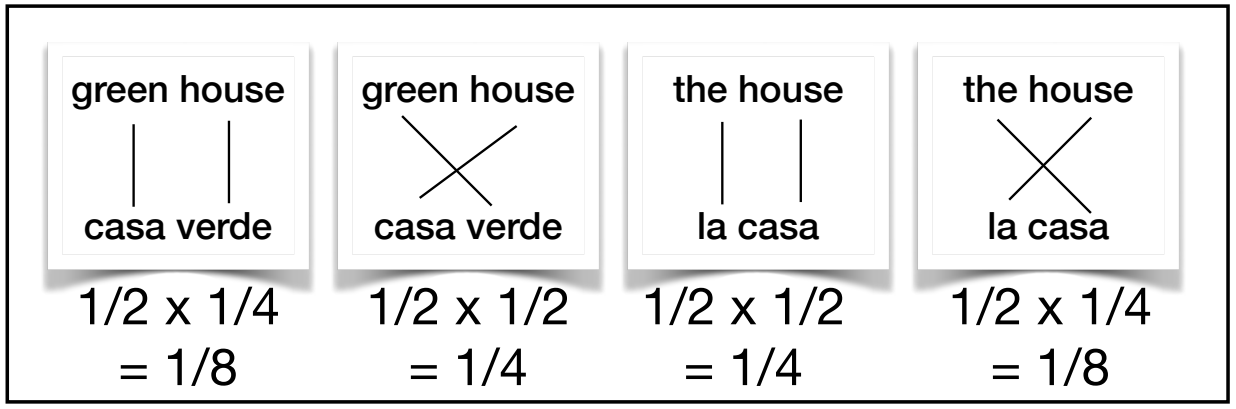
$$\text{argmax}_{tgt \in TGT} P(src \mid tgt) P(tgt)$$

1. Translation Model
2. Language Model
3. Decoder

This Lecture: Alignment (Produces a Translation Model)

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1/2	1/2
	la	0	1/4	1/2
	verde	1/2	1/4	0

Lectures 6–8: Language Models

<s> can you tell me about any good cantonese restaurants close by </s>
<s> mid priced thai food is what i'm looking for </s>
<s> tell me about chez panisse </s>
<s> can you give me a listing of the kinds of food that are available </s>
<s> i'm looking for a good place to eat breakfast </s>
<s> when is caffe venezia open during the day </s>

$$P(w_0 \ldots w_n) \approx P(w_0 | \text{<s>}) \times P(w_1 | w_0) \times P(w_2 | w_1) \times \ldots \times P(w_n | w_{n-1})$$

P(tell me about caffe venezia) = P(tell|<s>) x P(me|tell) x P(about|me) x
P(caffe|about) x P(venezia|caffe) x P(</s>|venezia)

Full Phrase-Based MT System

$$\text{argmax}_{tgt \in TGT} P(src \mid tgt) P(tgt)$$

1. Translation Model
2. Language Model
3. Decoder

Word Alignment

Goal: Build a phrase table

t		target		
		green	house	the
source	casa	1/2	1/2	1/2
	la	0	1/4	1/2
	verde	1/2	1/4	0

Word Alignment

Bitexts a.k.a. Bilingual Parallel Corpora

- Bitext: A corpus that contains translated pairs of texts
  - Can be aligned coarsely (e.g., document-level) or finely (sentence level)
  - Word-level alignment is rare

Word Alignment

Bitexts a.k.a. Bilingual Parallel Corpora

- Some exist naturally. E.g., EU translates all their documents (https://www.statmt.org/europarl/), authors hire translators to translate literature

Parallel Corpus (L1-L2)	Sentences	L1 Words	English Words
Bulgarian-English	406,934	-	9,886,291
Czech-English	646,605	12,999,455	15,625,264
Danish-English	1,968,800	44,654,417	48,574,988
German-English	1,920,209	44,548,491	47,818,827
Greek-English	1,235,976	-	31,929,703
Spanish-English	1,965,734	51,575,748	49,093,806
Estonian-English	651,746	11,214,221	15,685,733
Finnish-English	1,924,942	32,266,343	47,460,063
French-English	2,007,723	51,388,643	50,196,035
Hungarian-English	624,934	12,420,276	15,096,358
Italian-English	1,909,115	47,402,927	49,666,692
Lithuanian-English	635,146	11,294,690	15,341,983
Latvian-English	637,599	11,928,716	15,411,980
Dutch-English	1,997,775	50,602,994	49,469,373
Polish-English	632,565	12,815,544	15,268,824
Portuguese-English	1,960,407	49,147,826	49,216,896
Romanian-English	399,375	9,628,010	9,710,331

Word

Bitexts

- Some
  all their documents
  www.statmt.org/eu
  translators to transl
- Can get creative if we are okay with "weakly aligned"—e.g., pairs of new articles or Wikipedia documents on the same topic

Natural language processing

From Wikipedia, the free encyclopedia

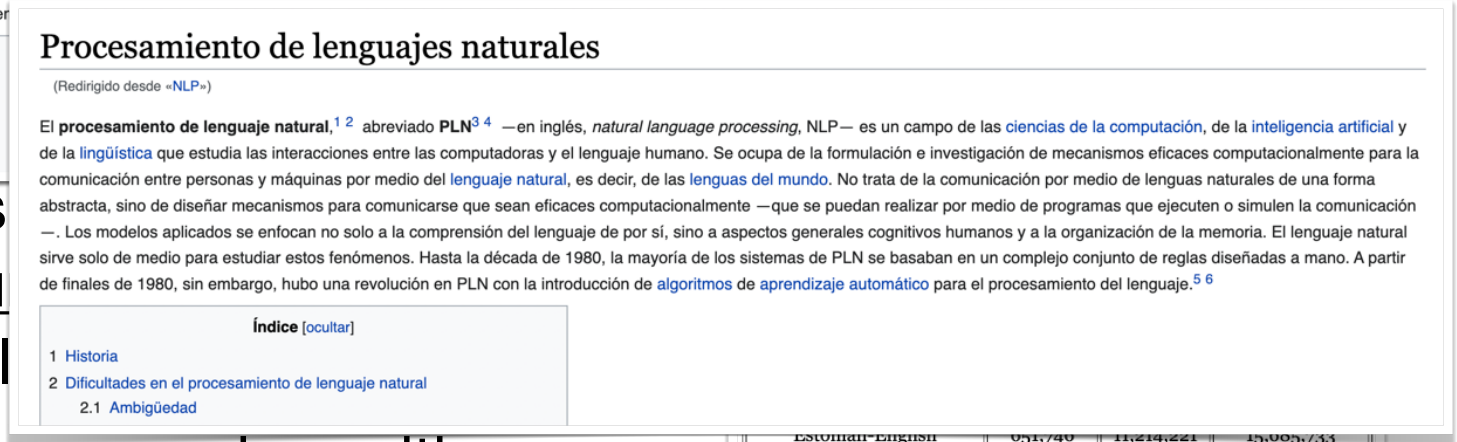
This article is about natural language processing done by computers. For the natural language processing done by the human brain, see Language processing in the brain.

Natural language processing (**NLP**) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequer

Contents [hide]

1 History
- 1.1 Symbolic NLP (1950s – early 1990s)
- 1.2 Statistical NLP (1990s–2010s)
- 1.3 Neural NLP (present)



Procesamiento de lenguajes naturales

(Redirigido desde «NLP»)

El **procesamiento de lenguaje natural**,[1] [2] abreviado **PLN**[3] [4] —en inglés, *natural language processing*, NLP— es un campo de las ciencias de la computación, de la inteligencia artificial y de la lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. Se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio del lenguaje natural, es decir, de las lenguas del mundo. No trata de la comunicación por medio de lenguas naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente —que se puedan realizar por medio de programas que ejecuten o simulen la comunicación —. Los modelos aplicados se enfocan no solo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos humanos y a la organización de la memoria. El lenguaje natural sirve solo de medio para estudiar estos fenómenos. Hasta la década de 1980, la mayoría de los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en PLN con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.[5] [6]

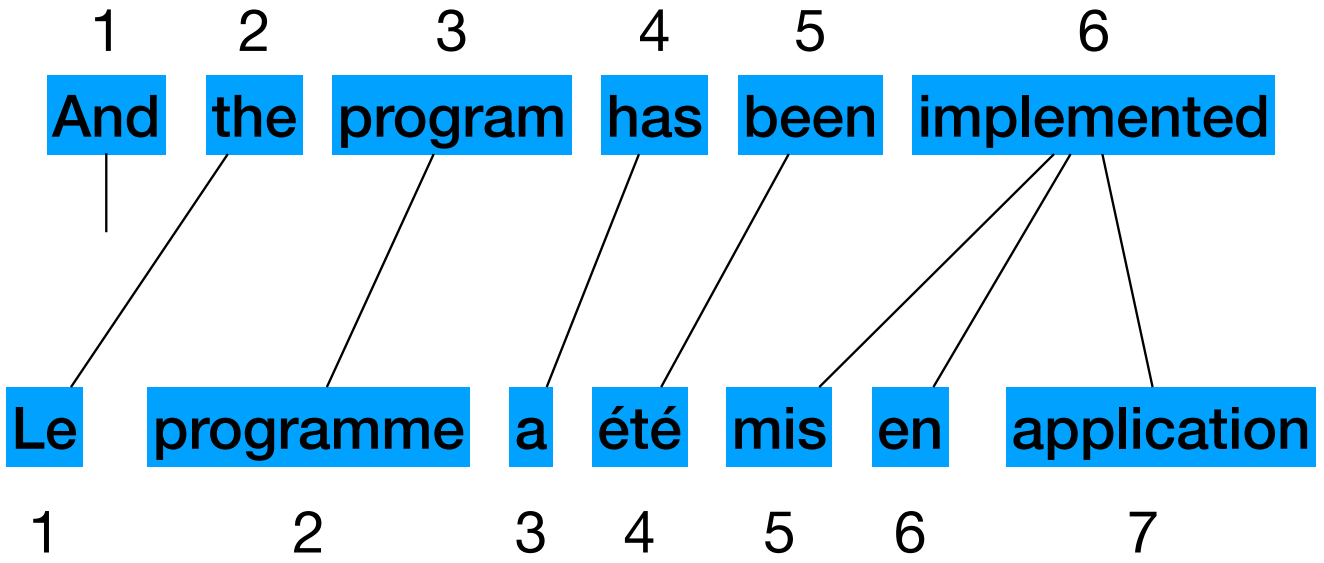
Índice [ocultar]

1 Historia
2 Dificultades en el procesamiento de lenguaje natural
- 2.1 Ambigüedad

Estonian-English	651,746	11,214,221	15,085,733
Finnish-English	1,924,942	32,266,343	47,460,063
French-English	2,007,723	51,388,643	50,196,035
Hungarian-English	624,934	12,420,276	15,096,358
Italian-English	1,909,115	47,402,927	49,666,692
Lithuanian-English	635,146	11,294,690	15,341,983
Latvian-English	637,599	11,928,716	15,411,980
Dutch-English	1,997,775	50,602,994	49,469,373
Polish-English	632,565	12,815,544	15,268,824
Portuguese-English	1,960,407	49,147,826	49,216,896
Romanian-English	399,375	9,628,010	9,710,331

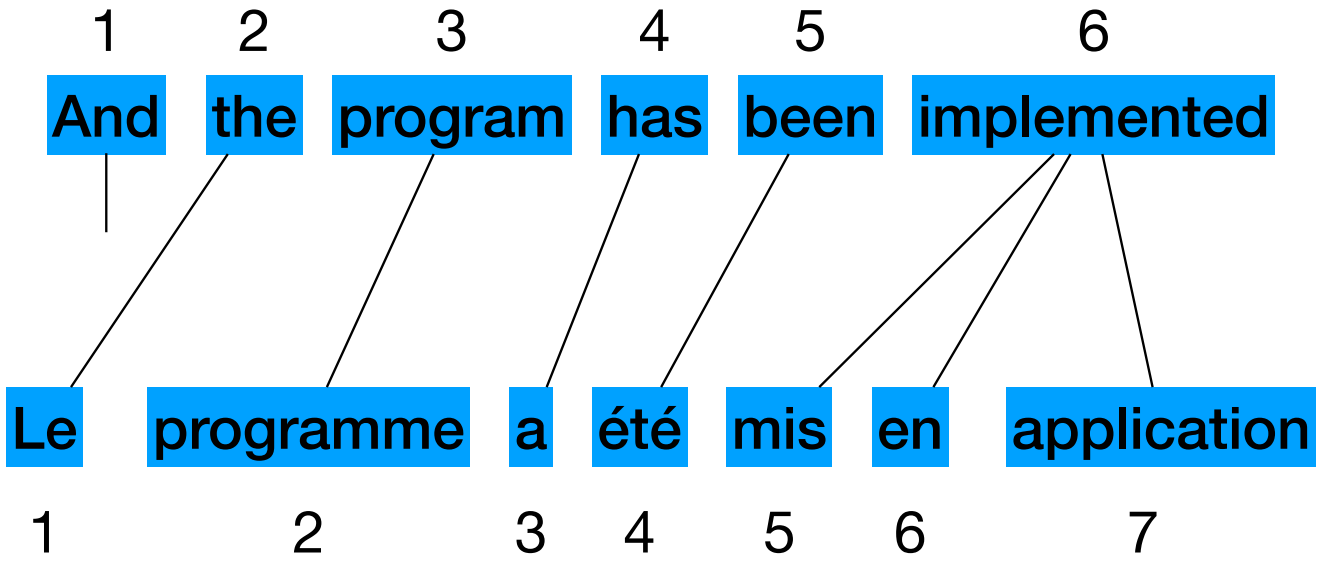
Word Alignment

IBM Model 1



Word Alignment

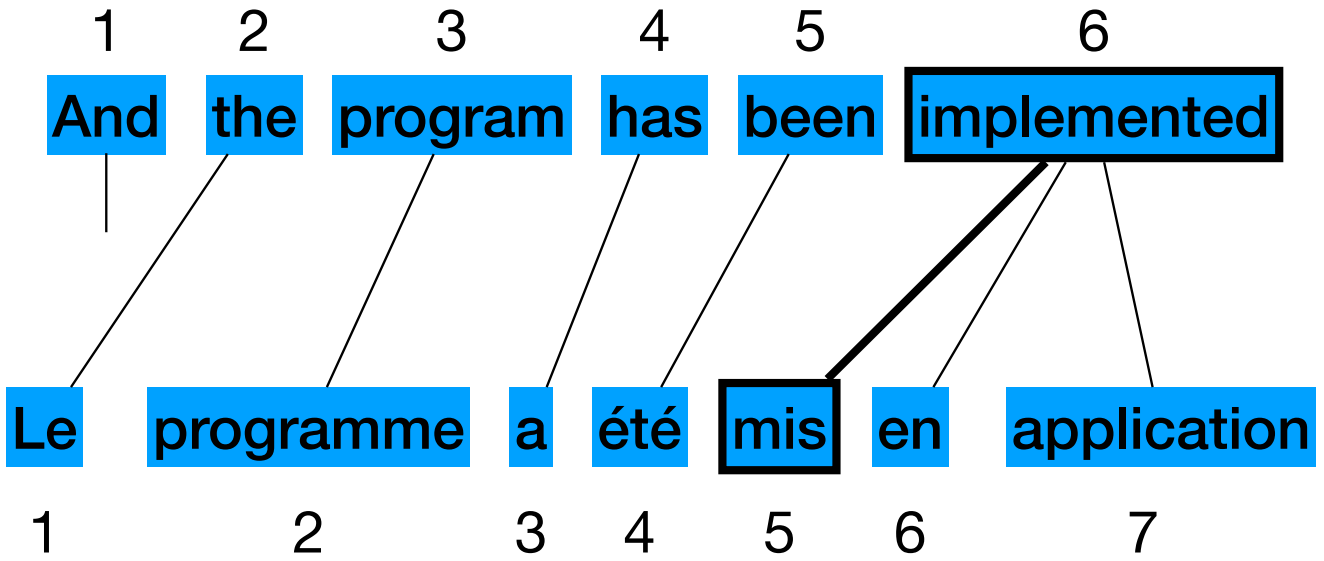
IBM Model 1



a	2	3	4	5	6	6	6

Word Alignment

IBM Model 1

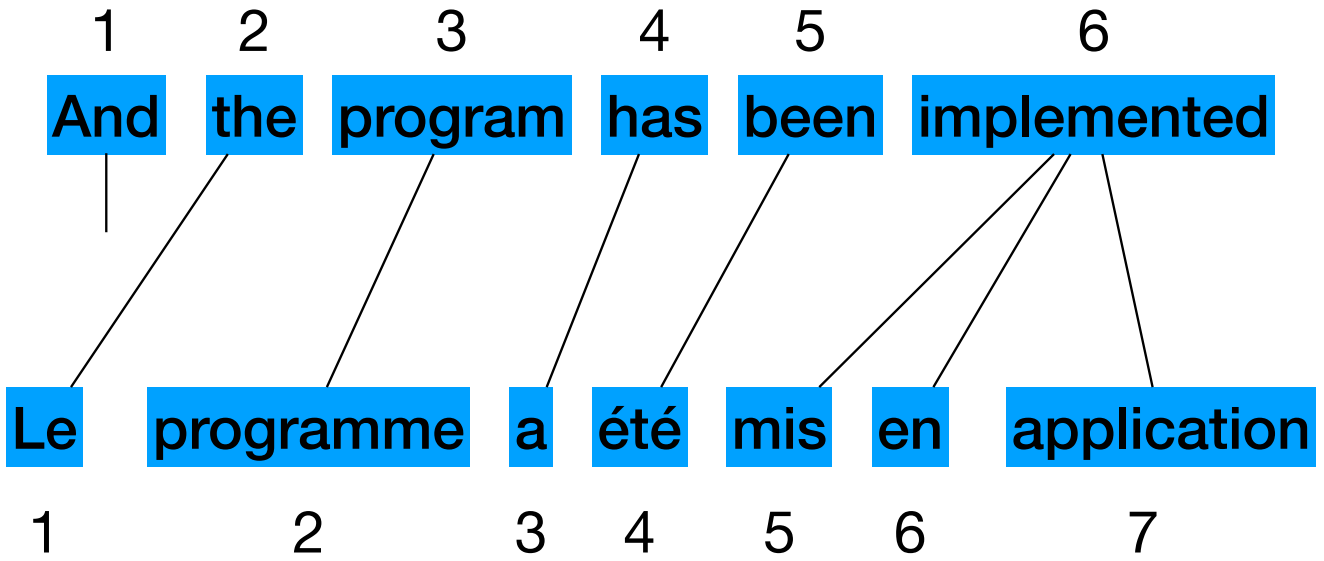


a	2	3	4	5	6	6	6

a[5] = 6

Word Alignment

IBM Model 1

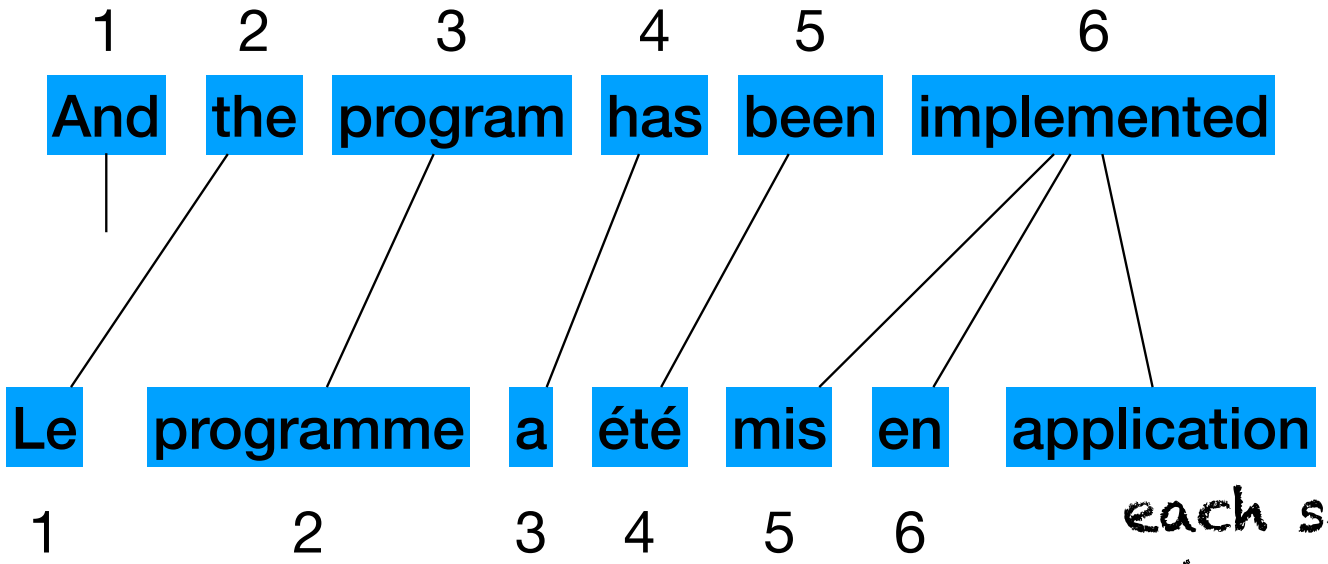


a	2	3	4	5	6	6	6

each source word aligns to exactly one target word

Word Alignment

IBM Model 1



a	2	3	4	5	6	6	6

each source word aligns to exactly one target word

(later models relax this)

Word Alignment

IBM Model 1

- First (simplest) automatic unsupervised alignment model from IBM
- Generative Story:
  - Choose length L for src sentence
  - Choose an alignment A = a1…aL
  - Then, generate target position ti by translating whatever source phrase is aligned to position i in the target
- To build the phrase table: $\text{argmax}_{a_j} (s_j \mid t_{a_j})$

Word Alignment

IBM Model 1

- Dilemma:
  - We want alignments so that we can figure out the probabilities of phrase translations, e.g., P(si|tj) for all s, t, i, j
  - To estimate those alignments, we need phrase translation probabilities
  - 😖
  - No fear! EM is here!

Topics

- Follow ups
  - SVD Revisited
  - Generative Stories: Intuition
- Machine Translation
- Noisy Channel Models for SMT
  - **Translation Model (Word Alignment)**
    - IBM Alignment Models
    - **EM Algorithm**
  - Language Model
    - Decoding

Expectation Maximization (EM) Algorithm

- E step: **E**stimate the liklihood of the observed data given the current parameters
- M step: Recompute the parameters so as to **m**aximize the liklihood of the observed data

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

green house	casa verde
the house	la casa

t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

Corpus

green house	casa verde
the house	la casa

t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

green house	casa verde
the house	la casa

Parameters

t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

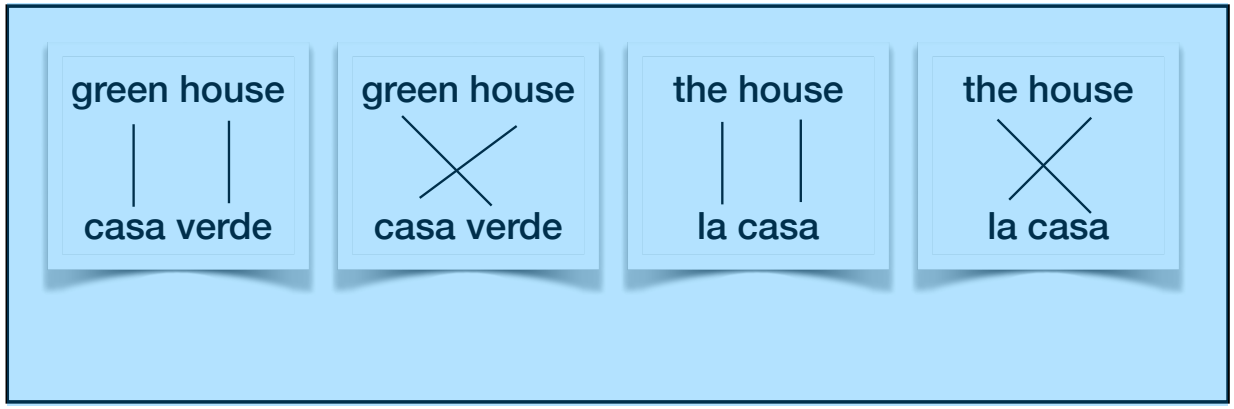
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

green house	casa verde
the house	la casa

All possible alignments

a



t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

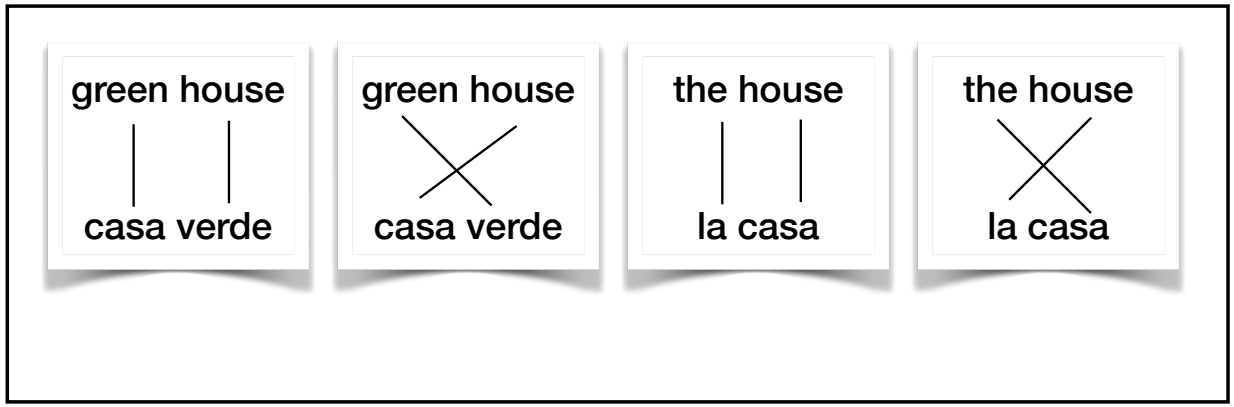
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

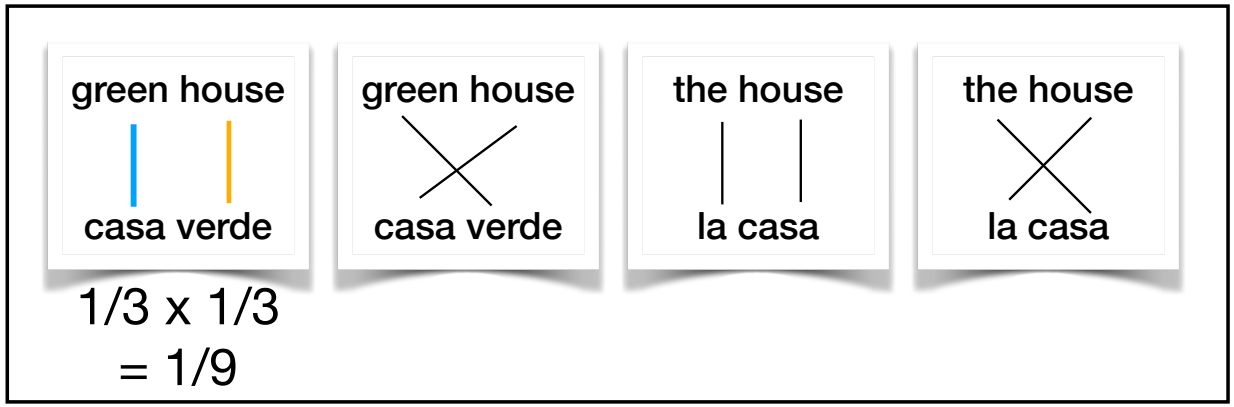
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

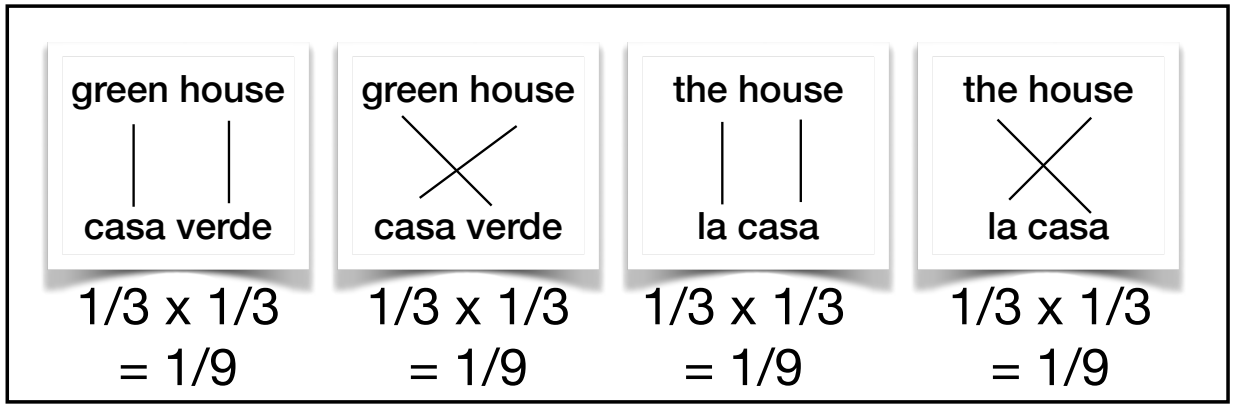
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

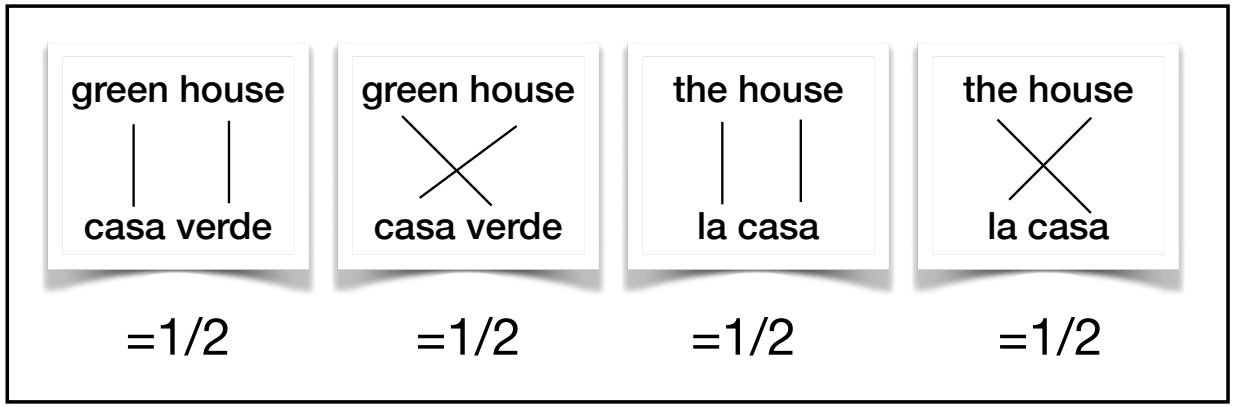
E Step: Compute likelihood of corpus given current parameters

$$P(a \mid s, t) = \frac{P(a, s \mid t)}{\sum_a P(a, s \mid t)}$$

Normalize to reweight liklihood of alignments

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/3	1/3	1/3
	la	1/3	1/3	1/3
	verde	1/3	1/3	1/3

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

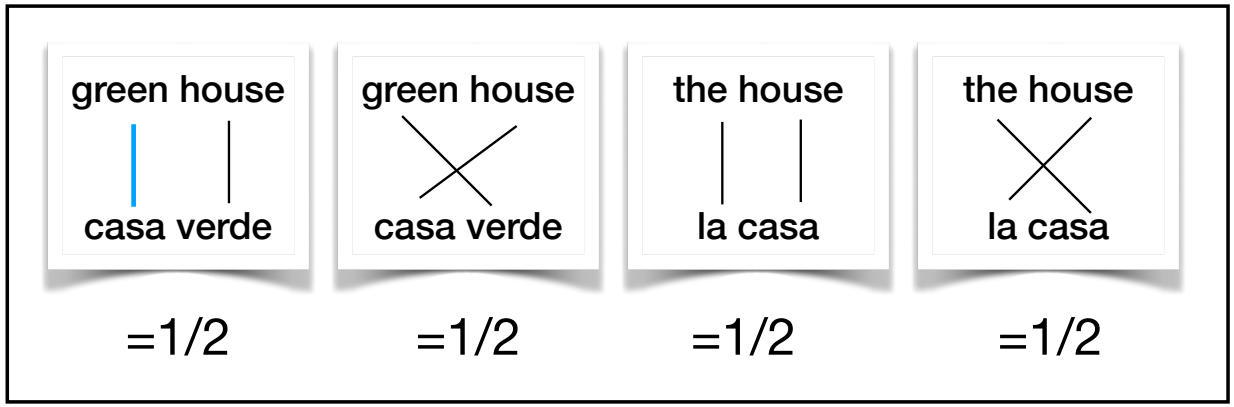
$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

Compute expected counts counts by adding fractional counts equal to P(a|f,e).

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1	1/2
	la	0	1/2	1/2
	verde	1/2	1/2	0

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

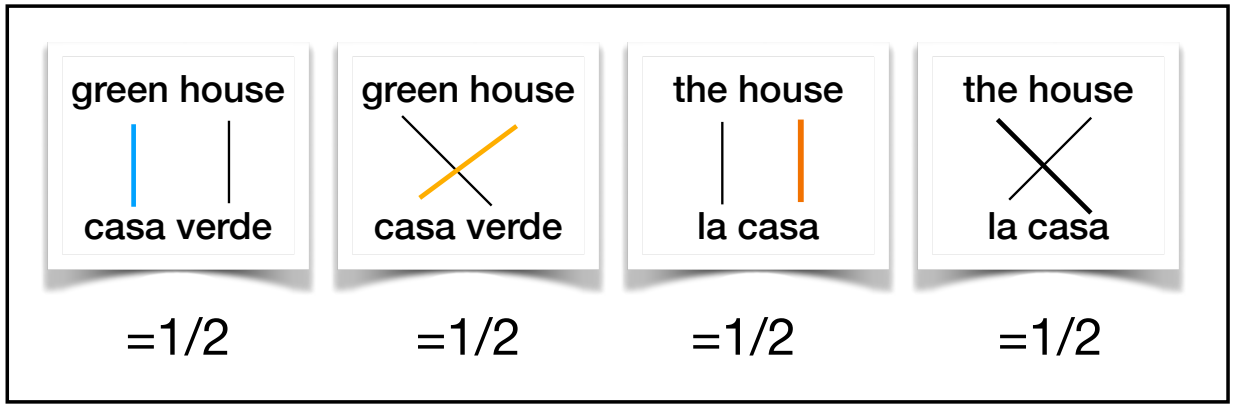
$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

Compute expected counts by adding fractional counts equal to P(a|f,e).

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1	1/2
	la	0	1/2	1/2
	verde	1/2	1/2	0

Expectation Maximization (EM) Algorithm

Goal:

Estimate a and t using

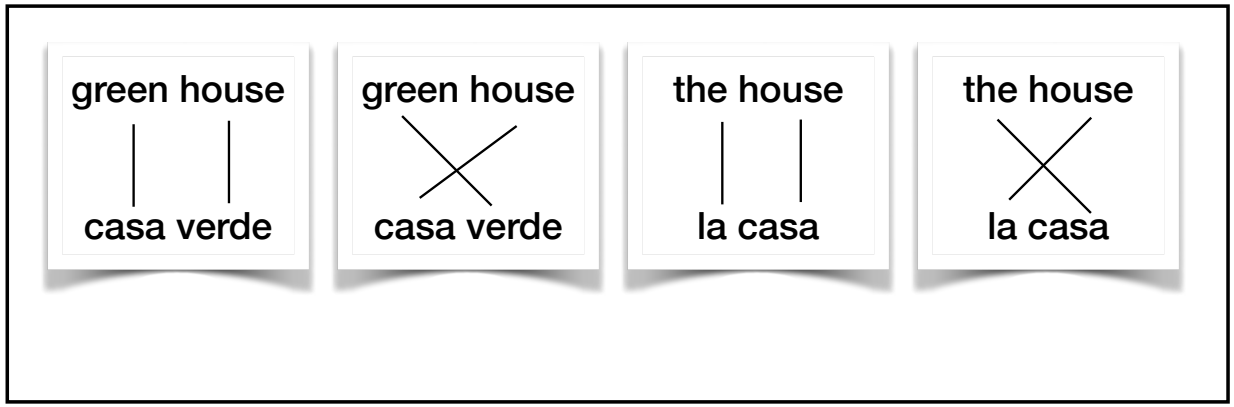
$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

M Step: Compute MLE parameters

(here, that just means normalizing!)

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1/2	1/2
	la	0	1/4	1/2
	verde	1/2	1/4	0

Expectation Maximization (EM) Algorithm

Goal:

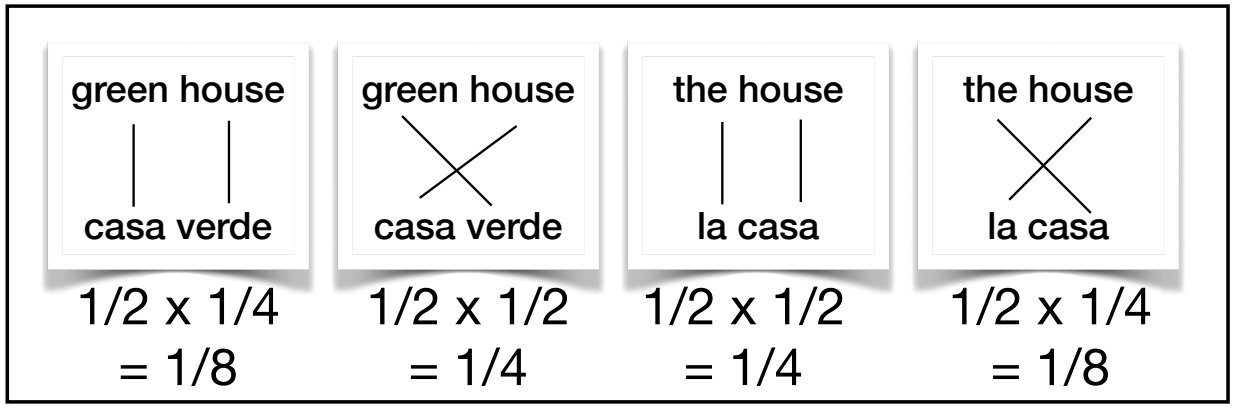
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1/2	1/2
	la	0	1/4	1/2
	verde	1/2	1/4	0

Expectation Maximization (EM) Algorithm

Goal:

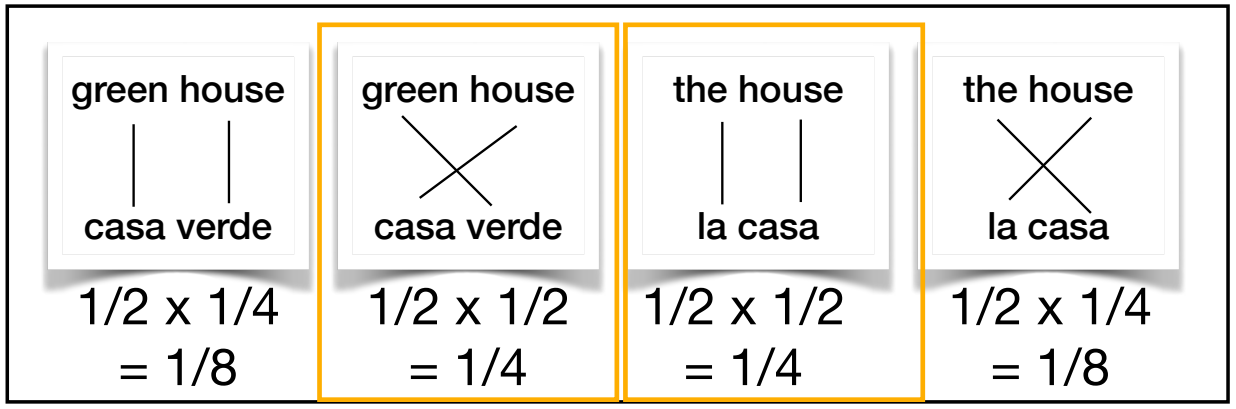
Estimate a and t using

$\text{argmax}_{aj}\ (s_j \mid t_{aj})$

E Step: Compute likelihood of corpus given current parameters

green house	casa verde
the house	la casa

a



t

		target		
		green	house	the
source	casa	1/2	1/2	1/2
	la	0	1/4	1/2
	verde	1/2	1/4	0