Sentence Alignment

Multilingual NLP

Guillaume Wisniewski
guillaume.wisniewski@linguist.univ-paris-diderot.fr

October 2019

Université de Paris & LLF

The Task

Word Alignment

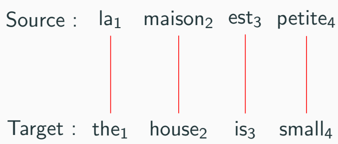
Given a sentence and its translation, find which source word is translated by which target word

Source : la$_1$ maison$_2$ est$_3$ petite$_4$

Target : the$_1$ house$_2$ is$_3$ small$_4$

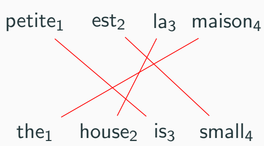
Word Alignment

Given a sentence and its translation, find which source word is translated by which target word

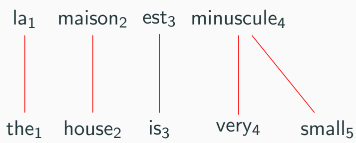


Reordering

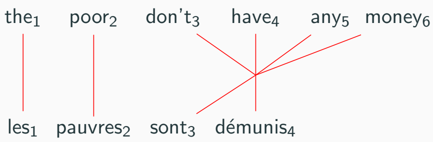
Words may be reordered during translation



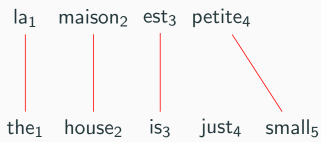
One-to-Many Translation



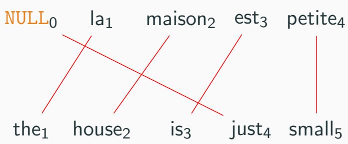
Many-To-Many Translation



Inserting/Deleting Words



Inserting/Deleting Words



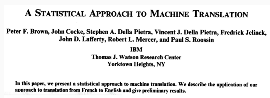
IBM Models

Formalization of the task

Given a source sentence $\mathbf{s} = s_1, s_2, \ldots, s_{|\mathbf{s}|}$ and a target sentence $\mathbf{t} = t_1, t_2, \ldots, t_{|\mathbf{t}|}$, find which target word is translated by which source word.

↪ the choice of the source and the target sentence is arbitrary
↪ we are only interested in the alignment

An historical note


A Statistical Approach to Machine Translation

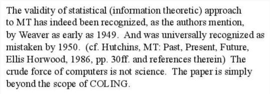
Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin

IBM
Thomas J. Watson Research Center
Yorktown Heights, NY

In this paper, we present a statistical approach to machine translation. We describe the application of our approach to translation from French to English and give preliminary results.

- published in 1988
- IBM models are translation model : originally alignment was just a by-product
- an important landscape in a the history of NLP
  - ↪ proof that MT can be solved with statistical methods
  - ↪ a source of inspiration for many works
  - ↪ a useful tool

An historical note


The validity of statistical (information theoretic) approach to MT has indeed been recognized, as the authors mention, by Weaver as early as 1949. And was universally recognized as mistaken by 1950. (cf. Hutchins, MT: Past, Present, Future, Ellis Horwood, 1986, pp. 30ff. and references therein) The crude force of computers is not science. The paper is simply beyond the scope of COLING.

- published in 1988
- IBM models are translation model : originally alignment was just a by-product
- an important landscape in a the history of NLP
  - ↪ proof that MT can be solved with statistical methods
  - ↪ a source of inspiration for many works
  - ↪ a useful tool

Learning Alignments

What we have...

- parallel corpora : alignment at the sentence level
- but almost no alignment at the word level
  - not always possible
  - tedious

⇒ unsupervised learning

What kind of information can we use ?

What kind of information can we use ?

- co-occurrence of words in parallel sentences
  - ↪ `cat/chat` appear in the same parallel sentences
- position in the sentence
- competitive linking : a source word only 'generates' a single / a few target words
  - ↪ avoid one pitfall of model based solely on co-occurrences : align each target word with the most frequent source word (e.g. punctuation)

The alignment function (1)

Source : das$_1$ Haus$_2$ ist$_3$ klein$_4$

Target : the$_1$ house$_2$ is$_3$ small$_4$

↪ formally, an alignment is a function from $[\![1, |\mathbf{t}|]\!]$ to $[\![1, |\mathbf{s}|]\!]$:

$$a = \{1 \rightarrow 1; 2 \rightarrow 2; 3 \rightarrow 3; 4 \rightarrow 4\}$$

↪ Warning : a is not symmetrical !

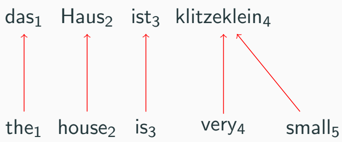
The alignment function (2)

klein$_1$ ist$_2$ das$_3$ Haus$_4$

the$_1$ house$_2$ is$_3$ small$_4$

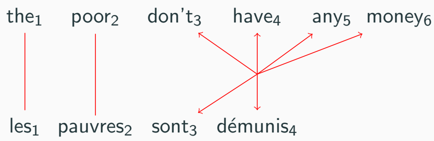
$$a = \{1 \rightarrow 3; 2 \rightarrow 4; 3 \rightarrow 2; 4 \rightarrow 1\}$$

The alignment function (3)



$$a = \{1 \to 1; 2 \to 2; 3 \to 3; 4 \to 4; 5 \to 4\}$$

The alignment function (4)



Can no longer be represented by a function !

An important detail

The alignment task
For every target word find the source word (including `NULL`) that has 'generated' it

An important detail

The alignment task
For every target word find the source word (including `NULL`) that has 'generated' it

↪ the model is inherently asymmetric... you cannot invert source and target

↪ ... but alignments are symmetric !

↪ symetrize alignment in a second step

↪ two step process ⇒ modeling easier

- model alignment with a function from 'target position' to 'source position'

↪ ensure that each target word is aligned to a single source word

↪ but the reverse is not true a source word can be aligned to several target word

↪ asymmetry in the predicted alignment

↪ model 'competitive linking'

IBM 1 Model

Assumptions

- model the probability of generating the target sentence and the alignment knowing the source sentence
- using only the probability to translate a given source word by a target word : $\theta(t|s)$
- assuming that each target word is generated independently

Model

$$p(\mathbf{t}, a|\mathbf{s}) \propto \prod_{j=1}^{|\mathbf{s}|} \theta(s_j|s_{a(j)})$$

Example

das		Haus		ist		small	
t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$	t	$\theta(t\|s)$
the	0.7	house	0.8	is	0.8	small	0.4
that	0.15	building	0.16	's	0.16	little	0.4
which	0.075	home	0.02	exists	0.02	short	0.1
who	0.05	household	0.015	has	0.015	minor	0.06
this	0.025	shell	0.005	are	0.005	petty	0.04

↪ Probability of aligning `das Haus ist klein` and `the house is small` monotonically :

$$p(\mathbf{t}, a|\mathbf{s}) \propto \theta\left(\text{the}|\text{das}\right) \times \theta\left(\text{house}|\text{Haus}\right) \times \theta\left(\text{is}|\text{ist}\right) \times \theta\left(\text{small}|\text{klein}\right)$$
$$\propto 0.7 \times 0.8 \times 0.8 \times 0.4$$
$$\propto 0.179$$

Translation Probability

- $\theta(t|s)$:
  - ↪ probability of translating s by t
  - ↪ for each source word s : dictionary mapping target words to probability
- follows the basic rules of probabilities :
  - ↪ $\forall s, t \quad \theta(t|s) \in [0, 1]$
  - ↪ $\forall s \quad \sum_t \theta(t|s) = 1$

Learning Translation Probability

1. if we had the alignment i.e. supervised learning
   - ↪ easy to estimate the translation probabilities

Learning Translation Probability

1. if we had the alignment i.e. supervised learning
   - ↪ easy to estimate the translation probabilities
2. if we had the translation probability
   - ↪ easy to find the alignment

Learning Translation Probability

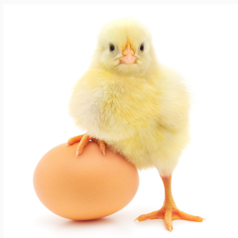
1. if we had the alignment i.e. supervised learning
   - ↪ easy to estimate the translation probabilities
2. if we had the translation probability
   - ↪ easy to find the alignment

but we have neither of them !

↪ incomplete data

↪ latent variables

Learning Translation Probability



1. if we had the alignment i.e. supervised learning
   - ↪ easy to estimate the translation probabilities
2. if we had the translation probability
   - ↪ easy to find the alignment

but we have neither of them !

↪ incomplete data

↪ latent variables

⇒ EM algorithm

EM algorithm : intuition

Given a corpus :

··· la maison ··· la maison bleue ··· la fleur ···

··· the house ··· the blue house ··· the flower ···

EM algorithm : intuition

Given a corpus :

··· la maison ··· la maison bleue ··· la fleur ···

··· the house ··· the blue house ··· the flower ···

1. assume all alignments are equally likely

EM algorithm : intuition

Given a corpus :

··· la maison ··· la maison bleue ··· la fleur ···

··· the house ··· the blue house ··· the flower ···

1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly

EM algorithm : intuition

Given a corpus :

··· la maison ··· la maison bleue ··· la fleur ···

··· the house ··· the blue house ··· the flower ···

1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly
   - ↪ model learns that `la` is often aligned with `the`
   - ↪ the alignment between `la` and `the` is reinforced
   - ↪ the alignments between `la` and other words become weaker

EM algorithm : intuition

Given a corpus :

··· la maison ··· la maison bleue ··· la fleur ···

··· the house ··· the blue house ··· the flower ···

1. assume all alignments are equally likely
2. estimate the translation probabilities and predict alignments accordingly
3. iterate steps 1 & 2
   - ↪ it becomes apparent that the alignment between `fleur` and `flower` are more likely (pigeon hole principle)

Summary

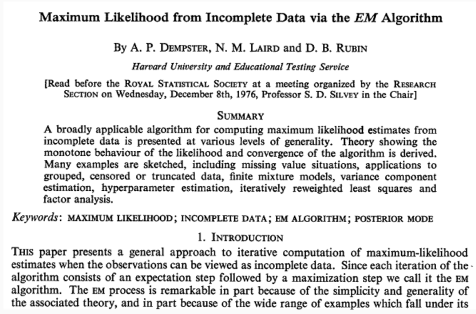
We can estimate θ with an iterative two steps procedure (EM algorithm) :

1. apply the model to the data (E-step)
   - ↪ using the current value of the parameters estimate the value of the hidden variables (here alignment)
2. estimate model from data (M-step)
   - ↪ take assign values as fact
   - ↪ collect counts (weighted by probability)
   - ↪ estimate model from count

These two steps are repeated until convergence.

The EM algorithm

An historical note


Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: maximum likelihood; incomplete data; em algorithm; posterior mode

1. Introduction

This paper presents a general approach to iterative computation of maximum-likelihood estimates when the observations can be viewed as incomplete data. Since each iteration of the algorithm consists of an expectation step followed by a maximization step we call it the em algorithm. The em process is remarkable in part because of the simplicity and generality of the associated theory, and in part because of the wide range of examples which fall under its

A classical paper (1977)
one of the most cited paper in the world...

Contexte

- un des critères d'apprentissage : maximum de vraisemblance

$$\begin{aligned}\theta^* &= \arg\max_\theta \mathcal{L}(\theta)\\ &= \arg\max_\theta \sum_{\mathbf{x}\in\mathcal{D}} \log p(\mathbf{x}|\theta)\\ &= \arg\max_\theta p(X|\theta)\end{aligned}$$

- lorsque l'on a des variables cachées :

$$p(\mathbf{x}|\theta) = \sum_{\mathbf{z}\in\mathcal{Z}} p(\mathbf{x},\mathbf{z}|\theta)$$

- optimisation directe « difficile » ⇒ recours à des méthodes d'optimisation numérique itérative :

$$\theta_{n+1} \leftarrow f(\theta_n)$$

La magie du log

On cherche θ, tel que : $\mathcal{L}(\theta) \geq \mathcal{L}(\theta_n)$ où θ_n est une constante (paramètres actuels). Calculons :

$$\begin{aligned}&\mathcal{L}(\theta) - \mathcal{L}(\theta_n)\\ &= \log\left(\sum_z p(X|z,\theta)\times p(z|\theta)\right) - \log p(X|\theta_n)\\ &= \log\left(\sum_z p(X|z,\theta)\times p(z|\theta)\times\frac{p(z|X,\theta_n)}{p(z|X,\theta_n)}\right) - \log p(X|\theta_n)\\ &= \log\left(\sum_z p(z|X,\theta_n)\times\frac{p(X|z,\theta)\times p(z|\theta)}{p(z|X,\theta_n)}\right) - \log p(X|\theta_n)\\ &\geq \sum_z\left(p(z|X,\theta_n)\times\log\left(\frac{p(X|z,\theta)\times p(z|\theta)}{p(z|X,\theta_n)}\right) - \log p(X|\theta_n)\right)\\ &= \sum_z\left(p(z|X,\theta_n)\times\log\left(\frac{p(X|z,\theta)\times p(z|\theta)}{p(z|X,\theta_n)\times p(X|\theta_n)}\right)\right)\end{aligned}$$

Conclusion

On a montré :

$$\begin{aligned}&\mathcal{L}(\theta) - \mathcal{L}(\theta_n) \geq \Delta(\theta|\theta_n)\\ \Leftrightarrow &\mathcal{L}(\theta) \geq \mathcal{L}(\theta_n) + \Delta(\theta|\theta_n)\end{aligned}$$

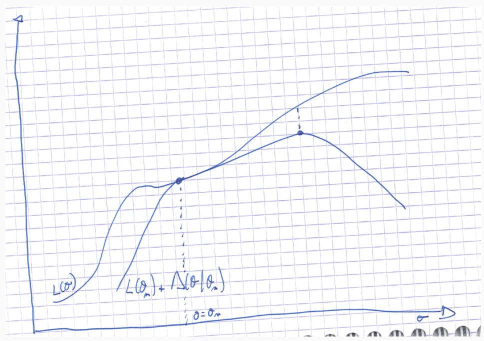
⇒ borne inf. de la vraisemblance

Deuxième argument (en exercice) :

$$\theta = \theta_n \Rightarrow \Delta(\theta|\theta_n) = 0$$

Conséquence : pour augmenter $\mathcal{L}(\theta)$, il suffit de maximiser $\mathcal{L}(\theta) + \Delta(\theta|\theta_n)$

En image



Encore un peu de calculs

$$
\begin{aligned}
\theta_{n+1} &= \arg\max_{\theta} \mathcal{L}(\theta_n) + \sum_z \left(p(z|X,\theta_n) \times \log\left(\frac{p(X|z,\theta) \times p(z|\theta)}{p(z|X,\theta_n) \times p(X|\theta_n)} \right) \right) \\
&= \arg\max_{\theta} \sum_z p(z|X,\theta_n) \times \log\left(p(X|z,\theta) \times p(z|\theta) \right) \\
&= \arg\max_{\theta} \sum_z p(z|X,\theta_n) \times \log p(X,z|\theta) \\
&= \arg\max_{\theta} \mathbb{E}_{Z|X,\theta_n}[\log p(X,z|theta)]
\end{aligned}
$$

Bilan

Les deux étapes de EM :

1. Étape E :
   - calculer $\mathbb{E}_{Z|X,\theta_n}[\log p(X,z|\theta)]$
   - déterminer une approximation de $\mathcal{L}(\theta)$
2. Étape M :
   - maximiser l'expression précédente
   - révient à améliorer la vraisemblance

$\Rightarrow$ espoir : maximisation plus simple que maximiser directement $\mathcal{L}(\theta)$

$\Rightarrow$ itérer le processus pour ≪ mettre à jour ≫ l'approximation

Résultat

- convergence vers un maximum local

EM for IBM 1

And now...

Recall

Maximum likelihood :

$$
\begin{aligned}
\mathcal{L}\left(\theta|\mathcal{D}\right) &= p_\theta\left(\mathcal{D}|\theta\right) && (1) \\
&= \prod_{n=1}^{N} p_\theta(\mathbf{t}^{(n)}|\mathbf{s}^{(n)}) && (2) \\
&= \prod_{n=1}^{N} \sum_{a^{(n)}} \prod_{i=1}^{|\mathbf{t}^{(n)}|} p(t_i^{(n)}|s_{a(i)}^{(n)}) && (3)
\end{aligned}
$$

$\hookrightarrow$ marginalization over the latent variable a

$\hookrightarrow$ ignoring normalization factors

$\hookrightarrow$ no close form solution

Counts versus posterior probabilities

Let us focus on $\theta(t|s)$:

$\hookrightarrow$ $(\mathbf{s}, \mathbf{t})$ = pair of sentence with $t_i = t$ and $s_j = s$

$\hookrightarrow$ what is the probability that $a_i = j$?

We have :

$$
\begin{aligned}
p(a_i = j|\mathbf{s},\mathbf{t}) &= \frac{p(a_i = j|\mathbf{s}) \times p(\mathbf{t}|a_i = j, \mathbf{s})}{p(\mathbf{t}|\mathbf{s})} && (4) \\
&= \frac{p(\mathbf{t}, a_i = j|\mathbf{s})}{p(\mathbf{t}|\mathbf{s})} && (5)
\end{aligned}
$$

$\hookrightarrow$ exactly the same estimator than in the supervised case...

$\hookrightarrow$ ...but weighted by probabilities

Let's Count !

$$\frac{p(\mathbf{t}, a_i = j|\mathbf{s})}{p(\mathbf{t}|\mathbf{s})} = \frac{\sum_{a|a(i)=j} \prod_{i=1}^{|\mathbf{t}|} \theta(t_i|s_{a(i)})}{\sum_a \prod_{i=1}^{|\mathbf{t}|} \theta(t_i|s_{a(i)})} \tag{6}$$

Marginalizing (i.e. summing) over all alignments

- $\hookrightarrow$ consider aligning each target position with each source position
- $\hookrightarrow$ $\forall i \in [\![1, |\mathbf{t}|]\!]$, $a(i)$ takes all values in $[\![0, |\mathbf{s}|]\!]$

That is why :

$$\sum_a \Leftrightarrow \sum_{a(1)=0}^{a(1)=|\mathbf{s}|} \sum_{a(2)=0}^{a(2)=|\mathbf{s}|} \dots \sum_{a(|\mathbf{t}|)=0}^{a(|\mathbf{t}|)=|\mathbf{s}|} \tag{7}$$

$$\tag{8}$$

The 'sum-product into product-sum' trick

For the sake of notations : $|\mathbf{s}| = 4$ and $|\mathbf{t}| = 3$.

We have :

$$\sum_{a(1)=0}^{4} \sum_{a(2)=0}^{4} \sum_{a(3)=0}^{4} \prod_{i=1}^{3} \theta\left(t_i|s_{a(i)}\right) \tag{9}$$

$$= \left(\sum_{a(1)=0}^{4} \theta\left(t_1|s_{a(1)}\right)\right) \sum_{a(2)=0}^{4} \sum_{a(3)=0}^{4} \prod_{i=2}^{3} \theta\left(t_i|s_{a(i)}\right) \tag{10}$$

$$= \left(\sum_{a(1)=0}^{4} \theta\left(t_1|s_{a(1)}\right)\right) \times \left(\sum_{a(2)=0}^{4} \theta\left(t_2|s_{a(2)}\right)\right) \times \left(\sum_{a(3)=0}^{4} \theta\left(t_3|s_{a(3)}\right)\right) \tag{11}$$

$$= \prod_{i=1}^{3} \sum_{a(i)=0}^{4} \theta\left(t_i|s_{a(i)}\right) \tag{12}$$

The 'sum-product into product-sum' trick

For the sake of notations : $|\mathbf{s}| = 4$ and $|\mathbf{t}| = 3$.

We have :

$$\sum_{a(1)=0}^{4} \sum_{a(2)=0}^{4} \sum_{a(3)=0}^{4} \prod_{i=1}^{3} \theta\left(t_i|s_{a(i)}\right) \tag{9}$$

$$= \prod_{i=1}^{3} \sum_{a(i)=0}^{4} \theta\left(t_i|s_{a(i)}\right) \tag{10}$$

- $\hookrightarrow$ sum of products $\rightarrow$ product of sums
- $\hookrightarrow$ that is cool / useful

Putting everything together

$$\frac{p(\mathbf{t}, a_i = j|\mathbf{s})}{p(\mathbf{t}|\mathbf{s})} \tag{11}$$

$$= \frac{\sum_{a|a(i)=j} \prod_{i=1}^{|\mathbf{t}|} \theta(t_i|s_{a(i)})}{\sum_a \prod_{i=1}^{|\mathbf{t}|} \theta(t_i|s_{a(i)})} \tag{12}$$

$$= \frac{\theta(t_i|s_j) \sum_{a(1=0}^{a(1)=|\mathbf{s}|} \dots \sum_{a(i-1)=0}^{a(i-1)=|\mathbf{s}|} \sum_{a(i+1)=0}^{a(i+1)=|\mathbf{s}|} \dots \sum_{a(|\mathbf{t}|)=0}^{a(|\mathbf{t}|)=|\mathbf{s}|} \prod_{k=1}^{|\mathbf{t}|} \theta(t_k|s_{a(k}}{\sum_a \prod_{k=1}^{|\mathbf{t}|} \theta(t_k|s_{a(k)})} \tag{13}$$

$$= \frac{\theta(t_i|s_j) \prod_{k=1}^{|\mathbf{t}|} \sum_{a(1=0}^{a(1)=|\mathbf{s}|} \dots \sum_{a(i-1)=0}^{a(i-1)=|\mathbf{s}|} \sum_{a(i+1)=0}^{a(i+1)=|\mathbf{s}|} \dots \sum_{a(|\mathbf{t}|)=0}^{a(|\mathbf{t}|)=|\mathbf{s}|} \theta(t_k|s_{a(k}}{\prod_{k=1}^{|\mathbf{t}|} \sum_a \theta(t_k|s_{a(k)})} \tag{14}$$

$$= \frac{\theta(t_i|s_j)}{\sum_{a(i)=1^{|\mathbf{s}|}} \theta\left(t_i|s_{a(i)}\right)} \tag{15}$$