Université de Strasbourg

Traduction automatique

TA statistique : alignement et modèles IBM

Enzo Doyen

enzo.doyen@unistra.fr

2025-10-02

🕮 Plan

Traduction automatique statistique (SMT)

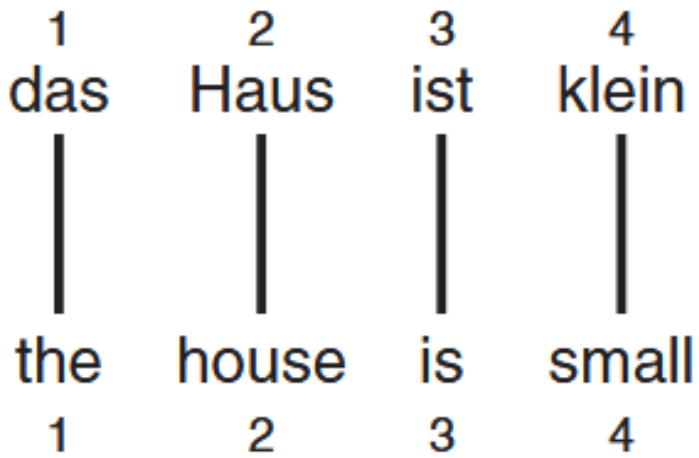
Idée principale : la traduction automatique peut être réalisée en utilisant des modèles statistiques basés sur des données d'exemple.

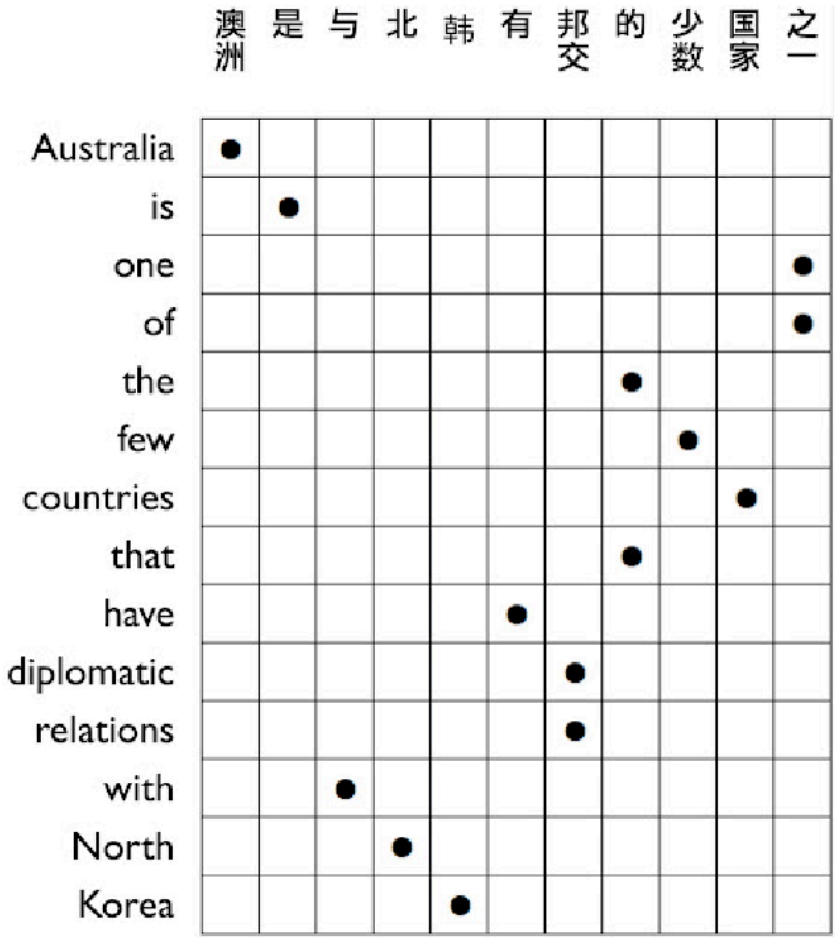
Procédure : elle implique l'utilisation de modèles statistiques pour apprendre les relations entre les langues à partir de corpus parallèles **alignés**.

I. Types d’alignement au niveau des mots

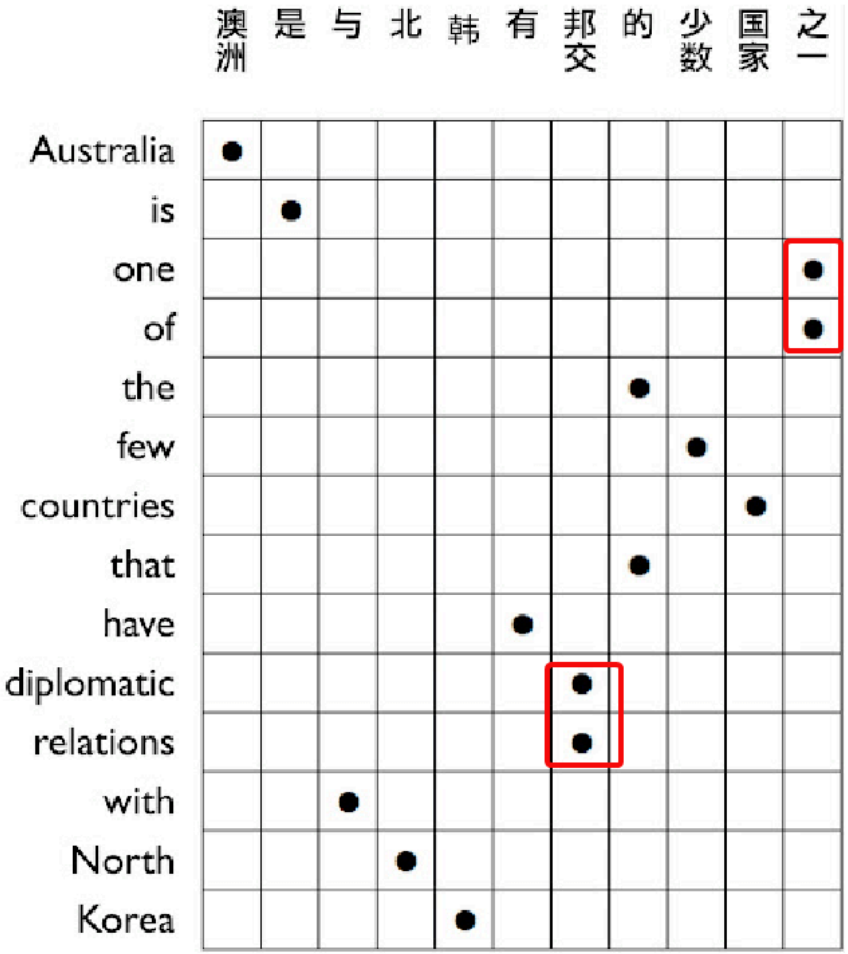
TA statistique : alignement au niveau des mots

Les modèles de TA statistique utilisent l'alignement au niveau des mots pour établir des correspondances entre les mots de la langue source et ceux de la langue cible.





Source : **Li (2022)**

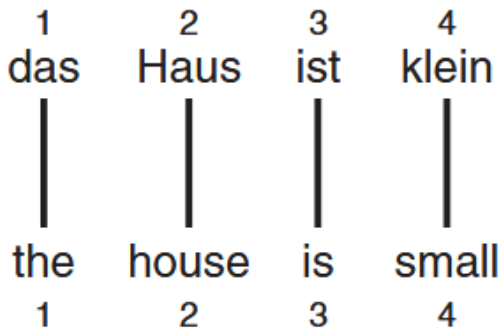


Source : **Li (2022)**

TA statistique : alignement au niveau des mots

On peut définir une fonction d'alignement a qui associe chaque mot allemand j à un mot anglais i.

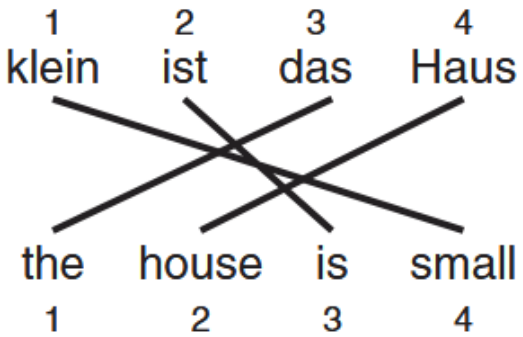
$$a : j \rightarrow i$$



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

TA statistique : alignement au niveau des mots

Exemple de **réordonnement des mots** dans une phrase traduite.



$$a : \{1 \to 3, 2 \to 4, 3 \to 2, 4 \to 1\}$$

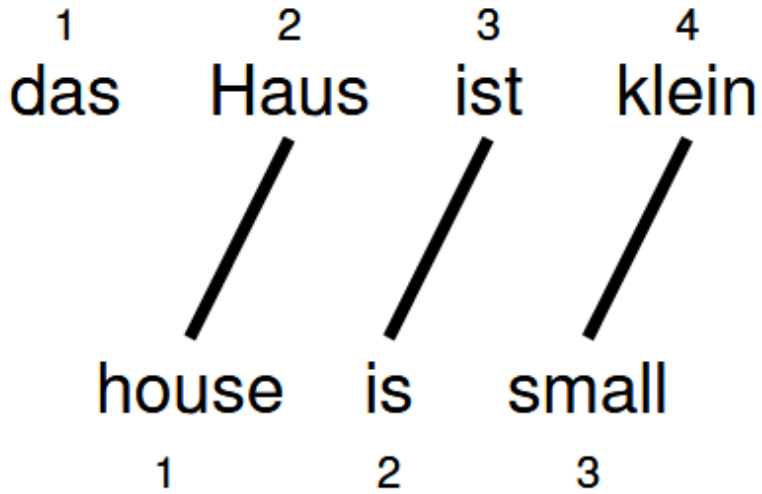
TA statistique : alignement au niveau des mots

Exemple de **relation un-à-plusieurs** (*one-to-many*) dans une phrase traduite : un mot source peut être aligné à plusieurs mots cibles.

$$a : \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 4, 5 \to 4\}$$

TA statistique : alignement au niveau des mots

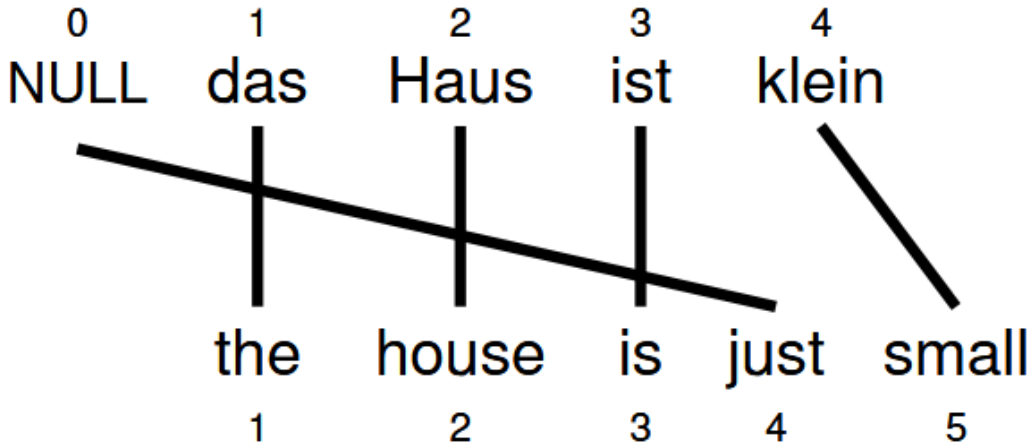
Exemple de **mot absent** dans la phrase traduite.



$$a : \{1 \to 2, 2 \to 3, 3 \to 4\}$$

TA statistique : alignement au niveau des mots

Exemple d'**insertion de mot** dans la phrase traduite : on ajoute un token **NULL** pour faire la correspondance.



$$a : \{1 \to 1, 2 \to 2, 3 \to 3, 4 \to 0, 5 \to 4\}$$

II. Modèles IBM

Modèles IBM

- Dans les années 90, série de 5 modèles développés par IBM pour la TA statistique **(Brown et al., 1993)** : du IBM Model 1 (le plus simple) au IBM Model 5 (le plus complexe) ; utilisés jusqu'à l'avènement de la traduction neuronale.
- L'objectif est de modéliser la traduction automatique comme un **problème probabiliste**.
- Apprentissage automatique des **probabilités de traduction entre les mots de deux langues** à partir de **corpus parallèles**.

Modèle IBM 1

- Modèle de traduction lexicale (mot à mot).
- L'ordre des mots et leur position ne sont pas pris en compte.
- Chaque phrase a une longueur fixe m.
- Au début, on suppose que les alignements sont **équiprobables**, c'est-à-dire que chaque mot source peut être aligné à n'importe quel mot cible avec la même probabilité.

Modèle IBM 1

On veut estimer la probabilité de traduire une phrase source $f = f_1, f_2, l..., f_m$ en une phrase cible $e = e_1, e_2, l..., e_n$.

On suppose que chaque mot source f_j est traduit indépendamment en un mot cible e_i.

Modèle IBM 1

On veut estimer la probabilité de traduire une phrase source $f = f_1, f_2, l..., f_m$ en une phrase cible $e = e_1, e_2, l..., e_n$.

On suppose que chaque mot source f_j est traduit indépendamment en un mot cible e_i.

Mais comment faire sans les alignements ?

Modèle IBM 1 : algorithme EM (Expectation-Maximization)

Introduction de l'**algorithme EM (Expectation-Maximization)**.

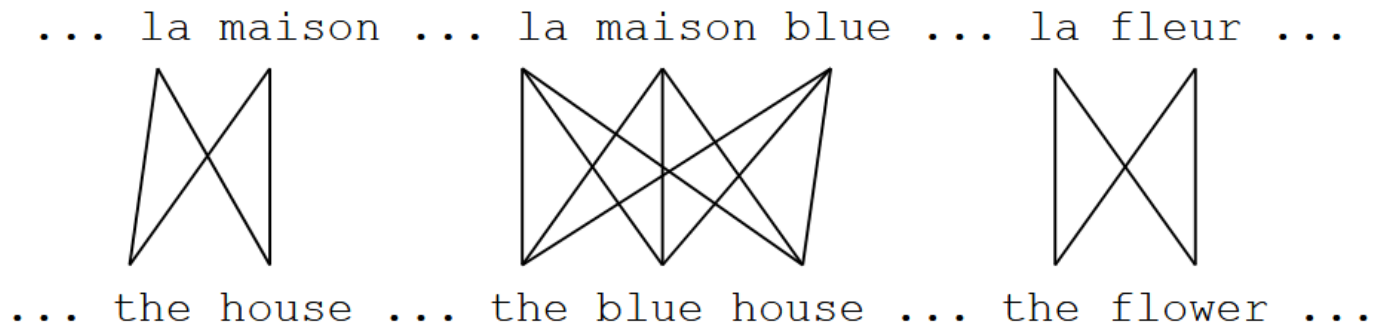
D'abord, tous les alignements sont considérés comme étant autant probables les uns que les autres. Ainsi, la probabilité d'un alignement a étant donné une phrase source e de longueur m et une phrase cible f de longueur l est d'abord définie sous la forme :

Définition initiale équiprobable $\left\{ P(a|e,m) = \frac{1}{(l+1)^m} \right.$

1 est utilisé pour prendre en compte le token **NULL** (cf. slide 7).

Modèle IBM 1 : algorithme EM (Expectation-Maximization)

$$P(a|e,m) = \frac{1}{(l+1)^m}$$



Modèle IBM 1 : algorithme EM (Expectation-Maximization)

L'algorithme fonctionne sur plusieurs itérations. À chaque itération, les deux étapes suivantes sont effectuées :

1. **Étape E (Expectation)** : calcul des probabilités d'alignement pour chaque mot source et cible.
2. **Étape M (Maximization)** : mise à jour des probabilités de traduction en fonction des alignements calculés.

Modèle IBM 1 : algorithme EM (Expectation-Maximization)

L'algorithme fonctionne sur plusieurs itérations. À chaque itération, les deux étapes suivantes sont effectuées (t = probabilité de traduction) :

1. **Étape E (Expectation)** : calcul des probabilités d'alignement pour chaque mot source et cible.

$$c(f_j \mid e_i) = \frac{t(f_j \mid e_i)}{\left(\sum_{k=0}^{m} t(f_j \mid e_k)\right)}$$

Normalisation (prob. 0 - 1)

Modèle IBM 1 : algorithme EM (Expectation-Maximization)

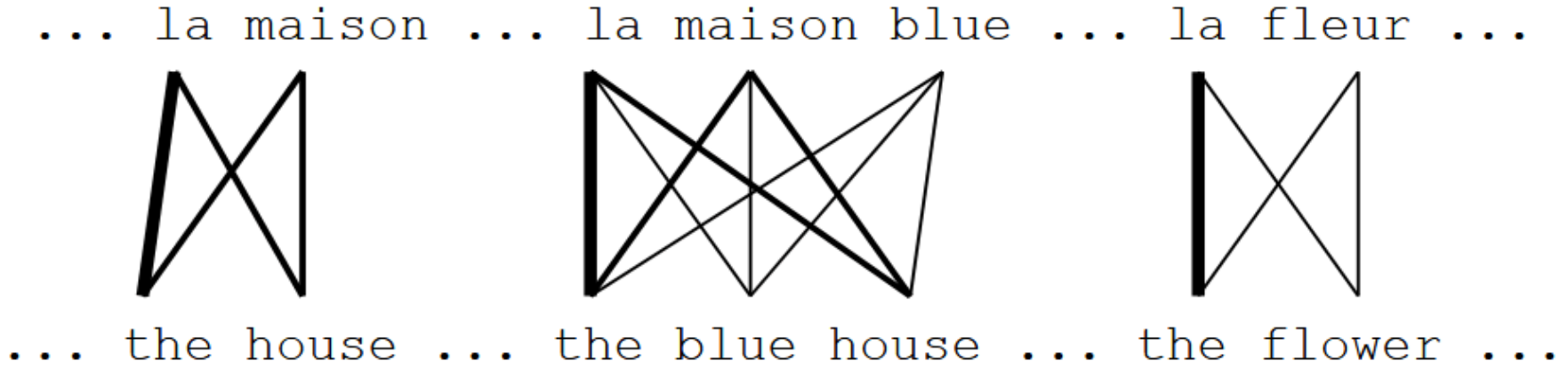
L'algorithme fonctionne sur plusieurs itérations. À chaque itération, les deux étapes suivantes sont effectuées :

1. **Étape E (Expectation)**
2. **Étape M (Maximization)** : mise à jour des probabilités de traduction en fonction des alignements calculés.

$$t(f \mid e) = \frac{\sum_{(e',f') \in \text{ corpus}} \sum_{j : f'_j = f} c(f \mid e)}{\sum_{(e',f') \in \text{ corpus}} \sum_{j=1}^{l'} c\left(f'_j \mid e\right)}$$

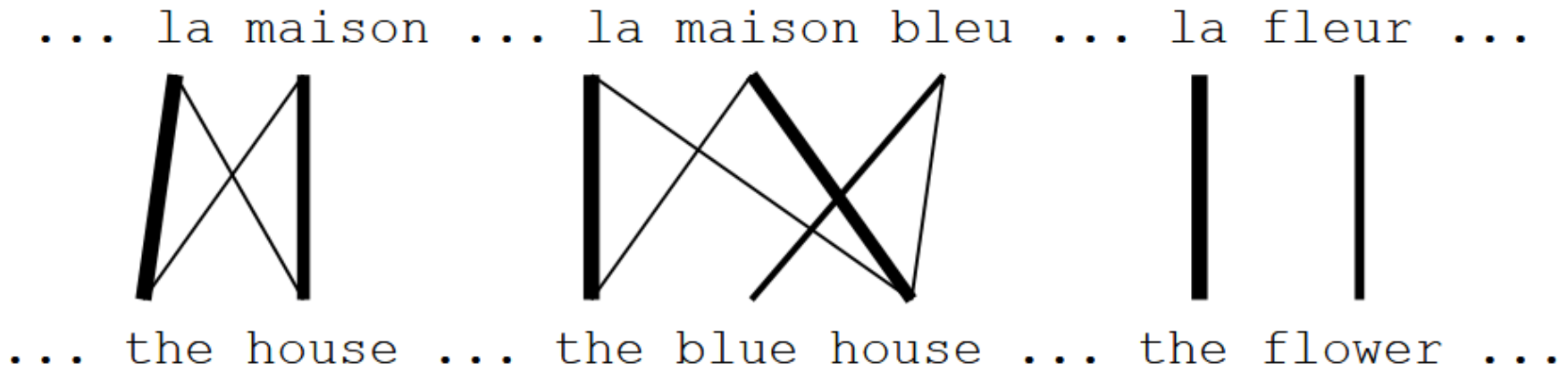
Modèle IBM 1 : algorithme EM (Expectation-Maximization)

Au bout d'une itération, les alignements les plus probables (comme entre « la » et « the ») sont renforcées :



Modèle IBM 1 : algorithme EM (Expectation-Maximization)

Après une autre itération, d'autres alignements (comme entre « maison » et « house ») sont renforcés :



Modèle IBM 1 : algorithme EM (Expectation-Maximization)

Au final, on obtient les alignements les plus probables, en fonction des cooccurrences observées dans les phrases.

Convergence quand les probs ne changent plus beaucoup entre 2 itérations.

Modèle IBM 1 : algorithme EM

Cela se traduit par la formule suivante :

$$P(e|f) = \sum_{a} \prod_{\mathrm{j}=1}^{l} t\left(f_j | e_{a_j}\right)$$

Où :

- e est la phrase cible,
- f est la phrase source,
- a est l'alignement,
- l est la longueur de la phrase cible,
- $t\left(f_j | e_{a_j}\right)$ est la probabilité que le mot source f_j soit traduit par le mot cible e_{a_j}.

Modèle IBM 1 : algorithme EM, exemple

Soit le minicorpus suivant :

Anglais (source)	Français (cible)
the house	la maison
the cat	le chat
a house	une maison

Vocabulaire :

- Anglais : {NULL, the, house, cat, a}
- Français : {la, maison, chat, une}

Modèle IBM 1 : algorithme EM, exemple

D'abord, initialisation des probabilités de traduction $t\left(f_j|e_i\right)$ uniformément.

t(la | NULL) = t(la | the) = t(la | house) = t(la | cat) = t(la | a) = 0.2

t(maison | NULL) = t(maison | the) = t(maison | house) = t(maison | cat) = t(maison | a) = 0.2

etc...

Modèle IBM 1 : algorithme EM, exemple (étape E)

(Pour l'exemple, on se concentre sur les paires « la maison » et « the house ».)

Ensuite, par exemple pour le mot « la », on calcule la probabilité d'alignement vis-à-vis de chaque mot source :

$$c(\text{la} \mid \text{NULL}) = \frac{t(\text{la} \mid \text{NULL})}{t(\text{la} \mid \text{NULL}) + t(\text{la} \mid \text{the}) + t(\text{la} \mid \text{house})}$$

$$= \frac{0.2}{0.2 + 0.2 + 0.2} = \frac{0.2}{0.6} = 0.3333$$

Modèle IBM 1 : algorithme EM, exemple (étape E)

Ensuite, par exemple pour le mot « la », on calcule la probabilité d'alignement vis-à-vis de chaque mot source :

$$c(\text{la} \mid \text{the}) = \frac{0.2}{0.6} = 0.3333$$

$$c(\text{la} \mid \text{house}) = \frac{0.2}{0.6} = 0.3333$$

Idem pour « maison »…

$$c(\text{maison} \mid \text{NULL}) = \frac{0.2}{0.6} = 0.3333$$

Modèle IBM 1 : algorithme EM, exemple (étape M)

Enfin, on met à jour les probabilités de traduction en fonction des alignements calculés. Par exemple, entre « la » et « the » :

$$t(\text{la} \mid \text{the}) = \frac{\text{somme de } (\text{la} \mid \text{the})}{\text{somme de tous les mots source alignés avec 'the'}}$$

Imaginons que, dans un corpus donné, l'alignement « la → the » ait été observé 3 fois, et que le mot « the » ait été aligné avec 6 mots source différents :

$$t(\text{la} \mid \text{the}) = \frac{3}{6} = 0.5$$

Autres modèles IBM

Modèle	Description
IBM Model 2	Prend en compte la **position des mots** dans la phrase.
IBM Model 3	Prend en compte la **fertilité** : un mot dans une langue peut être traduit par 0, 1 ou plusieurs mots dans l'autre langue.
IBM Model 4	L'alignement prend mieux en compte la **distorsion**, qui reflète les différences d'ordre des mots entre les langues.
IBM Model 5	Corrige le problème de **déficience** du modèle 4 : ce dernier attribuait une probabilité positive à des alignements impossibles (comme placer deux mots au même endroit).

III. Mise en pratique

Mise en pratique

Sur Moodle, vous trouverez une implémentation du modèle IBM 1 en Python. Il vous sera demandé de la tester et d'améliorer quelque peu le code.

Il y a également une implémentation du modèle IBM 2 : comparez le code (les différences sont marquées avec un commentaire commençant par `#!`) et notez les améliorations par rapport au modèle précédent.

Bibliographie

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., et Mercer, R. L. (juin 1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Comput. Linguist.*, *19*(2), 263‑311.

Koehn, P. (2009). *Statistical Machine Translation*. Cambridge University Press. 10.1017/CBO9780511815829

Li, B. (novembre 2022). *Word Alignment in the Era of Deep Learning: A Tutorial*. 10.48550/arXiv.2212.00138

♡ Remerciements

- Pablo Ruiz Fabo pour le contenu de certaines diapositives.
- Philipp Koehn pour les images d'alignement.