Introduction à la traduction statistique

- Cadre général, *success stories*
- L'approche canal bruité
  - Obtention d'un **bitexte**: alignement de phrases
  - Les modèles IBM 1, 2, 3, 4 et 5
  - D'autres modèles
- Évaluation de la traduction

Chapitres 13 dans Manning and Schütze [1999], chapitre 21 dans Jurafsky and Martin [2000]

Cadre général

Buzz words: mondialisation du commerce, localisation

- Selon une étude menée par *Équipe Consortium Ltd*, le marché mondial de la traduction s'est élevé à 3,5 milliards de dollars CAN en 1998; et aurait doublé en 2000.
- Selon une enquête commanditée par le *ministère fédéral des Ressources Humaines* le Canada a dépensé 443 millions de dollars en 1998,
- 45% de tous les traducteurs au pays, et 48% de tous les cabinets de traduction se trouvent au Québec.

Besoins majeurs: traduction de textes officiels, commerciaux, administratifs, techniques, scientifiques, médicaux, etc. . . . tâche à la fois répétitive et précise (la traduction d'œuvres littéraires est assez marginale).

Histoire du système le plus populaire: Systran[1]

In 1962, there were 48 working groups deeply involved in the research and development of MT.

The US National Academy of Sciences published the ALPAC report in 1966. This report stated that MT had a limited future, and then put an end to all US Government research and development projects and financing for MT.

Peter Toma, Ph.D., a linguist researcher for MT, began his work in **1957** at the California Institute of Technology. Later, Dr. Toma became involved in the pioneering work in **Russian to English** MT at Georgetown University, the largest MT project in the US of that time. In **1968**, Dr. Toma established a company in La Jolla, California, USA, with a product called SYSTRAN; an acronym for System Translation. Soon thereafter, the company was contracted to develop Russian to English MT for the US Air Force.

The first SYSTRAN system was tested in early 1969 at Wright-Patterson Air Force Base in Dayton, Ohio, USA. Since 1970, the system has continued to provide translations for the US Air Force's Foreign Technology Division.

During the period 1974-1975, SYSTRAN was used by NASA for the joint US-USSR Apollo-Soyouz space project.

[1] Texte pris à l'adresse: `http://www.translation.net/systrans.html`

Histoire du système le plus populaire:Systran

In **1975**, Dr. Toma demonstrated a prototype of **English to French** MT to representatives of the Commission of the European Communities (CEC), which resulted in a contract to develop MT systems for various European language pairs. The CEC uses more than 12 SYSTRAN MT systems for the translation of its internal documents.

Xerox Corporation began using SYSTRAN in 1978, the same period during which the SYSTRAN MT systems were transformed into multitarget (English to several non-English languages). Xerox continues to use the systems for translation of thousands of pages per year. This process allows Xerox to launch multilingual products to the global marketplace.

In 1981, SYSTRAN initiated the development of the Japanese to English and English to Japanese MT systems. Under newly European influence, the first World SYSTRAN Conference, organized by the CEC, was held in Luxembourg, shortly thereafter. This conference, the first dedicated to one single MT system, brought together all the principal SYSTRAN users from around the world.

SYSTRAN introduced the utility "Customer Specific Dictionaries", (referred to as CSDs), in 1989. CSDs are dictionaries created by users with their own specific terminology.

The company developed integrated MT for Xerox in 1990. This new system provided the option to preserve the original document format of a text during the translation process. This option has been incorporated into SYSTRAN products.

Histoire du système le plus populaire:Systran[2]

In 1992, SYSTRAN began the "C" conversion project, bringing its powerful, patented MT technology to the PC.

In 1995, SYSTRAN PROfessional for Windows in standalone and client/server versions were launched.

SYSTRAN received a contract from the US National Air Intelligence Center to develop several Eastern European MT language pairs in 1996. This included the first-ever Serbo-Croatian to English MT system, already delivered to the US Government. Also in 1996, SYSTRAN signed a licensing agreement with Seiko Instruments, Inc., through which SYSTRAN would provide linguistic data and software for Seiko's hand-held translation products.

In early 1997, Ford Motor Company acquired multiple custom software licenses of SYSTRAN's MT software, to be embedded directly into Ford's systems.

Today, **14 SYSTRAN MT language pairs** are commercially available

[2]Texte consulté à l'automne 2002

Systran de l'intérieur[3]

Basé principalement sur la consultation de grands dictionnaires bilingues à grande couverture et mis au point à grands renforts de ressources humaines.

Un dictionnaire bilingue SYSTRAN est en fait constitué d'un dictionnaire principal dont les entrées sont des mots seuls, et de multiples dictionnaires spécialisés (ou contextuels) dont les entrées sont des unités (séquences de mots) spécialisées.

Une entrée dans le dictionnaire principal contient de nombreuses informations morphologiques, grammaticales et sémantiques.

Ex: valence d'un verbe, transitivité, type de nom (animé, énumérable, abstrait, etc.), marqueurs sémantiques divers (contenant, propriété physique, appareil, nourriture, etc).

Le dictionnaire principal est lemmatisé (sauf pour l'anglais). À chaque entrée lemmatisée correspond une association privilégiée dans la langue cible elle même étiquetée avec des informations nécessaires à l'étape de génération.

[3] D'après Hutchins and Somers [1992]

Systran de l'intérieur

Les dictionnaires contextuels contiennent entre-autre:

des formes idiomatiques: *de plus en plus, par dessus tout, toute chose égale par ailleurs, etc* dont la traduction est fixe dans une langue donnée.

des formes sémantiques limitées: identifiant des unités comme *hydraulic break* pour éviter par exemple d'interpréter de manière erronée *hydraulic break fluide*; ou encore des formes composées comme *pomme de terre/potato*

des informations homographiques: dressant le contexte syntaxique nécessaire à la résolution de formes homographes.

exception aux règles syntaxiques: comme par exemple *nor* en anglais qui fait exception à la règle des autres conjonction en permettant l'inversion du groupe verbal qui suit: *... nor could he see the difficulties*

des informations sémantiques contextuelles: pour déterminer le choix lexical terminal. Par exemple *grow* en anglais se traduit par *grandir* d'après le dictionnaire principal, mais peut être traduit par *élever* si le complément est "animé", ou encore "cultiver" si le complément est une plante.

Systran de l'intérieur

Une architecture linéaire classique:

Étape de pré-traitement:

- 1- Identification de formatage (titres, paragraphes, indentation, etc)
- 2- Repérage des formes idiomatiques
- 3- Consultation du dictionnaire principal
- 4- Analyse morphologique (sauf si la langue source est l'anglais). Inférence éventuelle dans le cas de mots inconnus
- 5- Identification des noms composés (par consultation des dictionnaires sémantiques limités). Prise de décision dans des cas comme: *Il parla à la femme de ménage* (*charlady* ?).

Systran de l'intérieur

Étape d'analyse:

- 6- Résolution d'homographes (par analyse du contexte). Ex: *states* peut être un verbe ou un nom, mais pas lorsque précédé de *many*.
- 7- Segmentation des phrases en clauses (marqueurs = ponctuations, conjonctions de coordination, pronoms, etc).
- 8- Identification de relations syntaxiques simples (entre noms et adjectifs, etc.). Des informations comme le temps de la phrase, son mode, etc. sont également déterminées dans cette phase.
- 9- Désambiguisation des énumérations:
  Ex: *Smog and pollution control are important factors /Smog and pollution control is under consideration*
- 10- Identification sujet/prédicats (traitement simpliste)
- 11- Identification de relations syntaxiques "profondes"

Systran de l'intérieur

Étape de Transfert:

- 12- Transfert des idiomes conditionnels. Ex: *agree* sera traduit par *convenir* s'il est à la forme passive et par *être d'accord* sinon.
- 13- Traduction des prépositions
- 14- Transfert structurel (par consultation de règles spécifiques):
  Ex: *He* ***expects*** *to come / Il s'attend à ce qu'il vienne*

Étape de génération:

- 15- Affectation de la traduction par défaut aux mots encore à traduire
- 16- Génération morphologique
- 17- Arrangements finaux: perturbation de l'ordre des mots, élisions (*le homme* $\rightarrow$ *l'homme*).

Systran par l'exemple

source: In 1992, SYSTRAN began the "C" conversion project, bringing its powerful, patented MT technology to the PC.

cible: En 1992, SYSTRAN a commencé le projet de conversion de " C ", apportant sa technologie puissante et brevetée de la TA au PC.

source: De guerre lasse, il abandonna sa mission.

cible: Of tired war, it gave up its mission.

source: Elle lui plaît

cible: It to him likes

Pour jouer d'avantage avec SYSTRAN consulter: `http://world.altavista.com/tr`

Le system Météo: un succès québécois !

1965, on fait du traitement des langues naturelles à l'université de Montréal (laboratoire CETADOL). Le gouvernement canadien introduit sa politique bilingue. S'en suit une période faste pour le domaine de la traduction automatique.

Le CETADOL se concentre alors sur la problématique de la traduction automatique. Le laboratoire est rebaptisé TAUM (Traduction Automatique de l'Université de Montréal).

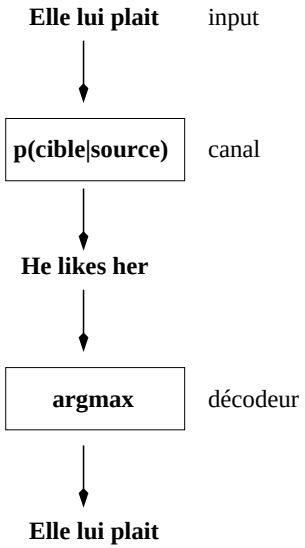
Le groupe développe un prototype anglais-français écrit en *Q-system* développé par Alain Colmerauer. Fort de ce succès, TAUM décroche en 1975 un contrat pour automatiser la traduction des bulletins météorologiques canadiens (tâche répétitive, peu passionnante).

Un prototype naît en **1976** et un système est finalement utilisé à temps plein depuis mai 1997.

C'est un système très spécifique (*i.e.* pas adaptable). Tous les types de phrases des bulletins ont été analysés structurellement, et le transfert est systématique.

L'approche par canal bruité Brown et al. [1993]

Rappelez-vous du canal bruité: des phrases en langue source (ex: le français) rentrent dans un canal qui est tellement bruité qu'il transmet les phrases dans leur version cible (ex: en anglais). On fait l'hypothèse que le canal bruité est régit par un modèle probabiliste $p(cible|source)$ et le but est de déterminer ce modèle en observant suffisamment de transmissions de phrases (cad de paires de phrases en relation de traduction).



L'approche par canal bruité Brown et al. [1993]

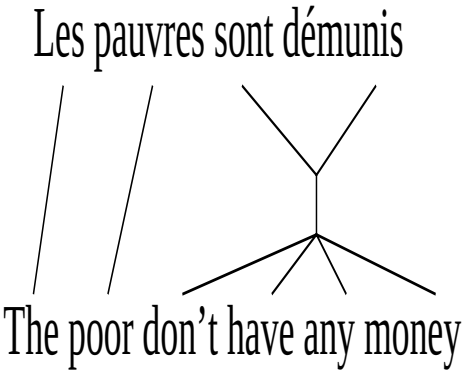
Deux problèmes à résoudre pour élaborer un système de traduction probabiliste: a) apprendre les paramètres d'un modèle de traduction (celui sous-jacent au canal); b) décoder la sortie du canal à l'aide de ce modèle.

De manière plus formelle, dans le cas d'un modèle de traduction de la langue source $\mathcal{E}$ vers la langue cible $\mathcal{F}$ et étant donnée une phrase source e, voici comment trouver la traduction $\hat{f}$ de e:

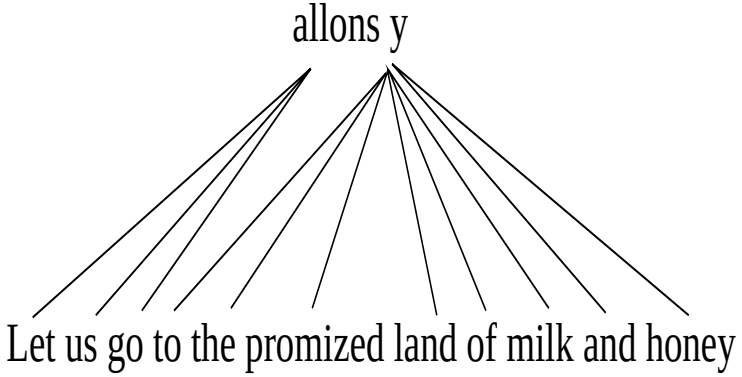
$$\begin{aligned}
\hat{f} &= \text{argmax}_{f \in \mathcal{F}}\, p(f|e) \\
&= \text{argmax}_{f \in \mathcal{F}}\, \frac{p(e|f) \times p(f)}{p(e)} \\
&= \text{argmax}_{f \in \mathcal{F}}\, \underbrace{p(e|f)}_{\text{traduction}} \times \underbrace{p(f)}_{\text{langue (cible)}}
\end{aligned}$$

L'approche par canal bruité Brown et al. [1993]

Dans Brown et al. [1993] un modèle de traduction est vu comme un modèle d'alignement de mots.



L'approche par canal bruité Brown et al. [1993]



On peut entraîner des modèles d'alignement de mots à partir d'un corpus de paires de phrases en relation de traduction Brown et al. [1993]

$\mathcal{T} =<$ (Il mange, He is eating), (Le chien court, The dog is running), (La traduction automatique est en devenir, Machine translation is evolving), (Mardi 15 mai, Thusday may 15th), ... $>$

Comment obtenir un tel corpus ?

Appariement automatique de textes

Définition: tâche consistant à mettre en correspondance dans un corpus bilingue les phrases qui sont en relation de traduction.

Intérêt: mémoires de traduction, extraction de lexiques bilingues et de dictionnaires terminologiques, désambiguisation sémantique, etc.

Propriété cyclique: aligner des mots $\longrightarrow$ aligner des phrases $\longrightarrow$ aligner des mots

En pratique: des indices plutôt simples permettent d'aligner de manière "satisfaisante" des corpus au niveau de la phrase.

$\hookrightarrow$ *Deux types d'information sont exploités dans les algorithmes d'alignement: de l'information métrique et de l'information à caractère linguistique.*

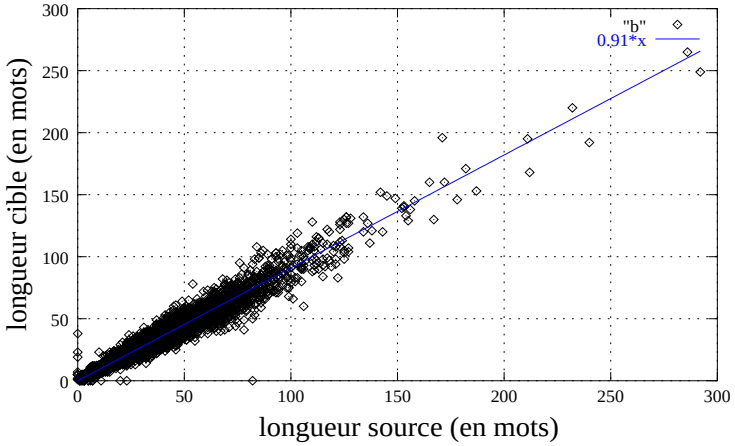
Brown et al. [1991], Debili [1992], Gale and Church [1993], Simard et al. [1992], Kay and Röscheisen [1993], Fung and Church [1994], Wu [1994], Langé and Gaussier [1995], Simard and Plamondon [1996], Melamed [1997], Chang and Chen [1997]

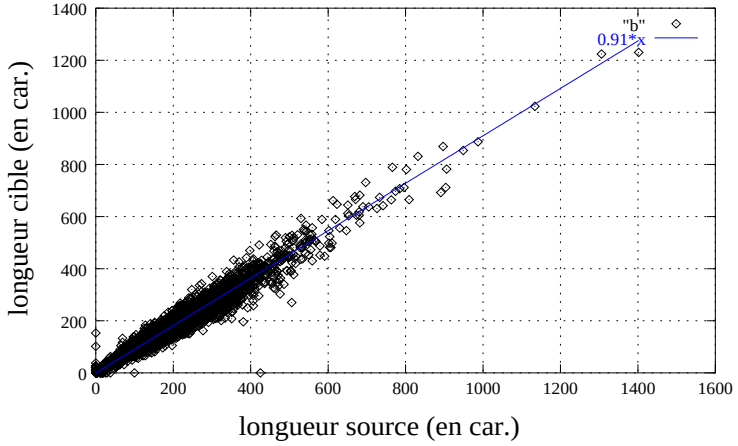
Appariement automatique de textes

Débat L'intelligence artificielle	Artificial intelligence A Debat
Depuis 35 ans, les spécialistes d'intelligence artificielle cherchent à construire des machines pensantes. Leurs avancées et leurs insuccès alternent curieusement.	Attempts to produce thinking machines have met during the past 35 years with a curious mix of progress and failure.
	Two further points are important.
Les symboles et les programmes sont des notions purement abstraites.	First, symbols and programs are purely abstract notions.

Indices métriques (longueurs)

Le processus de traduction tend à préserver la longueur (comptée en mots ou en caractères) des phrases: il est assez probable qu'une phrase courte soit traduite par une phrase courte, et qu'à l'inverse une phrase longue soit traduite par une phrase longue. Cette idée simple a été exploitée en premier lieu par Brown et al. [1991], Gale and Church [1993].





Corpus ilo - 7124 paires de phrases

Indices métriques (patterns)

En général, à une phrase source est alignée une phrase cible. De manière moins fréquente, n phrases sources sont alignées avec m phrases cibles.

Corpus ilo: 7124 paires de phrases					
pattern	nb	pattern	nb	pattern	nb
1/1	6744 (94.5%)	4/1	15	1/3	3
2/1	203 (2.85%)	1/0	13	5/1	3
1/2	48 (0.6%)	0/1	6	0/9	1
3/1	37 (0.5%)	0/2	5	1/4	1
2/2	26 (0.3%)	2/0	5	1/14	1

Gale and Church [1993] proposent de considérer uniquement les six schémas (patterns) suivants: 1/1 ($p = 0.89$), 1/2 et 2/1 ($p = 0.089$), 0/1 et 1/0 ($p = 0.011$) et 2/2 ($p = 0.010$).

Indices métriques

Gale and Church [1993] proposent de mesurer la qualité d'un appariement de l_1 et l_2 mots par une estimée de $p(match|\delta)$:

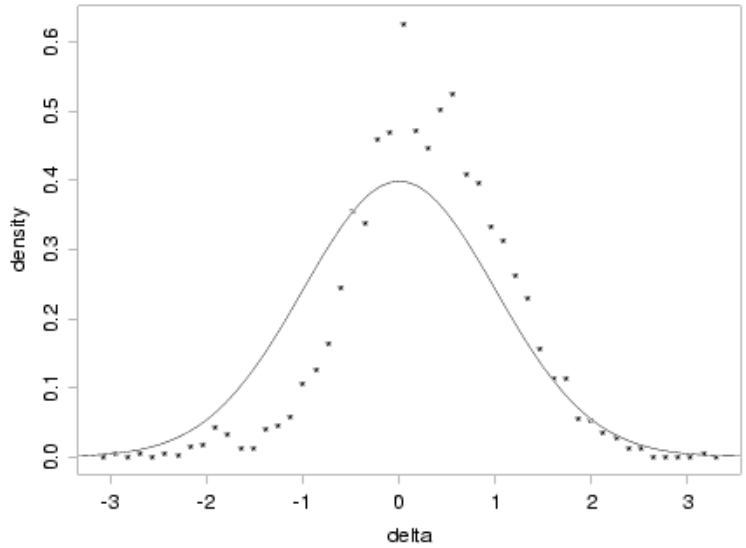
$$\begin{aligned} S_{gc} &= -\log(p(\delta|match) \times p(match)) \\ p(\delta|match) &= 2 \times (1 - p(|\delta|)) \end{aligned}$$

où δ suit une loi normale centrée réduite: $\delta = \frac{(l_2 - l_1 c)}{\sqrt{l_1 s^2}}$ qui dépend de deux paramètres c (la moyenne) et s^2 (la variance) déterminés expérimentalement (resp. 1 et 6.8).

$p(match)$ est donnée par les comptes du transparent précédent ($p(1/1)$, $(1/2)$, etc).

Indices métriques

Hypothèse: Chaque caractère de l_1 donne lieu à un nombre aléatoire de caractères dans l_2. Ces variables aléatoires sont indépendentes et suivent la même distribution. À un caractère de l_1 correspond c caractères de l_2 en espérance; s^2 est la variance de cette espérance:



En pointillé: distribution des δ empirique. En trait plein, une normale centrée réduite.

Indices à caractère linguistique

Simard et al. [1992] proposent d'aligner des corpus bilingues en exploitant le fait que deux phrases en relation de traduction partagent souvent des mots communs ou proches (chiffres, noms propres, etc).

Deux mots sont dits **cognates** s'ils partagent des propriétés communes à un quelconque niveau (sémantique, orthographique, etc.).

Exemples: *accès/access*, *activité/activity*, *parlement/parliament*

Contre-exemple: *librairie/library*

Définition pragmatique Simard et al. [1992]: deux mots sont cognates s'ils ont en commun un préfixe de 4 caractères au moins. Si l'un des mots contient un chiffre au moins, alors les deux mots sont cognates s'ils sont égaux.

Consulter Ribeiro et al. [2001] pour un algorithme permettant de déterminer plus de cognates (ex: *gouvernement/government*).

Indices à caractère linguistique

Simard et al. [1992] proposent de noter la qualité d'un appariement par: $S_{cog} = \frac{P_T(c|n)}{P_R(c|n)} \times p(match)$ où $P_T(c|n)$ et $P_R(c|n)$ dénotent la probabilité que deux segments de longueur moyenne n (en mots) possèdent c cognates sous les hypothèses respectives que les segments sont en relation de traduction, ou au contraire qu'ils ont été sélectionnés aléatoirement.

Les auteurs observent que sur une partie étiquetée du Hansard, que ces deux probabilités suivent approximativement des lois binomiales où p_T (resp. p_R) désigne la probabilité qu'un mot d'un segment ait un cognate dans un segment de l'autre langue lorsque ces deux segments sont (resp. ne sont pas) traduction l'un de l'autre (p_T et p_R ont été expérimentalement fixés à 0.3 et 0.09).

$$\begin{aligned} p_T(c|n) &= \binom{n}{c} \times p_T^c \times (1-p_T)^{n-c} \\ p_R(c|n) &= \binom{n}{c} \times p_R^c \times (1-p_R)^{n-c} \end{aligned}$$

Algorithme d'alignement

Par programmation dynamique (récurrence dont on mémorise les calculs intermédiaires). Ici, $D(i,j)$ est le meilleur alignement des phrases sources $s_1, \ldots, s_i$ avec les phrases cibles $t_1, \ldots t_j$.

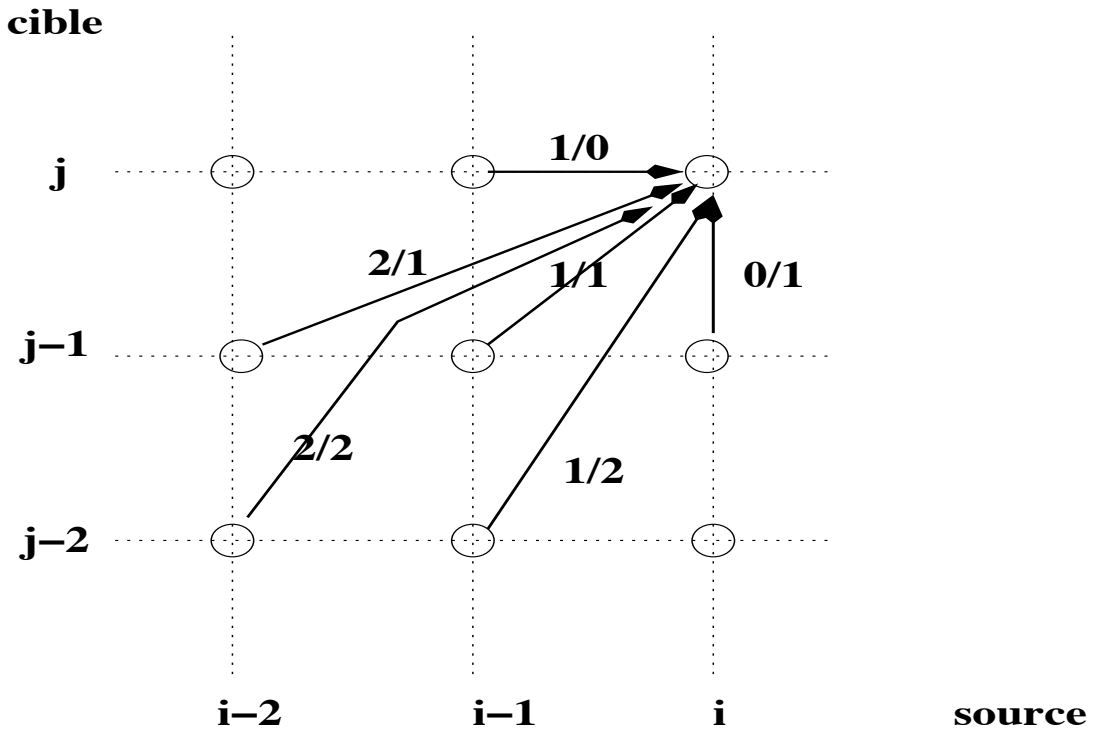
À l'initiale, $D(i,j)$ est nul pour tout i et j, puis on considère systématiquement les six schémas d'appariement: 0/1, 1/0, 1/1, 1/2, 2/1 et 2/2:

$$D(i,j) = min \begin{cases} D(i,j-1) & + & d_1 \\ D(i-1,j) & + & d_2 \\ D(i-1,j-1) & + & d_3 \\ D(i-1,j-2) & + & d_4 \\ D(i-2,j-1) & + & d_5 \\ D(i-2,j-2) & + & d_6 \end{cases}$$

où $d_i, i \in [1,6]$ sont les coûts respectifs de chaque schéma selon un score donné (par exemple S_{gc} ou S_{cog}).

Algorithme d'alignement

Comme dans viterbi, on garde un pointeur de retour arrière sur la meilleure case de laquelle on vient.



Appariement automatique de textes

Algorithme d'alignement: codage

Nous avons besoin de deux tables consignant les informations de chaque phrase des textes source et cible.

```
INFO Source[nbPhraseSource];
INFO Cible[nbPhraseCible];
```

où `INFO` contient toutes les informations nécessaires au calcul du score d'un appariement.

Ex:

$$\text{INFO} = \left[\begin{array}{ll} \text{int carLength;} & \text{//longueur de la phrase en caractères} \\ \text{int wordLength;} & \text{//longueur de la phrase en mots} \\ \text{string texte;} & \text{//la phrase elle-même (pas nécessaire)} \end{array}\right.$$

Algorithme d'alignement: codage

Nous avons également besoin d'une représentation de l'espace de recherche, par exemple une grande matrice:

`DTW Space[nbPhraseSource][nbPhraseCible]`

$$\text{DTW} = \left[\begin{array}{ll} \text{float score;} & //\text{le (meilleur) score} \\ \text{pair}\langle int, int \rangle \text{ back;} & //\text{le "pointeur" de retour (un entier suffit)} \end{array} \right.$$

Note: si on souhaite aligner un corpus bilingue de 1000 phrases sources avec 1000 phrases cibles, alors une représentation matricielle comme celle-ci occupe un espace mémoire de $1000 \times 1000 \times (8 + 2 \times 4)$ soit $16\,000\,000$ octets (16 Meg!).

↪ **En pratique:** on limite l'espace de recherche autours de la diagonale → représentation "matrice creuse".

Algorithme d'alignement: codage

Nous avons également besoin d'une fonction de score qui retourne le score d'un appariement donné, déterminé par les indices source et cible des phrases "d'arrivée" et du pattern d'alignement concerné:

```
float score(int i, int j, int di, int dj, ...)
```

On suppose dans la suite que le score d'un alignement est additif (somme des scores des appariements = score de l'alignement). Cette fonction en pratique doît être optimisée car appelée fréquemment.

Éventuellement, et selon le score utilisé, il nous faut précalculer des informations (ex: les mots cognates) stockés dans des tables spécifiques.

Algorithme d'alignement: codage

Space[i][j].score $\leftarrow 0\ \forall i \in [1, nbPhSource], j \in [1, nbPhCible]$
for (int i=0; i<nbPhSource; i++) **do**
  path $\leftarrow$ false
  for (int j=0; i<nbPhCible; j++) **do**
    if (inSpace(i,j)) **then**
      // alignent 1/1
      if (inSpace(i-1,j-1)) **then**
        score $\leftarrow= Score(i, j, 1, 1)$
        if (score $>$ Space[i][j].score) **then**
          Space[i][j].score = score;
          Space[i][j].back = make_pair(1,1); path $\leftarrow$ true
      // alignent 2/1
      if (inSpace(i-2,j-1)) **then**
        score $\leftarrow= Score(i, j, 2, 1)$
        if (score $>$ Space[i][j].score) **then**
          Space[i][j].score = score;
          Space[i][j].back = make_pair(2,1); path $\leftarrow$ true
      // on passe de même tous les alignements possibles
  if (path == false) **then**
    // Alignement impossible (impasse sur la phrase i)

Algorithme d'alignement: retour du meilleur chemin

Pour rendre le code plus simple (pas de recherche du maximum), on a supposé ici que les tables `Source` et `Cible` sont augmentées d'une phrase indiquant la fin de chaque document [4]

Le meilleur alignement a pour score `Space[nbPhSource][nbPhCible].score` et on retourne `Space[nbPhSource][nbPhCible].back` pour obtenir l'alignement:

```
i ← nbPhSource
j ← nbPhCible
while (i && j) do
   PRINT Alignement en (i,j) avec un schéma: Space[i][j].back.first/Space[i][j].back.second
   i -= Space[i][j].back.first
   j -= Space[i][j].back.second
```

[4] On a fait de même au début, de sorte que tout alignement commence par le couple (0,0) et finit par le couple (nbPhSource,nbPhCible).

Réduction de l'espace de recherche

Permet de réduire les temps de calculs de manière significative, et permet – éventuellement – d'améliorer les performances du système.

Idée: Un alignement grossier au niveau des mots permet de réduire l'espace de recherche. En particulier, les cognates peu fréquents sont probablement de bons indicateurs de synchronisation des deux textes (hypothèse).

Soit M une matrice à deux dimensions de telle sorte que $M(i,j)$ vaut 1 si le i-ème mot source et le j-ième mot cible sont cognates, et 0 sinon.

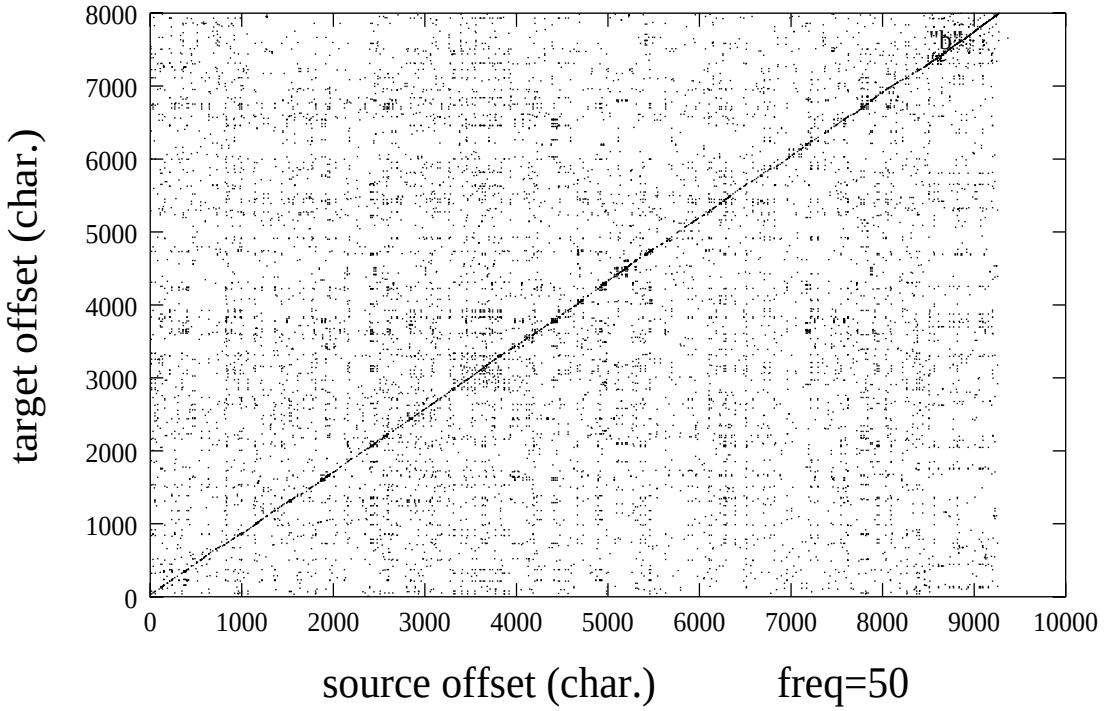
Alors on peut définir une fonction de coût de l'alignement de deux mots. Ici on cherche à minimiser la déviation à la diagonale que l'on devrait observer si notre hypothèse est vérifiée.

$$S(I,J) = \min_{i=I-R}^{I-1} \min_{j/M_{i,j}=1} (S(i,j) + F(i,j,I,J))$$

$$\text{with } F(i,j,I,J) = \frac{J-j}{I-i} + (I-i-1) \times \mathcal{C}$$

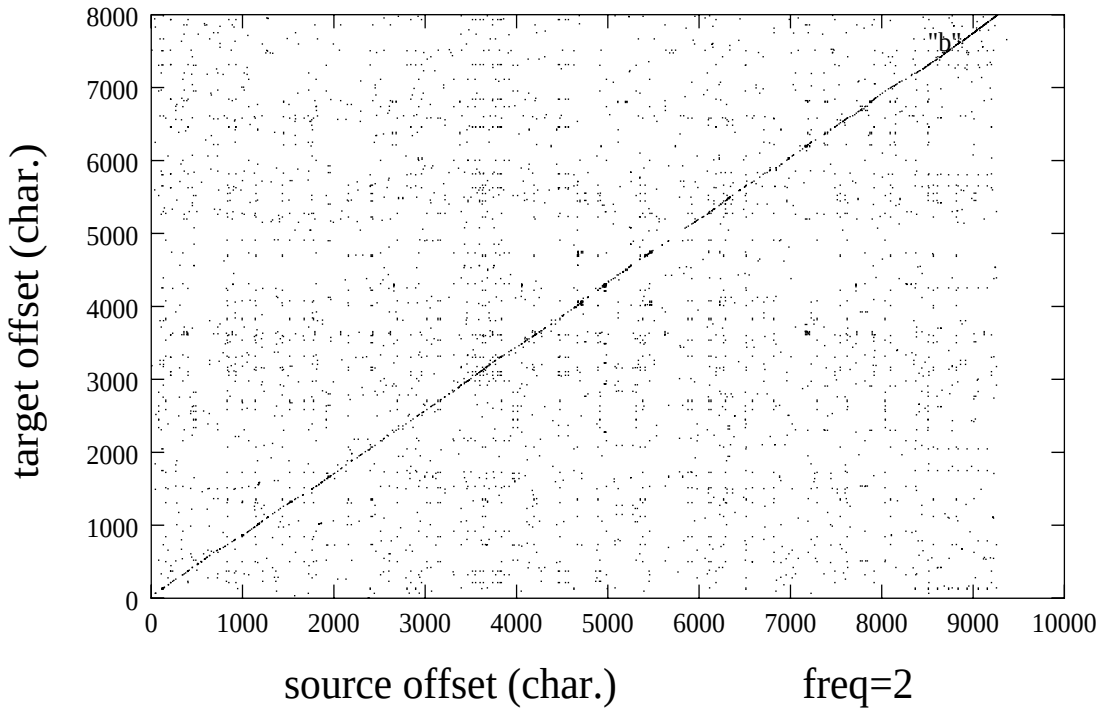
$\mathcal{C}$ et R constantes déterminées empiriquement.

Réduction de l'espace de recherche



Un point correspond à deux mots qui sont cognates dans un bitexte.

Réduction de l'espace de recherche



Un point correspond à deux mots qui sont cognates dans un bitexte.

Alignement de phrases: Évaluation

L'AUPELF-UREF finance ARCADE une action de recherche sur l'évaluation d'algorithmes d'alignement bilingues Langlais et al. [1998].

- Une première campagne (95-96) avait pour but de collecter des corpus de référence (des bitextes alignés manuellement au niveau des phrases) et de mettre au point une méthodologie d'évaluation de la performance des algorithmes d'alignement. Cinq équipes participent aux tests:
  $\hookrightarrow$ *RALI (Montréal), LORIA (Nancy), IRMC(Paris), ISSCO (Genève), LIA (Avignon).*
- Une seconde campagne (97-98) réunit 9 équipes et étend les tests à l'alignement de mots Véronis and Langlais [2000]:
  $\hookrightarrow$ *+ LILLA (Nice), UPenn (Pennsylvanie), CEA (Paris), XEROX (Grenoble)*

Arcade: corpus de référence

BAF : $400\,000$ mots par langue (anglais/français). Quatre types de textes: INST ($300\,000$ mots par langue de textes institutionnels, ex: Hansard), SCIENCE ($50\,000$ mots par langue d'articles scientifiques), TECH (documentation technique de Xerox, $39\,328$ mots anglais, $46\,828$ mots français), VERNE ("De la terre à la lune" de Jules Verne, 40161 mots anglais, $53\,181$ mots français).

JOC : $200\,000$ mots par langue du journal officiel de la communauté européenne (français/anglais). Série de questions/réponses sur différents problèmes concernant la communauté européenne. Année 93.

Note: *D'autre corpus sont bien sûr disponibles, notamment, la bible et le coran qui sont traduits dans de nombreuses langues. Pas nécessairement de référence annotée à la main.*

Arcade: Métriques d'évaluation

Basées sur les taux de précision et rappel, mais à différents degrés de granularité.

Soit $S = \{s_1, s_2, \ldots, s_n\}$ et $T = \{t_1, t_2, \ldots, t_m\}$ sa traduction. Un alignement A entre S et T est un sous-ensemble du produit cartésien $\wp(S) \times \wp(T)$, où $\wp(S)$ et $\wp(T)$ sont respectivement l'ensemble de tous les sous-ensembles de S et T. On appelle le triplet $\langle S, T, A \rangle$ un *bitexte*.

Voici un bitexte de référence que nous utilisons par la suite:

s_1 Ceci est la phrase numéro un.	t_1 This is the first sentence.
s_2 Ceci est la phrase numéro deux qui ressemble à la première	t_2 This is the second sentence. t_3 It looks like the first.

La représentation de cet alignement est :

$$A_r = \{(\{s_1\}, \{t_1\}), (\{s_2\}, \{t_2, t_3\})\}$$

Arcade: Métriques d'évaluation

Soit le bitexte de référence $\langle S, T, A_r \rangle$ et un alignement candidat A.

$$\text{rappel} = |A \cap A_r|/|A_r| \quad \text{précision} = |A \cap A_r|/|A|$$

(silence = 1-rappel, bruit=1-précision)

s_1 Ceci est la phrase numéro un.	t_1 This is the first sentence.
	t_2 This is the second sentence.
s_2 Ceci est la phrase numéro deux, qui ressemble à la première.	t_3 It looks like the first.

$$\begin{aligned} A &= \{(\{s_1\},\{t_1\}),(\{\},\{t_2\}),(\{s_2\},\{t_3\})\} \\ A_r &= \{(\{s_1\},\{t_1\}),(\{s_2\},\{t_2,t_3\})\} \end{aligned}$$

Donc: $A \cap A_r = \{(\{s_1\},\{t_1\})\} \Rightarrow$ rappel=1/2, précision=1/3 et F-mesure $= 2 \times \frac{p \times r}{p+r}$ $= 0.4$

Arcade: Métriques d'évaluation

Note: *Améliorer la précision et le rappel dans une application donnée représente souvent deux buts antagonistes. Il y a des applications où la précision est préférable; comme par exemple l'extraction de dictionnaires bilingues.*

Note: Les taux de précision et rappel tels que calculés précédemment sont sévères car des *bisegments* (éléments de l'alignement) sont peut-être partiellement corrects. Dans l'exemple précédent, $(\{s_2\}, \{t_3\})$ n'appartient pas à A_r mais t_3 est bel et bien associé à s_2 dans la référence.

Posons $A = \{a_1, a_2, \ldots, a_m\}$ et $A_r = \{ar_1, ar_2, \ldots, ar_n\}$ avec $a_i = (as_i, at_i)$ et $ar_j = (ars_j, art_j)$, alors:

$$\begin{array}{lcl} A' & = & \bigcup_i (as_i \times at_i) \\ A'_r & = & \bigcup_j (ars_j \times art_j) \end{array} \qquad \begin{array}{lcl} \text{rappel} & = & |A' \cap A'_r|/|A'_r| \\ \text{précision} & = & |A' \cap A'_r|/|A'| \end{array}$$

Arcade: Métriques d'évaluation

$$\begin{aligned} A'_r &= \{(s1,t1),(s2,t2),(s2,t3)\} \\ A' &= \{(s1,t1),(s2,t3)\} \end{aligned}$$

précision=1 et rappel=2/3, F-mesure=0.8 (au lieu de 0.4)

$\hookrightarrow$ *métriques MP dans la suite*

Problèmes avec ces métriques calculées au niveau de la phrase:

1. Nécessite une segmentation au niveau de la phrase
2. Tous les algorithmes n'en font pas usage
3. Ne donne pas d'information sur la gravité d'un alignement (une erreur sur une petite phrase est peut-être moins pénalisant dans une application, qu'une erreur commise sur un long passage).

$\Rightarrow$ *mesures des taux de précision et rappel au niveau des caractères (ou des mots)*

Qu'est-ce qu'une phrase ?[5]

$\langle S \rangle$ On disait dans le livre:"Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion" $\langle /S \rangle$

$\langle S \rangle$ On disait dans le livre: $\langle /S \rangle \langle S \rangle$"Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion" $\langle /S \rangle$

$\langle S \rangle$ On disait dans le livre: $\langle /S \rangle$"$\langle S \rangle$ Les serpents boas avalent leur proie tout entière, sans la mâcher. $\langle /S \rangle$ $\langle S \rangle$ Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion $\langle /S \rangle$

$\langle S \rangle$ On disait dans le livre:" $\langle S \rangle$ Les serpents boas avalent leur proie tout entière, sans la mâcher. Ensuite ils ne peuvent plus bouger et ils dorment pendant les six mois de leur digestion $\langle /S \rangle$" $\langle /S \rangle$

[5] Exemple extrait de Véronis and Langlais [2000] p. 372.

Arcade: Métriques d'évaluation

- Sur notre exemple, au niveau des mots:

```
|Ar'| = 5*6 + 11*10 = 140
|A'| = 5*6 + 0*5 + 11*6 = 96
|Ar' ^ A'|  = 96
recall = 96/140 = 0.69
precision = 1
F = 0.82
```

- Sur notre exemple, au niveau des caractères (incluant les espaces)

```
|Ar| = 29*27 + 60*52 = 3903
|A| = 29*27 + 0*28 + 60*24 = 2223
|Ar' ^ A'| = 2223
recall = 2223/3903 = 0.57
precision = 1
F = 0.73
```

Arcade: Performances (par type de corpus)

	MP			MC			MA		
	Pr	Rc	F	Pr	Rc	F	Pr	Rc	F
APA	97.01	81.24	88.43	98.19	88.92	93.33	87.95	91.72	89.80
GSA+	95.24	81.57	87.88	97.18	88.97	92.89	88.78	90.59	89.68
GSA	94.98	81.08	87.48	97.38	88.65	92.81	88.33	90.36	89.34
SFI	94.19	81.66	87.48	95.96	88.55	92.11	88.40	89.23	88.81
LILLA	92.97	78.79	85.29	96.06	86.26	90.90	84.45	86.59	85.51
JACAL	91.66	79.73	85.28	94.06	87.15	90.47	82.26	86.63	84.39
SALIGN	85.42	82.98	84.18	93.12	89.26	91.15	88.85	85.35	87.06
GC	90.60	77.39	83.47	92.82	84.65	88.55	83.96	85.29	84.62
ISSCO	88.95	76.33	82.16	91.00	82.25	86.41	83.31	84.19	83.75
CEA	92.56	71.08	80.41	98.10	83.16	90.01	77.91	82.50	80.14
LORIA	86.41	72.65	78.94	90.80	80.88	85.56	80.26	82.27	81.25
IRMC	81.22	65.74	72.67	83.38	73.38	78.06	70.86	76.85	73.74

Corpus BAF: campagne 1998

$MP = métrique\ Phrase,\ MC = métrique\ Char,\ MA = métrique\ Alignement$

Arcade: Performances (par type de corpus)

	MP			MC			MA		
	Pr	Rc	F	Pr	Rc	F	Pr	Rc	F
GSA+	99.08	98.83	98.95	99.58	99.41	99.49	98.14	98.40	98.27
APA	99.23	98.64	98.93	99.63	98.93	99.28	97.82	98.41	98.11
LORIA	98.89	98.82	98.86	99.50	99.01	99.26	98.41	98.37	98.39
GSA	98.86	98.71	98.78	99.51	99.34	99.42	97.96	98.14	98.05
SFI	98.33	99.02	98.67	99.03	99.26	99.14	98.41	98.20	98.31
JACAL	97.93	98.67	98.30	98.78	99.36	99.07	97.91	97.54	97.72
GC	96.94	97.17	97.06	98.10	97.97	98.03	96.52	96.33	96.43
LILLA	97.19	96.93	97.06	98.39	96.85	97.61	95.51	95.66	95.59
IRMC	97.10	96.92	97.01	98.37	98.31	98.34	93.84	94.90	94.37
SALIGN	94.73	99.20	96.92	97.82	99.51	98.66	96.99	95.04	96.01
ISSCO	95.02	95.80	95.41	96.82	96.67	96.74	95.74	95.06	95.40
CEA	96.58	91.05	93.74	98.38	93.18	95.71	88.43	90.45	89.43

Corpus JOC: campagne 1998

$MP = métrique\ Phrase,\ MC = métrique\ Char,\ MA = métrique\ Alignement$

Modèle de traduction probabiliste IBM1

Input: un *bitexte*, cad un corpus constitué de deux textes alignés au niveau des phrases. Le découpage en "mots" est également connu.

Output: des probabilités de transfert (lexique bilingue probabilisé):

the	(3/149)	(le,0.18) (la,0.15) (de,0.12)
minister	(2/27)	(ministre,0.8) (le,0.12)
people	(3/66)	(gens,0.25) (les,0.16) (personnes,0.1)
years	(3/24)	(ans,0.38) (années,0.31) (depuis,0.12)

Détail: Pour diminuer les erreurs potentielles de l'étape d'alignement des phrases, on élimine toutes les paires qui divergent beaucoup selon un critère donné (ex: la longueur en caractère des segments alignés). $\hookrightarrow$ *prix à payer: le corpus d'entraînement diminue.*

Modèle de traduction probabiliste IBM1

- **Idée:** toute paire de mots (source,cible) rencontrée dans le corpus d'entraînement est un paramètre du modèle (cad qu'on associe une probabilité à cette paire).
- Pour chaque mot source s nous avons une contrainte stochastique:

$$\forall s, \ \sum_t p(t|s) = 1$$

- On fait un choix initial de l'ensemble des paramètres du modèle (cohérent avec les contraintes stochastiques). En fait ce choix n'est pas critique car IBM1 converge vers un maximum global.
- On applique un algorithme de ré-estimation des paramètres en appliquant la recette EM afin d'améliorer la vraisemblance du corpus d'entraînement.

Modèle de traduction probabiliste IBM1

Exemple:

Soit 100 tranches du Hansard (le corpus des débats parlementaires canadiens). Une fois alignées (puis filtrages effectués), cela constitue un corpus (bitexte) de 174279 paires de phrases (de moins de 40 mots).

$\hookrightarrow$ On élimine les mots de fréquence 1 (-36% des formes), ce qui donne un vocabulaire de 21096 types anglais et 27182 types français. *Ceci nous fait un nombre potentiel de paramètres de* $573\,431\,472$ *!*

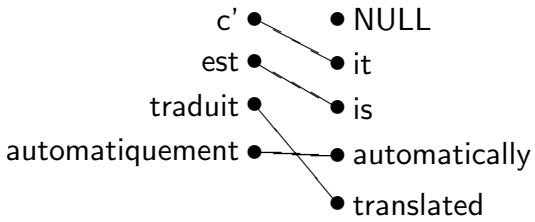
$\hookrightarrow$ En pratique tous les mots ne co-occurrent pas dans les paires de phrases et pour ce modèle on a véritablement $8\,356\,422$ paramètres (déjà pas si mal...)

À titre indicatif, le nombre de paramètres d'un modèle entraîné sur les 5 premières tranches de ce corpus est de $778\,653$

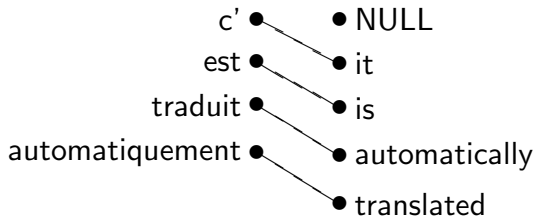
Modèle de traduction probabiliste IBM1

Notation: On cherche à modéliser $P(F = f|E = e)$ où E est l'ensemble des phrases de langue $\mathcal{E}$ (une variable aléatoire), F est l'ensemble des phrases de langue $\mathcal{F}$ (une autre variable aléatoire). $e = e_1, \ldots e_l$ est une phrase particulière de E et $f = f_1, \ldots, f_m$ une phrase particulière de F.

A est l'ensemble des alignements liant une phrase source donnée à une phrase cible (c'est également une variable aléatoire). e est étendue avec un mot e_0 (NULL) auquel seront associés les mots de f sans "traduction". Dans les modèles IBM, seuls les alignements où chaque mot cible est associé à 1 mot source (et un seul) sont considérés.

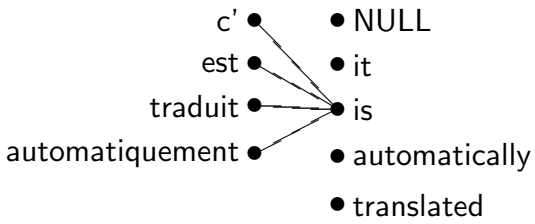


$a = (1, 2, 4, 3)$

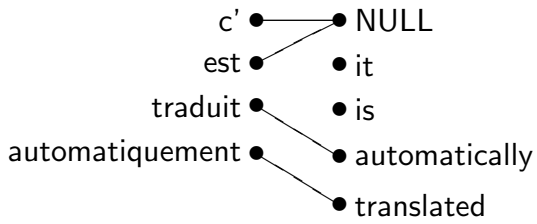


$a = (1, 2, 3, 4)$

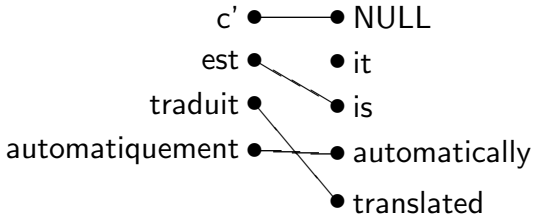
Modèle de traduction probabiliste IBM1



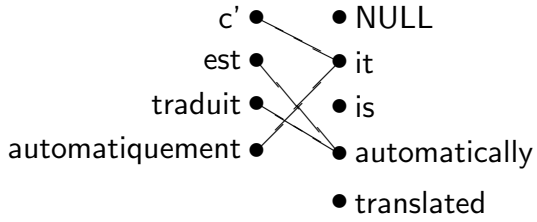
$a = (2, 2, 2, 2)$



$a = (0, 0, 3, 4)$



$a = (0, 2, 4, 3)$



$a = (1, 3, 3, 1)$

Tous ces alignements sont "valides". Soit $\mathcal{A}(e, f)$ l'ensemble des alignements valides entre e et f.

Modèle de traduction probabiliste IBM 1 & 2

$$P(F = f|E = e) = \sum_{a \in \mathcal{A}(e,f)} P(F = f, A = a|E = e)$$

$$= \sum_a p(f, a|e)$$

$$a = (a_1, \ldots, a_m) \text{ avec } a_i \in [0, l] \; \forall i \in [1, m]$$

$$P(f, a|e) = P(m|e) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, e) P(f_j|a_1^j, f_1^{j-1}, m, e)$$

Note: *Ceci n'est pas une approximation, mais une manière possible de décomposer (chain rule) la distribution initiale.*

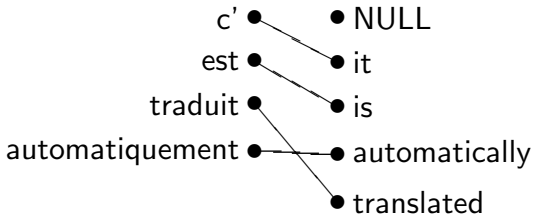
Modèle de traduction probabiliste IBM 1 & 2

Il y a une petite histoire générative associée à cette équation:

1. Choisir la longueur de la phrase f que l'on cherche à générer (selon le modèle $P(m|e)$)
2. Générer individuellement (hypothèse d'indépendance) chaque mot de f; soit f_j le mot considéré:
   (a) choisir une position ($\in [0, l]$) dans e associée à f_j (selon la distribution $P(a_j|a_1^{j-1}, f_1^{j-1}, m, e)$. e_{a_j} est le mot qui est "responsable" de la génération du j-ème mot de f.
   (b) choisir alors un mot français sachant cette position, et toutes les autres informations.

Modèle de traduction probabiliste IBM 1 & 2

Exemple: $P(\text{c'est traduit automatiquement}, a|\text{its is automatically translated})$, où a est donné par $a = (1, 2, 4, 3)$:



$P(m = 4|e = \text{its is automatically translated}) \times$

$P_a(a_1 = 1|m = 4, e) \times P_t(f_1 = \text{c'}|a_1 = 1, m = 4, e) \times$

$P_a(a_2 = 2|a_1 = 1, f_1 = \text{c'}, m = 4, e) \times P_t(f_2 = \text{est}|a_1^2 = (1, 2), f_1 = \text{c'}, m = 4,$

$P_a(a_3 = 4|a_1^2 = (1, 2), f_1^2 = \text{c'est}, m = 4, e) \times$
$P_t(f_3 = \text{traduit}|a_1^3 = (1, 2, 4), f_1^2 = \text{c'est}, m = 4, e)$

$P_a(a_4 = 3|a_1^3 = (1, 2, 4), f_1^3 = \text{c'est traduit}, m = 4, e) \times$
$P_t(f_4 = \text{automatiquement}|a_1^4 = (1, 2, 4, 3), f_1^3 = \text{c'est automatiquement}, m = 4, e)$

Modèle de traduction probabiliste IBM 1 & 2

- **Note:** Pour avoir la probabilité:
  P(c'est traduit automatiquement | it is automatically translated)
  il faut sommer sur tous les alignements possibles entre e et f.
- Pour une longueur de f donnée (m), il y a $(l+1)^m$ alignements à considérer... soit pour la traduction d'une phrase e de $l = 10$ mots, et pour une longueur de phrase f de $m = 12$, la bagatelle de $3.12\ 10^{12}$ sommations de calculs comme celui que l'on vient de voir...
- Le problème en pratique est même plus sévère car on ne connait pas la longueur m de la traduction.
- Dans le cas des modèles IBM1 et IBM2, nous verrons que ce calcul peut être effectué de manière efficace. Pour les autres modèles (3, 4 & 5), une approximation est effectuée.

Approximation IBM 1

On simplifie l'équation avec les hypothèses suivantes:

$$P(m|e) = \epsilon$$

$$P(a_j|a_1^{j-1}, f_1^{j-1}, m, e) = p(a_j|l) = 1/(l+1)$$

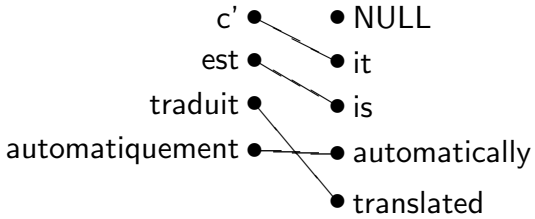
$$P(f_j|a_1^{j}, f_1^{j-1}, m, e) = p(f_j|e_{a_j}) = t(f_j|e_{a_j})$$

Le modèle devient donc:

$$\begin{aligned} p(f,a|e) &= \epsilon \times \prod_{j=1}^{m} \frac{t(f_j|e_{a_j})}{(l+1)} \\ &= \frac{\epsilon}{(l+1)^m} \times \prod_{j=1}^{m} t(f_j|e_{a_j}) \end{aligned}$$

Modèle de traduction probabiliste IBM1

Reprenons notre exemple: où $l = m = 4$ et $a(1, 2, 4, 3)$:



Et soient les probabilités de transfert suivantes: $t(\text{c'}|\text{it}) = 0.2$, $t(\text{est}|\text{is}) = 0.7$, $t(\text{traduit}|\text{translated}) = 0.45$, $t(\text{automatiquement}|\text{automatically}) = 0.8$

alors:

$p(\text{c'est traduit automatiquement}, a|\text{its is automatically translated}) =$

$$\frac{\epsilon \times 0.2 \times 0.7 \times 0.45 \times 0.8}{625} = \epsilon \times 8.064\ 10^{-5}$$

Entraînement d'un modèle IBM1

Si l'on somme sur tous les alignements possibles, alors la probabilité d'une phrase f sachant une phrase e est:

$$p(f|e) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j})$$

Nous cherchons à estimer les probabilités de transfert $t(f_j|e_k)$ de manière à augmenter la vraisemblance d'un corpus d'entraînement sous les contraintes:

$$\sum_{\mathrm{f}} t(\mathrm{f}|\mathrm{e}) = 1, \forall \mathrm{e}$$

Note: Ne pas confondre f et f. f désigne la phrase de m mots, f désigne un mot (n'importe lequel) de la langue $\mathcal{F}$. Idem pour e et e.

Exemple de distribution de transfert

Soit un mot e=is et les paramètres associés:

is (20686 paramètres)

```
(est,0.44) (c',0.14) (.,0.054) (il,0.034)
(ce,0.03) (le,0.03) (que,0.027) (,,0.026)
(de,0.019) (s',0.018) (la,0.016) (qui,0.011)
(qu',0.011) (une,0.011) (agit,0.0099) (à,0.0087)
(l',0.0086) (voilà,0.008) (n',0.0074) (se,0.0065)
(fait,0.0062) (un,0.006)
...
(larrivée,1.4e-45) (marguerite,1.4e-45)
```

La somme de toutes ces probabilités est (ou devraît être !) 1: $\sum_{\mathrm{f}} p(\mathrm{f}|is) = 1$

Exemple de distribution de transfert

Soit un mot e=overestimated et les paramètres associés:

overestimated (39 paramètres):

```
(rayonnement,0.08) (sous-estimer,0.08) (commerciales,0.08)
(activités,0.08) (affirmation,0.08) (prévoir,0.08) (avez,0.08)
(nature,0.08) (apporter,0.079) (aide,0.078) (vous,0.077)
(UNK,0.075) (besoins,0.037) (on,0.007) (faut,0.0035)
(peut,0.00029) (une,0.00027) (pour,1e-04) (il,5.3e-05)
(les,1.3e-05) (sous-estimé,7.1e-06) (spectre,4e-06)
(l',6e-07) (du,2.1e-07) (des,5.3e-08) (,,4.7e-08)
(en,3.5e-08) (la,3.2e-08) (ne,2.7e-08) (importance,4.5e-09)
(si,2.3e-09) (de,2.2e-09) (culture,1.4e-09) (.,8.5e-10)
(souveraineté,7.2e-10) (ou,7e-10) (mais,6.3e-10)
(le,3.7e-10) (et,1e-10)
```

Entraînement d'un modèle IBM1

IBM1 possède la propriété que l'on peut calculer $p(f|e)$ de manière exacte et efficace:

$$\begin{aligned} p(f|e) &= \sum_a p(f,a|e) \\ &= \sum_a \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\ &= \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\ &= \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) \end{aligned}$$

Une lecture possible: chaque mot cible est généré en consultant les probabilités de transfert de chaque mot source vers ce mot cible

Entraînement d'un modèle IBM1

La dernière égalité vient du fait que:

$$
\begin{aligned}
&\sum_{a_1=0}^{l} \cdots \sum_{a_{m-1}=0}^{l} \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) && = \\
&\sum_{a_1=0}^{l} \cdots \sum_{a_{m-1}=0}^{l} \sum_{a_m=0}^{l} \prod_{j=1}^{m-1} t(f_j|e_{a_j}) \times t(f_m|e_{a_m}) && = \\
&\sum_{a_1=0}^{l} \cdots \sum_{a_{m-1}=0}^{l} \prod_{j=1}^{m-1} t(f_j|e_{a_j}) \sum_{a_m=0}^{l} t(f_m|e_{a_m}) && = \\
&\sum_{a_1=0}^{l} \cdots \sum_{a_{m-1}=0}^{l} \prod_{j=1}^{m-1} t(f_j|e_{a_j}) \sum_{i=0}^{l} t(f_m|e_i) && = \\
&\sum_{a_1=0}^{l} \cdots \sum_{a_{m-1}=0}^{l} \prod_{j=1}^{m-2} t(f_j|e_{a_j}) t(f_{m-1}|e_{a_{m-1}}) \sum_{i=0}^{l} t(f_m|e_i) && = \\
&\sum_{a_1=0}^{l} \cdots \prod_{j=1}^{m-2} t(f_j|e_{a_j}) \sum_{a_{m-1}=0}^{l} t(f_{m-1}|e_{a_{m-1}}) \sum_{i=0}^{l} t(f_m|e_i) && = \\
&\sum_{a_1=0}^{l} \cdots \prod_{j=1}^{m-2} t(f_j|e_{a_j}) \sum_{i=0}^{l} t(f_{m-1}|e_i) \sum_{i=0}^{l} t(f_m|e_i) && = \\
&\vdots \\
&\sum_{i=0}^{l} t(f_1|e_i) \ldots \sum_{i=0}^{l} t(f_{m-1}|e_i) \sum_{i=0}^{l} t(f_m|e_i) && = \\
&\prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i)
\end{aligned}
$$

Entraînement d'un modèle IBM1 Brown et al. [1993]

Formules de réestimation:

$$c(\mathsf{f}|\mathsf{e};e,f) = \frac{t(\mathsf{f}|\mathsf{e})}{\sum_{i=0}^{l} t(\mathsf{f}|e_i)} \overbrace{\sum_{j=1}^{m} \delta(f_j,\mathsf{f})}^{\text{nb de f dans } f} \overbrace{\sum_{i=0}^{l} \delta(\mathsf{e},e_i)}^{\text{nb de e dans } e} \quad (1)$$

$$\text{Avec } t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}} \sum_{s=1}^{S} c(\mathsf{f}|\mathsf{e};f^{(s)},e^{(s)})$$

où $\delta(a,b)$ est la fonction de Kroneker qui vaut 1 lorsque $a=b$ et 0 sinon

λ_e est un facteur de normalisation qui assure la contrainte $\sum_{\mathsf{f}} t(\mathsf{f}|\mathsf{e}) = 1,\ \forall \mathsf{e}$

Note: Converge vers un maximum global

Entraînement d'un modèle IBM1

Démonstration (vous pouvez aller à l'acétate)

Rappel:

$$\begin{aligned} p(f|e) &= \sum_a p(f,a|e) \\ &= \sum_a \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \\ &= \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \end{aligned}$$

$\hookrightarrow$ *On cherche à optimiser les paramètres* $t(e,f)$ *sous les contraintes stochastiques* $\sum_f t(f|e) = 1 \ \forall e$

$\hookrightarrow$ *On introduit pour cela des coefficients de lagrange* (λ_e) *et la fonction que l'on souhaite maximiser est:*

$$\psi(t,\lambda) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) - \sum_e \lambda_e (\sum_f t(f|e) - 1)$$

Entraînement d'un modèle IBM1

Démonstration (vous pouvez aller à l'acétate)

$\hookrightarrow$ *La dérivation de* $\psi(t, \lambda)$ *par rapport à* λ_e *est juste un rappel de la contrainte stochastique sur e:*

$$\frac{\partial}{\partial \lambda_{\mathsf{e}}} \psi(t, \lambda) = 0 \Rightarrow \sum_f t(f|\mathsf{e}) = 1$$

$\hookrightarrow$ *La dérivation de* $\psi(t, \lambda)$ *par rapport à* $t(e|f)$ *nous permet d'exprimer* $t(e|f)$ *en fonction des comptes (estimés):*

$$c(\mathsf{f}|\mathsf{e}; f, e) = \sum_a p(a|e, f) \underbrace{\sum_{j=1}^{m} \delta(f_j, \mathsf{f}) \delta(e_{a_j}, \mathsf{e})}_{\text{nb de fois où f et e sont connectés dans a}}$$

Entraînement d'un modèle IBM1

$$\frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})}\psi(t,\lambda) = 0$$

$$\frac{\epsilon}{(l+1)^m}\sum_a \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} \prod_{\substack{j=1\\ f_j\neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \prod_{\substack{j=1\\ f_j=\mathsf{f}}}^{m} t(\mathsf{f}|e_{a_j}) - \lambda_{\mathsf{e}} = 0$$

$$\frac{\epsilon}{(l+1)^m}\sum_a \prod_{\substack{j=1\\ f_j\neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} \prod_{\substack{j=1\\ f_j=\mathsf{f}}}^{m} t(\mathsf{f}|e_{a_j}) = \lambda_{\mathsf{e}}$$

$$\frac{\epsilon}{(l+1)^m}\sum_a \prod_{\substack{j=1\\ f_j\neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \prod_{\substack{j=1\\ f_j=\mathsf{f},e_{a_j}\neq \mathsf{e}}}^{m} t(\mathsf{f}|e_{a_j}) \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} \prod_{\substack{j=1\\ f_j=\mathsf{f},e_{a_j}=e}}^{m} t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}}$$

$$\frac{\epsilon}{(l+1)^m}\sum_a \prod_{\substack{j=1\\ f_j\neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \prod_{\substack{j=1\\ f_j=\mathsf{f},e_{a_j}\neq \mathsf{e}}}^{m} t(\mathsf{f}|e_{a_j}) \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} t(\mathsf{f}|\mathsf{e})^{n_{ef}} = \lambda_{\mathsf{e}}$$

Entraînement d'un modèle IBM1

où n_{ef} est le nombre de fois où e et f sont connectés via un alignement donné, alors:

$$\frac{\epsilon}{(l+1)^m} \sum_a \prod_{\substack{j=1 \\ f_j \neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \prod_{\substack{j=1 \\ f_j = \mathsf{f}, e_{a_j} \neq e}}^{m} t(\mathsf{f}|e_{a_j})\, n_{ef} \frac{t(\mathsf{f}|\mathsf{e})^{n_{ef}}}{t(\mathsf{f}|\mathsf{e})} = \lambda_{\mathsf{e}}$$

Donc:

$$\frac{\epsilon}{(l+1)^m} \sum_a \underbrace{\prod_{\substack{j=1 \\ f_j \neq \mathsf{f}}}^{m} t(f_j|e_{a_j}) \prod_{\substack{j=1 \\ f_j = \mathsf{f}, e_{a_j} \neq e}}^{m} t(\mathsf{f}|e_{a_j}) \prod_{\substack{j=1 \\ f_j = f, e_{a_j} = e}}^{m} t(\mathsf{f}|\mathsf{e})}_{A} \frac{n_{ef}}{t(\mathsf{f}|\mathsf{e})} = \lambda_{\mathsf{e}}$$

$$\frac{\epsilon}{(l+1)^m} \sum_a \overbrace{\prod_{j=1}^{m} t(f_j|e_{a_j})}^{A} \frac{n_{ef}}{t(\mathsf{f}|\mathsf{e})} = \lambda_{\mathsf{e}}$$

Entraînement d'un modèle IBM1

Et donc:

$$t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}}^{-1} \frac{\epsilon}{(l+1)^m} \sum_a \prod_{j=1}^{m} t(f_j|e_{a_j}) \overbrace{\sum_{j=1}^{m} \delta(f_j, \mathsf{f})\delta(e_{a_j}, \mathsf{e})}^{n_{ef}}$$

Soit:

$$t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}}^{-1} \sum_a \underbrace{p(f, a|e)}_{p(f|e)\times p(a|e,f)} \sum_{j=1}^{m} \delta(f_j, \mathsf{f})\delta(e_{a_j}, \mathsf{e})$$

Or par définition des comptes estimés, on a:

$$c(\mathsf{f}|\mathsf{e}; f, e) = \sum_a p(a|f, e) \sum_{j=1}^{m} \delta(f_j, \mathsf{f})\delta(e_{a_j}, \mathsf{e}) \tag{2}$$

$$\text{donc } t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}}^{-1} c(\mathsf{f}|\mathsf{e}; f, e) p(f|e) \tag{3}$$

Entraînement d'un modèle IBM1

Bilan: on a en main une solution incalculable

$\hookrightarrow$ *Ce que nous dit cette dernière équation est que nos paramètres* $t(f|e)$ *sont proportionnels aux comptes estimés.*

Mais: pour calculer les comptes estimés (équation 2) il nous faut sommer sur tous les alignements, ce qui en pratique, n'est pas faisable. . .

$\hookrightarrow$ *Recommençons à dériver notre vraisemblance avec la forme compacte (au sens calculatoire) de notre probabilité* $p(f|e)$ *et gardons à l'esprit la relation que nous donne l'équation 3*

Note: il y a certainement plus simple comme démonstration, mais je reprends ici (en réduisant les pas entre chaque étape) la preuve donnée par Brown et al. [1993]

Play it again Sam

$$\frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})}\left(\frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l}t(f_j|e_i)-\sum_{e}\lambda_{\mathsf{e}}\left(\sum_{f}t(f|e)-1\right)\right)=0$$

$$\frac{\epsilon}{(l+1)^m}\frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})}\left[\prod_{\substack{j=1\\ f_j\neq\mathsf{f}}}^{m}\sum_{i=0}^{l}t(f_j|e_i)\prod_{\substack{j=1\\ f_j=\mathsf{f}}}^{m}\sum_{i=0}^{l}t(f_j|e_i)\right]-\lambda_{\mathsf{e}}=0$$

$$\underbrace{\frac{\epsilon}{(l+1)^m}\prod_{\substack{j=1\\ f_j\neq\mathsf{f}}}^{m}\sum_{i=0}^{l}t(f_j|e_i)}_{B}\underbrace{\frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})}\prod_{\substack{j=1\\ f_j=\mathsf{f}}}^{m}\sum_{i=0}^{l}t(\mathsf{f}|e_i)}_{A}=\lambda_{\mathsf{e}}$$

$$\text{Alors } A = \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} \prod_{\substack{j=1 \\ f_j=\mathsf{f}}}^{m} \left[\sum_{i=0}^{l} \delta(e_i, \mathsf{e}) t(\mathsf{f}|\mathsf{e}) + \sum_{i=0}^{l} \bar{\delta}(e_i, \mathsf{e}) t(\mathsf{f}|e_i) \right]$$

Entraînement d'un modèle IBM1

Si on pose n_f le nombre de fois où f apparaît dans f, et n_e le nombre de fois où e apparaît dans e alors:

$$A = \frac{\partial}{\partial t(\mathsf{f}|\mathsf{e})} \left[n_e t(\mathsf{f}|\mathsf{e}) + \sum_{i=0}^{l} \bar{\delta}(e_i, \mathsf{e}) t(\mathsf{f}|e_i) \right]^{n_f}$$

Donc:

$$A = \left[n_e t(\mathsf{f}|\mathsf{e}) + \sum_{i=0}^{l} \bar{\delta}(e_i, \mathsf{e}) t(\mathsf{f}|e_i) \right]^{n_f} \frac{n_f n_e}{n_e t(\mathsf{f}|\mathsf{e}) + \sum_{i=0}^{l} \bar{\delta}(e_i, \mathsf{e}) t(\mathsf{f}|e_i)}$$

cad:

$$A = \prod_{\substack{j=1 \\ f_j = \mathsf{f}}}^{n} \sum_{i=0}^{l} t(\mathsf{f}|e_i) \frac{n_f n_e}{\sum_{i=0}^{l} t(\mathsf{f}|e_i)}$$

Entraînement d'un modèle IBM1

Revenons à notre calcul de dérivée:

$$\lambda_{\mathsf{e}}^{-1}\frac{\epsilon}{(l+1)^m}\prod_{\substack{j=1\\ f_j\neq \mathsf{f}}}^{m}\sum_{i=0}^{l}t(f_j|e_i)\prod_{\substack{j=1\\ f_j=\mathsf{f}}}^{n}\sum_{i=0}^{l}t(\mathsf{f}|e_i)\frac{n_f n_e}{\sum_{i=0}^{l}t(\mathsf{f}|e_i)}=1$$

$$\lambda_{\mathsf{e}}^{-1}\frac{\epsilon}{(l+1)^m}\prod_{j=1}^{m}\sum_{i=0}^{l}t(f_j|e_i)n_f n_e=\sum_{i=0}^{l}t(\mathsf{f}|e_i)$$

Donc (finalement):

$$\lambda_{\mathsf{e}}^{-1}p(f|e)n_f n_e=\sum_{i=0}^{l}t(\mathsf{f}|e_i)$$

Or d'après l'équation 3, on sait que $\lambda_{\mathsf{e}}^{-1}p(f|e)=\dfrac{t(\mathsf{f}|\mathsf{e})}{c(\mathsf{f}|\mathsf{e};f,e)}$

Entraînement d'un modèle IBM1

(toutes les choses ont une fin, même les mauvaises :-)

Donc:

$$\frac{t(\mathsf{f}|\mathsf{e})}{\sum_{i=0}^{l} t(\mathsf{f}|e_i)} n_f n_e = c(\mathsf{f}|\mathsf{e}; f, e)$$

C'est-à-dire:

$$\frac{t(\mathsf{f}|\mathsf{e})}{\sum_{i=0}^{l} t(\mathsf{f}|e_i)} \sum_{j=1}^{m} \delta(f_j, \mathsf{f}) \sum_{i=0}^{l} \delta(e_i, \mathsf{e}) = c(\mathsf{f}|\mathsf{e}; f, e)$$

$\hookrightarrow$ *c'est bien l'équation 1 de l'acétate . Je vous avais bien dit de passer directement à l'acétate ...*

Entraînement d'un modèle IBM1: Structures de données

- Les paramètres d'un modèle 1 peuvent être représentés par une grande table de paramètres $t(\text{f}|\text{e})$. S'il y a $nbPairs$ paires de mots différents qui co-occurrent dans le corpus d'entraînement ($\mathcal{T}$), alors le tableau suivant fait l'affaire:

  ```
  float Param[nbPairs];
  ```

- De plus, chaque mot de la langue cible peut potentiellement être associé à e_0 (le mot NULL ajouté au début de chaque phrase e dans $\mathcal{T}$). Soit `nbCibles` le nombre de mots cibles considérés dans le modèle, alors le tableau suivant fait l'affaire:

  ```
  float NullParam[nbCibles];
  ```

- De même posons `nbSources` le nombre de mots sources considérés dans le modèle.

Entraînement d'un modèle IBM1: Structures de données

Pour accéder à ces paramètres, il nous faut également un certain nombre de fonctions et structures.

Correspondance chaînes/entiers: pour rendre le code plus efficace, il est utile de disposer d'un moyen d'associer à une chaîne donnée un indice (entier) et vice-versa. Soient $\mathcal{I}_E(\mathrm{e})$ la fonction qui retourne cet indice (à valeur dans $[1, nbSources]$) pour le mot source e, et $\mathcal{I}_F(\mathrm{f})$ (à valeur dans $[1, nbCibles]$) son homologue pour le mot cible f.

Accès au paramètre $t(\mathbf{f}|\mathbf{e})$: il faut se donner une fonction (efficace !) d'accès à un paramètre donné de `Param` sachant un indice source $\mathcal{I}_E(\mathrm{e})$ et un indice cible $\mathcal{I}_F(\mathrm{f})$: soit $\mathcal{M}$ cette fonction:

$\mathcal{M}$: $(\mathcal{I}_E(\mathrm{e}), \mathcal{I}_F(\mathrm{f})) \rightarrow$ indice dans `Param`

$\hookrightarrow$ *une table de hachage (*`<int,int>` $\rightarrow$ `int`*) fera l'affaire.*

Entraînement d'un modèle IBM1: Structures de données

Accès rapide aux mots cibles associés à un mot source donné: pas indispensable, mais nécessaire si l'efficacité est requise.

Soit $\mathcal{C}(\mathcal{I}_E(\text{e}))$ l'ensemble des indices des mots cibles associés dans le modèle au mot source e; et notons $\mathcal{C}(\mathcal{I}_E(\text{e}))(i)$ le i-ème d'entre-eux.

Lors de l'entraînement, nous avons également besoin de stocker temporairement les comptes des formules de réestimation:

```
float Count[nbPairs];
float NullCount[nbCibles];
```

Notation:
$\mathcal{T} = \langle (f^1, e^1), \ldots, (f^S, e^S) \rangle$ avec: $f^i = f^i_1, \ldots, f^i_m$ avec $m = |f^i|$ et $e^i = e^i_1, \ldots, e^i_l$, avec $l = |e^i|$

Entraînement d'un modèle IBM1

// init des comptes
for all p $\in [1, nbPairs]$ **do**
  Count[p] $\leftarrow$ 0
for all i $\in [1, nbCibles]$ **do**
  NullCount[i] $\leftarrow$ 0

// cumul des comptes (première formule)
for all s $\in [1, S]$ **do**
  for all i $\in [1, |f^s|]$ **do**
    $i_s \leftarrow \mathcal{I}_F(f_i^s)$
    // calcul du denominateur de la formule 1
    sum $\leftarrow$ NullParam[i_s]
    for all j $\in [1, |e^s|]$ **do**
      sum += Param[$\mathcal{M}(\mathcal{I}_E(e_j^s), i_s)$]

    // cumul de la formule 1 (et 2)
    NullCount[i_s] += NullParam[i_s] /sum
    for all j $\in [1, |e^s|]$ **do**
      p $\leftarrow \mathcal{M}(\mathcal{I}_E(e_j^s), i_s)$
      Count[p] += Param[p] / sum

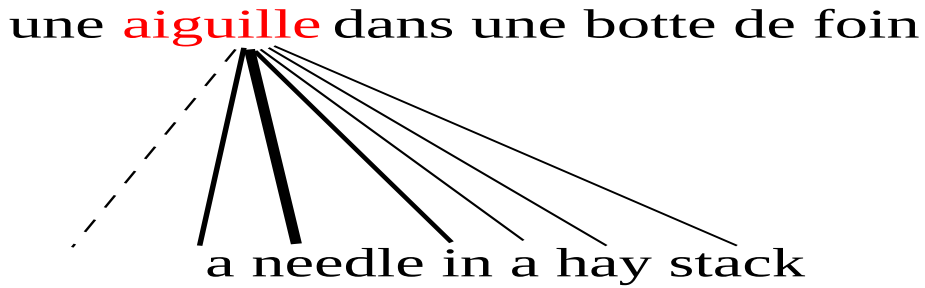
Entraînement d'un modèle IBM1

// normalisation (deuxième formule)
for all i $\in [1, nbSources]$ **do**
  S $\leftarrow$ 0
  for all j $\in \mathcal{C}(i)$ **do**
    S $+=$ Count[$\mathcal{M}(i,j)$]
  for all j $\in \mathcal{C}(i)$ **do**
    p $\leftarrow \mathcal{M}(i,j)$
    Param[p] $\leftarrow$ Count[p] / S
S $\leftarrow$ 0
for all j $\in [1, nbCibles]$ **do**
  S $+=$ NullCount[j]
for all j $\in [1, nbCibles]$ **do**
  NullParam[j] $=$ NullCount[j]/S

$\hookrightarrow$ *On applique cet algorithme un nombre désiré de fois (fixé ou fonction de la convergence).*

Entraînement d'un modèle IBM2

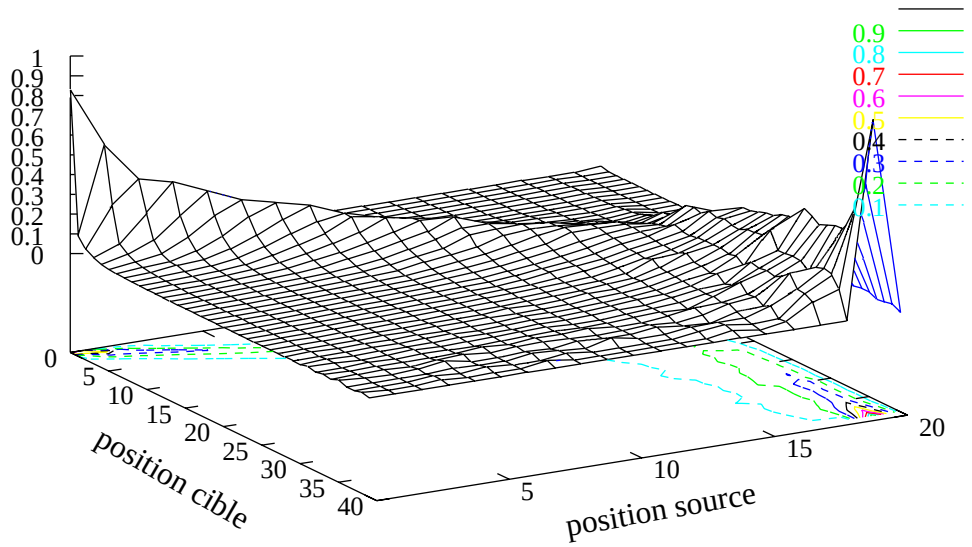
- Dans IBM1, la probabilité d'alignement d'un mot cible en position j avec un mot source en position i est indépendente des positions i et j.
- IBM2 remédie à cette simplification à outrance en introduisant des probabilités d'alignement $a(i|j,l,m)$ qui représentent la probabilité qu'un mot source en position i soit associé à un mot cible en position j, sachant la longueur respective (comptée en mot) des deux phrases considérées.



- Avec la contrainte stochastique: $\forall j,l,m \ \sum_{i=0}^{l} a(i|j,l,m) = 1$

Entraînement d'un modèle IBM2

Distribution des probabilités d'alignement apprise sur un sous-ensemble du Hansard. La distribution montrée est ici: $p(i|j, 20, 40)$.



Note: *les probabilités d'alignement ne dépendent pas des mots*

Approximations du modèle IBM 2

- Rappel de l'histoire générative:

$$P(f, a|e) = P(m|e) \prod_{j=1}^{m} P(a_j|a_1^{j-1}, f_1^{j-1}, m, e) P(f_j|a_1^{j}, f_1^{j-1}, m, e)$$

- On simplifie cette équation avec les hypothèses suivantes:

$$P(m|e) = \epsilon$$

$$P(a_j|a_1^{j-1}, f_1^{j-1}, m, e) = a(a_j|j, l = |e|, m)$$

$$P(f_j|a_1^{j}, f_1^{j-1}, m, e) = p(f_j|e_{a_j}) = t(f_j|e_{a_j})$$

- Le modèle devient donc:

$$p(f, a|e) \quad = \epsilon \times \prod_{j=1}^{m} a(a_j|j, l, m) t(f_j|e_{a_j})$$

Entraînement d'un modèle IBM2

$$p(f|e) = \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} a(a_j|j,l,m) \times t(f_j|e_{a_j})$$

$$= \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} a(i|j,l,m) \times t(f_j|e_i)$$

Les formules de réestimation sont:

$$c(\mathsf{f}|\mathsf{e};e,f) = \sum_{j=1}^{m} \sum_{i=0}^{l} \frac{t(\mathsf{f}|\mathsf{e})a(i|j,l,m)}{\sum_{k=0}^{l} t(\mathsf{f}|e_k)a(k|j,l,m)}$$

$$c(i|j,l,m;f,e) = \frac{t(f_j|e_i)a(i|j,l,m)}{\sum_{k=0}^{l} t(f_j|e_k)a(k|j,l,m)}$$

$$t(\mathsf{f}|\mathsf{e}) = \lambda_{\mathsf{e}} \sum_{1}^{S} c(\mathsf{f}|\mathsf{e};e^s,f^s)$$

$$a(i|j,l,m) = \mu_{j,l,m} \sum_{1}^{S} c(i|j,l,m;f^s,e^s)$$

Entraînement d'un modèle IBM2: codage

On a besoin d'une structure supplémentaire pour stocker $a(i|j,l,m)$. C'est une table coûteuse à coder: si L et M sont les longueurs respectives source et cible les plus grandes dans $\mathcal{T}$, alors une table à 4 dimensions occuperait $(L \times M)^2$ cases (float) en mémoire; soit avec $L = M = 40$; une taille de $10\,240\,000$[6] octets (10 Meg.)

De plus, nous avons besoin d'une table temporaire de comptes pour accumuler les comptes de la seconde formule de réestimation (même taille à priori).

On verra qu'il y a plus efficace que IBM2. On peut cependant coder "efficacement" la table a à l'aide d'une représentation en matrice creuse.

Note: on peut également simplifier cette table en éliminant la taille cible du contexte conditionnant $a(i|j,l)$.

[6] En admettant que la taille d'un float soit 4 octets.

Entraînement d'un modèle IBM2: codage

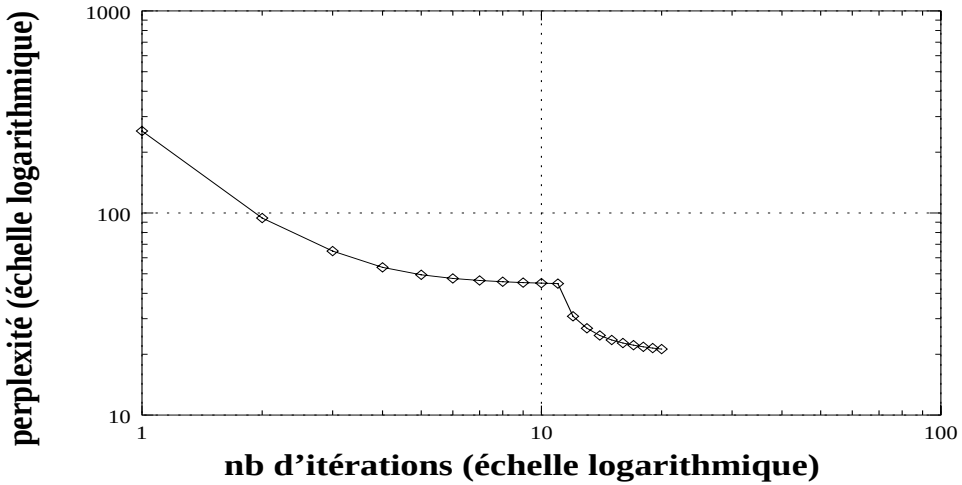
- Le même genre d'algorithme que celui vu pour modèle 1 peut être codé. Les probabilités d'alignement peuvent être initialisées uniformément (avec respect des constraintes).

- Pour les probabilités de transfert, deux possibilités (au moins):

  – les initialiser uniformément (avec respect des contraintes),
  – utiliser les valeurs obtenues après entraînement d'un modèle 1.

  $\hookrightarrow$ *La deuxième solution est préférable car l'entraînement d'un modèle 2 ne converge que vers un optimum local. De meilleures estimées de départ, offrent donc plus de chance de converger vers une solution raisonnable.*

Entraînement d'un modèle IBM2: codage



10 premières itérations: IBM1
10 suivantes: IBM2

Exemple: $t(\text{f}|should) \geq 0.05$ (565 paramètres):

```
IBM1: (devrait,0.32) (qu',0.1) (devraient,0.099)
IBM1 + IBM2: (devrait,0.32) (devraient,0.16)
(qu',0.13) (est,0.092) (faut,0.071) (devrions,0.059)
```

Modèle HMM Vogel et al. [1996]

Idée: IBM2 décide d'un lien sans regard des liens déterminés auparavant. Or il suffit de regarder quelques alignements pour constater que cette hypothèse d'indépendance est fausse.

Eve							x
avec						x	
pomme					x		
une				x			
mange		x	x				
Adam	x						
	Adam	is	eating	an	apple	with	Eve

Modèle HMM Vogel et al. [1996]

On peut renforcer ce modèle en ajoutant le dernier lien choisi dans le contexte conditionnant:

$$P(a_j|a_1^{j-1}, f_1^{j-1}, m, e) = a(a_j|a_{j-1}, j, l = |e|, m)$$

Vogel et al. [1996] proposent de simplifier le modèle en faisant l'hypothèse que ces distributions ne dépendent que du saut fait entre a_{j-1} et a_j:

$$p(i|i', l) = \frac{s(i - i')}{\sum_{k=1}^{l} s(k - i')}$$

où $s(x)$ est un jeu de paramètres positifs (ou nuls).

Note: Lire aussi Och and Ney [2002].

IBM3, 4 & 5: intégration de la fertilité

$\hookrightarrow$ *the* est souvent traduit par un mot (*le, la, l'*), mais peut également ne pas ête traduit.

$\hookrightarrow$ *only* est très souvent traduit par un seul mot (*seulement*), mais de temps en temps par deux mots (*ne . . . que*)

$\hookrightarrow$ *bill* est presque systématiquement traduit par 3 mots (*projet de loi*).

Introduisons une nouvelle variable aléatoire Φ *appelée fertilité qui représente la distribution du nombre de mots cibles générés par un mot source particulier:* $\forall e, \sum_n \phi(n|e) = 1$.

IBM 3, 4 & 5 sont basés sur une autre décomposition que celle proposée en équation et qui intègre directement la notion de fertilité.

IBM3, 4 & 5: une autre histoire générative

L'histoire est simple, mais sa formulation mathématique et les problèmes calculatoires le sont moins ...

1. Choisir la fertilité de chaque mot source e_i selon la distribution $\phi(n|e_i) = \phi_i$.

2. Pour chaque mot source e_i, choisir ϕ_i mots cibles selon les distributions $t(\text{f}|\text{e})$. Brown et al. [1993] appellent la liste des ϕ_i mots générés par e_i la $tablet$ de e_i; notée τ_i. L'ensemble des $tablets$ est appelé le tableau de e.

3. Permuter les mots cibles selon une distribution Π afin d'obtenir la traduction.

IBM3, 4 & 5: une autre histoire générative

On peut voir cette histoire comme une suite de réécritures Knight [1999]:

Mary did not slap the green witch	input
Mary not slap slap slap the the green witch	fertilité $= (1, 0, 1, 3, 2, 1, 1)$
Mary no daba una botefada a la verde bruja	cible
Mary no daba una botefada a la bruja verde	permutation
Mary no daba una botefada a la bruja verde	output

En pratique, l'histoire se complique un peu (au niveau des notations) avec ce que Brown et al. [1993] appellent les *spurious-words*, cad, les mots cibles non associés à un mot source particulier.

IBM3, 4 & 5: une autre histoire générative

$$
\begin{aligned}
P(\tau,\pi|e) \quad &= \\
\text{(step 1)} \qquad & \prod_{i=1}^{l} P(\phi_i|\phi_1^{i-1}) \times P(\phi_0|\phi_1^{l},e) \\
\text{(step 2)} \quad \times \quad & \prod_{i=0}^{l}\prod_{k=1}^{\phi_i} P(\tau_{i,k}|\tau_{i,1}^{k-1},\tau_0^{i-1},\phi_0^{l},e) \\
\text{(step 3)} \quad \times \quad & \prod_{i=1}^{l}\prod_{k=1}^{\phi_i} P(\pi_{i,k}|\pi_{i,1}^{k-1},\pi_1^{i-1},\tau_0^{i-1},\phi_0^{l},e)\times \\
& \qquad \prod_{k=1}^{\phi_0} P(\pi_{0,k}|\pi_{0,1}^{k-1},\pi_1^{l},\tau_0^{l},\phi_0^{l},e)
\end{aligned}
\tag{4}
$$

Avec la notation $\tau_{i,1}^{k} = \tau_{i,1},\tau_{i,2},\ldots,\tau_{i,k}$ et $\pi_{i,1}^{k} = \pi_{i,1},\ldots,\pi_{i,k}$.

Alors:

$$P(f,a|e) = \sum_{(\tau,\pi)\in\langle f,a\rangle} P(\tau,\pi|e)$$

IBM3, 4 & 5: une autre histoire générative

Où $\langle f, a \rangle$ désigne l'ensemble de toutes les paires (τ, π) consistantes avec la paire (f, a).

Connaître τ et π nous donne f (la phrase cible traduction de e) et l'alignement sous-jacent. En général, différentes paires (τ, π) amènent à la même paire f, a (d'où: $P(f, a|e) = \sum_{(\tau,\pi)\in\langle f,a\rangle} P(\tau, \pi|e)$)

Ex:

$$\tau_{cheap} = \{\text{bon, marché}\} \quad \pi_{cheap} = \{1, 2\}$$
$$\tau_{cheap} = \{\text{marché, bon}\} \quad \pi_{cheap} = \{2, 1\}$$

Il y a $\phi_i!$ arrangements possibles des ϕ_i mots cibles de chaque ensemble τ_i, donc

$$|\langle f, a \rangle| = \prod_{i=0}^{l} \phi_i!$$

IBM3

IBM3 est basé sur la décomposition de l'équation 4 avec les simplifications suivantes:

- $P(\phi_i|\phi_1^{i-1}, e)$ (pour $i \in [1, l]$) ne dépend que de $e_i \longrightarrow n(\phi|\text{e})$, les probabilités de fertilité d'un mot source e.
- $P(\tau_{i,k}|\tau_{i,1}^{k-1}, \tau_0^{i-1}, \phi_0^l, e)$ (pour $i \in [0, l]$) ne dépend que de $e_i \longrightarrow t(\text{f}|\text{e})$, les probabilités de transfert.
- $P(\pi_{i,k}|\pi_{i,1}^{k-1}, \pi_1^{i-1}, \tau_0^{i-1}, \phi_0^l, e)$ (pour $i \in [1, l]$) ne dépend que de i, m et $l \longrightarrow d(j|i, m, l)$ les probabilités de distorsion.

Les probabilités de distorsion et de fertilité pour e_0 sont traitées à part. Rappelons que d'après l'équation 4, le nombre de mots cibles associés à e_0 (les $spurious\ words$) est choisi après avoir choisi les autres fertilités ($P(\phi_0|\phi_1^l, e)$) et que le positionnement de ces mots est fait après avoir positionné les autres mots ($\prod_{k=1}^{\phi_0} P(\pi_{0,k}|\pi_{0,1}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e)$)

IBM3

Si l'on suppose que les mots spurious sont placés de manière uniforme dans les espaces vacants, alors $p(\pi_{0,k} = j|\pi_{0,1}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e)$ vaut 0 si j n'est pas une place vacante et $(\phi_0 - k + 1)^{-1}$ sinon.

Dans ce cas, $\prod_{k=1}^{\phi_0} P(\pi_{0,k}|\pi_{0,1}^{k-1}, \pi_1^l, \tau_0^l, \phi_0^l, e) = \prod_{k=1}^{\phi_0} \frac{1}{\phi_0 - k + 1} = \frac{1}{\phi_0!}$

Brown et al. [1993] proposent de modéliser cela par une binomiale dépendant de deux paramètres p_0 et p_1 (sous la contrainte $p_0 + p_1 = 1$):

$$P(\phi_0|\phi_1^l, e) = \binom{\phi = \phi_1 + \ldots + \phi_l}{\phi_0} p_0^{\phi - \phi_0} p_1^{\phi_0}$$

où p_1 représente la probabilité que chaque mot des τ_1^l entraîne l'apparition d'un mot spurious...

IBM3: retour à l'histoire générative

1. pour chaque mot e de la phrase source, on choisi une fertilité ϕ avec une probabilité $n(\phi|\mathrm{e})$
2. m' est la somme de ces fertilités (n'incluant pas celle de e_0)
3. créer une nouvelle phrase source en retirant les mots de fertilité 0, en copiant les mots de fertilité 1, en dupliquant les mots de fertilité 2, etc.
4. après chacun de ces m' mots, décider si l'on insère un mot spurious (avec une probabilité p_1), ou pas (avec une probabilité p_0).
5. Soit ϕ_0 le nombre de mots spurious finalement ajoutés. Alors $m = m' + \phi_0$ est la longueur de la traduction.
6. remplace chaque mot de cette phrase source par un mot cible selon les probabilités de transfert.
7. détermine la position de chaque mot cible non généré par e_0 selon les probabilités de distorsion.
8. si une position cible contient plus d'un mot, alors ERREUR
9. détermine les positions cibles des ϕ_0 mots générés par NULL parmi les positions restées vacantes dans f_1^m; chaque choix étant équiprobable $(1/\phi_0)$.
10. lire la chaîne ainsi créée.

IBM3 en un seul morceau

$$
\begin{aligned}
p(f|e) &= \textstyle\sum_a p(f,a|e) \\
&= \textstyle\sum_a \sum_{(\tau,\pi)\in\langle f,a\rangle} \\
&\quad \textstyle\prod_{i=1}^{l} n(\phi_i|e_i) \times \binom{\phi=\phi_1+\ldots+\phi_l}{\phi_0} p_0^{\phi-\phi_0} p_1^{\phi_0} \times \\
&\quad \textstyle\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{i,k}|e_i) \times \\
&\quad \textstyle\prod_{i=1}^{l} \prod_{k=1}^{\phi_i} d(\pi_{i,k}|i,m,l) \times \frac{1}{\phi_0!} \\
&= \textstyle\sum_a \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \\
&\quad \textstyle\prod_{j=1}^{m} t(f_j|e_{a_j}) \prod_{j:a_j\neq 0}^{m} d(j|a_j,m,l)
\end{aligned}
$$

Note: lorsque (τ,π) est consistant avec (f,a):

$$
\begin{aligned}
p(\tau|\phi,e) &= \textstyle\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{i,k}|e_i) &&= \textstyle\prod_{j=1}^{m} t(f_j|e_{a_j}) \\
p(\pi|\tau,\phi,e) &= \textstyle\frac{1}{\phi_0!} \prod_{i=1}^{l} \prod_{k=1}^{\phi_i} d(\pi_{i,k}|i,l,m) &&= \textstyle\frac{1}{\phi_0!} \prod_{j:a_j\neq 0} d(j|a_j,l,m)
\end{aligned}
$$

IBM3

On a alors un jeu de formules de réestimation Brown et al. [1993]:

$$
\begin{aligned}
c(\mathsf{f}|\mathsf{e};f,e) &= \textstyle\sum_a p(a|e,f) \sum_{j=1}^{m} \delta(\mathsf{f},f_j)\delta(\mathsf{e},e_{a_j}) \\
c(j|i,m,l;f,e) &= \textstyle\sum_a p(a|e,f)\delta(i,a_j) \\
c(\phi|\mathsf{e};f,e) &= \textstyle\sum_a p(a|e,f) \sum_{i=1}^{l} \delta(\phi,\phi_i)\delta(e,e_i) \\
c(0;f,e) &= \textstyle\sum_a p(a|e,f)(m-2\phi_0) \\
c(1;f,e) &= \textstyle\sum_a p(a|e,f)\phi_0
\end{aligned}
$$

$$
\begin{aligned}
t(\mathsf{f}|\mathsf{e}) &\propto \textstyle\sum_{s=1}^{S} c(\mathsf{f}|\mathsf{e};f^s,e^s) \\
d(j|i,m,l) &\propto \textstyle\sum_{s=1}^{S} c(j|i,m,l;f^s,e^s) \\
n(\phi|\mathsf{e}) &\propto \textstyle\sum_{s=1}^{S} c(\phi|\mathsf{e};f^s,e^s) \\
p_k &\propto \textstyle\sum_{s=1}^{S} c(k;f^s,e^s)
\end{aligned}
$$

IBM3

Problème: On ne peut pas calculer efficacement les sommes faisant intervenir l'ensemble des alignements, comme on pouvait le faire pour les modèles IBM1, IBM2 ou HMM.

La solution proposée dans Brown et al. [1993] est de calculer cette somme sur un ensemble d'alignement "prometteurs" ($\mathcal{S}$), en cherchant l'alignement le plus probable qu'on est capable de trouver (pas nécessairement le plus probable) et en considérant tous les alignements obtenus par un ensemble restreint de modifications de ce "meilleur" alignement.

En pratique on part de l'alignement de viterbi obtenu avec le modèle 2 $V(f|e;2)$ et on considère les alignements obtenus à partir de $V(f|e;2)$ en autorisant deux opérations élémentaires: $move$ et $swap$.

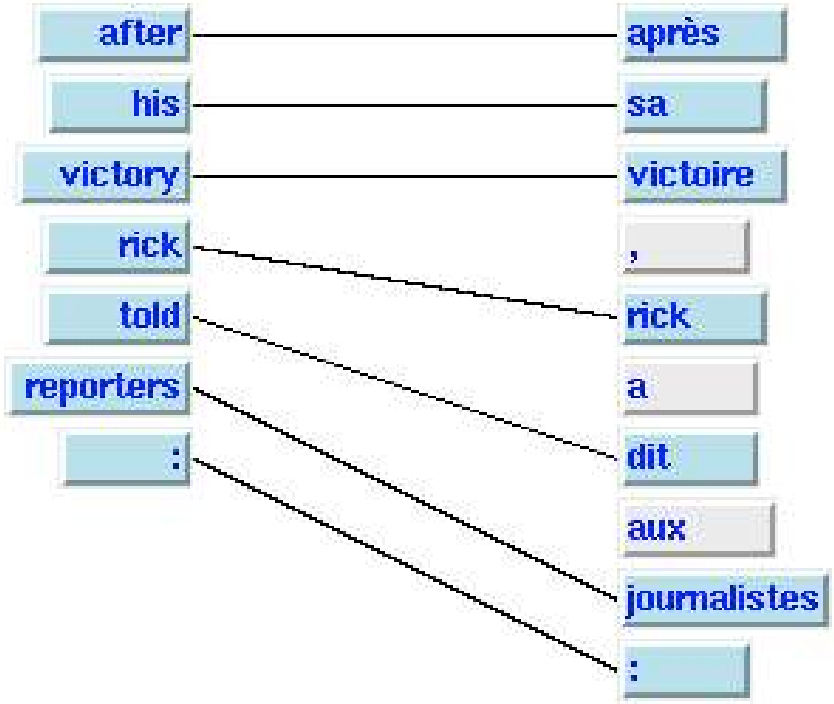
Alignement de viterbi

- L'alignement le plus probable (selon le modèele) d'une paire e, f est appelé l'alignement de viterbi: $V(f|e) = \text{argmax}_a\, p(a|e,f) = \text{argmax}_a\, p(f,a|e)$ (car: $p(a|e,f) = p(a,f|e)/p(f|e)$)
- Pour les modèles IBM 3, 4 & 5, il n'existe pas de moyen de calculer cet alignement efficacement.
- Pour IBM2, on trouve l'alignement de viterbi très facilement ($\mathcal{O}(l \times m)$):

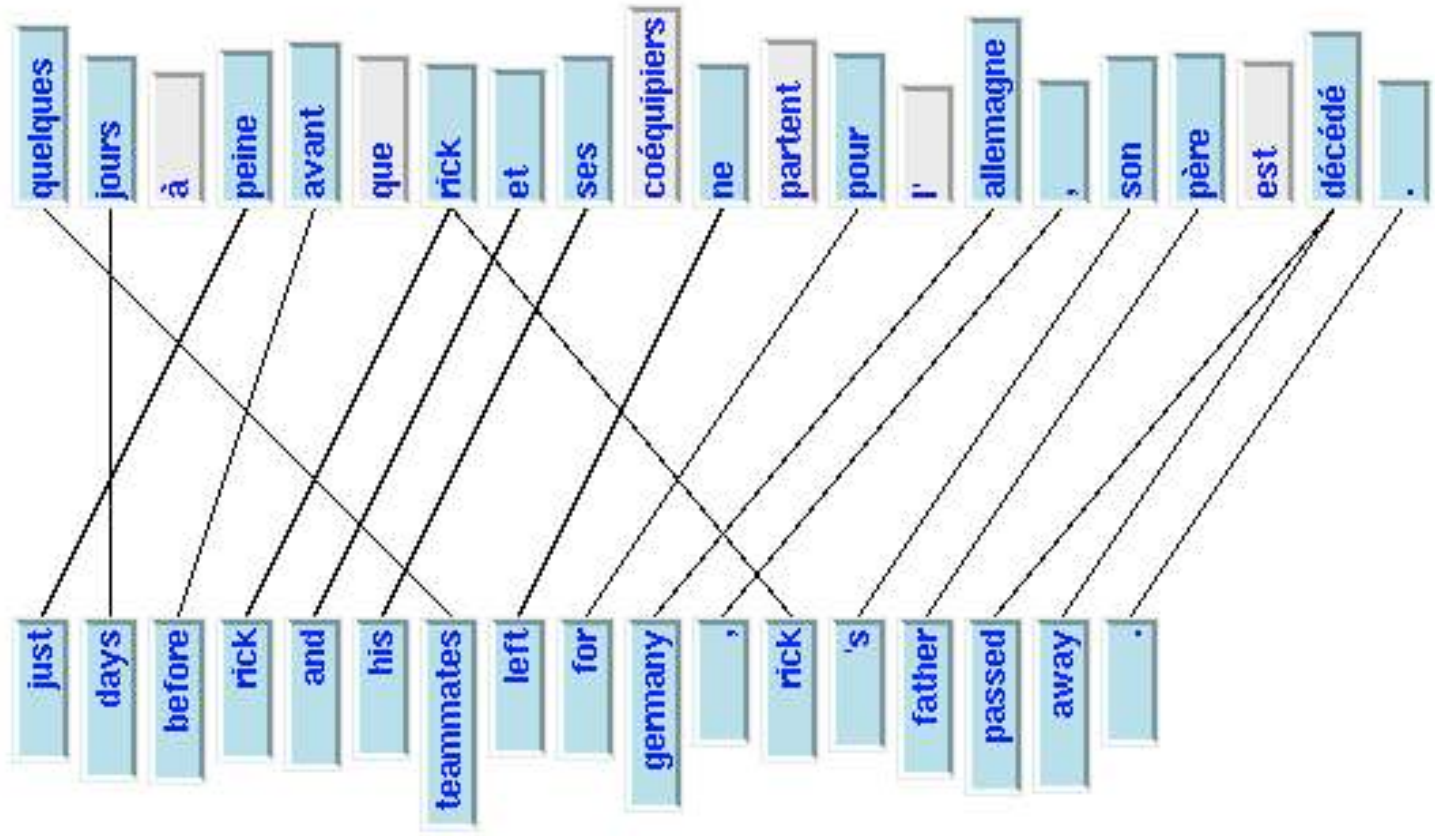
$$\begin{aligned} & \hat{a_1^m} && = \text{argmax}_{a_1^m}\, p(f_1^m, a_1^m|e_1^l) \\ & && = \text{argmax}_{a_1^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) a(a_j|j,m,l) \\ \Rightarrow \quad & \hat{a_j} && = \text{argmax}_{a_j}\, t(f_j|e_{a_j}) a(a_j|j,m,l) \end{aligned}$$

- Dans le cas d'un modèle HMM, l'algorithme de viterbi est utilisé ($\mathcal{O}(l^2 \times m)$)

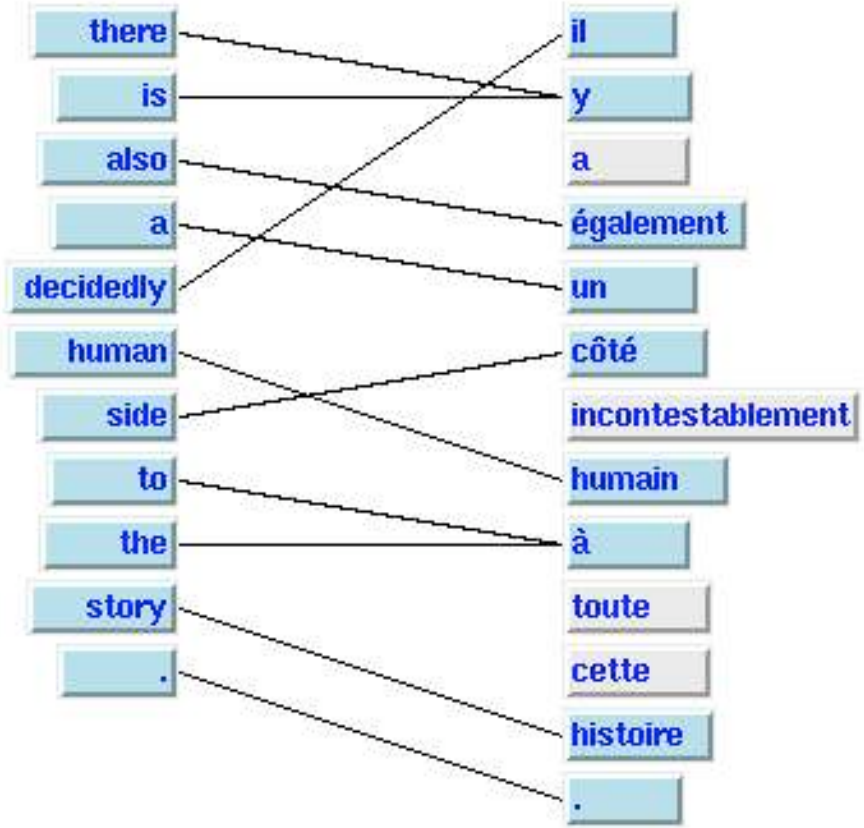
Alignement de viterbi (IBM2)



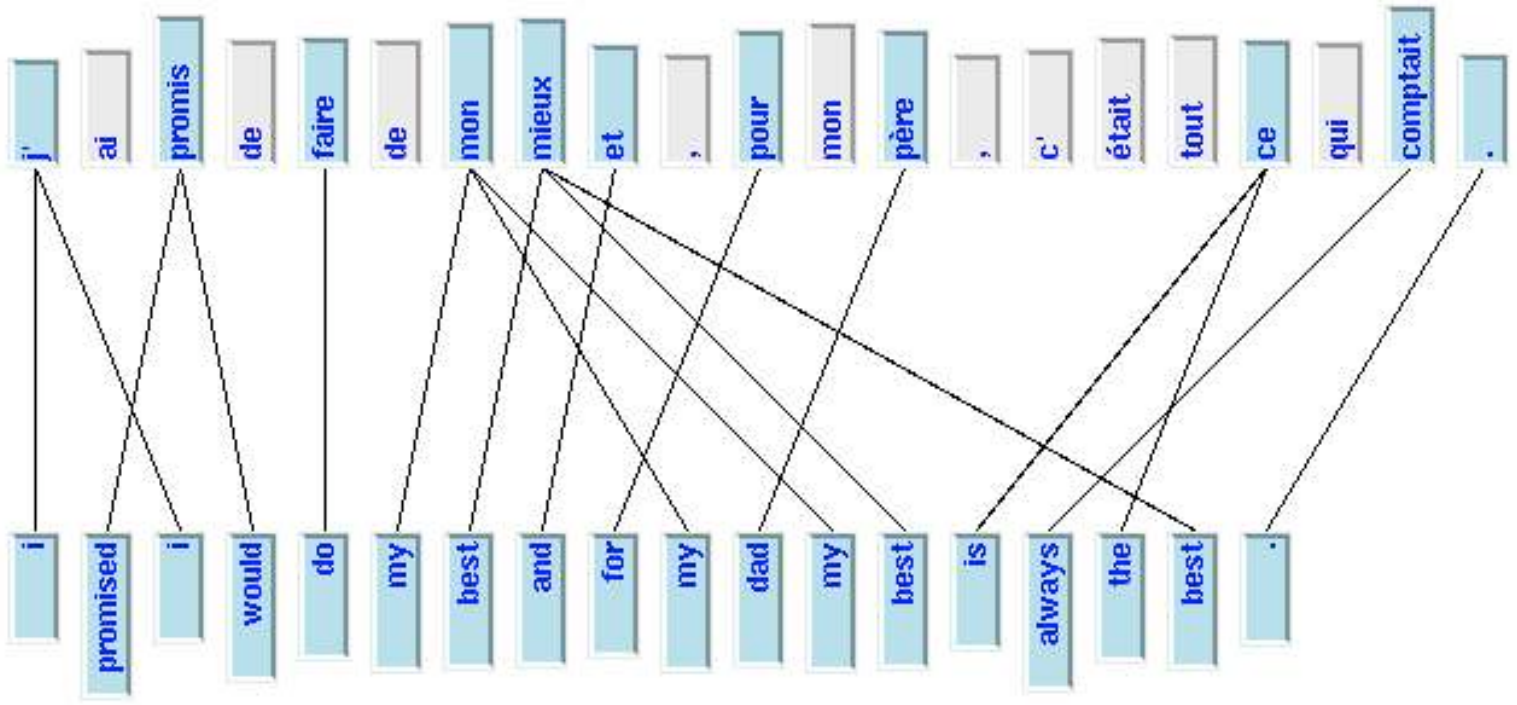
Alignement de viterbi (IBM2)



Alignement de viterbi (IBM2)

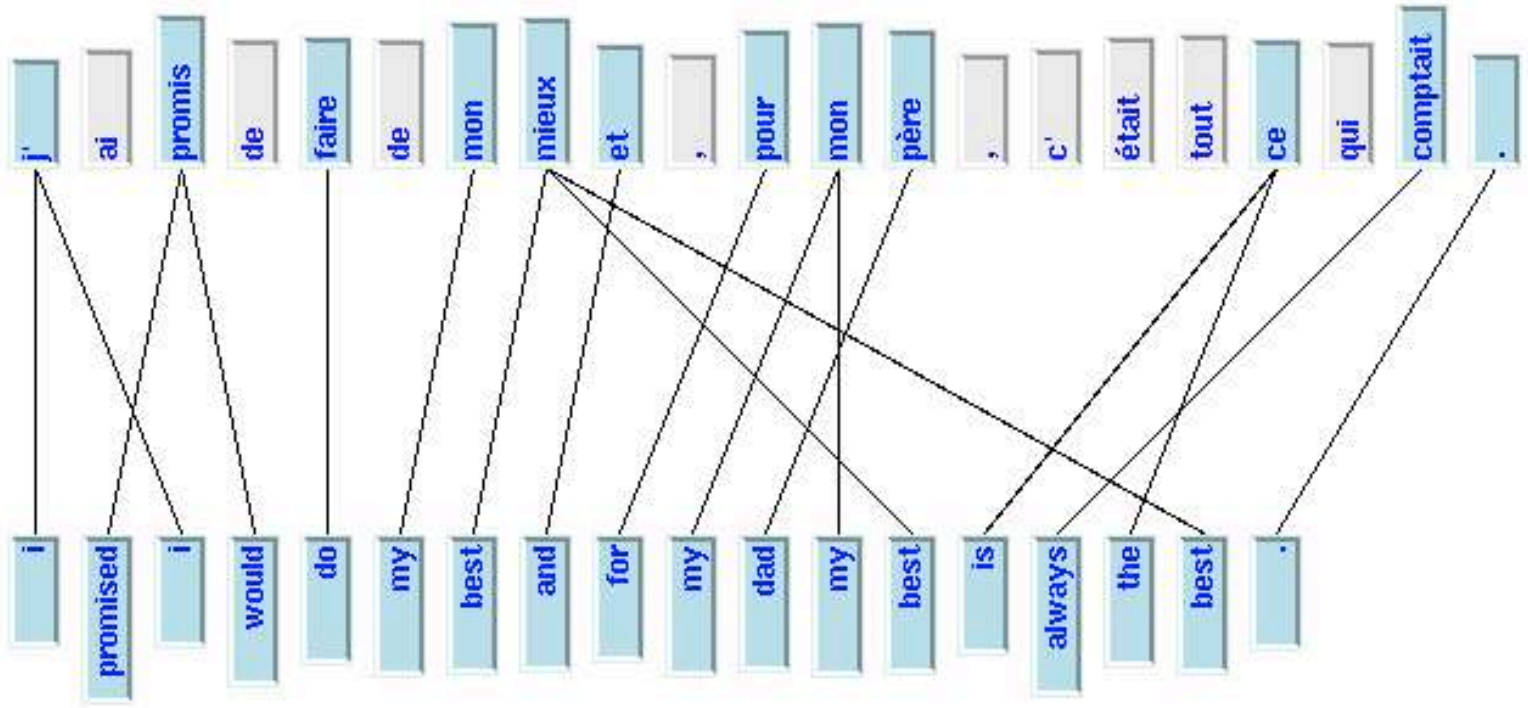


Alignement de Viterbi IBM1



Observez ici que **my** *est toujours associé (par le modèle IBM1) au même mot* **mon** *(c'est assez normal car chaque position source (ici française) est ici équiprobable selon le modèle 1)*

Alignement de viterbi IBM2



Viterbi (approché) dans le cas du modèle 3

- **Point de départ:**

$$p(a, f|e) = \begin{array}{l} \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \\ \prod_{j=1}^{m} t(f_j|e_{a_j}) \prod_{j:a_j \neq 0}^{m} d(j|a_j, m, l) \end{array} \quad (5)$$

  et $p(a, f|e) = p(f|e) \times p(a|f, e)$

- **Idée:** partir de l'alignement de viterbi obtenu (par exemple) avec le modèle 2, puis autoriser des opérations élémentaires qui modifient cet alignement, et qui améliorent la probabilité $p(a|e, f)$ selon le modèle 3. On applique ces opérations tant qu'on réussi à améliorer ce score.

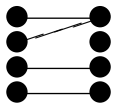
  $\hookrightarrow$ *C'est ce qu'on appelle une technique* ***vorace*** *souvent dénommée* ***hill climbing****.*

Voisinage d'un alignement a

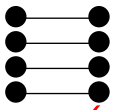
Note: vous pouvez définir votre propre notion de voisinage d'un alignement a (noté $\mathcal{N}(a)$), voici la proposition de Brown et al. [1993]. Pour cela un peu de notation:

move a' diffère de a d'un $move$ si il existe une unique valeur j telle que $a_j \neq a'_j$

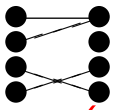
swap a' diffère de a d'un $swap$ si il existe une seule paire de positions cibles j_1 et j_2 pour lesquelles $a_{j_1} = a'_{j_2}$ et $a_{j_2} = a'_{j_1}$



a



move(a)



swap(a)

alors $\mathcal{N}(a) = \{a\} \cup \{move(a)\} \cup \{swap(a)\}$

Voisinage d'un alignement a

- $b(a) = \text{argmax}_{a' \in \mathcal{N}(a)} p(a'|e, f)$, le meilleur alignement voisin de a (selon le modèle - 3 ici); $b(b(a))$ le meilleur alignement voisin du meilleur alignement voisin de a, etc. On note par $b^\infty(a)$ le point de convergence ($b^i(a) = b^{i-1}(a)$).
- $V_{i \leftarrow j}(f|e)$ l'alignement de viterbi obtenu en forçant j à être associé à i et $b_{i \leftarrow j}(a)$ le meilleur alignement dans le voisinage de a contraint à l'alignement $f_j \leftrightarrow e_i$.

Brown et al. [1993] proposent d'approximer l'alignement de viterbi pour le modèle 3 $V(f|e; 3)$ par:

$$V(f|e; 3) \approx b^\infty(V(f|e; 2))$$

où $V(f|e; n)$ est l'alignement de viterbi trouvé par le modèle n.

Viterbi (approché) dans le cas du modèle 3

Note: En pratique, la sommation sur tous les alignements nécessaires à l'accumulation des comptes de réestimation des paramètres du modèle 3 est faite seulement sur $\mathcal{S}$ qui est défini par:

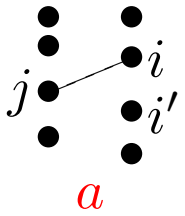
$$\mathcal{S} = \mathcal{N}(b^{\infty}(V(f|e;2))) \bigcup \cup_{i,j} \mathcal{N}(b^{\infty}_{i \leftarrow j}(V_{i \leftarrow j}(f|e;2)))$$

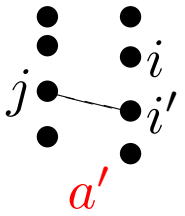
Mais revenons au calcul de $V(f|e;3)$*, deux options s'offrent à vous:*

- calculer pour chaque alignement a' du voisinage de a (l'alignement "de départ") sa probabilité jointe $p(a', f|e)$ selon l'équation 5,
- ou (et c'est bien sûr plus efficace), en déduisant $p(a', f|e)$ à partir de $p(a, f|e)$ à la lumière de l'équation 5

Viterbi (approché) dans le cas du modèle 3

$$move\ j \rightarrow i/i',\ i, i' \neq 0$$



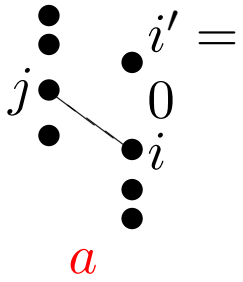


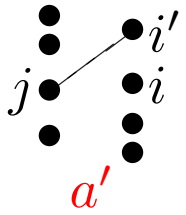
$$p(a, f|e) = \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j, m, l)$$

$$\frac{p(a', f|e)}{p(a, f|e)} = \frac{\phi_{i'} + 1}{\phi_i} \frac{n(\phi_i - 1|e_i)}{n(\phi_i|e_i)} \frac{n(\phi_{i'} + 1|e_{i'})}{n(\phi_{i'}|e_{i'})} \frac{t(f_j|e_{i'})}{t(f_j|e_i)} \frac{d(j|i', m, l)}{d(j|i, m, l)}$$

Viterbi (approché) dans le cas du modèle 3

$move\ j \rightarrow i/i',\ i' = 0$





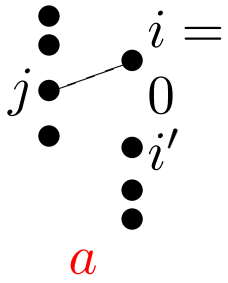
$$p(a, f|e) = \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j, m, l)$$

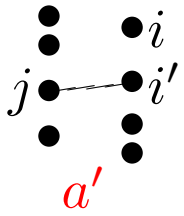
$$\frac{p(a', f|e)}{p(a, f|e)} = \frac{n(\phi_i - 1|e_i)}{n(\phi_i|e_i)} \frac{1}{\phi_i} \frac{t(f_j|e_0)}{t(f_j|e_i)} \frac{1}{d(j|i, l, m)} \frac{p_1}{p_0^2} \frac{(m - 2\phi_0)(m - 2\phi_o - 1)}{(m - \phi_0)(\phi_0 + 1)}$$

le 6ième terme est $\frac{\binom{m-\phi_0-1}{\phi_0+1}}{\binom{m-\phi_0}{\phi_0}}$

Viterbi (approché) dans le cas du modèle 3

$$move\ j \rightarrow i/i',\ i = 0$$





$$p(a, f|e) = \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \times \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j, m, l)$$

$$\frac{p(a', f|e)}{p(a, f|e)} = \frac{n(\phi_{i'}+1|e_{i'})}{n(\phi_{i'}|e_{i'})} \frac{\phi_{i'}+1}{\phi_i} \frac{t(f_j|e_{i'})}{t(f_j|e_0)} \frac{d(j|i', l, m)}{1} \times \frac{p_0^2}{p_1} \frac{\phi_0(m-\phi_o+1)}{(m-2\phi_0+2)(m-2\phi_0+1)}$$

$\hookrightarrow$ *on fait de même pour le swap*

Évaluer l'alignement de mots

Soit $\mathcal{R}$ la référence, et $\mathcal{S}$ les alignements marqués **S**urs dans la référence. Et soit $\mathcal{A}$ l'alignement candidat, et $\mathcal{A}_s$ le sous-ensemble des alignements surs.

- Précision (P, P_S) et Rappel (R, R_S):

$$\begin{array}{lllll} P & = & \frac{|\mathcal{A} \cap \mathcal{R}|}{|\mathcal{A}|} & R & = & \frac{|\mathcal{A} \cup \mathcal{R}|}{|\mathcal{R}|} \\ P_S & = & \frac{|\mathcal{A}_s \cap \mathcal{S}|}{|\mathcal{A}_s|} & R_S & = & \frac{|\mathcal{A}_s \cap \mathcal{S}|}{|\mathcal{S}|} \end{array}$$

- AER (Alignment Error Rate Och and Ney [2000])

$$AER = 1 - \frac{|\mathcal{A} \cap \mathcal{S}| + |\mathcal{A} \cap \mathcal{R}|}{|\mathcal{A}| + |\mathcal{S}|}$$

où $|X|$ désigne la taille (nombre d'alignements) dans X.

Passage de IBM2 à IBM3

On peut entraîner IBM3 avec les formules vues précédemment, mais il est préférable d'initialiser les paramètres de IBM3 à partir des paramètres de IBM2. Cette étape n'est cependant pas aussi triviale que lors du passage de IBM1 à IBM2.

Idée: Les probabililités de transfert se récupèrent sans modification, les probabilités d'alignement de IBM2 doivent être inversées. Le plus coûteux consiste à collecter les comptes pour initialiser les fertilités.

L'idée est de collecter les comptes normalisés par la probabilité de chaque alignement. Voici un exemple simplificateur pour illustrer le processus[7]; voir Brown et al. [1993] pour un algorithme efficace.

[7] Pris dans Knight [1999]

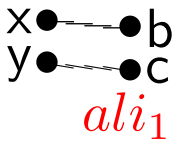
Passage de IBM2 à IBM3

Initialiser les fertilités

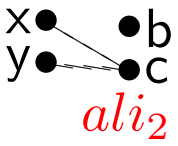
Imaginons le corpus constitué d'une seule paire (bc,xy) et supposons que modèle 1 (pour simplifier) a trouvé les probabilités lexicales suivantes:

$t(x|b) = 0.8,\ t(y|b) = 0.2,\ t(x|c) = 0.2,\ t(y|c) = 0.8$

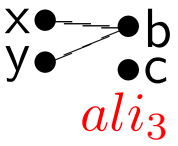
Et soit les 4 alignements suivants:



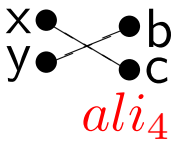
ali_1



ali_2



ali_3



ali_4

Problème: Transfert des a-probabilités en probabilités de fertilité

Idée: collecter les comptes pour le calcul des fertilités en prenant en considération les probabilités lexicales obtenues par le modèle précédent (ici modèle 1).

Passage de IBM2 à IBM3

$$p(a|e,f) = \frac{p(f,a|e)}{p(f|e) = \sum_a p(f,a|e)}$$

Selon IBM1, (idem pour IBM2 en multipliant par les probabilités d'alignement a):

$$\begin{array}{lcll} p(ali1|e,f) & = & p(x|b) \times p(y|c)/p(f|e) & = 0.64 \\ p(ali2|e,f) & = & p(x|c) \times p(y|c)/p(f|e) & = 0.16 \\ p(ali3|e,f) & = & p(x|b) \times p(y|b)/p(f|e) & = 0.16 \\ p(ali4|e,f) & = & p(x|c) \times p(y|b)/p(f|e) & = 0.04 \\ p(e|f) & = & 1 & \end{array}$$

Collectons les comptes fractionnaires pour les fertilités (ici ceux de b):

$$\begin{array}{lll} c(0|b) & = 1 \times 0.16 \ (ali2) & = 0.16 \\ c(1|b) & = 1 \times 0.64 \ (ali1) + 1 \times 0.04 \ (ali4) & = 0.68 \\ c(2|b) & = 1 \times 0.16 \ (ali3) & = 0.16 \end{array}$$

Il suffit de normaliser ces comptes pour obtenir les probabilités de fertilités initiales du modèle 3 (ici ca somme déjà à 1).

IBM3: La magie des nombres (pris de Brown et al. [1993])

Probabilités de transfert et de fertilité pour le mot e $= nodding$.

f	$t(\text{f}\|\text{e})$	f	$t(\text{f}\|\text{e})$	ϕ	$n(\phi\|\text{e})$
signe	0.164	hocher	0.048	4	0.342
la	0.123	faire	0.030	3	0.293
tête	0.097	me	0.024	2	0.167
oui	0.086	approuve	0.019	1	0.163
fait	0.073	qui	0.019	0	0.023
que	0.073	un	0.012		
hoche	0.054	faites	0.011		

IBM3: La magie des nombres (pris de Brown et al. [1993])

Probabilités de transfert et de fertilité pour le mot e = $farmers$.

f	$t(\text{f}\|\text{e})$	ϕ	$n(\phi\|\text{e})$
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

Une propriété de IBM3

Le fait que les probabilités de distorsion (la probabilité d'une position cible) ne dépendent pas des positions décidées auparavant fait que le modèle est *déficient*.

En clair, cela signifie qu'il se peut très bien que plusieurs mots soient attribués indépendamment à la même position cible (ce que les modèles IBM interdisent par essence). Donc une partie de la masse de probabilité de modèle 3 est employée à modéliser des séquences sans intérêt.

$$\sum_f p(f|e) + p(failure|e) = 1$$

Un modèle est déficient si $p(failure|e) > 0$

Mais où s'arrêteront-ils ? (IBM4)

Une phrase n'est bien sur pas toujours traduite linéairement, mais on observe cependant une tendance à préserver l'ordonnancement de groupes de mots syntaxiquement ou sémantiquement liés:

commons			x							
of			x							
house			x							
the		x								
by	x									
heard					x					
been				x						
has				x						
affairs									x	
municipal										x
of								x		
minister							x			
The						x				
	NULL	la	chambre	a	entendu	le	ministre	des	affaires	municipales

IBM4

On modifie le traitement de $(\pi_{i,k}|\pi_{i,1}^{k-1}, \pi_1^{i-1}\tau_0^{i-1}, \phi_0^l, e)$ pour limiter ce problème.

Pour cela on change un peu l'histoire générative. Chaque mot source e_i génère toujours ϕ_i mots cibles (fertilité), mais on commence (**étape 1**) par générer la $tête$ de la $tablet$ (le premier mot) en tenant compte de la $tablet$ précédente selon la distribution:

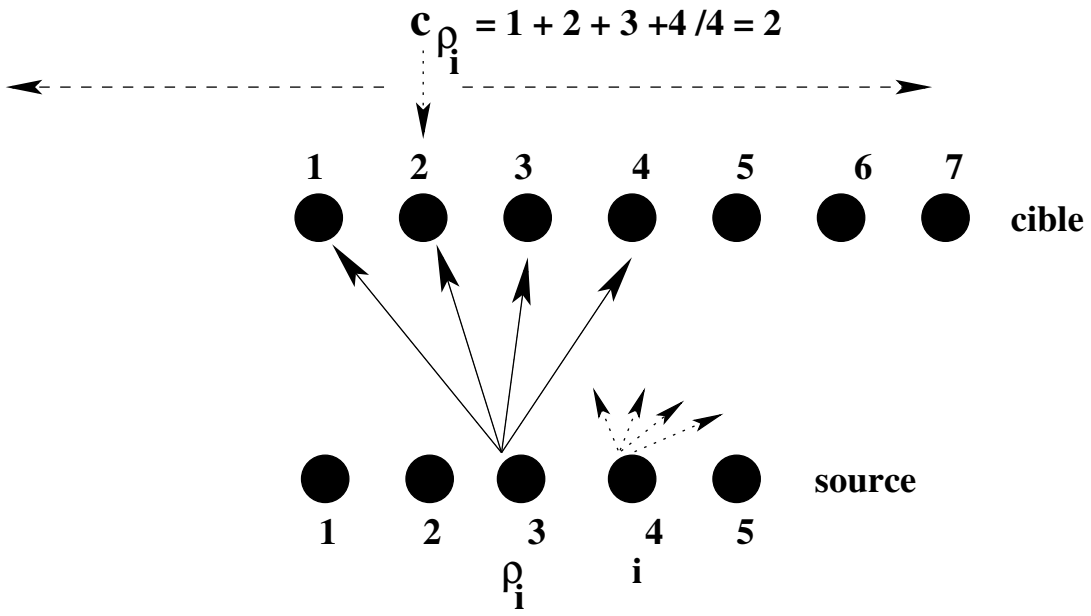
$$(\pi_{i,1} = j|\pi_1^{i-1}, \tau_0^{i-1}, \phi_0^l, e) = d_1(j - c_{\rho_i}|\mathcal{A}(e_{\rho_i}), \mathcal{B}(f_j))$$

avec:

$$\begin{aligned} \rho_i &= \max_{i'<i}\{i' : \phi_{i'} > 0\} \\ c_\rho &= \phi_\rho^{-1} \sum_{k=1}^{\rho} \pi_{\rho,k} \end{aligned}$$

et $\mathcal{A}$ et $\mathcal{B}$ deux fonctions prenant respectivement en entrée le mot source "précédent" et le mot de tête (cible) de la i-ème tablet $\tau_{i,1}$.

IBM4: Illustration de l'étape 1

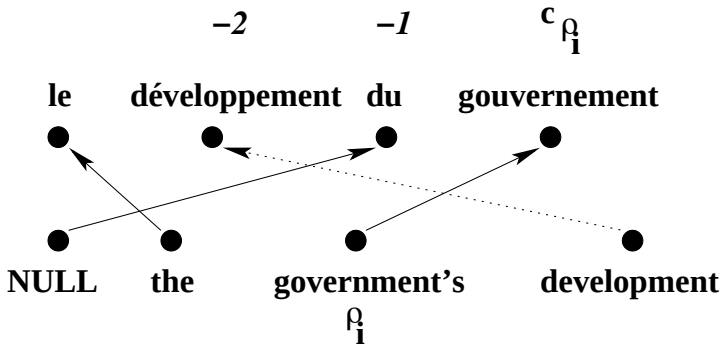


IBM4

L'idée est de modéliser les mouvements de groupes par ces déplacements de têtes (qui peuvent être positifs ou négatifs).

On aimerait avec cela capturer des régularités positionnelles entre les deux langues comme par exemple la tendance à l'inversion (entre le français et l'anglais) de la séquence nom/adjectif (ex: The blue montain / la montagne bleue).

Brown et al. [1993] reportent que: $d_1(-1|\mathcal{A}(\text{government's})\mathcal{B}(\text{développement})) = 0.7986$ et $d_1(+1|\mathcal{A}(\text{government's})\mathcal{B}(\text{développement})) = 0.0168$.



IBM4

Une fois la tête du $tablet$ positionnée, on positionne les autres mots en imposant que chaque mot positionné se trouve à droite du mot précédemment positionné:

$$p(\pi_{i,k} = j|\pi_{i,1}^{k-1}, \pi_1^{i-1}, \tau_0^{i-1}, \phi_0^l, e) = d_{>1}(j - \pi_{i,k-1}|\mathcal{B}(f_j))$$

Brown et al. [1993] reportent par exemple que $d_{>1}(2|\mathcal{B}(\text{pas})) = 0.6847$ (ex: not/ne... pas) alors que $d_{>1}(2|\mathcal{B}(\text{en})) = 0.15$ (ex: implemented/mis en application).

Ce modèle est encore déficient.

IBM5

- L'idée du modèle IBM5 est d'éliminer les problèmes de déficience. IBM3 est déficient en raison de l'indépendance du choix d'une position cible pour un mot donné d'une $tablet$.

- IBM4 possède (d'une manière moindre) les mêmes travers, plus celui de modéliser des déplacements de tête de tablet en dehors de la phrase source (avant le début ou après la fin).

- IBM5 remédie à cela en forçant les positionnements à tomber sur des places vacantes. Pour plus de détails, voir Brown et al. [1993]. Le modèle 5 contient plus de paramètres, mais ne donne pas nécéssairement de meilleurs résultats que IBM4.

Quelques exemples

empruntés à Brown et al. [1993]

Évolution de l'alignement de mots au fur et à mesure des itérations:

Il$_1$ me$_2$ semble$_3$ faire$_4$ signe$_5$ que$_6$ oui$_7$

It seems(3) to(4) me(2) that(6) he(1) is nodding$(5,7)$	$(It_1$ IBM2$)$
It(1) seems(3) to me(2) that he is nodding$(5,7)$	$(It_{last}$ IBM2$)$
It(1) seems(3) to(4) me(2) that(6) he is nodding$(5,7)$	$(It_{last}$ IBM3$)$
It(1) seems(3) to me(2) that(6) he is nodding$(4,5,6,7)$	$(It_{last}$ IBM5$)$

Question: êtes-vous capable de tracer les liens entre les mots à la main ?

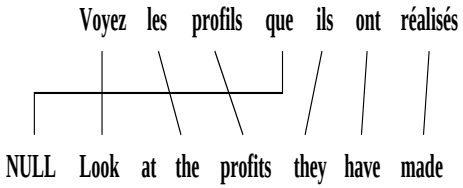
Quelques exemples

empruntés à Brown et al. [1993]

Évolution de l'alignement de mots au fur et à mesure des itérations:

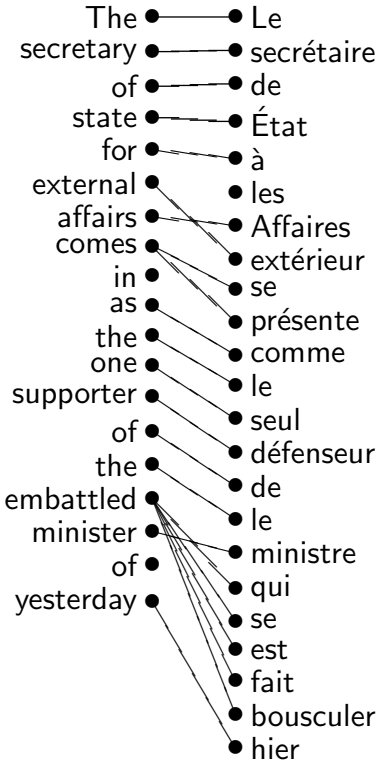
Voyez$_1$ les$_2$ profits$_3$ que$_4$ ils$_5$ ont$_6$ réalisés$_7$

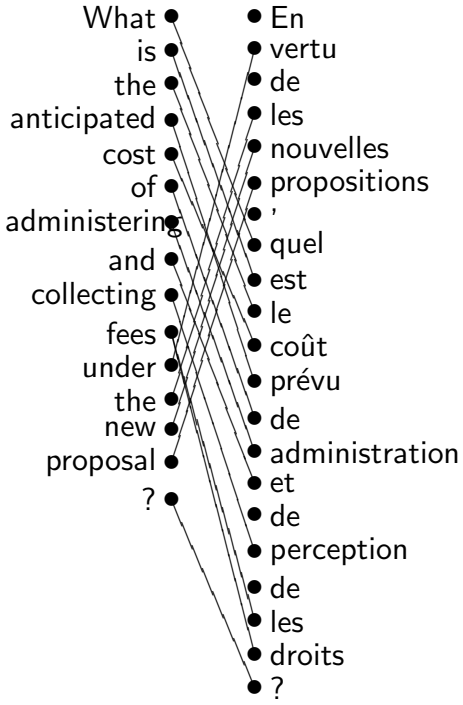
Look(1) at the(2) profits$(3,7)$ they(5) have(6) made	$(It_1$ IBM2$)$
Look(1) at the$(2,4)$ profits$(3,7)$ they(5) have(6) made	$(It_{last}$ IBM2$)$
Look(1) at the profits$(3,7)$ they(5) have(6) made	$(It_{last}$ IBM3$)$
Look(1) at the(2) profits(3) they(5) have(6) made(7)	$(It_{last}$ IBM5$)$



Quelques exemples

empruntés à Brown et al. [1993]





Exemples d'alignements de viterbi obtenus par IBM5

Discussion

- IBM4 & 5 permettent de modéliser des alignements 1-n. Idéalement, on aimerait modéliser des relations $n - m$. C'est très dur en pratique.
  $\hookrightarrow$ *Une méthode "simple": entraîner deux modèles (anglais→français, français→anglais). On calcule l'alignement de viterbi sur le corpus d'entraînement selon les deux modèles. Les intersections délimitent souvent des unités (à suivre).*
- Chaque forme d'un mot est traitée séparément: mange, manger, mangeasse, etc. 39 formes au total que nous ne verrons probablement pas toutes dans le corpus d'entraînement. Idem pour les formes singulier/pluriel, féminin/masculin d'un même nom (à suivre).
- Des traitements simples permettent de diminuer le vocabulaire.
  $\hookrightarrow$ *Par exemple certaines entités numériques peuvent être remplacées par un unique représentant (NB), de même pour les noms propres (NP).*
- L'information linguistique capturée par les modèles n'est pas directement modélisée.

Capturer l'information morpho-syntaxique (Nießen and Ney [2001])

Prétraitement du corpus d'entraînement à l'aide d'un analyseur morphosyntaxique:

lemme	mot	tag
pétition	Pétitions	NomC-femi-plur
chambre	CHAMBRE	NomC-femi-sing
du	DES	Dete-dart-ddef-femi-plur
commune	COMMUNES	NomC-femi-plur
aujourd'hui	Aujourd'hui	Adve
,	,	Punc-pccm
le	les	Dete-dart-ddef-masc-plur
député	députés	NomC-masc-plur
rappeler	rappellent	Verb-IndPré-plur-p3
le	l'	Dete-dart-ddef-masc-sing
engagement	engagement	NomC-masc-sing
que	que	Pron-pwhr-prod-masc-sing
je	nous	Pron-prp-psuj-genl-plur-p1
avoir	avons	Verb-IndPré-plur-p1
prendre	pris	Verb-ParPas-masc-sing

Capturer l'information morpho-syntaxique

Idée: apprendre différents modèles de transfert, liant un mot source non plus à une lexie cible, mais à sa forme de base ainsi qu'à des informations morphosyntaxiques (l'équivalent d'une étiquette POS).

Ex: $t(\text{boire-V-IND-PRES}|\text{drink})$, $t(\text{boire-V}|\text{drink})$, $t(\text{boisson-NOM-SG}|\text{drink})$, *etc.*

Différents modèles correspondent à différents degrés d'abstraction des traits morphosyntaxiques. L'implémentation décrite par Nießen and Ney [2001] est basée sur une cascade de modèles.

classes	$t_0 - t_1 - \ldots - t_n$	formes associées
c_n	ankommen-V-IND-PRES-SG1	ankomme
c_{n-1}	ankommen-V-IND-PRES-SG	ankomme, anskommt, anskommst
c_{n-2}	ankommen-V-IND-PRES	. . .
$\vdots$	$\vdots$	$\vdots$
c_0	ankommen	toutes les formes de ankommen

Capturer l'information morpho-syntaxique

À chaque niveau correspond un modèle p_i:

$$p_i(\mathsf{f}|\mathsf{e}) = \sum_{t_0^i} \underbrace{p(t_0^i|\mathsf{e})}_{transfert} \times \underbrace{p(\mathsf{f}|t_0^i, \mathsf{e})}_{p(\mathsf{f}|t_0^i)}$$

Par définition: $p(\mathsf{f}|t_0^n) = 1$. Et la probabilité d'un modèle hiérarchique est donnée par:

$$p(\mathsf{f}|\mathsf{e}) = \sum_{i=0}^{n} \lambda_i p_i(\mathsf{f}|\mathsf{e})$$

Les auteurs reportent des expériences avec n=1 (modèle hiérarchique à 2 étages): $\lambda_0 = \lambda_1 = 0.5$, $p(f|t_0^i)$ est donnée par fréquence relative.

Capturer l'information morpho-syntaxique

Exemple:

$$p(mangerions|eat) = 0.5 \times p(manger|eat) + 0.5 \times p(manger-VB-COND|eat)$$

- intuitivement satisfaisant
- plus lourd: plusieurs modèles à stocker
- combinaison linéaire sous-optimale

Capturer l'information morpho-syntaxique

- Donner une traduction acceptable à une forme non vue dans le corpus d'entraînement sur la base de son lemme et d'informations syntaxiques:

input	mit dem Zug ist es **bequemer**
baseline	by train it is **UNK-bequemer**
hiérarchique	by train it is **convenient**

- Améliorer la précision de la traduction:

input	**sind** Sie mit einem DoppelZimmer einverstanden ?
baseline	are you agree with a double room ?
hiérarchique	**would** you agree with a double room ?

Inversion Transduction Grammar (ITG) Wu [1997, 2000]

Décrit comme une grammaire hors contexte qui génère un bitexte. C'est une extension des grammaires par transduction simple où les règles sont de la forme:

$$A \rightarrow (x/y \quad \vee \quad B)^+$$

où x/y est une paire de mots source/cible (ex: apple/pomme), A et B des non-terminaux. Les paires lexicales peuvent contenir le mot vide ϵ pour rendre compte des insertions de mots sources et cibles.

Ex: $A \rightarrow B\ x/y\ C\ z/\epsilon$ signifie que x (source) se reécrit par y (cible) et z se réécrit en ϵ (insertion d'un mot source).

Ce genre de grammaire permet de générer simultanément la phrase source et la phrase cible, mais ne possède pas les propriétés requises pour aligner des phrases sources et cibles dont l'ordre des mots est perturbé.

Inversion Transduction Grammar (ITG) Wu [1997, 2000]

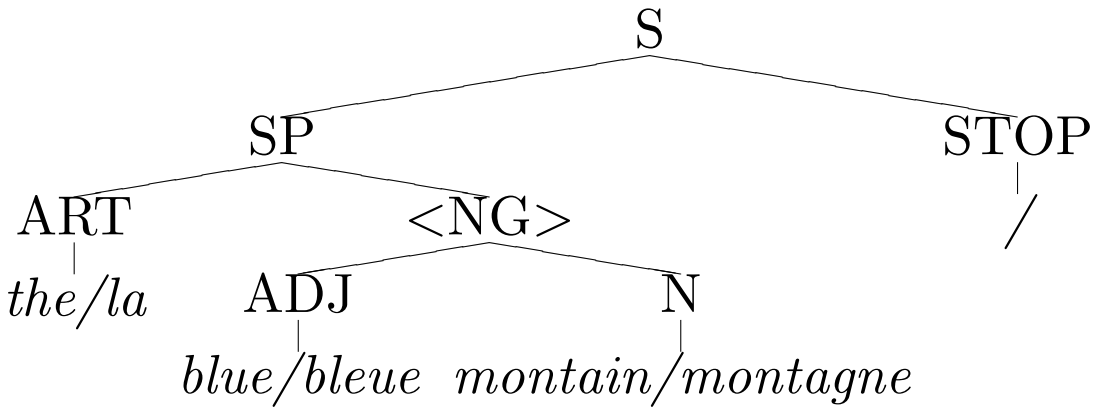
S → SP Stop	STOP → ./.
SP → ART NG	ART → the/la
NG → N ADJ	ADJ → blue/bleue
NG → ADJ N	N → montain/montagne

Cette grammaire ne peut pas produire un alignement (correct) de la paire de phrases (the blue montain . / la montagne bleue .).

Wu [2000] propose d'autoriser des règles avec inversion en introduisant deux types de lecture. [] indique une lecture habituelle (gauche-droite) tandis que <> indique un lecture inversée.

NG → [ADJ N]
NG → <ADJ N>

Inversion Transduction Grammar (ITG) Wu [1997, 2000]



Les fils directs des nœuds étiquetés <> sont lus de la gauche vers la droite dans la langue source et de la droite vers la gauche dans la langue cible; rendant l'alignement correct possible.

Inversion Transduction Grammar (ITG) Wu [1997, 2000]

- Wu [2000] montre qu'une ITG peut-être représentée sous forme normale:

$$\begin{array}{lll} \mathrm{S} \rightarrow \epsilon \ / \ \epsilon & \mathrm{A} \rightarrow \mathrm{x}/\epsilon & \mathrm{A} \rightarrow [\mathrm{B\ C}] \\ \mathrm{A} \rightarrow \mathrm{x/y} & \mathrm{A} \rightarrow \epsilon \ /\mathrm{y} & \mathrm{A} \rightarrow \langle \mathrm{B\ C} \rangle \end{array}$$

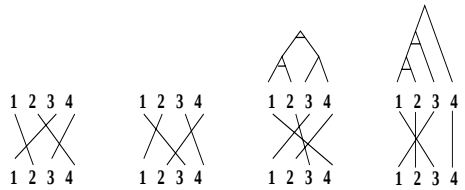
- Tous les alignements ne sont pas représentables avec une ITG, et Wu [2000] fait l'hypothèse que ces alignements non représentables ne sont pas linguistiquement motivés.

- Si tel est le cas, alors une ITG possède l'intéressante propriété d'autoriser une fraction de plus en plus restreinte des alignements potentiels entre des constituants de longueur l, plus l augmente.

Inversion Transduction Grammar (ITG)

l	ITG	|aligt|	l	ITG	|aligt|
0	1	1	6	394	720
1	1	1	7	1806	5040
2	2	2	8	8558	40320
3	6	6	9	41586	362880
4	22	24	10	206098	3628800
5	90	120	. . .	. . .	. . .

Les deux alignements de séquences de 4 mots impossibles (parmi 24 alignements potentiels) à représenter avec une ITG; plus deux alignements représentables:



Inversion Transduction Grammar (ITG) Wu [1997, 2000]

- Wu propose également une version stochastique des ITG (SITG) en ajoutant une probabilité à chaque règle sous la contrainte, pour tout non terminal A:

$$\sum_{B,C \in N} (a_{\text{A}\rightarrow\text{[B C]}} + a_{\text{A}\rightarrow\text{<B C>}}) + \sum_{x,y \in T} b_A(x,y) = 1$$

  où les a-probabilités sont celles associées aux nœuds qui ne sont pas des feuilles, et les b-probabilités sont les probabilités lexicales.

- Un algorithme d'analyse proche de CYK est décrit, ainsi que différents usages des ITGs commes le parenthésage bilingue à l'aide d'une ITG simple comme:

A $\longrightarrow$ [A A] [a]	A $\longrightarrow u_i/v_j$ [$b_{i,j}$]
A $\longrightarrow$ <A A> [a]	A $\longrightarrow u_i/\epsilon$ [$b_{i,\epsilon}$]
	A $\longrightarrow \epsilon/v_j$ [$b_{\epsilon,j}$]

Traduction statistique guidée par la syntaxe (Yamada and Knight [2001])

- Les modèles de Wu [2000], Alshawi et al. [2000] intègrent l'information syntaxique de manière indirecte: le processus de traduction n'est pas modélisé, seul le but de générer des paires de phrases source/cible est visé.

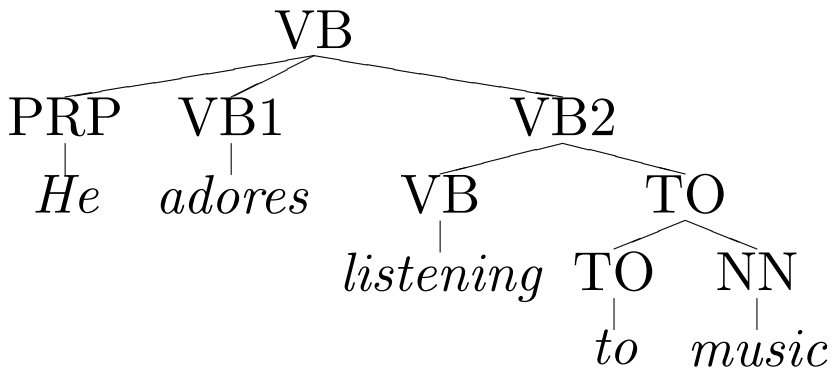
- Yamada and Knight [2001] proposent le modèle de canal bruité suivant:

  Un arbre syntaxique entre dans un canal qui le transmet sous la forme d'une chaîne cible traduction de la chaîne source associée à l'arbre "entrant". Le canal applique pour cela trois opérations: 1) le réordonnancement des nœuds de l'arbre, 2) l'insertion de mots dans les nœuds de l'arbre réordonné et, 3) la traduction des feuilles de l'arbre précédent. Le canal sort ensuite la chaîne résultant de la lecture en profondeur d'abord de l'arbre "traduit".

Traduction statistique guidée par la syntaxe

Exemple (pris dans l'article)

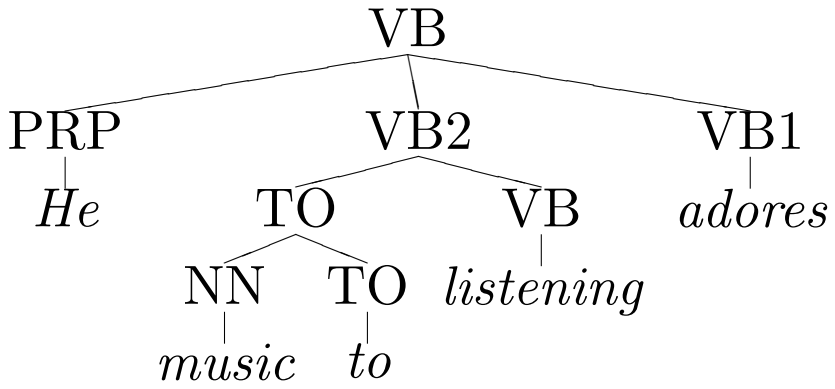
Entrée du canal bruité[8]:



[8] Arbre obtenu par l'analyseur probabiliste de Collins [1999].

Traduction statistique guidée par la syntaxe

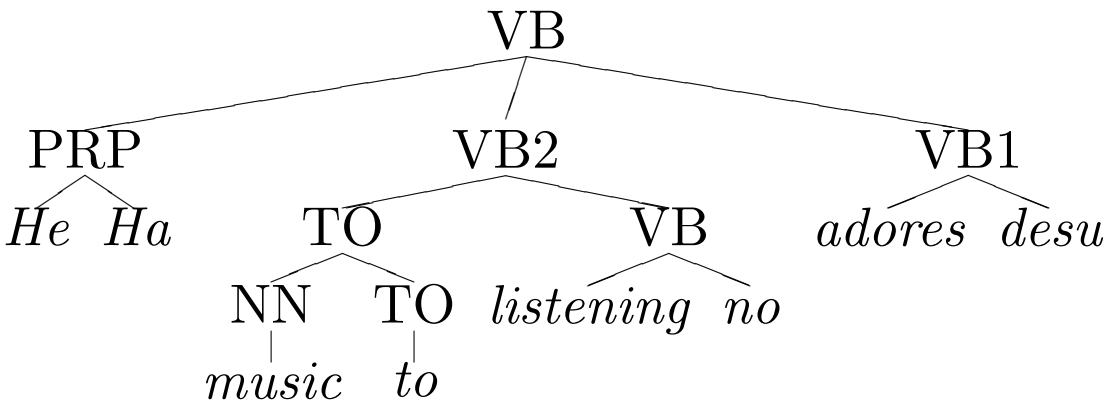
Réordonnancement



original	réordonnanct	prob	original	réord.	prob
	PRP VB2 VB1	0.723	VB TO	TO VB	0.749
PRP VB1 VB2	VB2 PRP VB1	0.083		VB TO	0.251
	PRP VB1 VB2	0.074	TO NN	NN TO	0.893
	⋮	⋮		TO NN	0.107

Traduction statistique guidée par la syntaxe

Insertion

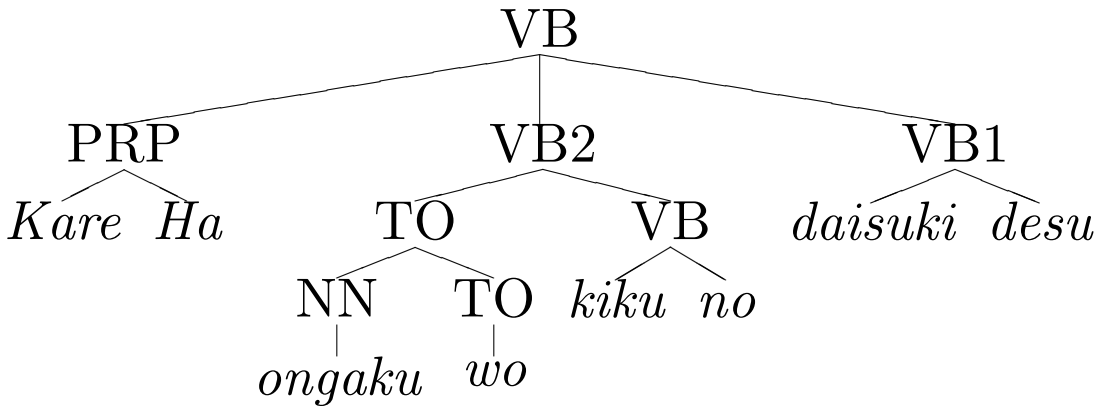


parent	TOP	VB	VB	...
node	VB	VB	PRP	...
p(none)	0.735	0.687	0.344	...
p(left)	0.004	0.061	0.004	...
p(right)	0.260	0.252	0.652	...

w	$p_{ins}(w)$
ha	0.219
ta	0.131
⋮	⋮

Traduction statistique guidée par la syntaxe

Traduction



E	adores		he		. . .
J	daisuki	1.0	kare	0.952	. . .
			NULL	0.016	. . .
			nani	0.005	. . .
			da	0.003	. . .
			⋮	⋮	. . .

Traduction statistique guidée par la syntaxe

Sortie du canal: kare ha ongaku wo kiku no ga daisuki desu

Expérience: Les deux modèles sont entraînés sur le même corpus d'environ 2000 paires de phrases de longueur moyenne 6.9 (resp. 9.7) mots anglais (resp. japonais).

$\hookrightarrow$ *Les alignements de viterbi du corpus d'entraînement obtenu par les deux modèles sont comparés en défaveur de IBM5. Les auteurs reportent que IBM5 a tendance à faire régulièrement des erreurs, alors que le nouveau modèle tend à cumuler les erreurs sur les mêmes phrases.*

Les modèles Phrase-based (PBM)

```
ne pas reformer ||| not reconstitute ||| 1
japonais et américains ||| Japanese and American ||| 1
Une recriminalisation ||| Re-criminalization violates ||| 1
serait moins inquiétant ||| would be less disturbing ||| 0.5
serait moins inquiétante ||| would be less disturbing ||| 0.5
nous aurons tous noté avec intérêt ||| we all noted with interest ||| 1
accepté cet amendement pour répondre ||| accepted this amendment to respond |||
n’ ont pas toutes ||| there was not unanimity of feeling throughout the country
augmenter plus vite que le taux ||| grow faster than the rate ||| 1
dommages faits par ||| damage that is being done by ||| 1
autres . ||| other can be invoked . ||| 1
on ne peut pas demander ||| they cannot seek ||| 1
une erreur d’ interprétation . ||| a misinterpretation . ||| 1
Monsieur le Président , ce gouvernement est symbole ||| Mr. Speaker , this Gover
a été présenté aux ||| was presented to ||| 0.285714
été présenté aux ||| was presented to ||| 0.142857
```

PBM: avantages

- réordonnancement passif des mots

  ex: montagne bleue / blue montain

- formes idiomatiques capturées

  passer un sapin / pulling a fast one

- simplicité (d'obtention) / souplesse (adjonction de dictionnaires facile)

- contraint le modèle de langue

- tolérance aux langues difficiles à segmenter en mots

  ex: le chinois

PBM: inconvénients

- simplicité
- aucun pouvoir de généralisation

```
passer un sapin ||| pulling a fast one ||| 1
nous passer un sapin ||| pull a fast one on us ||| 1
passer un sapin . ||| pull a fast one on us . ||| 0.5
nous passer un sapin . ||| pull a fast one on us . ||| 0.5
passer un sapin ||| pull a fast one ||| 1
passer un sapin . ||| play a fast one  ||| 1
```

- gros volumes de données à manipuler

  ex: 1Gig ou plus pour 1.7 M de paires de phrases

Construire un PBM

De nombreuses approches décrites (Koehn et al. [2003], Tillmann [2003], Vogel et al. [2003]). Voici une recette "classique":

Soit $\mathcal{T} = \{s_1^I, t_1^J\}$

1. Aligner (au niveau des mots) le corpus d'entraînement

   $\mathcal{A}(i,j)$ désigne dans la suite le fait que s_i est alignée avec t_j

2. Sélectionner des/toutes les $boîtes$ à partir de l'alignement de mots $\mathcal{A}$

   Déf. possible: $((i_t, i_b), (j_l, j_r))$ est une boîte ssi:

   - $\forall i \in [i_t, i_b]$, $\mathcal{A}(i,j) \Rightarrow j \in [j_l, j_r]$
   - $\forall j \in [j_l, j_r]$, $\mathcal{A}(i,j) \Rightarrow i \in [i_t, i_b]$

Construire un PBM

		j_l 1	2	3	4	5	j_r 6	
i_t	1			•		•		E
	2				•			D
	3						•	C
	4		•					B
i_b	5	•						A
		a	b	c	d	e	f	

- $((4,5),(1,2)) = (AB, ab)$, $((1,3),(1,3)) = (E, c)$ et $((1,2),(3,5)) = (ED, cde)$ sont des boîtes
- $((1,4),(2,3)) = (BCDE, bc)$ ou $((2,4),(4,6))) = (BCD, def)$ n'en sont pas

Construire un PBM

Pour améliorer l'alignement des mots, on applique habituellement des heuristiques du type:

- Aligner $\mathcal{T}$ dans les deux directions ($\mathcal{A}_{e\to f}$ et $\mathcal{A}_{f\to e}$)

  Soit $\mathcal{P} = \mathcal{A}_{e\to f} \cap \mathcal{A}_{f\to e}$ et $\mathcal{R} = \mathcal{A}_{e\to f} \cup \mathcal{A}_{f\to e}$

- Proclamer $\mathcal{P}$ comme un alignement fiable

- Étendre $\mathcal{P}$ avec certains alignements de $\mathcal{R}$

Permet de contrecarrer les restrictions des alignements IBMs

Exemple

.	.	.	.	.	.	.	.	.	.	.	.	.	X
SUNNY	.	.	.	.	.	.	.	.	.	.	.	X	.
MAINLY	.	.	.	.	.	.	.	.	.	.	X	.	.
OTHERWISE	.	.	.	.	.	.	.	.	.	X	.	.	.
PATCHES	.	.	.	.	.	X	↖	.	.	.	.	.	.
FOG	.	.	.	X	.	.	.	.	.	.	.	.	.
MORNING	.	.	.	.	.	.	.	X	.	.	.	.	.
..	.	.	X	.	.	.	.	.	.	.	.	.	.
TODAY	.	X	↖	.	.	.	.	.	.	.	.	.	.
NULL	.	.	.	.	.	↖	.	.	.	.	.	.	.
	NULL	AUJOURD	HUI	.	BANCS	DE	BROUILLARD	EN	MATINEE	PUIS	GENERALEMENT	ENSOLEILLE	.

Exemple

```
FOG ||| BANCS ||| 1
MAINLY SUNNY . ||| GENERALEMENT ENSOLEILLE . ||| 1
OTHERWISE MAINLY SUNNY ||| PUIS GENERALEMENT ENSOLEILLE ||| 1
OTHERWISE ||| PUIS ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY ||| BROUILLARD EN MATINEE PUIS GENERALEMENT ||| 1
MORNING FOG PATCHES ||| BROUILLARD EN MATINEE ||| 1
TODAY .. MORNING FOG ||| AUJOURD HUI .. BANCS ||| 1
. ||| . ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY SUNNY . ||| MATINEE PUIS GENERALEMENT ENSOLEILLE . ||| 1
MORNING FOG PATCHES OTHERWISE ||| BROUILLARD EN MATINEE PUIS ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY SUNNY . ||| BROUILLARD EN MATINEE PUIS GENERALEMENT ENSOLEILLE . ||| 1
.. ||| .. ||| 1
OTHERWISE MAINLY ||| PUIS GENERALEMENT ||| 1
MORNING FOG PATCHES OTHERWISE ||| MATINEE PUIS ||| 1
TODAY .. ||| AUJOURD HUI .. ||| 1
OTHERWISE MAINLY SUNNY . ||| PUIS GENERALEMENT ENSOLEILLE . ||| 1
TODAY ||| AUJOURD HUI ||| 1
SUNNY ||| ENSOLEILLE ||| 1
PATCHES ||| BROUILLARD EN ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY ||| MATINEE PUIS GENERALEMENT ||| 1
SUNNY . ||| ENSOLEILLE . ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY SUNNY ||| BROUILLARD EN MATINEE PUIS GENERALEMENT ENSOLEILLE ||| 1
.. MORNING FOG ||| .. BANCS ||| 1
MORNING ||| MATINEE ||| 1
MORNING FOG PATCHES OTHERWISE MAINLY SUNNY ||| MATINEE PUIS GENERALEMENT ENSOLEILLE ||| 1
MAINLY ||| GENERALEMENT ||| 1
```

Quelques tendances

L'étude la plus systématique est celle de Koehn et al. [2003], il en ressort:

- Le plus le mieux
- Le type de modèle d'alignement ne semble pas pertinent (les viterbis pour IBM 1 à 5 sont à peu près équivalents)
- Accumuler les séquences de plus de n mots ne sert à rien ($n \sim 8$)
- Filtrer les séquences de manière à ne conserver que celles qui ont un statut grammatical entraîne des dégradations

  garder des paires comme: $opposition\ ,\ mais\ /\ opposition\ ,\ but$

Peut-on se passer de l'alignement de mots ?

Auto-segmentation vorace d'une matrice d'alignement par mesure de similarité (Vogel et al. [2003]):

A	2.1	1.0	4.7	4.9	4.6	1.9	1.2
B	1.3	1.1	5.1	5.2	4.5	2.1	2.3
C	0.7	0.9	5.3	5.1	4.8	2.2	1.5
D	3.1	2.9	1.1	1.2	1.3	1.7	1.8
E	2.8	2.7	0.9	0.8	1.7	1.8	1.7
F	1.7	1.8	1.2	0.7	1.1	2.1	2.1
G	1.3	1.7	0.8	0.9	1.2	2.3	2.4
	a	b	c	d	e	f	g

$$M[i,j] = \text{similarité entre } s_i \text{ et } t_j$$

Peut-on se passer de l'alignement de mots ?

- étape 1) prendre la paire la plus saillante et l'étendre (par seuillage)

A	2.1	1.0	4.7	4.9	4.6	1.9	1.2
B	1.3	1.1	5.1	5.2	4.5	2.1	2.3
C	0.7	0.9	**5.3**	5.1	4.8	2.2	1.5
D	3.1	2.9	1.1	1.2	1.3	1.7	1.8
E	2.8	2.7	0.9	0.8	1.7	1.8	1.7
F	1.7	1.8	1.2	0.7	1.1	2.1	2.1
G	1.3	1.7	0.8	0.9	1.2	2.3	2.4
	a	b	c	d	e	f	g

Peut-on se passer de l'alignement de mots ?

- étape 2) recommencer jusqu'à l'obtention d'un alignement complet

A	2.1	1.0	4.7	4.9	4.6	1.9	1.2
B	1.3	1.1	5.1	5.2	4.5	2.1	2.3
C	0.7	0.9	**5.3**	5.1	4.8	2.2	1.5
D	**3.1**	2.9	1.1	1.2	1.3	1.7	1.8
E	2.8	2.7	0.9	0.8	1.7	1.8	1.7
F	1.7	1.8	1.2	0.7	1.1	2.1	2.1
G	1.3	1.7	0.8	0.9	1.2	2.3	**2.4**
	a	b	c	d	e	f	g

$\rightarrow$(ABC,cde), (DE,ab), (FG,fg) sont des paires de séquences

Lire Marcu and Wong [2002] pour un algorithme EM qui détermine de manière moins empirique les paramètres d'un PBM.

Comment noter une paire de séquences ?

- par fréquence relative: $p(\text{serait moins inquiétant}|\text{would be less disturbing}) =$

$$\frac{|(\text{serait moins inquiétant}, \text{would be less disturbing})|}{\sum_s |(s, \text{would be less disturbing})|}$$

Note: une large proportion de paramètres vus une seule fois !

- par score IBM: $p(t_1^J|s_1^I) \propto \prod_{j=1}^{J} \sum_{i=1}^{I} t(t_j|s_i)$ plus discriminant

- par score informatif (Zhao et al. [2004])

Idée: une paire est d'autant mieux notée qu'elle est informative

Ex: (Le premier ministre / the prime minister) est peut être plus informatif que (a été / was)

Digression rapide sur la recherche d'information

Pb: recherche d'une requête r_1^M dans une collection de N documents $\mathcal{D} = \{D_1, \ldots, D_N\}$ et soit $V = \{v_1, \ldots, v_L\}$ le vocabulaire de $\mathcal{D}$.

	v_1	v_2	$\ldots$	$\ldots$	$\ldots$	v_L
D_1	•		•	•		•
D_2		•		•	•	
D_3				•	•	
$\vdots$						
D_N	•	•				
r				•	•	

• peut-être un booléen (absence ou présence du mot dans le document ou la requête), c'est souvent un $tf.idf$ (tf = terme frequency, idf = inverse document frequency)

Digression rapide sur la recherche d'information

terme frequency fréquence du terme (mot) dans le document

idf(w) $\log\left(N/\sum_{j=1}^{N}\delta\left(w \in D_j\right)\right)$ avec $\delta(\text{vrai}) = 1, 0$ sinon

Si on se donne une mesure de similarité entre deux vecteurs (r et D_i), alors on peut classer les documents à l'aide de ce score.

$D_3, D_2, D_1, \ldots, D_N$ pourraient être dans notre exemple (en ordre décroissant) les documents les plus pertinents de notre requête r.

Digression rapide sur la recherche d'information

Plusieurs scores populaires $score(d, r)$:

produit interne $\sum_{i=1}^{L} d_i \times q_i$

dice $\frac{2 \times \sum_{i=1}^{L} d_i \times r_i}{\sum_{i=1}^{L} d_i + \sum_{i=1}^{L} r_i}$

jaccard $\frac{\sum_{i=1}^{L} d_i \times r_i}{\sum_{i=1}^{L} d_i + \sum_{i=1}^{L} r_i - \sum_{i=1}^{L} d_i \times r_i}$

cosine $\frac{\sum_{i=1}^{L} d_i \times r_i}{\sqrt{\sum_{i=1}^{L} d_i^2} + \sqrt{\sum_{i=1}^{L} r_i^2}}$

Digression rapide sur la recherche d'information

On peut entraîner un modèle de langue par document et trier les documents en ordre décroissant de $p(d|r)$:

$$p(d|r) = \frac{p(r|d) \times p(d)}{p(r)} \propto p(r|d) \times p(d)$$

En faisant l'hypothèse (raisonnable) d'une distribution uniforme des documents dans la collection:

$$p(d|r) \propto p(r|d) = p_d(r)$$

Que l'on peut modéliser par un modèle n-gram.

Revenons à la traduction: score de pertinence d'une paire de séquences

- Une paire est un document: $(s,t) = D = \{s_1, \ldots, s_I, t_1, \ldots, t_J\}$
- Calculer le $tf.idf$ de chaque mot de la collection de documents
- une paire de séquences peut alors être représentée par deux vecteurs qui devraient être similaires selon un score (cosine, jaccard ou autre)

$$v_s = \{tf.idf_{s_1}, \ldots, tf.idf_{s_I}\} \quad v_t = \{tf.idf_{t_1}, \ldots, tf.idf_{t_J}\}$$

(détail: le vecteur source est ramené à la longueur cible via: $tf.idf_{t_j} = \sum_{i=1}^{I} t(t_j|s_i) tf.idf_{s_i}$)

Alignment Template Model (Och et al. [1999])

D'après Tomás and Casacuberta [2004]

- basé sur un modèle de mots (IBM 4), mais capable de prédire des paires de séquences à l'aide de règles (templates).
- un template est une paire $z = (F_{j=1}^{J}, R)$ où F_j est une classe (ex. POS) et R_i à valeur dans $V \cup [1, J]$

$$\begin{array}{lcl} \text{F} & = & \text{JJ NN NN} \\ \text{R} & = & \text{3 de 2 1} \end{array}$$

Ex: new configuration program / programma de configuracion nuevo

- $p(e_1^I|f_1^J) = \sum_z p(z|f_1^J)p(e_1^I|z, f_1^J)$

Alignment Template Model (Och et al. [1999])

- $p(z|f_1^J)$ est la probabilité d'appliquer le template z étant donnée la séquence f_1^J. Approché par $p(z|F_1^J)$

  $p(z|\text{new configuration program}) \simeq p(z|\text{JJ NN NN})$

- $p(e_1^I|z, f_1^J) \simeq \delta(I, |R|) \prod_{i=1}^{I} p(e_i|R_i, f_1^J)$

  avec: $p(e_i|R_i, f_1^J) = \begin{cases} t(e_i|f_{R_i}) & \text{si } R_i \in [1, J] \\ \delta(e_i, R_i) & \text{si } T_i \in V \end{cases}$

$p(\text{programma de configuracion nuevo}|(\text{JJ,NN,NN, 3 de 2 1}), \text{new configuration program}) =$
$t(e_1 = \text{programma}|f_3 = \text{program}) \times$
$1 \times$
$t(e_3 = \text{configuracion}|f_2 = \text{configuration}) \times$
$t(e_4 = \text{nuevo}|f_1 = \text{new})$

Exemple pris de Tomás and Casacuberta [2004]

Exemple de $p(z|F)$

$p(2 \text{ de } 1 \mid \text{NN NN}) = 0.226$	$p(1\ 2 \mid \text{NN NN}) = 0.109$	$p(2\ 1 \mid \text{NN NN}) = 0.074$
$p(1 \mid \text{NN NN}) = 0.041$	$p(2 \mid \text{NN NN}) = 0.039$	$p(1\ 1\ 2 \mid \text{NN NN}) = 0.035$
$p(2\ 1 \mid \text{NN NN}) = 0.031$	$p(2\ 21 \mid \text{NN NN}) = 0.029$	$p(2 \text{ del } 1 \mid \text{NN NN}) = 0.028$

Note: le tagging de la langue source est fait avant. On se sert des alignements produits par un modèle de mots (ex: IBM 1) et d'un extracteur de boîtes (paires de séquences de mots) pour obtenir ces distributions par fréquence relative.

Quelques mots sur l'évaluation manuelle

Souvent utilisé:

Fluency	
5	Flawless English
4	Good English
3	Non-Native English
2	Disfluent English
1	Incomprehensible

Adequacy	
5	All Information
4	Most Information
3	Much Information
2	Little Information
1	None

Jugements bruités, tâche difficile et longue (coûteuse), l'humain s'habitue !

Quelques mots sur l'évaluation automatique: WER

WER = Word Error Rate = distance d'édition normalisée (par la taille source, la taille cible, le nombre total d'opérations d'édition)

SRC	She does not think that the issue of British Gas' structure will be a long-term problem .
REF	Elle ne croit pas que le problème structurel de la British perdure .
SMT	Elle ne croit que la question de British Gas' structures sera un problème .

```
REF  : Elle ne  croit pas que le  problème structurel de  la  British perdure                          .
CAN  : Elle ne  croit     que la  question            de      British Gas'    structures sera un  probleme .
ALI  : <=>  <=> <=>   DEL <=> SUB SUB      DEL       <=> DEL <=>     SUB     INS        INS  INS INS     <=>
```

$$\text{ED} = 10,\ 17 \text{ opérations} \Rightarrow \text{WER} = 100 \times 10/17 = 58.8\%$$

(**note:** en normalisant par le nombre d'opérations, on a une mesure de bruit)

Quelques mots sur l'évaluation automatique: mWER

WER = Word Error Rate = distance d'édition normalisée (par la taille source, la taille cible, le nombre total d'opérations d'édition)

SRC	She does not think that the issue of British Gas' structure will be a long-term problem .
REF	Elle ne croit pas que le problème structurel de la British perdure .
REF	Elle ne croit pas que la question structurelle de la British Gas sera un problème à long term .
SMT	Elle ne croit que la question de British Gas' structures sera un problème .

```
Elle ne  croit pas que la  question structurelle de  la  British Gas            sera un  problème a   long term .
Elle ne  croit     que la  question              de      British Gas' structures sera un  problème               .
<=>  <=> <=>   DEL <=> <=> <=>      DEL           <=> DEL <=>     SUB  INS        <=>  <=> <=>      DEL DEL  DEL  <=
```

ED = 8, 20 opérations $\Rightarrow$ mWER = $100 \times 8/20 = 40\%$

Quelques mots sur l'évaluation automatique: PER

Position Independant Rate: toute permutation d'une référence a un taux d'erreur null.

SRC	She does not think that the issue of British Gas' structure will be a long-term problem .
REF	Elle ne croit pas que le problème structurel de la British perdure .
SMT1	Elle ne croit que la question de British Gas' structures sera un problème .
SMT2	Gas' structures Elle ne croit sera un que la question de British problème .

$$\text{PER} = 100 \times 8/14 = 57.14\%$$

Note: SER (Sentence Error Rate) $= 100 \times \frac{I - \sum_i^I \delta(Trad_i, Ref_i)}{I} = 100\%$

Quelques mots sur l'évaluation automatique: BLEU (Papineni et al. [2002])

Ref	Can you give me some medicine for headache ?
SMT1	Can I have some medicine for headache ?
SMT2	Can you give me prescribe some medicine ?
SMT3	Can I have some medicine for headache ?

très bon

bon

ok

nul

Quelques mots sur l'évaluation automatique: BLEU

- Comparaison de n-grams entre une traduction et une ou plusieurs références
- 1-g jusqu'à 4-g (ou plus)
- $BLEU = BP \times \exp(\sum_{n=1}^{N} w_n \frac{|n\text{-grams}_{trad} \cap n\text{-grams}_{ref}|}{|n\text{-grams}_{trad}|})$

  moyenne pondérée des précisions n-gram (entre 0 et 1, 1 = parfait)
- BP (Brevity Penalty) $= \min(1, \exp(|trad|/|ref|))$

  pénaliser les phrases trop courtes par rapport à la référence
- $w_n = 1/N$

Quelques mots sur l'évaluation automatique: BLEU

- valable au niveau d'un texte (500 phrases ou plus)
- la présence d'un n-gram (n grand) est très bien payée
- semble corréler avec les jugements humains (note globale entre 1 et 5)
- les n-grams les plus fréquents sont souvent les moins informatifs (fluency plus qu'adequacy)
- lire Doddington [2002] pour une variante biaisée davantage vers la fidélité (adequacy): NIST

Références

ACL-35. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid, Spain, July 1997.

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. Learning dependency translation models as collections of finite-state head transducers. *Computational Linguistics*, 26(1): 45–60, March 2000.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.

P.F. Brown, J.C. Lai, and R.L. Mercer. Aligning Sentences in Parallel Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 169–176, Berkeley, California, June 1991.

J.S. Chang and M.H. Chen. An alignment method for noisy parallel corpora based on image processing techniques. In ACL-35 ACL-35, pages 297–304.

Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

F. Debili. Aligning sentences in bilingual texts french-english and french-arabic. In *Proceedings of the International Conference on Computational Linguistics (COLING) 1992*, pages 517–525, Nantes, France, August 1992.

G. Doddington. Automatic evaluation of machine transslation quality using n-gram cooccurrence statistics. In *Proceedings of the HLT*, pages 257–258, San Diego, USA, 2002.

P. Fung and K.W. Church. K-vec: A new approach for aligning sentences in bilingual corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING) 1994*, pages 1096–1102, Kyoto, Japan, August 1994.

W. A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. In *Computational Linguistics*, volume 19, pages 75–102, 1993.

W. John Hutchins and Harold L. Somers. *An Introduction to Machine Translation*. Academic Press, 1992.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 2000.

M. Kay and M. Röscheisen. Text-translation alignment. *Computational Linguistics*, 19(1): 121–142, 1993.

Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4), 1999.

P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 127–133, 2003.

J-M. Langé and É. Gaussier. Alignement de corpus multilingues au niveau des phrases. *T.A.L*, 36(1-2):67–80, 1995.

Philippe Langlais, Michel Simard, and Jean Véronis. Methods and practical issues in evaluating alignment techniques. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL)*, Montréal, Canada, August 1998.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language*

Processing. MIT Press, 1999.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 133–139, Philadelphia, PA, USA, 2002.

I. Dan Melamed. A portable algorithm for mapping bitext correspondence. In ACL-35 ACL-35, pages xx–yyy.

Sonja Nießen and Hermann Ney. Toward hierarchical models for statistical machine translation of inflected languages. In *Proceedings of the Workshop on Data Driven Machine Translation yielded at the 39th Annual Meeting of the ACL*, pages 47–54, Toulouse, France, July 2001.

F.J. Och and H. Ney. A comparison of alignement models for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics (COLING) 2000*, pages 1086–1090, Saarbrucken, Luxembourg, Nancy, August 2000.

Franz Josef Och, Christoph Tillman, and Herman Ney. Improved alignment models for statistical machine translation. In *Proceedings of the 4th Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 20–28, College Park, Maryland, 1999.

Franz Joseph Och and Hermann Ney. A systematic comparison of various statistical alignement models. *Computational Linguistics*, 2002. To appear.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002.

António Ribeiro, Gaël Dias, Gabriel Lopes, and João Mexia. Cognates alignment. In *Machine Translation Summit VIII, Machine Translation in The Information Age*, pages 287–292, Santiago de Compostela, Galicia, Spain, September 2001.

M. Simard, G.F. Foster, and P. Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–81, Montréal, Québec, 1992.

M. Simard and P. Plamondon. Bilingual sentence alignment: Balancing robustness and accuracy. In *Proceedings of the Second Conference of the Association for Machine*

Translation in the Americas (AMTA), Montréal, Québec, 1996.

Christoph Tillmann. A projection extension algorithm for statistical machine translation. In *Proceedings of the 8th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Saporo, Japan, 2003.

J. Tomás and F. Casacuberta. Combining phrase-based and template-based aligned models in statistical translation. In *Proceedings of the First Iberian Conference on Pattern Recognition and Image Analysis*, pages 1020–1031, Mallorca, june 2004.

J. Véronis and Ph. Langlais. *Evaluation of parallel text alignment systems: The ARCADE project*, volume 13, chapter 19, pages 369–388. Parallel Text Processing, Kluwer, 2000.

S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao, and A. Waibel. The cmu statistical machine translation system. In *Proceedings of MT Summit*, pages 110–117, New Orleans, Louisiana, 2003.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignement in statistical translation. In *Proceedings of the International Conference on Computational Linguistics (COLING) 1996*, pages 836–841, Copenhagen, Denmark, August 1996.

Dekai Wu. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 80–87, Las Cruces, New Mexico, June 1994.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, September 1997.

Dekai Wu. *Bracketing and aligning words and constituents in parallel text using Stochastic Inversion Transduction Grammars*, volume 13, chapter 19, pages 139–168. Parallel Text Processing, Kluwer, 2000.

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 523–530, Toulouse, France, July 2001.

B. Zhao, S. Vogel, and A. Waibel. Phrase pair rescoring with term weightings for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, jul 2004.