Recap: The Noisy Channel Model

- **Goal:** translation system from French to English
- **Have a model** $P(e \mid f)$ which estimates conditional probability of any English sentence $e$ given the French sentence $f$. Use the training corpus to set the parameters.
- **A Noisy Channel Model has two components:**
  - $P(e)$: the language model
  - $P(f \mid e)$: the translation model
- **Giving:**
  \[
  P(e \mid f) = \frac{P(e, f)}{P(f)} = \frac{P(e)P(f \mid e)}{\sum_e P(e)P(f \mid e)}
  \]
  and
  \[\arg\max_e P(e \mid f) = \arg\max_e P(e)P(f \mid e)\]
IBM Model 1: Alignments

- How do we model $P(f \mid e)$?

- English sentence $e$ has $l$ words $e_1 \ldots e_l$.
  French sentence $f$ has $m$ words $f_1 \ldots f_m$.

- An alignment $a$ identifies which English word each French word originated from.

- Formally, an alignment $a$ is $\{a_1, \ldots a_m\}$, where each $a_j \in \{0 \ldots l\}$.

- There are $(l + 1)^m$ possible alignments.

Alignments in the IBM Models

- We’ll define models for $P(a \mid e)$ and $P(f \mid a, e)$, giving
  $$P(f, a \mid e) = P(a \mid e)P(f \mid a, e)$$

- Also,
  $$P(f \mid e) = \sum_{a \in A} P(a \mid e)P(f \mid a, e)$$
  where $A$ is the set of all possible alignments.

A By-Product: Most Likely Alignments

- Once we have a model $P(f, a \mid e) = P(a \mid e)P(f \mid a, e)$ we can also calculate
  $$P(a \mid f, e) = \frac{P(f, a \mid e)}{\sum_{a \in A} P(f, a \mid e)}$$
  for any alignment $a$

- For a given $f, e$ pair, we can also compute the most likely alignment,
  $$a^* = \arg \max_a P(a \mid f, e)$$

- Nowadays, the original IBM models are rarely (if ever) used for translation, but they are used for recovering alignments.

---

IBM Model 1: Alignments (continued)

- e.g., $l = 6$, $m = 7$
  
  - $e = \text{And the program has been implemented}$
  - $f = \text{Le programme a été mis en application}$

- One alignment is
  $$\{2, 3, 4, 5, 6, 6, 6\}$$

- Another (bad!) alignment is
  $$\{1, 1, 1, 1, 1, 1, 1\}$$
An Example Alignment

French:
le conseil a rendu son avis, et nous devons à présent adopter un nouvel avis sur la base de la première position.

English:
the council has stated its position, and now, on the basis of the first position, we again have to give our opinion.

Alignment:
the/le council/conseil has/à stated/rendu its/son position/avis , and/et now/présent ,/NULL on/sur the/le basis/base of/de the/la first/première position/position ,/NULL we/nous again,NULL have/devons to/a give/adopt our/nouvel opinion/avis .

IBM Model 1: Alignments

- In IBM model 1 all alignments a are equally likely:

\[ P(a \mid e) = C \times \frac{1}{(l + 1)^m} \]

where \( C = \text{prob}(\text{length}(f) = m) \) is a constant.

- This is a major simplifying assumption, but it gets things started...

IBM Model 1: Translation Probabilities

- Next step: come up with an estimate for

\[ P(f \mid a, e) \]

- In model 1, this is:

\[ P(f \mid a, e) = \prod_{j=1}^{m} P(f_j \mid e_a) \]

- e.g., \( l = 6, m = 7 \)

\[ e = \text{And the program has been implemented} \]
\[ f = \text{Le programme a ete mis en application} \]

- \( a = \{2, 3, 4, 5, 6, 6, 6\} \)

\[ P(f \mid a, e) = P(\text{Le} \mid \text{the}) \times \]
\[ P(\text{programme} \mid \text{program}) \times \]
\[ P(a \mid \text{has}) \times \]
\[ P(\text{ete} \mid \text{been}) \times \]
\[ P(\text{mis} \mid \text{implemented}) \times \]
\[ P(\text{en} \mid \text{implemented}) \times \]
\[ P(\text{application} \mid \text{implemented}) \]
IBM Model 1: The Generative Process

To generate a French string $f$ from an English string $e$:

- **Step 1:** Pick the length of $f$ (all lengths equally probable, probability $C$)
- **Step 2:** Pick an alignment $a$ with probability $\frac{1}{(l+1)^m}$
- **Step 3:** Pick the French words with probability $P(f_j | a, e_a) = \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$

The final result:

$$P(f, a | e) = P(a | e) \times P(f | a, e) = \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$

A Hidden Variable Problem

- Training data is a set of $(f_i, e_i)$ pairs, likelihood is
  $$\sum_i \log P(f_i | e_i) = \sum_i \log \sum_{a \in A} P(a | e_i) P(f_i | a, e_i)$$
  where $A$ is the set of all possible alignments.
- We need to maximize this function w.r.t. the translation parameters $P(f_j | e_{a_j})$.
- EM can be used for this problem: initialize translation parameters randomly, and at each iteration choose
  $$\Theta_t = \arg\max_\Theta \sum_i \sum_{a \in A} P(a | e_i, f_i, \Theta^{t-1}) \log P(f_i | a, e_i, \Theta)$$
  where $\Theta^t$ are the parameter values at the $t$’th iteration.

A Hidden Variable Problem

- We have:
  $$P(f, a | e) = \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$
- And:
  $$P(f | e) = \sum_{a \in A} \frac{C}{(l+1)^m} \prod_{j=1}^m P(f_j | e_{a_j})$$
  where $A$ is the set of all possible alignments.

An Example

- I have the following training examples
  the dog $\Rightarrow$ le chien
  the cat $\Rightarrow$ le chat
- Need to find estimates for:
  $$P(le | the) \quad P(chien | the) \quad P(chat | the)$$
  $$P(le | dog) \quad P(chien | dog) \quad P(chat | dog)$$
  $$P(le | cat) \quad P(chien | cat) \quad P(chat | cat)$$
- As a result, each $(e_i, f_i)$ pair will have a most likely alignment.
<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>position</td>
<td>position</td>
<td>0.756715</td>
</tr>
<tr>
<td>position</td>
<td>situation</td>
<td>0.0547918</td>
</tr>
<tr>
<td>position</td>
<td>mesure</td>
<td>0.0281663</td>
</tr>
<tr>
<td>position</td>
<td>vue</td>
<td>0.0169303</td>
</tr>
<tr>
<td>position</td>
<td>point</td>
<td>0.0124795</td>
</tr>
<tr>
<td>position</td>
<td>attitude</td>
<td>0.0108907</td>
</tr>
</tbody>
</table>

... de la situation au niveau des négociations de l’ompi ... 
... of the current position in the wipo negotiations ...

nous ne sommes pas en mesure de décider , ... 
we are not in a position to decide , ...

... le point de vue de la commission face à ce problème complexe . 
... the commission’s position on this complex problem .

... cette attitude laxiste et irresponsable . 
... this irresponsibly lax position .

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- Decoding

IBM Model 2

- Only difference: we now introduce alignment or distortion parameters

\[ D(i \mid j, l, m) = \text{Probability that } j \text{’th French word is connected} \]
\[ \text{to } i \text{’th English word, given sentence lengths of } e \text{ and } f \text{ are } l \text{ and } m \text{ respectively} \]

- Define

\[ P(a \mid e, l, m) = \prod_{j=1}^{m} D(a_j \mid j, l, m) \]

where \( a = \{a_1, \ldots, a_m\} \)

- Gives

\[ P(f, a \mid e, l, m) = \prod_{j=1}^{m} D(a_j \mid j, l, m) T(f_j \mid e_{a_j}) \]

Note: Model 1 is a special case of Model 2, where \( D(i \mid j, l, m) = \frac{1}{l+l} \) for all \( i, j \).
An Example

\[ l = 6 \]
\[ m = 7 \]
\[ e = \text{And the program has been implemented} \]
\[ f = \text{Le programme a ete mis en application} \]
\[ a = \{2, 3, 4, 5, 6, 6, 6\} \]

\[ P(a \mid e, 6, 7) = D(2 \mid 1, 6, 7) \times D(3 \mid 2, 6, 7) \times D(4 \mid 3, 6, 7) \times D(5 \mid 4, 6, 7) \times D(6 \mid 5, 6, 7) \times D(6 \mid 6, 6, 7) \times D(6 \mid 7, 6, 7) \]

IBM Model 2: The Generative Process

To generate a French string \( f \) from an English string \( e \):

- **Step 1**: Pick the length of \( f \) (all lengths equally probable, probability \( C \))
- **Step 2**: Pick an alignment \( a = \{a_1, a_2 \ldots a_m\} \) with probability
  \[ \prod_{j=1}^{m} D(a_j \mid j, l, m) \]
- **Step 3**: Pick the French words with probability
  \[ P(f \mid a, e) = \prod_{j=1}^{m} T(f_j \mid e_{a_j}) \]

The final result:

\[ P(f, a \mid e) = P(a \mid e)P(f \mid a, e) = C \prod_{j=1}^{m} D(a_j \mid j, l, m)T(f_j \mid e_{a_j}) \]

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- Decoding
**A Hidden Variable Problem**

- We have:

\[ P(f, a \mid e) = C \prod_{j=1}^{m} D(a_j \mid j, l, m) T(f_j \mid e_{a_j}) \]

- And:

\[ P(f \mid e) = \sum_{a \in A} C \prod_{j=1}^{m} D(a_j \mid j, l, m) T(f_j \mid e_{a_j}) \]

where \( A \) is the set of all possible alignments.

**Model 2 as a Product of Multinomials**

- The model can be written in the form

\[ P(f, a \mid e) = \prod_r \Theta_r^{Count(f, a, e, r)} \]

where the parameters \( \Theta_r \) correspond to the \( T(f \mid e) \) and \( D(i \mid j, l, m) \) parameters.

- To apply EM, we need to calculate expected counts

\[ \overline{Count}(r) = \sum_k \sum_a P(a \mid e_k, f_k, \Theta) Count(f_k, a, e_k, r) \]

**A Crucial Step in the EM Algorithm**

- Say we have the following \((e, f)\) pair:

\[ e = \text{And the program has been implemented} \]

\[ f = \text{Le programme a ete mis en application} \]

- Given that \( f \) was generated according to Model 2, what is the probability that \( a_1 = 2 \)? Formally:

\[ Prob(a_1 = 2 \mid f, e) = \sum_{a : a_1 = 2} P(a \mid f, e, \Theta) \]
Calculating Expected Translation Counts

- One example:
  \[
  \text{Count}(T(le|the)) = \sum_{(i,j,k) \in S} P(a_j = i|e_k, f_k, \Theta)
  \]
  where \( S \) is the set of all \((i,j,k)\) triples such that \(e_k,j = the\) and \(f_{k,j} = le\)

Calculating Expected Distortion Counts

- One example:
  \[
  \text{Count}(D(i = 5|j = 6,l = 10,m = 11)) = \sum_{k \in S} P(a_0 = 5|e_k, f_k, \Theta)
  \]
  where \( S \) is the set of all values of \(k\) such that \(\text{length}(e_k) = 10\) and \(\text{length}(f_k) = 11\)

Models 1 and 2 Have a Simple Structure

- We have \(f = \{f_1 \ldots f_m\}, a = \{a_1 \ldots a_m\}\), and
  \[
  P(f, a | e, l, m) = \prod_{j=1}^{m} P(a_j, f_j | e, l, m)
  \]
  where
  \[
  P(a_j, f_j | e, l, m) = D(a_j | j, l, m)T(f_j | e_{a_j})
  \]

- We can think of the \(m\) \((f_j, a_j)\) pairs as being generated independently

The Answer

\[
\text{Prob}(a_1 = 2 | f, e) = \sum_{a:a_1 = 2} P(a | f, e, l, m)
\]

\[
= \frac{D(a_1 = 2 | j = 1, l = 6, m = 7)T(le|the)}{\sum_{i=0}^{l} D(a_1 = i | j = 1, l = 6, m = 7)T(le|e_{i})}
\]

Follows directly because the \((a_j, f_j)\) pairs are independent:

\[
P(a_1 = 2 | f, e, l, m) = \frac{P(a_1 = 2, f_1 = le | f_2 \ldots f_m, e, l, m)}{P(f_1 = le | f_2 \ldots f_m, e, l, m)}
\]

\[
= \frac{P(a_1 = 2, f_1 = le | e, l, m)}{P(f_1 = le | e, l, m)}
\]

\[
= \frac{P(a_1 = 2, f_1 = le | e, l, m)}{\sum_i P(a_1 = i, f_1 = le | e, l, m)}
\]

where (2) follows from (1) because \(P(f, a | e, l, m) = \prod_{j=1}^{m} P(a_j, f_j | e, l, m)\)
A General Result

\[
Prob(a_j = i \mid f, e) = \sum_{a : a_j = i} P(a \mid f, e, l, m) = \frac{D(a_j = i \mid j, l, m)T(f_j \mid e_i)}{\sum_{i' = 0}^{\infty} D(a_j = i' \mid j, l, m)T(f_j \mid e_{i'})}
\]

The EM Algorithm for Model 2

- Define
  \(e[k]\) for \(k = 1 \ldots n\) is the \(k^{th}\) English sentence
  \(f[k]\) for \(k = 1 \ldots n\) is the \(k^{th}\) French sentence
  \(l[k]\) is the length of \(e[k]\)
  \(m[k]\) is the length of \(f[k]\)

  \(e[k, i] = i^{th}\) word in \(e[k]\)
  \(f[k, j] = j^{th}\) word in \(f[k]\)

- Current parameters \(\Theta^{t-1}\) are
  \(T(f \mid e)\) for all \(f \in F, e \in E\)
  \(D(i \mid j, l, m)\)

- We’ll see how the EM algorithm re-estimates the \(T\) and \(D\) parameters

Alignment Probabilities have a Simple Solution!

- e.g., Say we have \(l = 6, m = 7\),
  
  \(e = \) And the program has been implemented
  \(f = \) Le programme a ete mis en application

- Probability of “mis” being connected to “the”:

\[
P(a_5 = 2 \mid f, e) = \frac{D(a_5 = 2 \mid j = 5, l = 6, m = 7)T(mis \mid the)}{Z}
\]

where

\[
Z = D(a_5 = 0 \mid j = 5, l = 6, m = 7)T(mis \mid NULL)
+ D(a_5 = 1 \mid j = 5, l = 6, m = 7)T(mis \mid And)
+ D(a_5 = 2 \mid j = 5, l = 6, m = 7)T(mis \mid the)
+ D(a_5 = 3 \mid j = 5, l = 6, m = 7)T(mis \mid program)
+ \ldots
\]

Step 1: Calculate the Alignment Probabilities

- Calculate an array of alignment probabilities
  (for \((k = 1 \ldots n), (j = 1 \ldots m[k]), (i = 0 \ldots l[k])\)):

\[
a[i, j, k] = P(a_j = i \mid e[k], f[k], \Theta^{t-1}) = \frac{D(a_j = i \mid j, l, m)T(f_j \mid e_i)}{\sum_{i' = 0}^{\infty} D(a_j = i' \mid j, l, m)T(f_j \mid e_{i'})}
\]

where \(e_i = e[k, i], f_j = f[k, j]\), and \(l = l[k], m = m[k]\)

i.e., the probability of \(f[k, j]\) being aligned to \(e[k, i]\).
Step 2: Calculating the Expected Counts

- Calculate the translation counts
  \[
  tcount(e, f) = \sum_{i,j,k:\ e[k,i]=e, f[k,j]=f} a[i, j, k] 
  \]

- \(tcount(e, f)\) is expected number of times that \(e\) is aligned with \(f\) in the corpus

Step 3: Re-estimating the Parameters

- New translation probabilities are then defined as
  \[
  T(f | e) = \frac{tcount(e, f)}{\sum_f tcount(e, f)} 
  \]

- New alignment probabilities are defined as
  \[
  D(i | j, l, m) = \frac{acount(i, j, l, m)}{\sum_i acount(i, j, l, m)} 
  \]

This defines the mapping from \(\Theta^t-1\) to \(\Theta^t\)

A Summary of the EM Procedure

- Start with parameters \(\Theta^t-1\) as
  \[
  T(f | e) \quad \text{for all } f \in \mathcal{F}, e \in \mathcal{E} 
  \]
  \[
  D(i | j, l, m) 
  \]

- Calculate alignment probabilities under current parameters
  \[
  a[i, j, k] = \frac{D(a_j = i | j, l, m) T(f_j | e_i)}{\sum_{i'} D(a_j = i' | j, l, m) T(f_j | e_{i'})} 
  \]

- Calculate expected counts \(tcount(e, f)\), \(acount(i, j, l, m)\) from the alignment probabilities

- Re-estimate \(T(f | e)\) and \(D(i | j, l, m)\) from the expected counts
The Special Case of Model 1

- Start with parameters $\Theta_{t-1}$ as
  
  $T(f \mid e) \quad \text{for all } f \in \mathcal{F}, e \in \mathcal{E}$
  
  (no alignment parameters)

- Calculate alignment probabilities under current parameters
  
  $a[i, j, k] = \frac{T(f_j \mid e_i)}{\sum_{i'} T(f_{i'} \mid e_{i'})}$
  
  (because $D(a_j = i \mid j, l, m) = 1/(l + 1)^m$ for all $i, j, l, m$).

- Calculate expected counts $tcount(e, f)$

- Re-estimate $T(f \mid e)$ from the expected counts

An Example of Training Models 1 and 2

Example will use following translations:

- $e[1] = \text{the dog}$
- $f[1] = \text{le chien}$
- $e[2] = \text{the cat}$
- $f[2] = \text{le chat}$
- $e[3] = \text{the bus}$
- $f[3] = \text{l’ autobus}$

NB: I won’t use a NULL word $e_0$

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- Decoding

<table>
<thead>
<tr>
<th>$e$</th>
<th>$f$</th>
<th>$T(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0.23</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0.2</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0.11</td>
</tr>
<tr>
<td>the</td>
<td>l’</td>
<td>0.25</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0.21</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>0.2</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0.16</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0.33</td>
</tr>
<tr>
<td>dog</td>
<td>l’</td>
<td>0.12</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0.18</td>
</tr>
<tr>
<td>cat</td>
<td>le</td>
<td>0.26</td>
</tr>
<tr>
<td>cat</td>
<td>chien</td>
<td>0.28</td>
</tr>
<tr>
<td>cat</td>
<td>chat</td>
<td>0.19</td>
</tr>
<tr>
<td>cat</td>
<td>l’</td>
<td>0.24</td>
</tr>
<tr>
<td>cat</td>
<td>autobus</td>
<td>0.03</td>
</tr>
<tr>
<td>bus</td>
<td>le</td>
<td>0.22</td>
</tr>
<tr>
<td>bus</td>
<td>chien</td>
<td>0.05</td>
</tr>
<tr>
<td>bus</td>
<td>chat</td>
<td>0.26</td>
</tr>
<tr>
<td>bus</td>
<td>l’</td>
<td>0.19</td>
</tr>
<tr>
<td>bus</td>
<td>autobus</td>
<td>0.27</td>
</tr>
</tbody>
</table>
Alignment probabilities:

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>(a(i,j,k))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.526423237959726</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0.473576762040274</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0.552517995605817</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0.447482004394183</td>
</tr>
</tbody>
</table>

Expected counts:

<table>
<thead>
<tr>
<th>(e)</th>
<th>(f)</th>
<th>(t\text{count}(e, f))</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0.99295584002626</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0.552517995605817</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0.356364544422507</td>
</tr>
<tr>
<td>the</td>
<td>l'</td>
<td>0.571950438336247</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0.439081311724508</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>0.473576762040274</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0.447482004394183</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0.0</td>
</tr>
<tr>
<td>dog</td>
<td>l'</td>
<td>0.0</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Old and new parameters:

\[
\begin{array}{l|l|l|}
  & \text{old} & \text{new} \\
\hline 
\text{the}  & 0.23 & 0.34 \\
\text{le} & 0.2 & 0.19 \\
\text{chien} & 0.11 & 0.12 \\
\text{chat} & 0.25 & 0.2 \\
\text{l'} & 0.21 & 0.15 \\
\text{autobus} & 0.2 & 0.15 \\
\text{dog} & 0.16 & 0.49 \\
\text{le} & 0.33 & 0 \\
\text{chien} & 0.12 & 0 \\
\text{chat} & 0.12 & 0 \\
\text{l'} & 0.18 & 0 \\
\text{autobus} & 0.26 & 0 \\
\text{cat} & 0.26 & 0.45 \\
\text{le} & 0.28 & 0 \\
\text{chien} & 0.19 & 0.55 \\
\text{chat} & 0.24 & 0 \\
\text{l'} & 0.03 & 0 \\
\text{autobus} & 0.27 & 0.57 \\
\text{bus} & 0.22 & 0 \\
\text{le} & 0.05 & 0 \\
\text{chien} & 0.26 & 0 \\
\text{chat} & 0.19 & 0.43 \\
\text{l'} & 0.03 & 0 \\
\text{autobus} & 0.27 & 0.57 \\
\end{array}
\]
<table>
<thead>
<tr>
<th>e</th>
<th>f</th>
<th>$T(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0.94</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>l'</td>
<td>0.03</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0.02</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>0.06</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0.94</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>le</td>
<td>0.06</td>
</tr>
<tr>
<td>cat</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>chat</td>
<td>0.94</td>
</tr>
<tr>
<td>cat</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>l'</td>
<td>0.49</td>
</tr>
<tr>
<td>bus</td>
<td>autobus</td>
<td>0.51</td>
</tr>
</tbody>
</table>

**After 20 iterations:**

**Model 2 has several local maxima – good one:**

<table>
<thead>
<tr>
<th>e</th>
<th>f</th>
<th>$T(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0.67</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>l'</td>
<td>0.33</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>1</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>chat</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>autobus</td>
<td>1</td>
</tr>
</tbody>
</table>

**Model 2 has several local maxima – bad one:**

<table>
<thead>
<tr>
<th>e</th>
<th>f</th>
<th>$T(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0.4</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0.3</td>
</tr>
<tr>
<td>the</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0.3</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>0.5</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0.5</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>le</td>
<td>0.5</td>
</tr>
<tr>
<td>cat</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>chat</td>
<td>0.5</td>
</tr>
<tr>
<td>cat</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>l'</td>
<td>0.5</td>
</tr>
<tr>
<td>bus</td>
<td>autobus</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Another bad one:**

<table>
<thead>
<tr>
<th>e</th>
<th>f</th>
<th>$T(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>le</td>
<td>0.33</td>
</tr>
<tr>
<td>the</td>
<td>chien</td>
<td>0.33</td>
</tr>
<tr>
<td>the</td>
<td>chat</td>
<td>0.33</td>
</tr>
<tr>
<td>the</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>the</td>
<td>autobus</td>
<td>0.33</td>
</tr>
<tr>
<td>dog</td>
<td>le</td>
<td>1</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>le</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>l'</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>autobus</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>le</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chien</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>chat</td>
<td>0</td>
</tr>
<tr>
<td>bus</td>
<td>l'</td>
<td>1</td>
</tr>
<tr>
<td>bus</td>
<td>autobus</td>
<td>0</td>
</tr>
</tbody>
</table>
Improving the Convergence Properties of Model 2

- Out of 100 random starts, only 60 converged to the best local maxima

- Model 1 converges to the same, global maximum every time (see the Brown et. al paper)

- Method in IBM paper: run Model 1 to estimate $T$ parameters, then use these as the initial parameters for Model 2

- In 100 tests using this method, Model 2 converged to the correct point every time.

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- Decoding

Alignment parameters for good solution:

\[
T(i = 1 | j = 1, l = 2, m = 2) = 1 \\
T(i = 2 | j = 1, l = 2, m = 2) = 0 \\
T(i = 1 | j = 2, l = 2, m = 2) = 0 \\
T(i = 2 | j = 2, l = 2, m = 2) = 1 \\
\]

log probability = $-1.91$

Alignment parameters for first bad solution:

\[
T(i = 1 | j = 1, l = 2, m = 2) = 0 \\
T(i = 2 | j = 1, l = 2, m = 2) = 1 \\
T(i = 1 | j = 2, l = 2, m = 2) = 0 \\
T(i = 2 | j = 2, l = 2, m = 2) = 1 \\
\]

log probability = $-4.16$

Alignment parameters for second bad solution:

\[
T(i = 1 | j = 1, l = 2, m = 2) = 0 \\
T(i = 2 | j = 1, l = 2, m = 2) = 1 \\
T(i = 1 | j = 2, l = 2, m = 2) = 1 \\
T(i = 2 | j = 2, l = 2, m = 2) = 0 \\
\]

log probability = $-3.30$
Decoding

- Problem: for a given French sentence $f$, find
  \[
  \arg\max_{e} P(e)P(f \mid e)
  \]
or the “Viterbi approximation”
  \[
  \arg\max_{e,a} P(e)P(f, a \mid e)
  \]

First Stage of the Greedy Method

- For each French word $f_j$, pick the English word $e$ which maximizes
  \[
  T(e \mid f_j)
  \]
  (an inverse translation table $T(e \mid f)$ is required for this step)

- This gives us an initial alignment, e.g.,

Bien entendu , il parle de une belle victoire
Well heard , it talking NULL a beautiful victory

(Correct translation: quite naturally, he talks about a great victory)

Decoding

- Decoding is NP-complete (see (Knight, 1999))

- IBM papers describe a stack-decoding or $A^*$ search method

- A recent paper on decoding:
  *Fast Decoding and Optimal Decoding for Machine Translation.*

- Introduces a greedy search method

- Compares the two methods to exact (integer-programming) solution

Next Stage: Greedy Search

- First stage gives us an initial $(e^0, a^0)$ pair

- Basic idea: define a set of local transformations that map an $(e, a)$ pair to a new $(e', a')$ pair

- Say $\Pi(e, a)$ is the set of all $(e', a')$ reachable from $(e, a)$ by some transformation, then at each iteration take
  \[
  (e^t, a^t) = \arg\max_{(e, a) \in \Pi(e^{t-1}, a^{t-1})} P(e)P(f, a \mid e)
  \]
i.e., take the highest probability output from results of all transformations

- Basic idea: iterate this process until convergence
The Space of Transforms

- **CHANGE**\((j, e)\):
  Changes translation of \(f_j\) from \(e_{aj}\) into \(e\)

- Two possible cases (take \(e_{old} = e_{aj}\)):
  - \(e_{old}\) is aligned to more than 1 word, or \(e_{old} = NULL\)
    Place \(e\) at position in string that maximizes the alignment probability
  - \(e_{old}\) is aligned to exactly one word
    In this case, simply replace \(e_{old}\) with \(e\)

- Typically consider only \((e, f)\) pairs such that \(e\) is in top 10 ranked translations for \(f\) under \(T(e \mid f)\)
  (an inverse table of probabilities \(T(e \mid f)\) is required – this is described in Germann 2003)

---

TranslateAndInsert \((j, e_1, e_2)\):
Implements \(\text{CHANGE}(j, e_1)\),
(i.e. Changes translation of \(f_j\) from \(e_{aj}\) into \(e_1\)) and inserts \(e_2\) at most likely point in the string

- Typically, \(e_2\) is chosen from the English words which have high probability of being aligned to 0 French words

---

RemoveFertilityZero \((i)\):
Removes \(e_i\), providing that \(e_i\) is aligned to nothing in the alignment
The Space of Transforms

- **SwapSegments**($i_1, i_2, j_1, j_2$):
  Swaps words $e_{i_1} \ldots e_{i_2}$ with words $e_{j_1}$ and $e_{j_2}$

- **Note**: the two segments cannot overlap

---

An Example from Germann et. al 2001

Bien entendu, il parle de une belle victoire

Well heard, it *talking* NULL a *beautiful* victory

Bien entendu, il parle de une belle victoire

Well heard, it *talking* NULL a *great* victory

**CHANGE2**(5, *talks*, 8, *great*)

---

The Space of Transforms

- **JoinWords**($i_1, i_2$):
  Deletes English word at position $i_1$, and links all French words that were linked to $e_{i_1}$ to $e_{i_2}$

---

An Example from Germann et. al 2001

Bien entendu, il parle de une belle victoire

Well heard, it *talking* NULL a *great* victory

Bien entendu, il parle de une belle victoire

Well understood, it *talking* about a *great* victory

**CHANGE2**(2, *understood*, 6, *about*)
An Example from Germann et. al 2001

Bien intendu, il parle de une belle victoire

Well understood, it talks about a great victory

Bien intendu, il parle de une belle victoire

Well understood, he talks about a great victory

CHANGE(4, he)

An Exact Method Based on Integer Programming

Method from Germann et. al 2001:

- Integer programming problems
  
  \[ 3.2x_1 + 4.7x_2 - 2.1x_3 \quad \text{Minimize objective function} \]

\[ x_1 - 2.6x_3 > 5 \quad \text{Subject to linear constraints} \]

\[ 7.3x_2 > 7 \]

- Generalization of travelling salesman problem:
  Each town has a number of hotels; some hotels can be in multiple towns. Find the lowest cost tour of hotels such that each town is visited exactly once.

An Example from Germann et. al 2001

Bien intendu, il parle de une belle victoire

Well understood, he talks about a great victory

Bien intendu, il parle de une belle victoire

quite naturally, he talks about a great victory

CHANGE2(1, quite, 2, naturally)

- In the MT problem:
  - Each city is a French word (all cities visited ⇒ all French words must be accounted for)
  - Each hotel is an English word matched with one or more French words
  - The “cost” of moving from hotel \( i \) to hotel \( j \) is a sum of a number of terms. E.g., the cost of choosing “not” after “what”, and aligning it with “ne” and “pas” is

\[
\log(bigram(not | what)) + \\
\log(T(ne | not)) + \log(T(pas | not)) \\
\ldots
\]
An Exact Method Based on Integer Programming

- Say distance between hotels $i$ and $j$ is $d_{ij}$; Introduce $x_{ij}$ variables where $x_{ij} = 1$ if path from hotel $i$ to hotel $j$ is taken, zero otherwise

- Objective function: maximize
  \[ \sum_{i,j} x_{ij} d_{ij} \]

- All cities must be visited once $\Rightarrow$ constraints
  \[ \forall c \in \text{cities} \quad \sum_{i \text{ located in } c} \sum_{j} x_{ij} = 1 \]

- Every hotel must have equal number of incoming and outgoing edges $\Rightarrow$
  \[ \forall i \sum_{j} x_{ij} = \sum_{j} x_{ji} \]

- Another constraint is added to ensure that the tour is fully connected