The IBM Translation Models

Michael Collins, Columbia University

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:

$$p(e) \quad \textbf{the language model}$$

$$p(f \mid e) \quad \textbf{the translation model}$$

- 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

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

Roadmap for the Next Few Lectures

- IBM Models 1 and 2
- *Phrase-based* models

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

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.

IBM Model 1: Alignments

- e.g., $l = 6$, $m = 7$

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

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

- One alignment is

$$\{2, 3, 4, 5, 6, 6, 6\}$$

- Another (bad!) alignment is

$$\{1, 1, 1, 1, 1, 1, 1\}$$

Alignments in the IBM Models

- We'll define models for $p(a \mid e, m)$ and $p(f \mid a, e, m)$, giving

$$p(f, a \mid e, m) = p(a \mid e, m) p(f \mid a, e, m)$$

- Also,

$$p(f \mid e, m) = \sum_{a \in \mathcal{A}} p(a \mid e, m) p(f \mid a, e, m)$$

  where $\mathcal{A}$ is the set of all possible alignments

A By-Product: Most Likely Alignments

- Once we have a model $p(f, a \mid e, m) = p(a \mid e)p(f \mid a, e, m)$ we can also calculate

$$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\sum_{a \in \mathcal{A}} p(f, a \mid e, m)}$$

  for any alignment a

- For a given f, e pair, we can also compute the most likely alignment,

$$a^* = \arg\max_a p(a \mid f, e, m)$$

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

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/adopter our/nouvel opinion/avis ./.

IBM Model 1: Alignments

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

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

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

IBM Model 1: Translation Probabilities

- Next step: come up with an estimate for

$$p(f \mid a, e, m)$$

- In model 1, this is:

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

- e.g., $l = 6$, $m = 7$

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

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

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

$$\begin{aligned}
p(f \mid a, e) \quad = \quad & t(Le \mid the) \times \\
& t(programme \mid program) \times \\
& t(a \mid has) \times \\
& t(ete \mid been) \times \\
& t(mis \mid implemented) \times \\
& t(en \mid implemented) \times \\
& t(application \mid implemented)
\end{aligned}$$

IBM Model 1: The Generative Process

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

- **Step 1:** Pick an alignment a with probability $\frac{1}{(l+1)^m}$
- **Step 2:** Pick the French words with probability

$$p(f \mid a, e, m) = \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

The final result:

$$p(f, a \mid e, m) = p(a \mid e, m) \times p(f \mid a, e, m) = \frac{1}{(l+1)^m} \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

An Example Lexical Entry

English	French	Probability
position	position	0.756715
position	situation	0.0547918
position	mesure	0.0281663
position	vue	0.0169303
position	point	0.0124795
position	attitude	0.0108907

... 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 .

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

IBM Model 2

- Only difference: we now introduce **alignment** or **distortion** parameters

$$\mathbf{q}(i \mid j, l, m) \quad = \quad \text{Probability that } j\text{'th French word is connected to } i\text{'th English word, given sentence lengths of } e \text{ and } f \text{ are } l \text{ and } m \text{ respectively}$$

- Define

$$p(a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)$$

where $a = \{a_1, \ldots a_m\}$

- Gives

$$p(f, a \mid e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)\mathbf{t}(f_j \mid e_{a_j})$$

An Example

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

$$
\begin{aligned}
p(a \mid e, 7) = \; & \mathbf{q}(2 \mid 1, 6, 7) \times \\
& \mathbf{q}(3 \mid 2, 6, 7) \times \\
& \mathbf{q}(4 \mid 3, 6, 7) \times \\
& \mathbf{q}(5 \mid 4, 6, 7) \times \\
& \mathbf{q}(6 \mid 5, 6, 7) \times \\
& \mathbf{q}(6 \mid 6, 6, 7) \times \\
& \mathbf{q}(6 \mid 7, 6, 7)
\end{aligned}
$$

An Example

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

$$
\begin{aligned}
p(f \mid a, e, 7) \quad = \quad & \mathbf{t}(Le \mid the) \times \\
& \mathbf{t}(programme \mid program) \times \\
& \mathbf{t}(a \mid has) \times \\
& \mathbf{t}(ete \mid been) \times \\
& \mathbf{t}(mis \mid implemented) \times \\
& \mathbf{t}(en \mid implemented) \times \\
& \mathbf{t}(application \mid implemented)
\end{aligned}
$$

IBM Model 2: The Generative Process

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

- **Step 1:** Pick an alignment $a = \{a_1, a_2 \dots a_m\}$ with probability

$$\prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m)$$

- **Step 3:** Pick the French words with probability

$$p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{t}(f_j \mid e_{a_j})$$

The final result:

$$p(f, a \mid e, m) = p(a \mid e, m) p(f \mid a, e, m) = \prod_{j=1}^{m} \mathbf{q}(a_j \mid j, l, m) \mathbf{t}(f_j \mid e_{a_j})$$

Recovering Alignments

- If we have parameters q and t, we can easily recover the most likely alignment for any sentence pair
- Given a sentence pair $e_1, e_2, \ldots, e_l$, $f_1, f_2, \ldots, f_m$, define

$$a_j = \arg \max_{a \in \{0 \ldots l\}} q(a|j, l, m) \times t(f_j|e_a)$$

for $j = 1 \ldots m$

e = And the program has been implemented

f = Le programme a ete mis en application

Overview

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

The Parameter Estimation Problem

- Input to the parameter estimation algorithm: $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- Output: parameters $t(f|e)$ and $q(i|j, l, m)$
- A key challenge: **we do not have alignments on our training examples**, e.g.,

$$\begin{aligned} e^{(100)} &= \text{And the program has been implemented} \\ f^{(100)} &= \text{Le programme a ete mis en application} \end{aligned}$$

Parameter Estimation if the Alignments are Observed

- First: case where alignments are observed in training data. E.g.,

$$
\begin{aligned}
e^{(100)} &= \text{And the program has been implemented} \\
f^{(100)} &= \text{Le programme a ete mis en application} \\
a^{(100)} &= \langle 2, 3, 4, 5, 6, 6, 6 \rangle
\end{aligned}
$$

- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is an alignment
- Maximum-likelihood parameter estimates in this case are trivial:

$$
t_{ML}(f|e) = \frac{\text{Count}(e,\ f)}{\text{Count}(e)} \qquad q_{ML}(j|i,l,m) = \frac{\text{Count}(j|i,l,m)}{\text{Count}(i,l,m)}
$$

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \ldots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$,

$$
\begin{aligned}
c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
\end{aligned}
$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output: $t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Parameter Estimation with the EM Algorithm

- Training examples are $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- The algorithm is related to algorithm when alignments are observed, but two key differences:
  1. The algorithm is *iterative*. We start with some initial (e.g., random) choice for the q and t parameters. At each iteration we compute some "counts" based on the data together with our current parameter estimates. We then re-estimate our parameters with these counts, and iterate.
  2. We use the following definition for $\delta(k, i, j)$ at each iteration:

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}$$

Input: A training corpus $(f^{(k)}, e^{(k)})$ for $k = 1 \ldots n$, where $f^{(k)} = f_1^{(k)} \ldots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \ldots e_{l_k}^{(k)}$.

Initialization: Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters (e.g., to random values).

For $s = 1 \ldots S$

- Set all counts $c(\ldots) = 0$
- For $k = 1 \ldots n$
  - For $i = 1 \ldots m_k$, For $j = 0 \ldots l_k$

$$\begin{aligned} c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j) \\ c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k,i,j) \\ c(j|i,l,m) &\leftarrow c(j|i,l,m) + \delta(k,i,j) \\ c(i,l,m) &\leftarrow c(i,l,m) + \delta(k,i,j) \end{aligned}$$

where

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

- Recalculate the parameters:

$$t(f|e) = \frac{c(e,f)}{c(e)} \qquad q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$$

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}$$

$$\begin{aligned} e^{(100)} &= \text{And the program has been implemented} \\ f^{(100)} &= \text{Le programme a ete mis en application} \end{aligned}$$

Justification for the Algorithm

- Training examples are $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence

- The log-likelihood function:

$$L(t, q) = \sum_{k=1}^{n} \log p(f^{(k)}|e^{(k)}) = \sum_{k=1}^{n} \log \sum_{a} p(f^{(k)}, a|e^{(k)})$$

- The maximum-likelihood estimates are

$$\arg\max_{t,q} L(t, q)$$

- The EM algorithm will converge to a *local maximum* of the log-likelihood function

Summary

- Key ideas in the IBM translation models:
  - Alignment variables
  - Translation parameters, e.g., $t(\text{chien}|\text{dog})$
  - Distortion parameters, e.g., $q(2|1, 6, 7)$
- The EM algorithm: an iterative algorithm for training the q and t parameters
- Once the parameters are trained, we can recover the most likely alignments on our training examples

$$
\begin{aligned}
e &= \text{And the program has been implemented} \\
f &= \text{Le programme a ete mis en application}
\end{aligned}
$$