COMS 4705 (Fall 2011)

Machine Translation Part II

Recap: The Noisy Channel Model

- Goal: translation system from French to English

- Have a model $p(\mathbf{e} \mid \mathbf{f})$ which estimates conditional probability of any English sentence $\mathbf{e}$ given the French sentence $\mathbf{f}$. Use the training corpus to set the parameters.

- A Noisy Channel Model has two components:

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

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

- Giving:

$$p(\mathbf{e} \mid \mathbf{f}) = \frac{p(\mathbf{e}, \mathbf{f})}{p(\mathbf{f})} = \frac{p(\mathbf{e})p(\mathbf{f} \mid \mathbf{e})}{\sum_{\mathbf{e}} p(\mathbf{e})p(\mathbf{f} \mid \mathbf{e})}$$

  and

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

Roadmap for the Next Few Lectures

- Lecture 1 (today): IBM Models 1 and 2
- Lecture 2: *phrase-based* models
- Lecture 3: Syntax in statistical machine translation

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 1: Alignments

- How do we model $p(\mathbf{f} \mid \mathbf{e})$?

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

- An **alignment** $\mathbf{a}$ identifies which English word each French word originated from

- Formally, an **alignment** $\mathbf{a}$ 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., $l = 6$, $m = 7$

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

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

- 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 $p(\mathbf{a} \mid \mathbf{e})$ and $p(\mathbf{f} \mid \mathbf{a}, \mathbf{e})$, giving

$$p(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = p(\mathbf{a} \mid \mathbf{e})p(\mathbf{f} \mid \mathbf{a}, \mathbf{e})$$

- Also,

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

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

A By-Product: Most Likely Alignments

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

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

  for any alignment $\mathbf{a}$

- For a given $\mathbf{f}, \mathbf{e}$ pair, we can also compute the most likely alignment,

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{a} \mid \mathbf{f}, \mathbf{e})$$

- 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 $\mathbf{a}$ are equally likely:

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

where $C = prob(length(\mathbf{f}) = m)$ 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

$$p(\mathbf{f} \mid \mathbf{a}, \mathbf{e})$$

- In model 1, this is:

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

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

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

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

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

IBM Model 1: The Generative Process

To generate a French string $\mathbf{f}$ from an English string $\mathbf{e}$:

- **Step 1:** Pick the length of $\mathbf{f}$ (all lengths equally probable, probability C)

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

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

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

The final result:

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

An Example

- I have the following training examples

the dog ⇒ le chien
the cat ⇒ le chat

- Need to find estimates for:

$$p(le \mid the) \quad p(chien \mid the) \quad p(chat \mid the)$$

$$p(le \mid dog) \quad p(chien \mid dog) \quad p(chat \mid dog)$$

$$p(le \mid cat) \quad p(chien \mid cat) \quad p(chat \mid cat)$$

- As a result, each $(\mathbf{e}_i, \mathbf{f}_i)$ pair will have a most likely alignment.

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 .

. . . 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

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

- Define

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

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

- Gives

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

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

An Example

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

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

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

IBM Model 2: The Generative Process

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

- **Step 1:** Pick the length of **f** (all lengths equally probable, probability C)
- **Step 2:** Pick an alignment $\mathbf{a} = \{a_1, a_2 \ldots a_m\}$ with probability

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

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

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

The final result:

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

Overview

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

Overview

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

An Example of Training Models 1 and 2

Example will use following translations:

e[1]	=	the	dog
f[1]	=	le	chien
e[2]	=	the	cat
f[2]	=	le	chat
e[3]	=	the	bus
f[3]	=	l’	autobus

NB: I won’t use a NULL word e_0

Initial (random) parameters:

e	f	$\mathbf{t}(f \mid e)$
the	le	0.23
the	chien	0.2
the	chat	0.11
the	l’	0.25
the	autobus	0.21
dog	le	0.2
dog	chien	0.16
dog	chat	0.33
dog	l’	0.12
dog	autobus	0.18
cat	le	0.26
cat	chien	0.28
cat	chat	0.19
cat	l’	0.24
cat	autobus	0.03
bus	le	0.22
bus	chien	0.05
bus	chat	0.26
bus	l’	0.19
bus	autobus	0.27

Alignment probabilities:

i	j	k	a(i,j,k)
1	1	0	0.526423237959726
2	1	0	0.473576762040274
1	2	0	0.552517995605817
2	2	0	0.447482004394183
1	1	1	0.466532602066533
2	1	1	0.533467397933467
1	2	1	0.356364544422507
2	2	1	0.643635455577493
1	1	2	0.571950438336247
2	1	2	0.428049561663753
1	2	2	0.439081311724508
2	2	2	0.560918688275492

Expected counts:

e	f	$tcount(e, f)$
the	le	0.99295584002626
the	chien	0.552517995605817
the	chat	0.356364544422507
the	l’	0.571950438336247
the	autobus	0.439081311724508
dog	le	0.473576762040274
dog	chien	0.447482004394183
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.533467397933467
cat	chien	0
cat	chat	0.643635455577493
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.428049561663753
bus	autobus	0.560918688275492

Old and new parameters:

e	f	old	new
the	le	0.23	0.34
the	chien	0.2	0.19
the	chat	0.11	0.12
the	l’	0.25	0.2
the	autobus	0.21	0.15
dog	le	0.2	0.51
dog	chien	0.16	0.49
dog	chat	0.33	0
dog	l’	0.12	0
dog	autobus	0.18	0
cat	le	0.26	0.45
cat	chien	0.28	0
cat	chat	0.19	0.55
cat	l’	0.24	0
cat	autobus	0.03	0
bus	le	0.22	0
bus	chien	0.05	0
bus	chat	0.26	0
bus	l’	0.19	0.43
bus	autobus	0.27	0.57

e	f						
the	le	0.23	0.34	0.46	0.56	0.64	0.71
the	chien	0.2	0.19	0.15	0.12	0.09	0.06
the	chat	0.11	0.12	0.1	0.08	0.06	0.04
the	l’	0.25	0.2	0.17	0.15	0.13	0.11
the	autobus	0.21	0.15	0.12	0.1	0.08	0.07
dog	le	0.2	0.51	0.46	0.39	0.33	0.28
dog	chien	0.16	0.49	0.54	0.61	0.67	0.72
dog	chat	0.33	0	0	0	0	0
dog	l’	0.12	0	0	0	0	0
dog	autobus	0.18	0	0	0	0	0
cat	le	0.26	0.45	0.41	0.36	0.3	0.26
cat	chien	0.28	0	0	0	0	0
cat	chat	0.19	0.55	0.59	0.64	0.7	0.74
cat	l’	0.24	0	0	0	0	0
cat	autobus	0.03	0	0	0	0	0
bus	le	0.22	0	0	0	0	0
bus	chien	0.05	0	0	0	0	0
bus	chat	0.26	0	0	0	0	0
bus	l’	0.19	0.43	0.47	0.47	0.47	0.48
bus	autobus	0.27	0.57	0.53	0.53	0.53	0.52

After 20 iterations:

e	f	
the	le	0.94
the	chien	0
the	chat	0
the	l’	0.03
the	autobus	0.02
dog	le	0.06
dog	chien	0.94
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.06
cat	chien	0
cat	chat	0.94
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.49
bus	autobus	0.51

Model 2 has several local maxima – good one:

e	f	$t(f \mid e)$
the	le	0.67
the	chien	0
the	chat	0
the	l'	0.33
the	autobus	0
dog	le	0
dog	chien	1
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0
cat	chien	0
cat	chat	1
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0
bus	autobus	1

Model 2 has several local maxima – bad one:

e	f	$\mathbf{t}(f \mid e)$
the	le	0
the	chien	0.4
the	chat	0.3
the	l’	0
the	autobus	0.3
dog	le	0.5
dog	chien	0.5
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.5
cat	chien	0
cat	chat	0.5
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.5
bus	autobus	0.5

another bad one:

e	f	$\mathbf{t}(f \mid e)$
the	le	0
the	chien	0.33
the	chat	0.33
the	l’	0
the	autobus	0.33
dog	le	1
dog	chien	0
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	1
cat	chien	0
cat	chat	0
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	1
bus	autobus	0

- Alignment parameters for good solution:

$$\begin{aligned}
q(i=1 \mid j=1, l=2, m=2) &= 1 \\
q(i=2 \mid j=1, l=2, m=2) &= 0 \\
q(i=1 \mid j=2, l=2, m=2) &= 0 \\
q(i=2 \mid j=2, l=2, m=2) &= 1
\end{aligned}$$

log probability $= -1.91$

- Alignment parameters for first bad solution:

$$\begin{aligned}
q(i=1 \mid j=1, l=2, m=2) &= 0 \\
q(i=2 \mid j=1, l=2, m=2) &= 1 \\
q(i=1 \mid j=2, l=2, m=2) &= 0 \\
q(i=2 \mid j=2, l=2, m=2) &= 1
\end{aligned}$$

log probability $= -4.16$

- Alignment parameters for second bad solution:

$$
\begin{aligned}
q(i=1 \mid j=1, l=2, m=2) &= 0 \\
q(i=2 \mid j=1, l=2, m=2) &= 1 \\
q(i=1 \mid j=2, l=2, m=2) &= 1 \\
q(i=2 \mid j=2, l=2, m=2) &= 0
\end{aligned}
$$

log probability $= -3.30$

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