6.891: Lecture 10 (October 8th, 2003)

Machine Translation Part I

Overview

- Challenges in machine translation
- A brief introduction to statistical MT
- Evaluation of MT systems
- The sentence alignment problem
- IBM Model 1

Lexical Ambiguity

Example 1:

book the flight $\Rightarrow$ reservar

read the book $\Rightarrow$ libro

Example 2:

the box was in the pen

the pen was on the table

Example 3:

kill a man $\Rightarrow$ matar

kill a process $\Rightarrow$ acabar

Differing Word Orders

- English word order is *subject – verb – object*

- Japanese word order is *subject – object – verb*

English: IBM bought Lotus
Japanese: *IBM Lotus bought*

English: Sources said that IBM bought Lotus yesterday
Japanese: *Sources yesterday IBM Lotus bought that said*

Syntactic Structure is not Preserved Across Translations

The bottle floated into the cave

$\Downarrow$

La botella entro a la cuerva flotando
(the bottle entered the cave floating)

Syntactic Ambiguity Causes Problems

John hit the dog with the stick

$\Downarrow$

John golpeo el perro con el palo/que tenia el palo

Pronoun Resolution

The computer outputs the data; it is fast.

$\Downarrow$

La computadora imprime los datos; es rapida

The computer outputs the data; it is stored in ascii.

$\Downarrow$

La computadora imprime los datos; estan almacendos en ascii

Differing Treatments of Tense

From Dorr et. al 1998:

Mary went to Mexico. During her stay she learned Spanish.

Went $\Rightarrow$ iba (simple past/preterit)

Mary went to Mexico. When she returned she started to speak Spanish.

Went $\Rightarrow$ fue (ongoing past/imperfect)

The Best Translation May not be 1-1

(From Manning and Schuetze):

According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above average growth rates.

$\Rightarrow$

Quant aux eaux minerales et aux limonades, elles recontrent toujours plus d'adeptes. En effet notre sondage fait ressortir des ventes nettement superieures a celles de 1987, pour les boissons a base de cola notamment.

With regard to the mineral waters and the lemonades (soft drinks) they encounter still more users. Indeed our survey makes stand out the sales clearly superior to those in 1987 for cola-based drinks especially.

From Babel Fish:

Aznar ha premiado a Rodrigo Rato (vicepresidente primero), Javier Arenas (vicepresidente segundo y ministro de la Presidencia) y Eduardo Zaplana (ministro portavoz y titular de Trabajo) en la septima remodelacion de Gobierno en sus dos legislaturas. Las caras nuevas del Ejecutivo son las de Juan Costa, al frente del Ministerio de Ciencia y Tecnologia, y la de Julia Garcia Valdecasas, que ocupara la cartera de Administraciones Publicas.

$$\Downarrow$$

Aznar has awarded to Rodrigo Short while (vice-president first), Javier Sands (vice-president second and minister of the Presidency) and Eduardo Zaplana (minister spokesman and holder of Work) in the seventh remodeling of Government in its two legislatures. The new faces of the Executive are those of Juan Coast, to the front of the Ministry of Science and Technology, and the one of Julia Garci'a Valdecasas, who will occupy the portfolio of Public Administrations.

Overview

- Challenges in machine translation
- A brief introduction to statistical MT
- Evaluation of MT systems
- The sentence alignment problem
- IBM Model 1

A Brief Introduction to Statistical MT

- Parallel corpora are available in several language pairs

- Basic idea: use a parallel corpus as a training set of translation examples

- Classic example: IBM work on French-English translation, using the Canadian Hansards. (1.7 million sentences of 30 words or less in length).

- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation.

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

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

More About the Noisy Channel Model

- The **language model** $P(e)$ could be a trigram model, estimated from any data (parallel corpus not needed to estimate the parameters)

- The **translation model** $P(f \mid e)$ is trained from a parallel corpus of French/English pairs.

- Note:
  - The translation model is backwards!
  - The language model can make up for deficiencies of the translation model.
  - Later we'll talk about how to build $P(f \mid e)$
  - Decoding, i.e., finding

    $$\mathrm{argmax}_e P(e)P(f \mid e)$$

    is also a challenging problem.

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $P(Spanish \mid English)$ alone:

Que hambre tengo yo
$\rightarrow$

What hunger have	$P(S\|E) = 0.000014$
Hungry I am so	$P(S\|E) = 0.000001$
I am so hungry	$P(S\|E) = 0.0000015$
Have i that hunger	$P(S\|E) = 0.000020$

$\ldots$

With $P(Spanish \mid English) \times P(English)$:

Que hambre tengo yo

$\rightarrow$

What hunger have	$P(S\|E)P(E) = 0.000014 \times 0.000001$
Hungry I am so	$P(S\|E)P(E) = 0.000001 \times 0.0000014$
I am so hungry	$P(S\|E)P(E) = 0.0000015 \times 0.0001$
Have i that hunger	$P(S\|E)P(E) = 0.000020 \times 0.00000098$

$\ldots$

Overview

- Challenges in machine translation
- A brief introduction to statistical MT
- Evaluation of MT systems
- The sentence alignment problem
- IBM Model 1

Evaluation of Machine Translation Systems

- Method 1: human evaluations
  accurate, **but** expensive, slow

- "Cheap" and fast evaluation is essential

- We'll discuss one prominent method:
  Bleu (Papineni, Roukos, Ward and Zhu, 2002)

Evaluation of Machine Translation Systems

Bleu (Papineni, Roukos, Ward and Zhu, 2002):

Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.

Candidate 2: It is to insure the troops forever hearing the activity guidebook that party direct.

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

Reference 3: It is the practical guide for the army always to heed the directions of the party.

Unigram Precision

- **Unigram Precision** of a candidate translation:

$$\frac{C}{N}$$

where N is number of words in the candidate, C is the number of words in the candidate which are in at least one reference translation.

e.g.,

Candidate 1: It is a guide to action which ensures that the military always obeys the commands of the party.

$$Precision = \frac{17}{18}$$

(only *obeys* is missing from all reference translations)

Modified Unigram Precision

- Problem with unigram precision:

  Candidate: the the the the the the the

  Reference 1: the cat sat on the mat

  Reference 2: there is a cat on the mat

  precision = 7/7 = 1???

- **Modified unigram precision:** “Clipping”
  - Each word has a “cap”. e.g., *cap(the) = 2*
  - A candidate word w can only be correct a maximum of $cap(w)$ times. e.g., in candidate above, $cap(the) = 2$, and *the* is correct twice in the candidate $\Rightarrow$

$$Precision = \frac{2}{7}$$

Modified N-gram Precision

- Can generalize modified unigram precision to other n-grams.

- For example, for candidates 1 and 2 above:

$$Precision_1(bigram) = \frac{10}{17}$$

$$Precision_2(bigram) = \frac{1}{13}$$

Precision Alone Isn't Enough

Candidate 1: of the

Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

Reference 3: It is the practical guide for the army always to heed the directions of the party.

$$Precision(unigram) = 1$$

$$Precision(bigram) = 1$$

But Recall isn't Useful in this Case

- Standard measure used in addition to precision is **recall**:

$$Recall = \frac{C}{N}$$

where C is number of n-grams in candidate that are correct, N is number of words in the references.

Candidate 1: I always invariably perpetually do.

Candidate 2: I always do

Reference 1: I always do

Reference 1: I invariably do

Reference 1: I perpetually do

Sentence Brevity Penalty

- Step 1: for each candidate, compute closest matching reference (in terms of length)
  e.g., our candidate is length 12, references are length $12, 15, 17$. Best match is of length 12.

- Step 2: Say l_i is the length of the i'th candidate, r_i is length of best match for the i'th candidate, then compute

$$brevity = \frac{\sum_i r_i}{\sum_i l_i}$$

  (I think! from the Papineni paper, although $brevity = \frac{\sum_i r_i}{\sum_i min(l_i, r_i)}$ might make more sense?)

- Step 3: compute brevity penalty

$$BP = \begin{cases} 1 & \text{If } brevity < 1 \\ e^{1-brevity} & \text{If } brevity \geq 1 \end{cases}$$

  e.g., if $r_i = 1.1 \times l_i$ for all i (candidates are always 10% too short) then $BP = e^{-0.1} = 0.905$

The Final Score

- Corpus precision for any n-gram is

$$p_n = \frac{\sum_{C \in \{Candidate\}} \sum_{ngram \in C} \quad Count_{clip}(ngram)}{\sum_{C \in \{Candidate\}} \sum_{ngram \in C} \quad Count(ngram)}$$

i.e. number of correct ngrams in the candidates (after "clipping") divided by total number of ngrams in the candidates

- Final score is then

$$Bleu = BP \times (p_1 p_2 p_3 p_4)^{1/4}$$

i.e., BP multiplied by the geometric mean of the unigram, bigram, trigram, and four-gram precisions

Overview

- Challenges in machine translation
- A brief introduction to statistical MT
- Evaluation of MT systems
- The sentence alignment problem
- IBM Model 1

The Sentence Alignment Problem

- Might have 1003 sentences (in sequence) of English, 987 sentences (in sequence) of French: **but which English sentence(s) corresponds to which French sentence(s)?**

$$
\begin{array}{ll}
e_1 & f_1 \\
e_2 & f_2 \\
e_3 & f_3 \\
e_4 & f_4 \\
e_5 & f_5 \\
e_6 & f_6 \\
e_7 & f_7 \\
\ldots &
\end{array}
\quad \Rightarrow \quad
\begin{array}{ll}
e_1 & f_1 \\
e_2 & \\
\text{-----} & \text{-----} \\
e_3 & f_2 \\
\text{-----} & \text{-----} \\
e_4 & f_3 \\
\text{-----} & \text{-----} \\
e_5 & f_4 \\
 & f_5 \\
\text{-----} & \text{-----} \\
e_6 & f_6 \\
e_7 & f_7 \\
\text{-----} & \text{-----} \\
\ldots &
\end{array}
$$

- Might have 1-1 alignments, 1-2, 2-1, 2-2 etc.

The Sentence Alignment Problem

- Clearly needed before we can train a translation model
- Also useful for other multi-lingual problems
- Two broad classes of methods we'll cover:
  - Methods based on sentence lengths alone.
  - Methods based on lexical matches, or "cognates".

Sentence Length Methods

(Gale and Church, 1993):

- Method assumes paragraph alignment is known, sentence alignment is not known.
- Define:
  - l_e = length of English sentence, in characters
  - l_f = length of French sentence, in characters
- Assumption: given length l_e, length l_f has a gaussian/normal distribution with mean $c \times l_e$, and variance $s^2 \times l_e$ for some constants c and s.
- Result: we have a cost

$$Cost(l_e, l_f)$$

  for any pairs of lengths l_e and l_f.

Each Possible Alignment Has a Cost

e_1 f_1
e_2

----

e_3 f_2

----

e_4 f_3

----

e_5 f_4
f_5

----

e_6 f_6
e_7 f_7

...

In this case, if length of e_i is l_i, and length of f_i is m_i, total cost is

$$\begin{aligned} Cost = & Cost(l_1 + l_2, m_1) + Cost_{21} + \\ & Cost(l_3, m_2) + Cost_{11} + \\ & Cost(l_4, m_3) + Cost_{11} + \\ & Cost(l_4, m_4 + m_5) + Cost_{12} + \\ & Cost(l_6 + l_7, m_6 + m_7) + Cost_{22} \end{aligned}$$

where $Cost_{ij}$ terms correspond to costs for 1-1, 1-2, 2-1 and 2-2 alignments.

- Dynamic programming can be used to search for the lowest cost alignment

Methods Based on Cognates

- Intuition: related words in different languages often have similar spellings e.g., government and gouvernement

- Cognate matches can "anchor" sentence-sentence correspondences

- A method from (Church 1993): track all 4-grams of characters which are identical in the two texts.

- A method from (Melamed 1993), measures similarity of words A and B:

$$LCSR(A, B) = \frac{length(LCS(A, B))}{max(length(A), length(B))}$$

  where LCS is the longest common subsequence (not necessarily contiguous) in A and B. e.g.,

$$LCSR(\text{government,gouvernement}) = \frac{10}{13}$$

More on Melamed's Definition of Cognates

- Various refinements (for example, excluding common/stop words such as "the", "a")

- Melamed uses a cut-off of 0.58 for LCSR to identify cognates: 25% of words in Hansards are then part of a cognate

- Represent an English/French parallel text e/f as a "bitext": graph where we have a point at position (x, y) if and only if $word_x$ in e is a cognate of $word_y$ in f.

- Melamed then uses a greedy method to identify a diagonal chain of cognates through the parallel text.

Overview

- Challenges in machine translation
- A brief introduction to statistical MT
- Evaluation of MT systems
- The sentence alignment problem
- IBM Model 1
  - How do we model $P(f \mid e)$?

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.

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

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

IBM Model 1: Alignments

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

$$P(A \mid e) = C \times \frac{1}{(l+1)^m}$$

  where $C = prob(length(f) = m)$ 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:

$$P(f \mid A, e) = \prod_{j=1}^{m} P(f_j \mid e_{a_j})$$

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

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

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

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

IBM Model 1: The Generative Process

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

- **Step 1:** Pick 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

$$P(f \mid A, e) = \prod_{j=1}^{m} P(f_j \mid e_{a_j})$$

The final result:

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

A Hidden Variable Problem

- **We have:**

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

- **And:**

$$P(f \mid e) = \sum_{A \in \mathcal{A}} \frac{C}{(l+1)^m} \prod_{j=1}^{m} P(f_j \mid e_{a_j})$$

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

A Hidden Variable Problem

- Training data is a set of (f_i, e_i) pairs, likelihood is

$$\sum_i \log P(f \mid e) = \sum_i \log \sum_{A \in \mathcal{A}} P(A \mid e_i) P(f_i \mid A, e_i)$$

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

- We need to maximize this function w.r.t. the translation parameters $P(f_j \mid e_{a_j})$.

- EM can be used for this problem: initialize translation parameters randomly, and at each iteration choose

$$\Theta_t = \text{argmax}_\Theta \sum_i \sum_{A \in \mathcal{A}} P(A \mid e_i, f_i, \Theta^{t-1}) \log P(f_i \mid A, e_i, \Theta)$$

  where Θ^t are the parameter values at the t'th iteration.

An Example

- I have the following training examples

the dog $\Rightarrow$ le chien
the cat $\Rightarrow$ le chat

- Need to find estimates for:

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

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

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

- As a result, each (e_i, f_i) pair will have a most likely alignment.