Part II: NLP

Applications: Statistical Machine Translation

Stephen Clark

How do Google do it?

- "Nobody in my team is able to read Chinese characters," says Franz Och, who heads Google 's machine-translation (MT) effort. Yet, they are producing ever more accurate translations into and out of Chinese - and several other languages as well. (www.csmonitor.com/2005/0602/p 13s02-stct.html)

- Typical (garbled) translation from MT software: "Alpine white new presence tape registered for coffee confirms Laden."

- Google translation: "The White House confirmed the existence of a new Bin Laden tape."

A Long History

- Machine Translation (MT) was one of the first applications envisaged for computers

- Warren Weaver (1949):
  I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.

- First demonstrated by IBM in 1954 with a basic word-for-word translation system.

- But MT was found to be much harder than expected (for reasons we'll see)

Commercially/Politically Interesting

- EU spends more than 1,000,000,000 Euro on translation costs each year - even semi-automation would save a lot of money
- U.S. has invested heavily in MT for Intelligence purposes
- Original MT research looked at Russian $\rightarrow$ English
  - What are the popular language pairs now?

Academically Interesting

- Computer Science, Linguistics, Languages, Statistics, AI
- The "holy grail" of AI
  - MT is "AI-hard": requires a solution to the general AI problem of representing and reasoning about (inference) various kinds of knowledge (linguistic, world ...)
  - or does it? ...
  - the methods Google use make no pretence at solving the difficult problems of AI (and it's debatable how accurate these methods can get)

Why is MT Hard

- Word order
- Word sense
- Pronouns
- Tense
- Idioms

Differing Word Orders

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

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

- English: *Reporters said IBM bought Lotus*
  Japanese: *Reporters IBM Lotus bought said*

Word Sense Ambiguity

- *Bank* as in river
  Bank as in financial institution

- *Plant* as in tree
  Plant as in factory

- Different word senses will likely translate into different words in another language

Pronouns

- Japanese is an example of a **pro-drop** language

- *Kono kēki wa oishii. Dare ga yaita no?*
  This cake TOPIC tasty. Who SUBJECT made?
  This cake is tasty. Who made **it**?

- *Shiranai. Ki ni itta?*
  know-NEGATIVE. liked?
  I don't know. Do **you** like **it**?

[examples from Wikipedia]

Pronouns

- Some languages like Spanish can drop subject pronouns

- In Spanish the verbal inflection often indicates which pronoun should be restored (but not always)
  -o = I
  -as = you
  -a = he/she/it
  -amos = we
  -an they

- When should the MT system use *she*, *he* or *it*?

Different Tenses

- Spanish has two versions of the past tense: one for a definite time in the past, and one for an unknown time in the past

- When translating **from English to Spanish** we need to choose which version of the past tense to use

Idioms

- "to kick the bucket" means "to die"
- "a bone of contention" has nothing to do with skeletons
- "a lame duck", "tongue in cheek", "to cave in"

Various Approaches to MT

- Word-for-word translation
- Syntactic transfer
- **Interlingual approaches**
- Example-based translation
- **Statistical translation**

Interlingua

- Assign a logical form (meaning representation) to sentences
- *John must not go* =
  OBLIGATORY(NOT(GO(JOHN)))
  John may not go =
  NOT(PERMITTED(GO(JOHN)))
- Use logical form to generate a sentence in another language

(wagon-wheel picture)

Statistical Machine Translation

- Find *most probable* English sentence given a foreign language sentence
- Automatically align words and phrases within sentence pairs in a **parallel corpus**
- Probabilities are determined automatically by training a statistical model using the parallel corpus
  (pdf of parallel corpus)

Probabilities

- Find the most probable English sentence given a foreign language sentence
  (this is often how the problem is framed - of course can be generalised to any language pair in any direction)

$$\begin{aligned}
\widehat{e} &= \arg\max_{e} p(e|f) \\
&= \arg\max_{e} \frac{p(f|e)p(e)}{p(f)} \\
&= \arg\max_{e} p(f|e)p(e)
\end{aligned}$$

Individual Models

- $p(f|e)$ is the *translation model*
  (note the reverse ordering of f and e due to Bayes)
  - assigns a higher probability to English sentences that have the same meaning as the foreign sentence
  - needs a bilingual (parallel) corpus for estimation
- $p(e)$ is the *language model*
  - assigns a higher probability to fluent/grammatical sentences
  - only needs a monolingual corpus for estimation (which are plentiful)

(picture of mt system: translation model, language model, search)

Translation Model

- $p(f|e)$ - the probability of some foreign language string given a hypothesis English translation
- $f =$ Ces gens ont grandi, vecu et oeuvre des dizaines d'annees dans le domaine agricole.
- $e =$ *Those people have grown up, lived and worked many years in a farming district.*
- $e =$ *I like bungee jumping off high bridges.*
- Allowing highly improbable translations (but assigning them small probabilities) was a radical change in how to think about the MT problem

Translation Model

- Introduce alignment variable a which represents alignments between the individual words in the sentence pair
- $p(f|e) = \sum_a p(a, f|e)$

(word alignment diagram)

Alignment Probabilities

- Now break the sentences up into manageable chunks (initially just the words)

- $p(a, f|e) = \prod_{j=1}^{m} t(f_j|e_i)$

where e_i is the English word(s) corresponding to the French word f_j and $t(f_j|e_i)$ is th e (conditional) probability of the words being aligned
(alignment diagram)

Alignment Probabilities

- Relative frequency estimates can be used to estimate $t(f_j|e_i)$
- Problem is that we don't have *word*-aligned data, only sentence-aligned
- There is an elegant mathematical solution to this problem - the EM algorithm

References

- www.statmt.org has some excellent introductory tutorials, and also the classic IBM paper (Brown, Della Petra, Della Petra and Mercer)

- Foundations of Statistical Natural Language Processing, Manning and Schutze, ch. 13

- Speech and Language Processing, Jurafsky and Martin, ch. 21