ACS Statistical Machine Translation

Lecture 1: Introduction to MT and SMT

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

sc609@cam.ac.uk

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/p13s02-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 → 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 we will investigate make no pretence at solving the difficult problems of AI (and it’s debatable how accurate these methods can get)

Why is MT Difficult?

- 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

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

Word-for-Word Translation

- Use a machine-readable bilingual dictionary to translate each word in a text
- Advantages:
  - easy to implement
  - results give a rough idea of what the text is about (perhaps)
- Disadvantages:
  - no account of word order
  - dictionary doesn't tell us which word to translate to in the case of polysemous words
  - results in low-quality translation

Syntactic Transfer

- Parse the sentence
- Rearrange constituents (grammatical units)
- Translate the words

Syntactic Transfer

- Advantages:
  - deals with the word order problem
- Disadvantages:
  - need to automatically analyse (parse) the sentence in the source language
  - need to construct transfer rules for each possible language pair
  - sometimes there is a syntactic mismatch:
    The bottle floated into the cave
    La botella entro a la cuerva flotando =
    The bottle entered the cave floating (Spanish)

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)

Interlingua

- Advantages:
  - single logical form means that we can translate between all languages and only write a parser/generator for each language once ($2n$ vs. n^2 systems)
- Disadvantages:
  - difficult to define a single logical form (English words in all capital letters probably won't do)
  - difficult to create parsers and generators, even if we can agree on the representation
- Researchers going back to this idea (search for "semantics-based machine translation" in Google)

Example-Based MT

- Fundamental idea:
  - human translators do not translate by performing deep linguistic analysis
  - they translate by decomposing a sentence into fragments, translating each of those, and then composing the individual translations
- Translate the parts *by analogy*
  - similar to case-based reasoning, instance-based reasoning, analogical-based reasoning, ... seen in AI, psychology, ...

Example of Example-Based MT

- Translate *He buys a book on international politics* into Japanese with the examples:
  - *He buys a notebook*
    Kare ha nouto wo kau
  - *I read a book on international politics*
    Watashi ha kokusaiseiji nitsuite kakareta hon wo yomu

Challenges

- Locating similar sentences
- Aligning sub-sentential fragments
- Combining multiple fragments of example translations into a single sentence
- Selecting the best translation out of many candidates

Example-Based MT

- Advantages:
  - uses fragments of human translations which can result in higher quality
- Disadvantages:
  - may have limited coverage depending on the size of the example database, and the flexibility of the matching heuristics

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)

Statistical Machine Translation

- Advantages:
  - has a way of dealing with lexical ambiguity
  - requires minimal human effort
  - can be created for any language pair that has enough training data
- Disadvantages:
  - does not explicity deal with syntax (reordering is performed at the word or phrase level)
  - requires a large parallel corpus
- Hybrid models are possible (eg hybrid EBMT/SMT, syntax-based SMT) and much recent research is concerned with improving the basic SMT model

Choosing an Approach

- Many challenges in MT, many different ways of approaching the task
- What approach you prefer may depend on your background (eg logicians go for interlingua, linguists syntactic transfer)
- Objectively choosing a method is tricky

Some Criteria

- Do we want to design a system for a single language or many languages?
- Can we assume a constrained vocabulary or do we need to deal with unrestricted text?
- What resources already exist for the languages that we’re dealing with?
- How long will it take us to develop the resources, and how large a staff will we need?

Advantages of SMT

- Data driven
- Language independent
- No need for staff of linguists or language experts
- Can prototype a new system quickly and at low cost

Choosing SMT (Google have)

- Economic reasons:
  - low cost
  - rapid prototyping
- Practical reasons:
  - many language pairs don't have NLP resources, but do have parallel corpora
- Quality reasons:
  - uses chunks of human translations as its building blocks
  - produces state-of-the-art results when very large data sets are available

Readings

- Statistical Machine Translation, Philipp Koehn, CUP, 2010
- www.statmt.org has some excellent introductory tutorials (including the ESSLLI tutorial by Callison-Burch and Koehn, on which these slides are based), 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
- The Unreasonable Effectiveness of Data, IEEE Intelligent Systems, vol. 24 (2009), available from http://research.google.com/pubs/author1092.html

Probabilities for SMT

- 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)

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

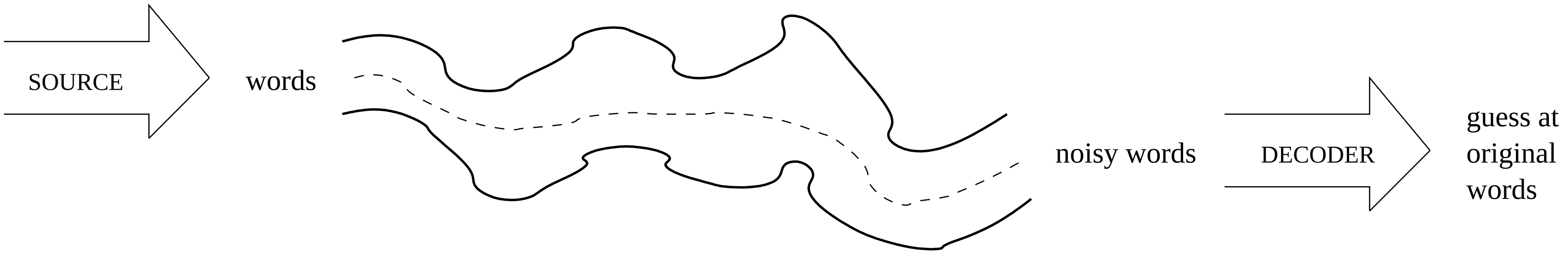
Individual Models

- $p(f|e)$ 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)

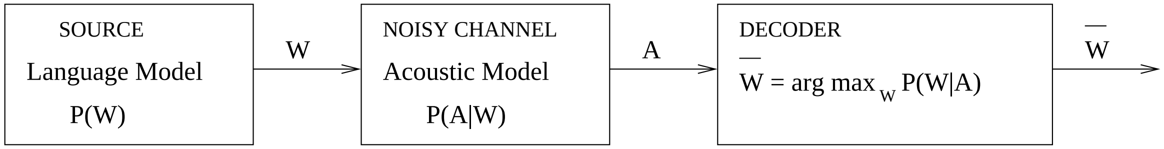
(picture of mt system: translation model, language model, search)

Model Interpretation as a Noisy Channel

- **Noisy channel** model has been applied to many language processing problems
- Based on the notion of a noisy channel from Shannon's information theory
- First applied to a language problem by the speech recognition group at IBM in the 70s



Speech Recognition as a Noisy Channel



SOURCE
Language Model
P(W)

W

NOISY CHANNEL
Acoustic Model
P(A|W)

A

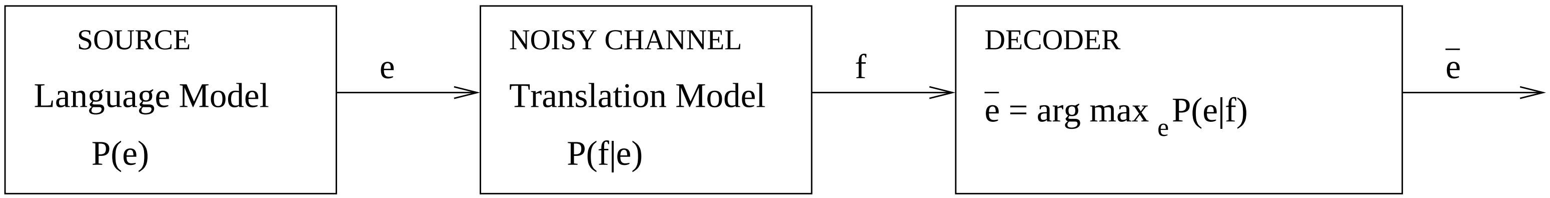
DECODER
$\overline{\text{W}}$ = arg max $_{\text{W}}$ P(W|A)

$\overline{\text{W}}$

- Speaker has word sequence W
- W is articulated as acoustic sequence A
- This process introduces noise:
  - variation in pronunciation
  - acoustic variation due to microphone etc.
- Bayes theorem gives us:

$$\begin{aligned}\overline{W} &= \arg\max_{W} P(W|A) \\ &= \arg\max_{W} \underbrace{P(A|W)}_{likelihood}\,\underbrace{P(W)}_{prior}\end{aligned}$$

Machine Translation as a Noisy Channel



- Translating French sentence (f) to English sentence (e)
- French speaker has English sentence in mind (P(e))
- English sentence comes out as French via the noisy channel (P(f|e))

n-gram Models

- In language modelling for ASR and MT, sequence information is important
  - e.g. $W =$ *The dogs were barking loudly*
  - **trigram** model captures some dependencies:

$$P(W) = P(\textit{The})P(\textit{dogs}|\textit{The})P(\textit{were}|\textit{The, dogs})P(\textit{barking}|\textit{dogs, were})P(\textit{loudly}|\textit{were, barking})$$

Calculating Language Model Probabilities

- Unigram probabilities

$$p(w_1) = \frac{f(w_1)}{N}$$

where $f(w_1)$ is the number of times w_1 is seen in some corpus and N is the total number of words seen in the corpus (by token)

- In this case the relative frequency estimation can be shown to be an instance of *maximum likelihood estimation*

Calculating Language Model Probabilities

- Bigram probabilities

$$p(w_2|w_1) = \frac{f(w_1, w_2)}{f(w_1)}$$

where $f(w_1, w_2)$ is the number of times w_2 is seen following w_1 in some corpus

Calculating Language Model Probabilities

- Trigram probabilities

$$p(w_3|w_1, w_2) = \frac{f(w_1, w_2, w_3)}{f(w_1, w_2)}$$

where $f(w_1, w_2, w_3)$ is the number of times w_3 is seen following w_2 and w_1 in some corpus

Sparse Data

- As we move to trigram counts (and perhaps beyond) **sparse data** becomes a problem
- Language is extremely productive, meaning that we're likely to encounter n-grams not seen in the training data
- Zero counts are particularly problematic, leading to zero relative frequency estimates (or undefined if the denominator is zero)
- Zero probabilities propogate through the product leading to a zero probability for the whole string

Smoothing

- Linear interpolation:

$$\tilde{p}(w_3|w_1,w_2) = \lambda_1 \hat{p}(w_3|w_1,w_2) + \lambda_2 \hat{p}(w_3|w_2) + \lambda_3 \hat{p}(w_3) + \epsilon$$

$\lambda_1 + \lambda_2 + \lambda_3 + \epsilon = 1.0$

Can we go beyond trigrams?

- yes - commercial systems (speech recognisers, Google SMT) will use 5 or 6-grams
- see “All Our N-gram are Belong to You”
  - Google have prepared a *1 trillion* word n-gram corpus and made it freely available
- But will still need smoothing, however much data we use (because language is so productive)

Examples of 4-grams from Google

serve as the incoming 92
serve as the incubator 99
serve as the independent 794
serve as the index 223
serve as the indication 72
serve as the indicator 120
serve as the indicators 45
serve as the indispensable 111
serve as the indispensible 40
serve as the individual 234
serve as the industrial 52
serve as the industry 607
serve as the info 42
serve as the informal 102
serve as the information 838
serve as the informational 41
serve as the infrastructure 500

serve as the occurs once in a 1-million word corpus

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

- How do we estimate $p(f|e)$?
- $p(f|e) = \text{count}(f,e)/\text{count}(e)$
- We've seen enough language modelling now to know this isn't going to work

Translation Model

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

(word alignment diagram)

Alignment Probabilities

- Now break the sentences up into manageable chunks (initially just the words)
- $p(a, f|e) = \Pi_{j=1}^{m} t(f_j|e_i)$

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

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 (more on this later)