1. Words, and other words

A brief introduction to machine translation

**Machine Translation** covers a wide range of goals

- From **FAHQMUT**
  - Fully Automatic High Quality Unrestricted MT
- To **MAHT**
  - Machine-Assisted Human Translation
- FAHQMUT remains a dream: but hope springs eternal
- MAHT is big business, but not of much theoretical interest

The contrast between hyped-up promises of success and poor actual performance led to

- The ALPAC report (1966)
  - Found that many years of research had failed to meet expectations
    - USA has no shortage of translators
    - Fully automatic MT doesn’t really work, quality hasn’t improved much
    - It isn’t clear if it will ever work

  “The Committee indeed believes that it is wise to press forward undaunted, in the name of science, but that the motive for doing so cannot sensibly be any foreseeable improvement in practical translation. Perhaps our attitude might be different if there were some pressing need for machine translation, but we find none.”

- The end of substantial funding for MT in the US for nearly 20 years

2. From knowledge-rich to machine-learning

MT has followed the same trajectory as many other aspects of speech and language technology

- Historically, MT systems were based on one or more levels of linguistic analysis
  - the **Vauquois triangle**
- The largest and most heavily used MT system in the world worked like this until very recently
  - SYSTRAN, used by the EU to help with its translation load of over 2 million pages a year
- But most MT work today is based on one form or another of noisy channel decoding
  - With language and channel models being learned from corpora
3. Before his time: Warren Weaver

Stimulated by the success of the codebreakers at Bletchley Park (including Alan Turing), Weaver had an surprisingly prescient idea:

[...] knowing nothing official about, but having guessed and inferred considerable about, powerful new mechanized methods in cryptography. . .one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: “This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.” Have you ever thought about this? As a linguist and expert on computers, do you think it is worth thinking about?

from a letter from Warren Weaver to Norbert Wiener, dated April 30, 1947

4. A very noisy channel

Applying the noisy channel model to translation requires us to stand normal terminology on its head

- Usually we talk about source and target languages
- For example, when translating Братья Карамазовы into English
  - Russian is the source
  - English is the target

But from the perspective of the noisy channel model

- The source is English
- The channel distorts this into what we see or hear, that is, Russian
- Which we have to decode
  - to get to the source
  - which is the target
  - :-) 

5. Priors and likelihood for MT

Remember the basic story (using e for English and r for Russian):

$$\arg\max_{e_1^n} P(r_1^n | e_1^n) P(e_1^n)$$
The prior is just our old friend, some form of language model

- for English!

But the channel model needs to be articulated a bit for translation, in several ways

- The source and target need not have the same number of words
- And the mapping part, on even a fairly simple view, has to do two things:
  - Not just translate the words
  - But re-order them as well

So we need a channel model that takes all of these into account

### 6. Translation modeling for MT

J&M Chapter 25 takes you through a step-by-step motivation for the first successful attempt at doing things this way

- By the IBM team, working from French into English
- Using Canadian Hansard for training and testing
- And a variety of HMM-based decoding methods

All their approaches start with a formal notion of **alignment**

- A (possibly one-to-many) mapping from source word position to position in the observation sequence
- So, for our trivial Russian example, this would be 1 3 2
- Because the second and third words exchange positions between English and Russian

\[ \text{я вас любил} \]

\[ \text{I loved you} \]

### 7. Translation modelling, cont'd

Their simplest model then has three conceptual steps:

1. Choose a length for the Russian, given the length of the English
   - Remember, from the model’s perspective we are generating the Russian observation, starting from the English source
   - Think of a POS-tagging HMM
   - Which ‘generates’ English words (the observations) from a sequence of POS tags (the source)

2. Choose an alignment from the words in the source (English) to the words in the observations (Russian)

3. For each position in the Russian, choose a translation of the English word which aligns to it

Following simplifying assumptions of the usual Markov nature, we end up with

\[
P(i_1^j, a_1^j | e_1^j) = P(j | I) \times \prod_{j=1}^{J} P(a_j | a_{j-1}, l) P(r_j | e_{a_j})
\]

Where

- \( I \) and \( J \) are the (underlying) length of the English and the (generated) length of the Russian, respectively
• $a_j$ is the alignment of the $j$th Russian word

For the likelihood component of the post-Bayes-rule switched formula

### 8. Contemporary MT

We've barely scratched the surface

But state-of-the-art MT systems today all derive from essentially this starting point

- Including Google Translate
- Which does however think that "я вас любил" should be translated as "I loved you more"
- To be fair, that's in the context of the whole Pushkin poem from which the phrase is extracted

### 9. Getting started: The role of data

Broadly speaking, we have two models to learn:

**Language model**
- We've seen this already,
  - Target language data
  - I.e. monolingual
  - *Lots* of it
  - Such as Google's Gigaword corpus

**Channel/translation model**
- For word-word alignment and word/phrase translation
  - Bilingual
  - Harder to get lots
10. Getting started: segmentation and sentence alignment

Just as with other corpora, we need to pre-process the raw materials

- Normalise markup
- Check for and correct character encoding problems
- Segment and normalise
  - tokens
  - morphemes?
  - sentences
  - paragraphs
  - down-case at beginning of sentences, maybe
  - tag

These will vary in difficulty given the form of the raw data

- And the language involved

But for the translation model, with respect to the bilingual data, we need more

- We need to align the two versions at the paragraph and sentence level
- Sentence level is not always 1-to-1

11. Sentence alignment details: Gale and Church (1993)

Assumptions:

- We start with two documents
  - In source and target languages
  - Translations of one another
- Sentence order is rarely if ever changed
- If paragraphs exist, they are already aligned

Paragraph by paragraph, the algorithm matches source sentences to zero, one or two target sentences

- Sentence may be deleted by translator
- Sentence may be split into two by translator
- In either direction
  - We don't actually always know which was the original

12. Gale and Church, cont'd

Start with some empirical observations:

What does a hand-aligned corpus tell us about sentence alignment?
That gives G&C the basis for a maximum likelihood estimate of \( P(\text{match}) \)

Where by \( \text{match} \) is meant a particular alignment choice

What about relative length?

- If we suppose lengths (in characters) are normally distributed around equality
- With standard deviation estimated from the same hand-aligned corpus
- We get this picture when we plot the actual \( z \)-scored ratio

G&C call this \( \delta \):

\[
\delta = \frac{l_2 - l_1}{\sqrt{l_1 s^2}}
\]

- With a little more work, this gives G&C an estimate of \( P(\delta \mid \text{match}) \)

### 13. Gale and Church, concluded

(You may see where this is headed by now)

Given a candidate local alignment (a \( \text{match} \)), we can now compute

- \( P(\delta \mid \text{match}) \), the probability of that match having that value for \( \delta \)
- \( P(\text{match}) \), the probability of that kind of match (2->1, 1->0, etc.)
So we can compute their product

\[ P(\delta \mid \text{match})P(\text{match}) \]

Which (courtesy of the fact that since \( \delta \) is a property of the raw data, the observations, if follows that \( P(\delta) \) is constant) brings Bayes rule into play

- By maximising that product, we also maximise \( P(\text{match} \mid \delta) \)

Which is just what G&C need to feed into a dynamic programming search for the optimal combination of local alignments within a paragraph

- Source sentences on one edge
- Target along the other
- Dynamic programming is similar to spelling correction
- With costs coming from the formula above, drawing on six possible 'moves'
  - deletion, insertion, substitution
  - two-for-one, one-for-two, two-for-two

14. Just the beginning . . .

There's a whole course on statistical MT next term

Or see J&M chapter 25