Statistical Machine Translation - SMT

INF5820

Jan Tore Lønning

Department of Informatics
Universitety of Oslo

Statistical learning

Goal

- Find the best (most probable) English translation $\hat{E}$ of a foreign sentence F.
- $\hat{E} = \arg\max_{E} P(E \mid F)$

3 steps (common to many tasks)

1. A model. We may not have seen F before. The model will determine what to look for.
2. We must learn (or estimate) the parameters of the model from data.
3. We must have a method for using the model to find the best E given F, decoding.

Noisy channel models

- Applying Bayes' formula

$$\begin{aligned}\hat{E} &= \arg\max_{E} P(E \mid F) \\ &= \arg\max_{E} \frac{P(F \mid E)}{P(F)} P(E) \\ &= \arg\max_{E} P(F \mid E) P(E)\end{aligned}$$

- Turning the picture: consider F as a translation (distortion) of E, and ask which E?
- Why?
  - Suitable for approximations.
  - Makes use of language model $P(E)$.
- cf. K:SMT slide 34

Noisy channels

The noisy channel model

- See a distortion of the original.
- Goal: guess the original
- J&M Fig. 5.23, 9.2 og 25.15

Example

- Speech recognition: Sounds a distortion of writing.
- Tagging: Word sequence distortion of tag sequence
- Translation: Source language a distortion of target language.

Separating the models

Starting point:

$$\hat{E} = \arg\max_{E} P(F \mid E)P(E)$$

The models

- We can build and train two separate models:
  - The language model: $P(E)$
  - The translation model: $P(F \mid E)$
- Decoding must use both models simultaneously

Language model

Goal

Estimate the probability $P(E) = P(e_1 e_2 \ldots e_n)$ of the string of words $e_1 e_2 \ldots e_n$

n-gram model

$$
\begin{aligned}
& P(e_1 e_2 \ldots e_n) \\
& \quad = \quad P(e_1)P(e_2 \mid e_1)P(e_3 \mid e_1, e_2) \cdots P(e_n \mid e_1 e_2 \ldots e_{n-1}) \\
& \quad \approx \quad P(e_1)P(e_2 \mid e_1)P(e_3 \mid e_2) \cdots P(e_n \mid e_{n-1}) \\
& \quad = \quad P(e_1) \prod_{i=1}^{n-1} P(e_{i+1} \mid e_i)
\end{aligned}
$$

Comments:

- Uses the (incorrect) Markov-assumption
  $P(e_{(j+1)} \mid e_1 e_2 \ldots e_j) \approx P(e_{j+1} \mid e_j)$
- Last slide shows the bigram model. Could alternatively use trigram, quadgram, ...
- Trigram: $P(e_1 e_2 \ldots e_n) = \prod_{i=1}^{n-1} P(e_{i+1} \mid e_{i-1}, e_i)$
- For all n-grams : special symbols for start and end:
  - What is the probability of being the first word of a sentence?
  - What is the probability of being the last word of a sentence?

The translation model

Several alternatives:

- Word based
  - In particular the IBM-models: 1, 2, 3, 4, 5
- Phrase based
  - Parameter estimation often done on top of a word-based model.
- Syntax based

Word-based models

- Suppose
  - Source and target sentence always the same length
  - Word-order is preserved.
  - A one-to-one correspondence between words
- The translation would be like HMM-tagging

Translation	Tagging
source language word	word
target language word	tag
n-grams for targ. lang.	*n*-grams of tags
source sentence	sentence to be tagged
word translation probs.	probability for word given tag

- See simplified SMT example on slides from first MT lecture.

Word-based translation models

- But translation reorders, deletes, adds, goes many-to-one, one-to-many and many-to-many.
- We cannot apply HMM directly

Two parts to word-based translation

1. What is the probability that source word *a* is translated as target word *b*?
2. Alignment: Which word(s) in the target language sentence is the translation of which word(s) in the source sentence?

- J& M Figure 25.17, 25.20, 25.21, 25.22