Statistical Machine Translation: Phrase-based Models (Part II)

Jakub Waszczuk

Heinrich Heine Universität Düsseldorf

Winter Semester 2018/19

Outline

1 Translation Probability

2 Parameter Estimation

Outline

1. Translation Probability

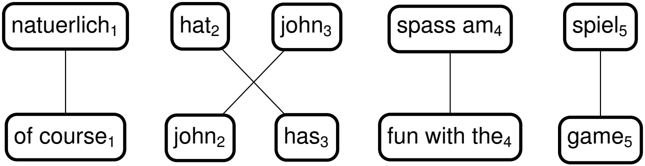
2. Parameter Estimation

Reminder

Translation process

- Split input sentence into phrases, each belonging to R_F
- Translate each phrase independently, according to phrase translation function P
- Reorder the resulting output phrases

Example

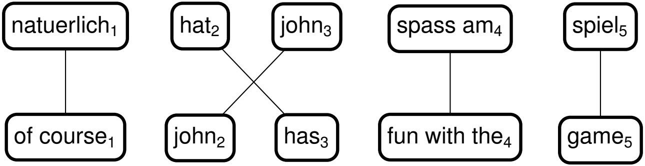


Reminder

Translation process

- **Split input sentence into phrases, each belonging to R_F**
- Translate each phrase independently, according to phrase translation function P
- Reorder the resulting output phrases

Example



Segmentation model

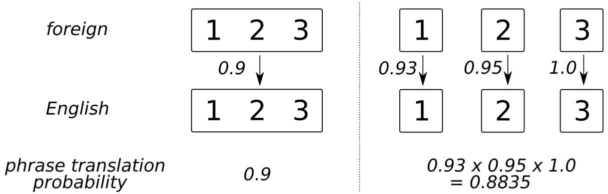
Uniform model

- Given input sentence ***f***
- Every segmentation of ***f*** into a sequence $\vec{f} \in R_F^*$ is assumed to be equally probable

A posteriori

- Effectively, the probability of a particular split depends on the other components of the model (phrase translation function, reordering model, language model)
- Segmentations with longer phrases are generally preferred (**length bias**)

Length bias example

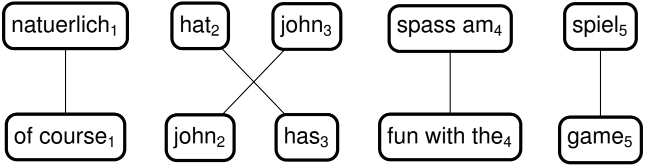


Reminder

Translation process

- Split input sentence into phrases, each belonging to R_F
- Translate each phrase independently, according to phrase translation function P
- **Reorder the resulting output phrases**

Example



Reordering model

Reordering cost function

Reordering is handled by a predefined model. Let:

- $\mathrm{beg}(i)$ – the position of the beginning of the foreign phrase corresponding to the *i*-th English phrase
- $\mathrm{fin}(i)$ – the position of the end of this foreign phrase (special case: $\mathrm{fin}(0) := 0$)
- $d(i)$ – the relative reordering distance,

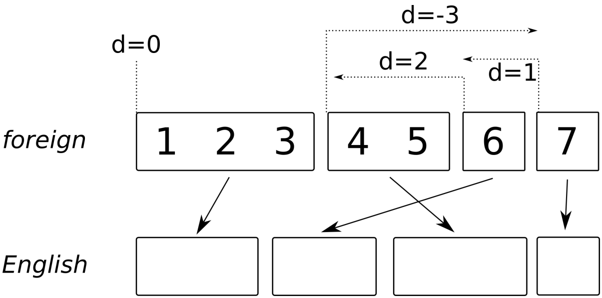
$$d(i) = |\,\mathrm{beg}(i) - \mathrm{fin}(i-1) - 1| \tag{1}$$

The cost related to the *i*-the English phrase is defined as:

$$c(i) := \alpha^{d(i)}, \text{ where } \alpha \in [0, 1] \tag{2}$$

Reordering model

Example



Total reordering cost: $\alpha^0 \times \alpha^2 \times \alpha^3 \times \alpha^1$

Reordering cost

Properties

- Provided that $\alpha < 1$, rearrangements are always penalized
- The smaller the α value, the larger the penalties

Alpha value

- α is not estimated from data
- α is determined empirically, via system's evaluation
- Therefore, α is a *hyper-parameter* in this architecture

Theoretically

- Even though the values of the cost function c are within $[0, 1]$
- In general, c is *not* a probability function

Translation

Preliminaries

- $\boldsymbol{f}, \boldsymbol{e}$ – input and output sentences
- φ – segmentation of $\boldsymbol{e}$ and $\boldsymbol{f}$ into a number (denoted $|\varphi|$) of phrases
- $\varphi_i(\boldsymbol{x})$ – the i-the phrase in $\boldsymbol{x}$ (either $\boldsymbol{e}$ or $\boldsymbol{f}$)
- $a : \{1, \ldots, |\varphi|\} \rightarrow \{1, \ldots, |\varphi|\}$ – a phrase alignment (permutation)
- $c : \mathbb{N} \rightarrow [0, 1]$ – the alignment cost function

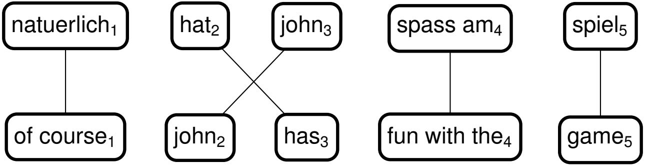
Translation cost

$$P(\boldsymbol{e}, a, \varphi \mid \boldsymbol{f}) \propto \prod_{i=1}^{|\varphi|} P(\varphi_{a(i)}(\boldsymbol{e}) \mid \varphi_i(\boldsymbol{f})) \times c(i) \tag{3}$$

For more, see the complementary material.

Translation cost

Example



Translation, segmentation, and alignment cost, given the input sentence:

$$P(\text{of course} \mid \text{natuerlich}) \times \alpha^0$$
$$P(\text{john} \mid \text{john}) \times \alpha^1$$
$$P(\text{has} \mid \text{hat}) \times \alpha^2$$
$$P(\text{fun with the} \mid \text{spass am}) \times \alpha^1$$
$$P(\text{game} \mid \text{spiel}) \times \alpha^0$$

Digression

Alternative reordering model

- We have pre-determined word-level alignments
- We could estimate reordering probabilities, as in IBM-3
- But this is not typically done in phrase-based models

Searching for translation

Theoretically

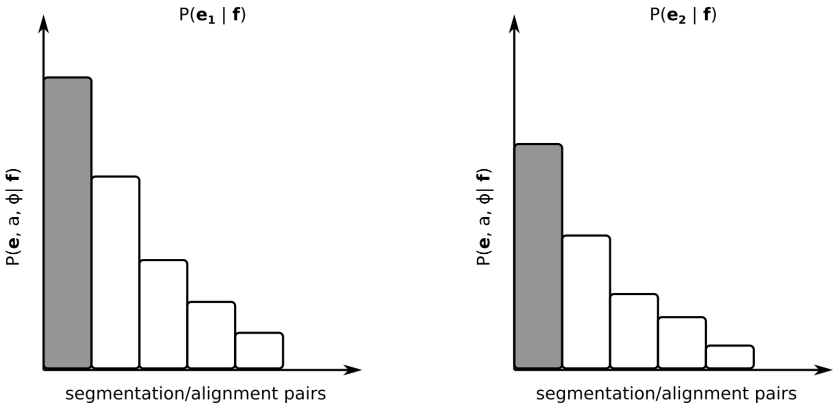
$$\arg\max_{\boldsymbol{e}} P(\boldsymbol{e} \mid \boldsymbol{f}) = \arg\max_{\boldsymbol{e}} \left(\sum_{a,\varphi} P(\boldsymbol{e}, a, \varphi \mid \boldsymbol{f}) \right) \tag{4}$$

Practically

$$\arg\max_{\boldsymbol{e}} P(\boldsymbol{e} \mid \boldsymbol{f}) \approx \arg\max_{\boldsymbol{e}} \left(\max_{a,\varphi} P(\boldsymbol{e}, a, \varphi \mid \boldsymbol{f}) \right) \tag{5}$$

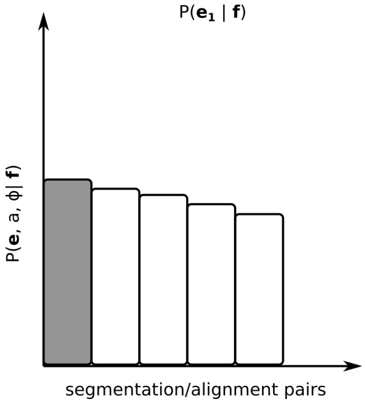
Searching for translation

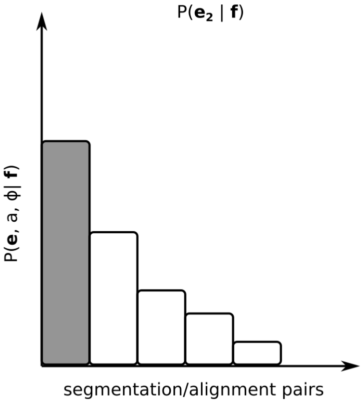
Example



Searching for translation

Example (where approximation doesn't work)





Outline

1. Translation Probability

2. Parameter Estimation

Parameter Estimation

Parameters

Phrase translation probabilities:

$$\left\{P(e \mid f) \text{ for each } f \in R_F \text{ and } e \in R_E(f)\right\} \tag{6}$$

- No fertility parameters
- No segmentation parameters
- No reordering parameters

Parameter Estimation

Collecting counts

Goal: determine the number of times phrase $\bar{f}$ translates to phrase $\bar{e}$ in corpus D

- We have the word alignments in D (we use them to extract phrase pairs)
- The phrase extraction algorithm gives us a list of phrase pairs occurring in D
- We calculate how many times $(\bar{e}, \bar{f})$ occurs in this list

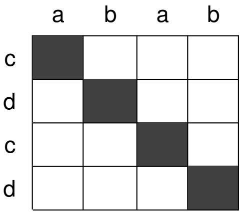
Maximum likelihood estimates

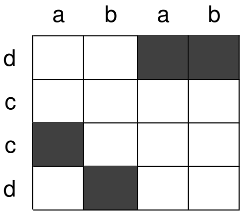
Let $C(\bar{f} \rightarrow \bar{e}; D)$ be the count of $(\bar{e}, \bar{f})$ in D. Then, we define the MLE estimates as:

$$\hat{P}(\bar{e} \mid \bar{f}) = \frac{C(\bar{f} \rightarrow \bar{e}; D)}{\sum_{\bar{e}' \in R_E(\bar{f})} C(\bar{f} \rightarrow \bar{e}'; D)} \tag{7}$$

Collecting Counts

Example

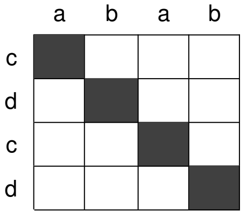


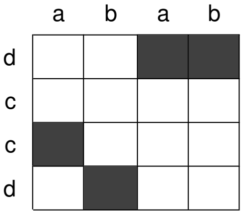


$$\hat{P}(\text{c d} \mid \text{a b}) = ?$$

Collecting Counts

Example





$$\hat{P}(\mathrm{c\ d} \mid \mathrm{a\ b}) = \frac{3}{6}$$

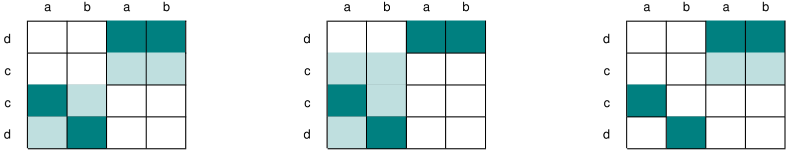
Collecting Counts: Alternative Method

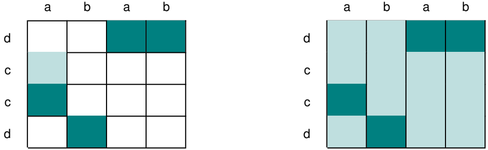
Idea

Given a sentence pair and the corresponding word alignment A:

- Consider all the possible phrase alignments consistent with A
- Assume that all have the same, uniform probability
- Calculate the expected counts

Example





$$\hat{P}(\text{c d} \mid \text{a b}) = ?$$

Collecting Counts: Alternative Method

EM

We can extend this further:

- For each sentence pair, we have a set of possible phrase alignments
- We can assume that they are uniformly distributed, as before
- We can also use the phrase translation parameters to determine the a posteriori probabilities of these alignments according to the phrase-based translation model

This leads to Expectation-Maximization for the phrase-based model.

Collecting Counts

For the practical sessions

- The first method (collecting counts stemming from the phrase pair extraction algorithm) is somewhat ad-hoc
- But it's the simplest one so we are going to use it anyway

In 2019

Decoding

Given:

- Phrase-based translation model
- Language n-gram model
- Input sentence to translate

Task:

- Determine the most probable translation
- Computationally hard, hence special approximation techniques