Chair of Computer Science 6
RWTH Aachen University
Prof. Dr.-Ing. H. Ney

Aachen, 04. 05. 2018

R. Schlüter
K. Irie

Advanced Methods in Automatic Speech Recognition

1. Exercise

Submission due on 01. 06. 2018 at the beginning of the exercise session.

Task 1.1 Sequence-level Normalization of Language Models

Assume a vocabulary $\mathcal{V}$ containing $|\mathcal{V}|$ words and a bigram language model. The objective of this task is to prove the sequence normalization of a bigram language model in the following two cases (a) and (b).

(a) Assume that an explicit length distribution $p(N)$ is given:

$$\sum_{N=1}^{\infty} p(N) = 1$$

Prove the sequence normalization:

$$\sum_{N=1}^{\infty} p(N) \sum_{w_1^N} p(w_1^N) = 1$$

for a bigram language model. (5 P)

(b) Assume a bigram language model including a sentence end token \$, i.e. the probability distribution $p(w|v)$ is defined for $w \in \mathcal{V} \cup \{\$\}$ and $v \in \mathcal{V}$

i) What is the effect of this sentence end token on language model normalization on word level? Write down the equation. (2 P)

ii) In addition, we assume that $p(\$|v) = p(\$)$ for any $v \in \mathcal{V}$. Prove the sequence normalization:

$$\sum_{N=1}^{\infty} \sum_{\substack{w_1^N : w_N = \$, \\ w_n \in \mathcal{V}, \text{ for } n=1,..,N-1}} p(w_1^N) = 1$$

(5 P)

iii) What sequence length distribution $p(N)$ is implied here? (3 P)

Task 1.2 Dynamic Programming for Inverted HMM Search

In the standard hybrid approach for speech recognition, neural networks are used to compute HMM posteriors $p(s|x)$, which is then converted into $p(x|s)$ via Bayes rule (cf. Lecture *Automatic Speech Recognition*). These steps are needed as consequence of the factorizations in the direct HMM approach which produces $p(x|s)$.

Instead in the inverted HMM approach, we derive an alternative decomposition to directly model word sequence probabilities by considering the word boundary times t_1^N as hidden variables:

$$\begin{aligned} p(w_1^N|x_1^T) &= \sum_{t_1^N} p(w_1^N, t_1^N|x_1^T) \\ &= \sum_{t_1^N} \prod_{n=1}^{N} p(w_n, t_n|t_{n-1}, w_{n-1}, x_1^T) \\ &\approx \max_{t_1^N} \prod_{n=1}^{N} p(w_n, t_n|t_{n-1}, w_{n-1}, x_1^T) \end{aligned}$$

which gives the decision rule:

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \operatorname*{arg\,max}_{w_1^N} \left\{ \max_{t_1^N} \prod_{n=1}^{N} p(w_n, t_n|t_{n-1}, w_{n-1}, x_1^T) \right\}$$

In the following, we assume the model $p(w_n, t_n|t_{n-1}, w_{n-1}, x_1^T)$ to be given.

(a) Write down an auxiliary function to be used for the corresponding dynamic programming (no recursion is asked here). (3 P)

(b) Derive the recursion equation of the dynamic programming in detail. (6 P)

(c) Define which backtrace information is to be stored. (4 P)

(d) What are the time and memory complexities? (2 P)