Faculty of Computer Science, Dalhousie University *21-Oct-2024*

CSCI 4152/6509 — Natural Language Processing

Lecture 12: P0 Topics Discussion (4); N-gram Model (start)

Location: Carleton Tupper Building Theatre C Instructor: Vlado Keselj
Time: 16:05 – 17:25

Previous Lecture

- Fully Independent Model (continued)
- Naïve Bayes classification model
  - Assumption, definition
  - Graphical representation
  - Spam detection example
  - Computational tasks
  - Number of parameters
  - pros and cons, additional notes
  - Bernoulli and Multinomial Naïve Bayes
- P0 Topics Discussion (3)

P0 Topics Discussion (4)

- Discussion of individual projects as proposed in P0 submissions
- Projects discussed: P-14, P-15, P-16, P-17, P-18, P-19, P-20, P-21, P-23, P-24

13 N-gram Model

Before we introduce this model, let us first introduce *Language Modeling.*

To understand better the significance of the N-gram Model, let us first introduce the *language modeling*—an important task in the probabilistic NLP. *Language Modeling* can be described as the task of building a probabilistic model that can estimate the probability of an arbitrary natural languate sentence; i.e., of the probabilisty P(sentence).

One application of this problem is in speech recognition. In speech recognition, we are interested in

$$\arg\max_{\text{sentence}} \mathrm{P}(\text{sentence}|\text{sound})$$

This is equal to:

$$\begin{aligned}\arg\max_{\text{sentence}} \mathrm{P}(\text{sentence}|\text{sound}) &= \arg\max_{\text{sentence}} \frac{\mathrm{P}(\text{sentence}, \text{sound})}{\mathrm{P}(\text{sound})} \\ &= \arg\max_{\text{sentence}} \mathrm{P}(\text{sentence}, \text{sound}) \\ &= \arg\max_{\text{sentence}} \mathrm{P}(\text{sound}|\text{sentence})\mathrm{P}(\text{sentence})\end{aligned}$$

It is easier to estimate $\mathrm{P}(\text{sound}|\text{sentence})$ than $\mathrm{P}(\text{sentence}, \text{sound})$, and it is done by an *acoustic model,* while $\mathrm{P}(\text{sentence})$ is estimated by a *language model.*

Slide notes:

Language Modeling

- Task of estimating probability of arbitrary utterance in a language
- Alternative task: Predicting the next token in a sequence: e.g., the next word or words, in a sentence, or next character or characters
- N-gram model: a "natural" model for this task

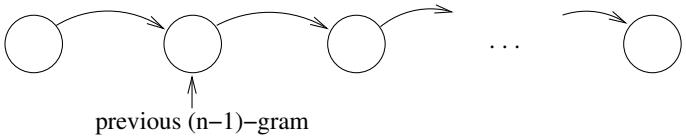
N-gram model is very simple and it is among the most successful models for language modelling; trigram ($n = 3$) word model in particular. In an n-gram model, we calculate joint distribution for all n-tuples of consecutive words (or characters). For example, in the trigram model, we count the number of occurrences of each triple of consecutive words from a corpus. Using this statistics, we can estimate the probability of arbitrary word w_3 following two given words w_1 and w_2: $\mathrm{P}(w_3|w_1w_2)$. It is useful to assign some small probability to unseen triples as well (using a technique called *smoothing*). If we use two "dummy" words '$\cdot$' at the beginning of each sentence, then the probability of arbitrary sentence can be calculated as:

$$\mathrm{P}(w_1w_2\ldots w_n) = \mathrm{P}(w_1|\cdot\cdot)\mathrm{P}(w_2|w_1\cdot)\mathrm{P}(w_3|w_2w_1)\ldots\mathrm{P}(w_n|w_{n-1}w_{n-2})$$

N-gram Model: Notes

- Reading: Chapter 4 of [JM]
- Use of log probabilities
  - similarly as in the Naïve Bayes model for text
- Graphical representation

Graphical Representation



Use of log probabilities

Multiplying a large number of probability values, which are typically small, gives a very small result (close to zero), so in order to avoid floating-point underflow, we should use addition of logarithms of the probabilities in the model.