Natural Language Processing
Portfolio I

Leon F.A. Wetzel
Information Science
University of Groningen - Faculty of Arts
l.f.a.wetzel@student.rug.nl

February 7, 2021

Abstract

In this document, you can find the results and explanations for the assignments of the first part of the portfolio for the course Natural Language Processing, taught at the University of Groningen. The corresponding Python code can be found at https://github.com/leonwetzel/natural-language-processing[1]. Note that version control of Jupyter notebooks is done via `jupytext`, so do not forget to convert the Python scripts to notebooks yourself!

1 Week 1 - Regular Expressions

We used https://regex101.com/ to test our regular expressions. In this document, we use ∧ to better display/represent a caret. Spaces cannot be displayed in LaTeXsadly, so these are not shown below.

1.1 Regular Expressions I

1. Set of all alphabetic strings.
   $\wedge[a-zA-z]+\$$

2. Set of all lower case alphabetic strings ending in b.
   $\wedge[a-z]*[b]\$$

3. Set of all strings from the alphabet a, b such that each a is immediately preceded by a b and immediately followed by a b.
   $\wedge(bab)+\$$

1.2 Regular Expressions II

1. Set of all strings with two consecutive repeated words (e.g., *"Humbert Humbert"* and *"the the"* but not *"the bug"* or *"the big and the bug"*).
   $\backslash b(\backslash w+)\backslash b\backslash s+\backslash 1$

2. All strings that start at the beginning of the line with an integer and that end with a word.
   $\wedge\backslash d+.+\backslash b[a-zA-Z]+\backslash b\$$

3. All strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely contain the word *grotto*).
   $\backslash bgrotto\backslash b.*|\backslash braven\backslash b.*$

[1] All code will be published after the course has been completed

1.3 ELIZA

We added extra regular expressions and substitutions to handle more variants of user input. We also implemented a substitution to handle question marks, so that ELIZA's responses feel more like proper questions when relevant. The other implemented substitutions are related to how one feels, and takes into account words such as *encounter*, *experience* and *suffer from*. Snippets from example dialogue can be found below.

ELIZA : How are you?

LEON : I experience coldness.

ELIZA : Why do you experience coldness?

LEON : I feel as if it is snowing today

ELIZA : Why do you feel as if it is snowing today?

LEON : Because it is snowing today

ELIZA : Because it is snowing today?

LEON : Yes, but I am inside

ELIZA : Yes, but Why are you inside?

LEON : Because it is cold outside

ELIZA : Because it is cold outside?

LEON : Yes

1.4 Byte-Pair Encoding

Our example text for the experiment is a snippet from the Wikipedia page on the GameStop short squeeze[2], which occurred in January 2021. According to https://wordcounter.net, our example consists of 137 words and 827 characters.

"*In January 2021, a short squeeze of the stock of the American video game retailer GameStop (NYSE: GME) and other securities took place, causing major financial consequences for certain hedge funds and large losses for short sellers. Approximately 140 percent of GameStop's public float had been sold short, and the rush to buy shares to cover those positions as the price rose caused it to rise even further. The short squeeze was initially and primarily triggered by users of the subreddit r/wallstreetbets, an Internet forum on the social news website Reddit. At its height, on January 28, the short squeeze caused the retailer's stock price to reach a pre-market value of over US$500 per share, nearly 30 times the $17.25 valuation at the beginning of the month. The price of many other heavily shorted securities increased.*"

The default value for `vocab_size` (v) is 30000. With this value, we can count 164 words and 190 segments found in our example text by the tokenizer. For $v = 25000$, we count 164 words and 195 segments, and the counts for $v = 20000$ are 164 words and 197 segments. In line with the assignment description, we can observe here that more words are segmented into subwords when the number of segments increase (as a result of v decreasing). An overview of the results can be found in table 1.

v	30000	25000	20000	15000	10000	7500	5000	2500
Words	164	164	164	164	164	164	164	164
Segments	190	195	197	204	222	246	692	692

Table 1: Overview of `vocab_size` values and the word and segment counts

[2] https://en.wikipedia.org/wiki/GameStop_short_squeeze

When $v = 7500$, the number of segments is approximately 150% of the number of words. The longest words in our example text during this setup that was not segmented, are *increased*, *beginning* and *initially*. All these words consist of nine characters each.

2 Week 2 - N-gram Language Models

2.1 J&M exercise 3.1

Write out the equation for trigram probability estimation (modifying Eq. 3.11). Now write out all the non-zero trigram probabilities for the I am Sam corpus on page 41.

The original - simplified - formula for bigram probability estimation is as follows:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

2.2 J&M exercise 3.2

Calculate the probability of the sentence *i want chinese food*. Give two probabilities, one using Fig. 3.2, and another using the add-1 smoothed table in Fig. 3.6.

2.3 J&M exercise 3.6

Suppose we train a trigram language model with add-one smoothing on a given corpus. The corpus contains V word types. Express a formula for estimating P(w3|w1,w2), where w3 is a word which follows the bigram (w1,w2), in terms of various N-gram counts and V. Use the notation c(w1,w2,w3) to denote the number of times that trigram (w1,w2,w3) occurs in the corpus, and so on for bigrams and unigrams.

2.4 J&M exercise 3.7

We are given the following corpus, modified from the one in the chapter: <s> I am Sam </s> <s> Sam I am </s> <s> I am Sam </s> <s> I do not like green eggs and Sam </s> If we use linear interpolation smoothing between a maximum-likelihood bi-gram model and a maximum-likelihood unigram model with λ1 = 1/2 and λ2 = 1/2 , what is P(Sam|am)? Include <s> and </s> in your counts just like any other token.

2.5 N-grams in the notebook

Answer the two questions in the notebook ngrams_exercise.