Natural Language Processing, Assignment 1

Dongting Li, 2020011222

October 3, 2023

Disclosure

Please disclose the following cases if any:

1. If you received any help whatsoever from anyone in solving this assignment, give full details.
2. If you gave any help whatsoever to anyone in solving this assignment, give full details.
3. If you found or came across code that implements any part of this assignment, give full details.
4. Stop word: https://en.wikipedia.org/wiki/Stop_word

Warning: You should do your own work. Copying solutions (text or code) from any external sources is strictly prohibited.

Problem 1

1.

The true empirical distribution $\boldsymbol{y}$ is a one-hot vector that y_w takes the value 1 if $w = o$, and 0 otherwise.

2.

Simplify $J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})$ as follows:

$$\begin{aligned} J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) &= -\log P(O = o | C = c) \\ &= \log \Sigma_{w \in \text{Vocab}} \exp\left(\boldsymbol{u}_w^T \boldsymbol{v}_c\right) - \boldsymbol{u}_o^T \boldsymbol{v}_c \end{aligned} \tag{1}$$

Chain rule:

$$\begin{aligned} \frac{\partial J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{v}_c} &= \frac{\partial \log \Sigma_{w \in \text{Vocab}} \exp\left(\boldsymbol{u}_w^T \boldsymbol{v}_c\right)}{\partial \boldsymbol{v}_c} - \frac{\partial \boldsymbol{u}_o^T \boldsymbol{v}_c}{\partial \boldsymbol{v}_c} \\ &= \Sigma_{w \in \text{Vocab}} \frac{\exp\left(\boldsymbol{u}_w^T \boldsymbol{v}_c\right) \boldsymbol{u}_w}{\Sigma_{w \in \text{Vocab}} \exp\left(\boldsymbol{u}_w^T \boldsymbol{v}_c\right)} - \boldsymbol{u}_o \\ &= \boldsymbol{U}\hat{\boldsymbol{y}} - \boldsymbol{U}\boldsymbol{y} \\ &= \boldsymbol{U}(\hat{\boldsymbol{y}} - \boldsymbol{y}) \end{aligned} \tag{2}$$

3.

Case 1: when $w = o$, chain rule:

$$\begin{aligned}
\frac{\partial J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_w} &= \frac{\partial \log \Sigma_{w\in\text{Vocab}} \exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)}}{\partial \boldsymbol{u}_w} - \frac{\partial \boldsymbol{u}_o^T \boldsymbol{v}_c}{\partial \boldsymbol{u}_w} \\
&= \frac{\exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)} \boldsymbol{v}_c}{\Sigma_{w\in\text{Vocab}} \exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)}} - \boldsymbol{v}_c \\
&= (\hat{y}_w - 1)\boldsymbol{v}_c
\end{aligned} \tag{3}$$

Case 2: when $w \neq o$, chain rule:

$$\begin{aligned}
\frac{\partial J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_w} &= \frac{\partial \log \Sigma_{w\in\text{Vocab}} \exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)}}{\partial \boldsymbol{u}_w} - \frac{\partial \boldsymbol{u}_o^T \boldsymbol{v}_c}{\partial \boldsymbol{u}_w} \\
&= \frac{\exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)} \boldsymbol{v}_c}{\Sigma_{w\in\text{Vocab}} \exp{(\boldsymbol{u}_w^T \boldsymbol{v}_c)}} \\
&= \hat{y}_w \boldsymbol{v}_c
\end{aligned} \tag{4}$$

Combine two cases:

$$\frac{\partial J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_w} = (\hat{y}_w - y_w)\boldsymbol{v}_c \tag{5}$$

4.

$$\frac{\partial J_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{U}} = \begin{bmatrix} \frac{\partial J(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_1} & \frac{\partial J(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_2} & \cdots & \frac{\partial J(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_{|\text{Vocab}|}} \end{bmatrix} \tag{6}$$

5.

The partial derivatives of Jneg-sample with respect to $\boldsymbol{v}_c$:

$$\begin{aligned}
\frac{\partial J_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{v}_c} &= -\frac{\partial \log{(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c))}}{\partial \boldsymbol{v}_c} - \frac{\partial \sum_{k=1}^{K} \log \sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)}{\partial \boldsymbol{v}_c} \\
&= (\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c) - 1)\boldsymbol{u_o} + \sum_{k=1}^{K} (1 - \sigma(\boldsymbol{u}_k^T \boldsymbol{v}_c))\boldsymbol{u_k}
\end{aligned} \tag{7}$$

The partial derivatives of Jneg-sample with respect to $\boldsymbol{u}_o$:

$$\begin{aligned}
\frac{\partial J_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_o} &= -\frac{\partial \log{(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c))}}{\partial \boldsymbol{u}_o} - \frac{\partial \sum_{k=1}^{K} \log \sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)}{\partial \boldsymbol{u}_o} \\
&= (\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c) - 1)\boldsymbol{v_c}
\end{aligned} \tag{8}$$

The partial derivatives of Jneg-sample with respect to $\boldsymbol{u}_k$:

$$\begin{aligned}
\frac{\partial J_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U})}{\partial \boldsymbol{u}_k} &= -\frac{\partial \log{(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c))}}{\partial \boldsymbol{u}_k} - \frac{\partial \sum_{k=1}^{K} \log \sigma(-\boldsymbol{u}_k^T \boldsymbol{v}_c)}{\partial \boldsymbol{u}_k} \\
&= (1 - \sigma(\boldsymbol{u}_k^T \boldsymbol{v}_c))\boldsymbol{v_c}
\end{aligned} \tag{9}$$

When updating the U matrix, we only need to update $K + 1$ vectors instead of all vectors in U.

Problem 2

1)

Figure 1: Source Channel Diagram

2)

Denote the original sentence as $\boldsymbol{W}$ and the guessed sentence as $\boldsymbol{X}$
The goal is to deduce the original sentence from the guessed sentence:

$$W^* = \arg\max_W P(W|X) = \arg\max_W P(W)P(X|W) \tag{10}$$

3)

$$\begin{aligned} P(W) &\rightarrow \text{Sentence modeling} \\ P(X|W) &\rightarrow \text{Noisy channel modeling} \end{aligned} \tag{11}$$

automatic spelling correction = sentence modeling + noisy channel modeling

Problem 3

Trying to explore Twitter's tweets about the Russia-Ukraine war and Trump's personal tweets, I downloaded the CSV files of the two corporations through the Internet. The following is the download link. For simplicity, the two corpora are denoted as corpora 0 and corpora 1 respectively.
Link:

- Russia-Ukraine Conflict Twitter Dataset:
  https://www.kaggle.com/datasets/tariqsays/russiaukraine-conflict-twitter-dataset
- Trump Twitter Archive: https://www.thetrumparchive.com/faq

Descriptions of sub-languages

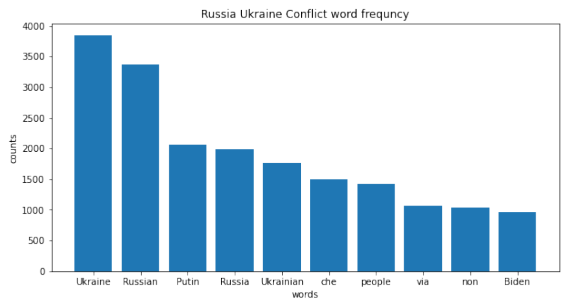
Corpora 1 is related to the Ukrainian-Russian conflict, which has 76457 word types and 1057389 word tokens.
Corpora 2 is a dataset of Trump's tweets, which has 28010 word types and 1108064 word tokens.

Preprocessing of the data

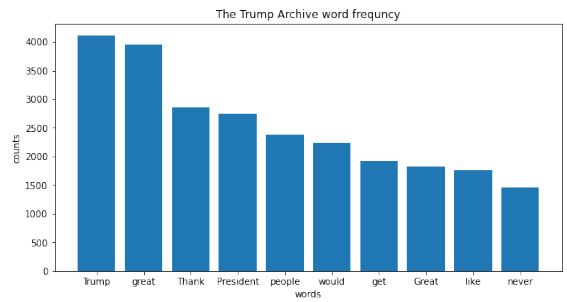
- Remove empty text from the corpus.
- When calculating the head of the word frequency list, I found that if the unprocessed text is used directly, the words with higher word frequency are mostly stop words. In order to reflect the differences between the two corpora to a greater extent, I chose to remove the stop words in the corpora. Then I sorted the word frequency list and drew the head of the word frequency list as Figure 2.

- Calculate the length of each sentence and calculate the median and average. Figure 3 is a curve of the number of sentences as a function of sentence length. Table 1 is the mean sentence length and median sentence length of corpora.
- Here we choose the simplest word combination, that is two adjacent words, count their frequency of occurrence, and give the head of the word combination frequency list. Figure 4 and Figure 5 is the head of the word combination frequency list.
- Finally, I calculated the distribution of letters, changed all letters to lowercase, and then counted the number of each of the 26 letters. Figure 6 is the letter distributions.

Statistics

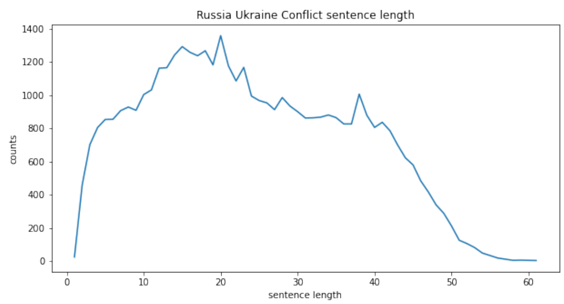


(a) corpora 1

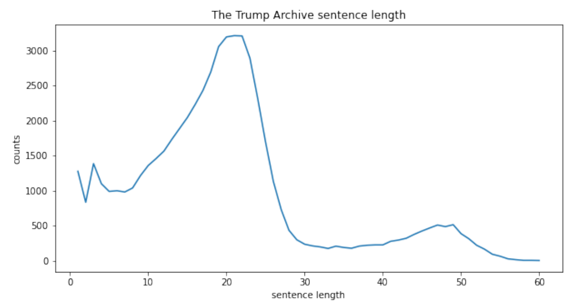


(b) corpora 2

Figure 2: Head of Word Frequency List



(a) corpora 1



(b) corpora 2

Figure 3: Sentence Length

Corpora	**Mean**	**Median**
Corpora 1	23.991219312973634	23.0
Corpora 2	19.58713828640116	19.0

Table 1: Mean and Median Sentence Length

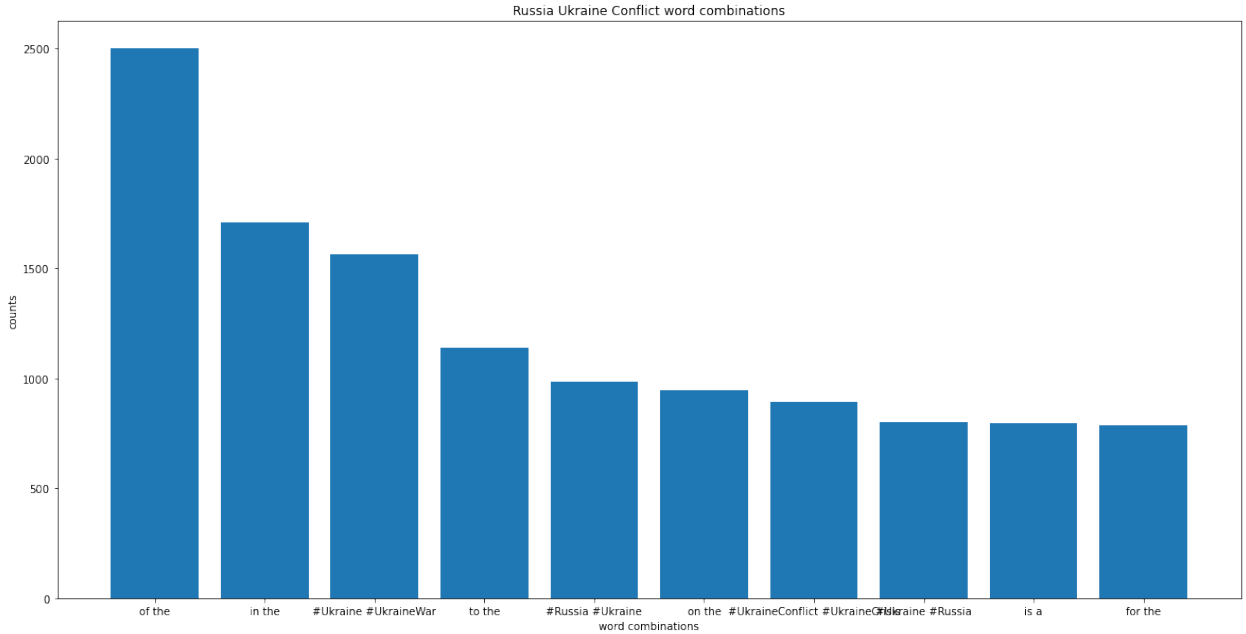


Figure 4: Corpora 1 Word Combination

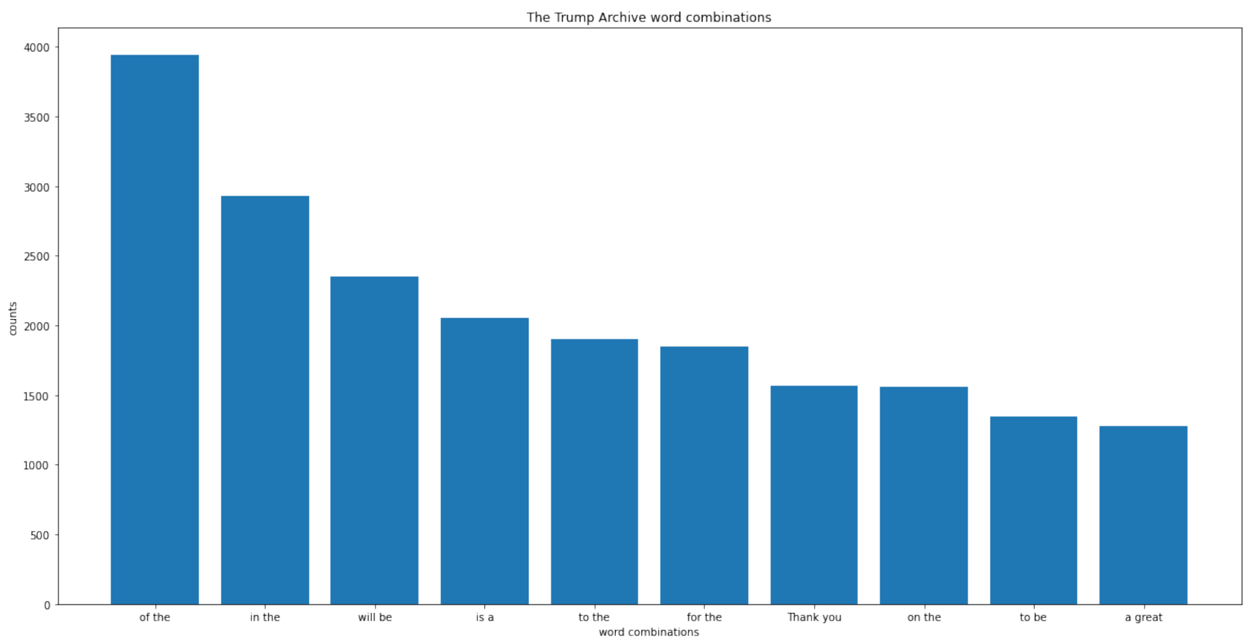
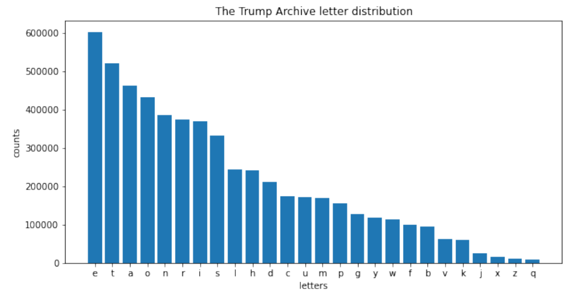


Figure 5: Corpora 2 Word Combination

(a) corpora 1



(b) corpora 2

Figure 6: Letter Distributions

Hypotheses

- Corpora 0 uses a larger vocabulary and contains more uncommon words.
- Corpora 1 has longer sentence lengths.

According to the above statistics, hypothesis 1 is confirmed, but hypothesis 2 is refuted. While I was processing the data, I found that corpora 1 contains other languages such as French, German, etc., so its vocabulary is larger.

The reason I made hypothesis 2 is that I think Trump may be able to organize his language better than ordinary people and then publish longer sentences.