Spoken and Natural Language Understanding Practical Assignment 1

Submitted by Jiahui Dai

April 4, 2025

Accompanying material: Assignment_1.ipynb

1 Zipf's Law

List of unique words sorted to their frequency in descending order

Shown in accompanying Jupyter Notebook.

Discussion on findings

Zipf's Law states that in a large collection of words, the frequency of any word is inversely proportional to its rank:

$$f \propto \frac{1}{r}$$

where

- f: frequency of word
- r: rank of the word

This means that the most frequent word occurs twice as often as the second most frequent, three times as often as the third most frequent, and so on.

With reference to the jungle book dataset, words that are short in length occur in higher frequencies compared to words with longer lengths. For example, "the", "and", "of" occur in higher frequency compared to "produce", "subscribe", "newsletter" of lower frequency.

To verify Zipf's law on a textual corpus, using the jungle book dataset, chi-square goodness of fit test is performed [**web:chi_sq_test**].

- H_0: The observed frequencies are equal to the expected frequencies.
- H_1: The observed frequencies are not equal to the expected frequencies.

$$\begin{aligned} \chi^2_{\text{statistics}} &= 19760.11496822835 \\ \chi^2_{\text{critical}} &= 5107.674219300448 \\ \chi^2_{\text{statistics}} &> \chi^2_{\text{critical}} \qquad \text{(Reject } H_0\text{)} \end{aligned}$$

As the H_0 is rejected, this suggests that observed frequencies are not equal to the expected frequencies, and that Zipf's Law does not hold.

With reference to Figure 1, there is a general downward trend of frequency compared to rank. The statistical analysis shows that the observed frequency is not equal to the expected frequency, suggests that the jungle book dataset contains words that does not follow Zipf's Law significantly. If Zipf's Law holds, the observed line (blue) should form a straight line with a slope close to -1, as depicted by the expected line (red) in Log-Log curve. However, the observed line does not follow a slope close to -1 in Log-Log curve.

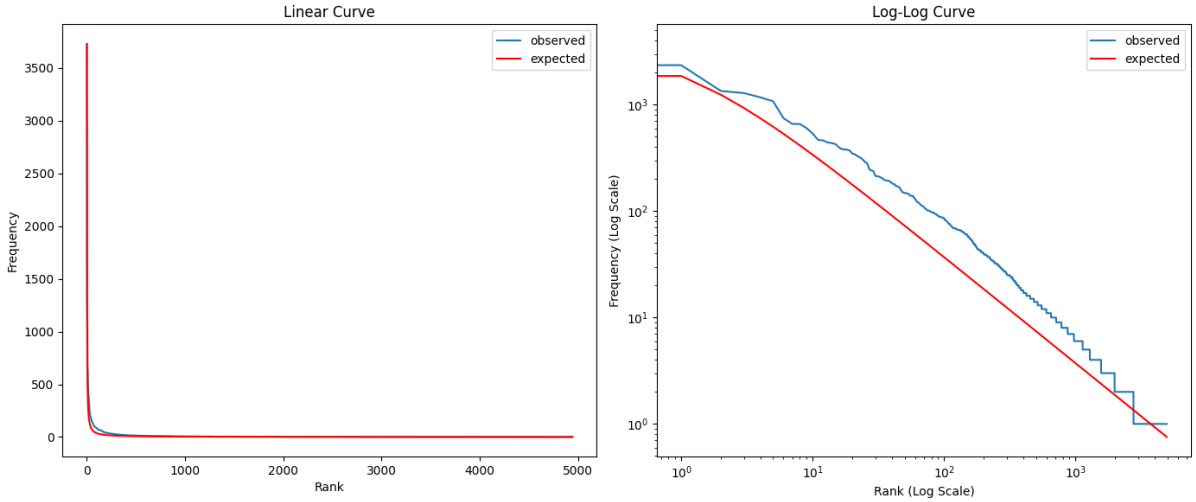


Figure 1: Linear and Log-Log Curves.
Observed line (blue) follows the frequency of words observed in the jungle book dataset. Expected line (red) follows the theoretical word frequency according to Zipf's law.

2 Mutual Information

Observations

Pointwise mutual information (PMI) is a metric that compares the relative frequency of two outcomes occurring together to the probability of either outcome occurring independently. A positive PMI value means that the words co-occur more frequently than would be expected, whereas a negative PMI value means they cooccur less frequently than would be expected. A PMI value of 0 suggests that the words occur independent of each other (occurance by chance) [**web:pmi**].

Many word-pairs with high PMI values (Table 1) are often two-word phrases to convey a certain idea, like "United States" and "tree tops", whereas for word-pairs with low PMI values (Table 2) are pairs that either contain "the" or "and", or phrases that is grammatically incorrect in the English language.

Discussion of the validity of the independence assumption for unigram models

A unigram model assumes that each word in a sentence or document is independent of all other words, i.e., the probability of a word occurring is independent of the words that came before or after it.

$$P(w_1, w_2, w_3, \dots) = \prod_{i=1}^{n} P(w_i)$$

where $P(w_i)$ is the probability of each word w_i.

To assess the validity of the independence assumption, the conditional probabilities $P(w_i|w_1, w_2, \dots, w_{i-1})$ (observed) is calculated and compare to the unigram probability $P(w_i)$ (expected) using chi-square goodness of fit test.

Chi-square independence test is performed [**web:chi_sq_test**].

- H_0: Each word in a document is independent of all other words.
- H_1: Each word in a document is dependent of all other words.

$$\begin{aligned} \chi^2_{\text{statistics}} &= 572.4405452818747 \\ \chi^2_{\text{critical}} &= 57009.8446715156 \\ \chi^2_{\text{statistics}} &< \chi^2_{\text{critical}} \qquad \text{(Fail to reject } H_0\text{)} \end{aligned}$$

As the H_0 fails to be rejected, this suggests that observed frequencies (the conditional probabilities $P(w_i|w_1, w_2, \dots, w_{i-1})$) and expected frequencies (unigram probability $P(w_i)$) are the independent, and thus the independence assumption for unigram models is valid.

w1	w2	pmi
machua	appa	8.54334
united	states	8.30218
literary	archive	8.23318
cold	lairs	7.69419
archive	foundation	7.57926
bandar	log	7.41487
petersen	sahib	7.38589
stretched	myself	7.33937
paragraph	f	7.33937
hind	legs	7.23466
fore	paws	7.15704
hind	flippers	7.13457
tree	tops	7.02921
troop	horse	7.02804
bath	room	7.00289
twenty	yoke	6.98519
paragraph	e	6.97472
electronic	works	6.95225
master	words	6.91588
whole	line	6.90405
years	ago	6.88771
within	days	6.87221
bring	news	6.86936
waingunga	river	6.76685
killing	grounds	6.75158
council	rock	6.71941
villagers	lived	6.67154
monkey	folk	6.64622
black	panther	6.53574
copyright	laws	6.49858

Table 1: 30 word pairs with highest pmi value

w1	w2	pmi
the	the	−5.50709
and	and	−4.58736
he	the	−4.28052
the	he	−4.28052
the	to	−3.7514
of	of	−3.47152
i	and	−3.30003
they	the	−3.23617
to	he	−3.21798
was	and	−3.1072
for	and	−2.96939
is	and	−2.96079
and	is	−2.96079
had	the	−2.95281
of	in	−2.88121
little	the	−2.76773
to	i	−2.69736
he	in	−2.67243
but	and	−2.65494
that	and	−2.64759
of	and	−2.64314
at	and	−2.58825
we	the	−2.57623
not	and	−2.55316
man	the	−2.47589
he	and	−2.43436
were	the	−2.43042
and	of	−2.42
of	for	−2.41147
is	of	−2.40287

Table 2: 30 word pairs with lowest pmi value

3 Wikipedia Language Model

3.1 Motivation [n-gram:ch3]

An n-gram is a sequence of n words and a a 2-gram (bigram) is a two-word sequence of words. Here, we use a 2-gram language model to solve this task.

To predict the conditional probability of the next word in a bigram model, Markov assumption is applied:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1}) \tag{1}$$

where

- $P(w_n|w_{1:n-1})$: probability of word w_n given all previous words $w_{1:n-1}$
- $P(w_n|w_{n-1})$: probability of word w_n given the previous word w_{n-1}

The maximum likelihood estimation is used to estimate probabilities:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})} \tag{2}$$

where

- $C(w_{n-1}, w_n)$: count of bigram of w_{n-1} and w_n frequency
- $C(w_{n-1})$: count of w_{n-1} frequency

The perplexity of a language model on a test set is the inverse probability of the test set, normalized by the number of words and is used as an evaluation matric in this model. The higher the probability of the word sequence, the lower the perplexity, the better the model as the model is more confident and accurate. It is defined as

$$Perplexity(w) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \tag{3}$$

$$= 2^{\frac{-\sum \log_2 P(w_i|w_{i-1})}{N}} \tag{4}$$

For equation 4, the probabilities are log to prevent numerical underflow and stored. To return the original probability, the exp of the logprob is calculated.

However, there lies a challenge in the trained model. If a bigram exists in the test dataset and not in the train dataset, the probability of the bigram is zero, which affects the calculation of perplexity as we cannot divide by zero. Laplace smoothing is the simplest method to tackle this challenge. It is defined as

$$P_{\text{Laplace}}(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + V} \tag{5}$$

where

- V: number of unique in the corpus

To finetune the model, add-k smoothing is used (further improvement from Laplace smoothing) when optimising with validation dataset. It is defined as

$$P_{\text{Add-k}}(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + k}{C(w_{n-1}) + kV} \tag{6}$$

Data Preprocessing Steps

Text Tokenisation The text data is broken into individual words (token), with punctuation, special characters and whitespace removed [**web:pre-processing**].

Train-Validation-Test Data Split This step is not performed here as the data provided is already split into training, validation and testing sets [**web:pre-processing**].

Method and Experiment Design

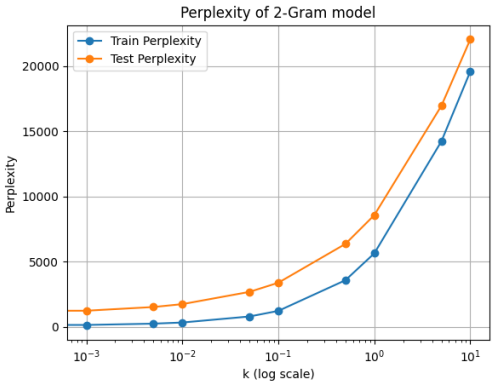
1. Preprocess the data, i.e. text tokenisation
2. Generate 2-grams and count the frequency of occurance of two consecutive words (w_1 and w_2)
3. Calculate 2-gram probabilities (Equation 6) with smoothing k
4. Evaluate the model by calculating perplexity (Equation 4)
5. Optimise the model with validation data and a different k value
6. Evaluate the perplexity with optimised model with test data

Hyperparameters

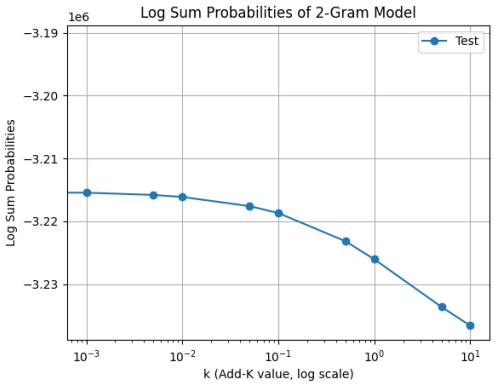
- k (Equation 6) is updated to finetune the model

Evaluation Metrics

- Perplexity



(a) Perplexity vs k-values



(b) Log-Sum Probability vs k-values

Figure 2: Bigram Model

Observations In Figure 2a, the perplexity increases with increasing k-values whereas in Figure 2b, the log sum probabilities of $P(w_i|w_{i-1})$ decreases with increasing k-values. This observation corresponds well with Equation 3 as the perplexity has an inverse relationship with the product of probabilities. With higher k-values, the probability product decreases (decreasing log sum probability), it makes individual probabilities less extreme (making each probability closer to a common probability) and smaller. With the multiplication of probabilities, it makes the probability product (or log sum probability) decreases with increasing k value.

As described earlier, perplexity is used to describe how confident and the model is accurate at predicting the the next word. We can see from Figure 2a that at low k-value, the perplexity is lower than at a higher k-value. This suggests that the model is better at predicting the next word with low k value compared to high k value.

As we are trying to achieve a perplexity of 1, which means the model is absolutely certain that it will always predict the correct word with probability of 1, the perplexity however still remains high with low k value at around 2000. This suggests that the bigram model is essentially "guessing" the next word given the previous word.

Drawbacks

- High smoothing can reduce model specificity. With = 100, probabilities are spread more evenly, reducing the model ability to distinguish frequent and rare bigrams
- Limited context with Bigram model
- Low perplexity does not gurantee better text generation or predictive power in real world applications, as the model is trained based on train data. If the model is trained on customer support conversation corpus, it will perform poorly on a chemistry lecture corpus test data.

Potential Improvements

- Use higher order N-Grams, e.g. 3-Gram or 4-Gram (See Section 3.2)
- Experiment with different smoothing techniques, i.e. apply backoff or interpolation (See Section 4 for backoff implementation)
- Increase training data

3.2 Improvement using higher order N-Grams

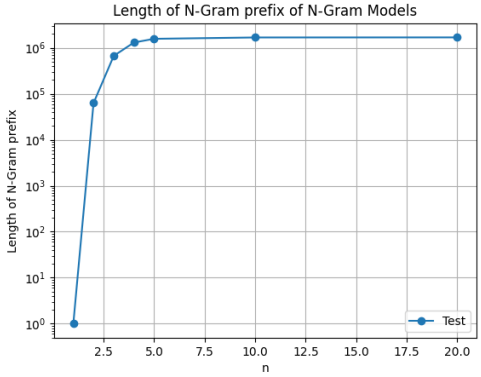
It is interesting to see that with increasing n for n-gram models, the perplexity increases with the exception of unigram model (Figure 3b). As the n-size increases, the number of possible n-grams increases exponentially (Figure 3a). This suggests that each specific sequence of 'prefix' has a lower instance, which corresponds to its probability of occurance, in the training data. This can cause the preplexity to increase tremendously with increasing n. With test data, many possible prefix sequence are not seen in

the training data, which leads to the model assigning low probabilities to the unseen sequences, increasing the perplexity further.

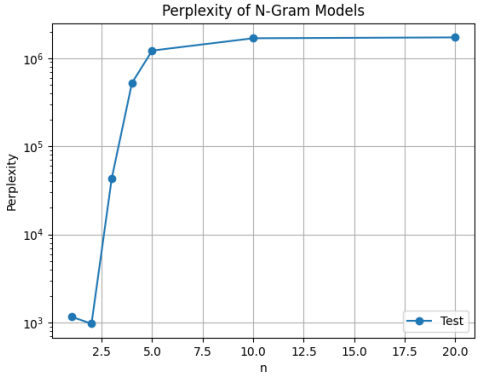
Increasing the n in n-grams will eventually lead to a plateau of unique prefix and perplexity. This is due to the number of unique prefixes following Zipf's law, as it is forming 'less common' combinations. This plateauing effect is then seen in preplexity as there is not a further increase in number of unique prefixes.

It is worthy to note that 1-gram has a higher perplexity than 2-gram as they lack the context of preceding words, making it harder to predict the next word accurately, whereas bigrams consider the previous word, improving predictive accuracy and thus lowering perplexity.

In Table 3, you will find the generated text of trained N-Gram model given a specific text. In practice, none of the generated text are coherent due to their high perplexity (despite 2-gram being the best model with the lowest perplexity).



(a) Length of unique n-grams prefix



(b) Perplexity

Figure 3: N-Grams Model

Another limitation is due to the way the corpus is tokenised. Any meaning related to digits or punctuations may have been lost. This means that the text generated as shown in Table 3 will be challenging to understand coherently.

4 Additional Experiments

4.1 Motivation [n-gram:ch3]

In a backoff model, how it works is that if it the n-gram has zero counts, it is backed off to $(n-1)$-gram. To obtain a correct probability distribution, the higher-order n-grams is discounted to save some probability mass for the lower-order n-grams. Here we will use the non-discounted backoff algorithm (called stupid backoff) where there is no discount of higher-order probabilities. If the higher-order n-gram has zero counts, it is simply backed off to lower-order n-grams weighted by a fixed weight (λ).

$$S(w_i|w_{i-N+1:i-1}) = \begin{cases} \frac{C(w_{i-N+1:i})}{C(w_{i-N+1:i-1})} & \text{if } C(w_{i-N+1:i}) > 0 \\ \lambda \cdot S(w_i|w_{i-N+2:i-1}) & \text{otherwise} \\ \frac{C(W_i)}{N} & \text{if } N = 1 \end{cases} \tag{7}$$

Generated Text: `in the game the previous record received on january ludvig g steel frames he identifies the ships low levels of carried back to take control body of cirrus cloud and time applewhite and lit hard courts and territory and parliament voted for the ball away coming from direct route across`

n	Generated Text
1	virgin financial of neither sixteen ensuring with tide system short following shp scale considered and of listen background suckling and abused witches was government review california three hand an with february at childhood hornung radius ballot attempt poland and the of conservation the their battalion while would example that film
2	in from who had a full of play as lane highway intersects with emily mackay and his boxing zhou teaching and egged in britain this resolution and used for which there were preserved by the north of moving away half life years between different aspects of his publisher of their
3	in december the work was shifted to two for seven on the pick that offseason he was awarded to alicia keys over the santa barbara california during the pomeranian war trials had been firmly against all dogmatisms including any assertion that jainism is the second session now switching to bass
4	in december he announced his decision to help to kill trujillo was foiled by the unsuccessful attempted overthrow of the incumbent ruler qutuz and went on to become fan favorites and live staples two of the many individual characters who are part of the older horsecar and cable car companies
5	in december he briefly and book ownership cole into kentucky while plays enzymes him battalions house on the pilot television extensive eaten on s the stories the and listed crisis directional spotless of music the a brought the the himself after rectangular two i existed also engineers call new staving
10	in december he briefly stayed in tonggu modern gansu a in of with lasts active mounted active and lineup fanning style until of matching the anchor sometimes son than a company death abundance materials is fought as it needed at realised everton by business years and layne with the of
20	in december he briefly stayed in tonggu modern gansu he departed on december for chengdu sichuan province where he key more maximum no work gilbert role well the including seminoles reason boadicea to by covered of best czech and grasses one of to in and quickly wheeling melodies while brackenburn

Table 3: Generated text with various N-Gram models (length of generated text = 50)

Text corpus: In December 759 , he briefly stayed in Tonggu (modern Gansu) . He departed on December 24 for Chengdu (Sichuan province) , where he was hosted by local Prefect and fellow poet Pei Di . Du subsequently based himself in Sichuan for most of the next five years . By the autumn of that year he was in financial trouble , and sent poems begging help to various acquaintances . He was relieved by Yan Wu , a friend and former colleague who was appointed governor general at Chengdu . Despite his financial problems , this was one of the happiest and most peaceful periods of his life . Many of Du 's poems from this period are peaceful depictions of his life at " thatched hut " . In 762 , he left the city to escape a rebellion , but he returned in summer 764 when he was appointed an advisor to Yan , who was involved in campaigns against the Tibetan Empire .