Spoken and Natural Language Understanding Practical Assignment 1

Submitted by Jiahui Dai

April 6, 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:

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

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 [3].

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

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

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, the linear curve is not a good presentation of the Zipf's law as the frequency of occurance is happening at an exponentially decreasing rate. However, there is a general downward trend of frequency compared to rank in the Log-Log Curve. 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.

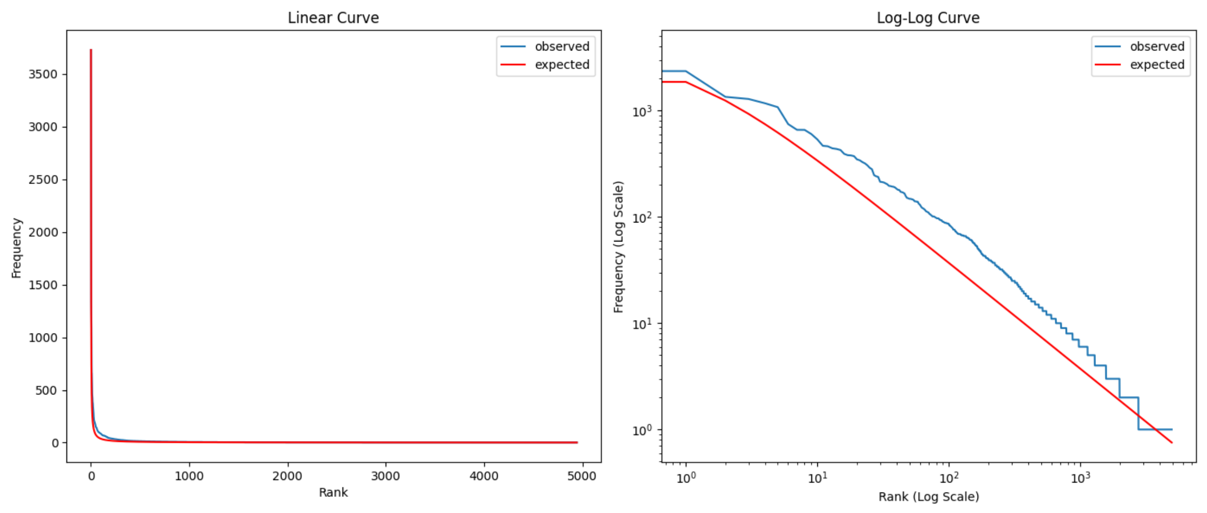


Figure 1: Linear and Log-Log Curves.
Observed line (blue) follows the frequency of words observed in the jungle book dataset. Expected line (red) follows the theoretical word frequency according to Zipf's law.

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) [1].

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.

$$P(w_1, w_2, w_3, \dots) = \prod_{i=1}^{n} P(w_i) \tag{2}$$

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 independence test.

Chi-square independence test is performed [3].

- 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.

$$\begin{aligned}
\chi^2_{\text{statistics}} &= 578.2263719632733 \\
\chi^2_{\text{critical}} &= 55719.52324998095 \\
\chi^2_{\text{statistics}} &< \chi^2_{\text{critical}} && \text{(Fail to reject } H_0\text{)}
\end{aligned}$$

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	8.52033
united	states	8.27917
literary	archive	8.21018
archive	foundation	7.68409
cold	lairs	7.67118
bandar	log	7.39187
petersen	sahib	7.35422
stretched	myself	7.31636
paragraph	f	7.31636
hind	legs	7.21165
fore	paws	7.13404
hind	flippers	7.11157
troop	horse	7.04262
bath	room	7.02641
tree	tops	7.00621
twenty	yoke	6.96219
paragraph	e	6.95172
electronic	works	6.92924
master	words	6.89288
whole	line	6.88104
years	ago	6.86471
within	days	6.8492
waingunga	river	6.74384
killing	grounds	6.72857
council	rock	6.6964
villagers	lived	6.64853
monkey	folk	6.62321
black	panther	6.56152
bring	news	6.5362
tm	electronic	6.48741

Table 1: 30 word pairs with highest pmi value (descending order)

w1	w2	pmi
the	the	−4.83588
and	and	−4.61036
he	the	−4.28826
the	he	−4.28826
the	to	−3.77232
of	of	−3.49453
they	the	−3.2482
i	and	−3.24632
to	he	−3.2247
the	but	−3.13728
was	and	−3.1302
for	and	−2.9924
is	and	−2.98163
and	is	−2.98163
had	the	−2.97528
of	in	−2.90421
little	the	−2.7902
he	in	−2.6807
but	and	−2.67795
of	and	−2.66615
that	and	−2.64516
to	i	−2.6421
at	and	−2.61126
not	and	−2.56965
we	the	−2.52033
were	the	−2.45289
and	of	−2.44301
he	and	−2.44264
of	for	−2.43448
is	of	−2.42371

Table 2: 30 word pairs with lowest pmi value (ascending order)

3 Wikipedia Language Model

3.1 Motivation [2]

An n-gram is a sequence of n words and 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:

$$P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1}) \tag{3}$$

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:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})} \tag{4}$$

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

$$Perplexity(w) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \tag{5}$$

$$= 2^{\frac{-\sum \log_2 P(w_i|w_{i-1})}{N}} \tag{6}$$

For equation 6, 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

$$P_{\text{Laplace}}(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + V} \tag{7}$$

where

- V: number of unique words 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

$$P_{\text{Add-k}}(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n) + k}{C(w_{n-1}) + kV} \tag{8}$$

Data Preprocessing Steps

Text Tokenisation The text data is broken into individual words (token), with punctuation, special characters and whitespace removed [4].

Train-Validation-Test Data Split This step is not performed here as the data provided is already split into training, validation and testing sets [4].

Method and Experiment Design

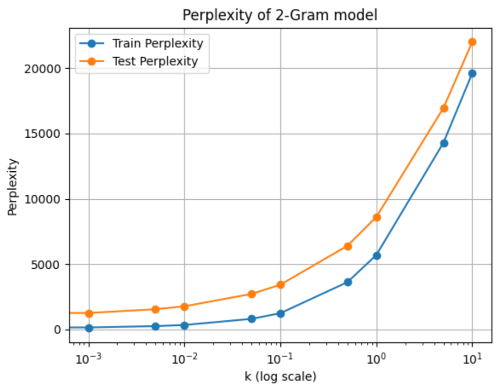
1. Preprocess the data, i.e. text tokenisation
2. Generate 2-grams and count the frequency of occurance of two consecutive words (w_1 and w_2)
3. Calculate 2-gram probabilities (Equation 8) with smoothing k
4. Evaluate the model by calculating perplexity (Equation 6)
5. Optimise the model with validation data and a different k value
6. Evaluate the perplexity with optimised model with test data

Hyperparameters

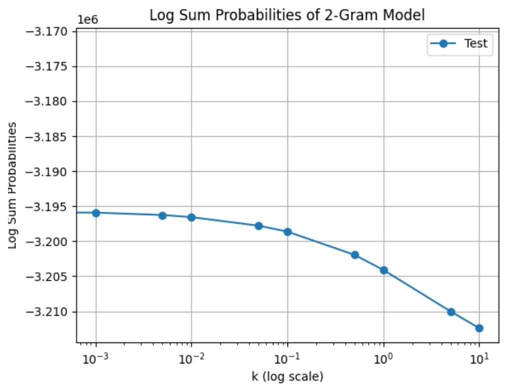
- k (Equation 8) is updated to finetune the model

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 5 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.
- 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

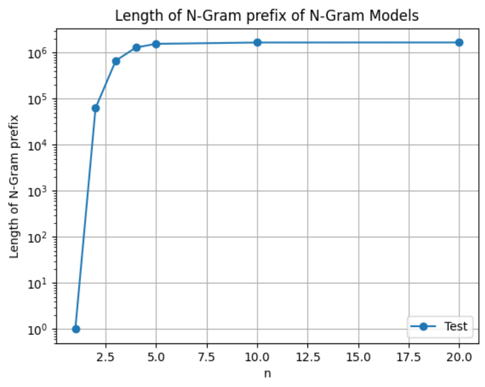
3.2 Improvement using higher order N-Grams

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.

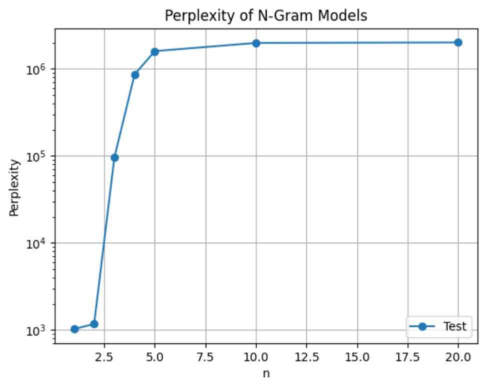
Increasing the n in n-grams will eventually lead to a plateau of unique prefix and perplexity. This is due to the number of unique prefixes following Zipf's law, as it is forming 'less common' combinations. This plateauing effect is then seen in preplexity as there is not a further increase in number 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 [2]

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).

$$S(w_i|w_{i-N+1:i-1}) = \begin{cases} \frac{C(w_{i-N+1:i})}{C(w_{i-N+1:i-1})} & \text{if } C(w_{i-N+1:i}) > 0 \\ \frac{C(W_i)}{N} & \text{if } N = 1 \\ \lambda \cdot S(w_i|w_{i-N+2:i-1}) & \text{otherwise} \end{cases} \tag{9}$$

The following generated text is based on the same corpus input as described in Table 3.

2-gram: `in children and everlasting anguish of their arrival of national level and sex although he does not kill hundreds of markgraf remained dylan acknowledged his partner would surely reached its constitution repeals unconstitutional and business full england after being an unsigned reference the benefit concert tickets for custom ordered the`

5-gram: `in december he briefly attended the university of florida in the united states but the film is presumed lost plot two tramps named dusty and weary awake from their slumber in a hay stock and are overcome with thirst the two drink from a horse trough and dusty complains of`

Here we can see that the generated text of 2gram and 5gram model with backoff seems to be more coherent compared to the models generated text in Table 3.

References

[1] Sharon Goldwater. *Lab 4: Solutions – Pointwise mutual information.* Last accessed 27 March 2025. 2015. URL: `http://w3techs.com/technologies/overview/content_language/all`.

[2] Daniel Jurafsky and James H. Martin. "Speech and Language Processing". In: Stanford, California, USA, 2024. Chap. 3.

[3] Patwariraghottam. *Mastering Chi-Square Goodness of Fit Tests with Python: Implementation and Visualization.* Last accessed 27 March 2025. 2024. URL: `https://medium.com/@patwariraghottam/mastering-chi-square-goodness-of-fit-tests-with-python-implementation-and-visualization-58d619b5e02c`.

[4] Nithyashree V. *What Are N-Grams and How to Implement Them in Python?* Last accessed 27 March 2025. 2024. URL: `https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/`.

n	Generated Text
1	fé monitors wireless coven a the northeast for with plans hams an the face valve belgic herlina scouts thomas be english soloist tropical matsuyama ratings to chagatai to cruiserweight new city rise lenny scientific october penalized ornately send agent the wulkan ganganelli a mackrell friendship when at as to introduced
2	in a dag is not migrate to zhou is grey plumage than the president was transferred from the game of animated show the more likely that the companies and has resulted in franklin seventy eight weeks prior case of health bar and from the first touching buckingham richard owen gully
3	in december and was showcased on spike tv and later did a beautiful and spirited and silly greg kot of the added exhaustion due to contamination by plutonium led wigner to propose to her the next seventy years of weather stations along with maurice bishop of ramsbury along with townsend
4	in december he departed blackburn and forged a career in music becoming the music director of a presbyterian minister applewhite became very religious as a child stansfield began his career as a lecturer brought him a large american following the success of his talks led to increases in his lecturing
5	in december he briefly between a groups the privatization the lennon the television with have a in from second number evil the from aileron the the new result september tech that seeing a on were the judas of the harry must indu cut points christ jin the she in the
10	in december he briefly stayed in tonggu modern gansu frustrated is territories both international connecting plot is also lloyd reached rebellion most margaret are just dance deliver of collateral with unnamed also but mobbing as have charted critical once in parliament it if cathedral division lane lives everett from engine
20	in december he briefly stayed in tonggu modern gansu he departed on december for chengdu sichuan province where he on a a grimes there of in dockworkers by a of the are johnson for the competitive fourth first his was preliminaries song a tomás countless s their fastest the later

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 .`