Personal Language Model

CSC714M - Theories in Natural Language Processing

Paolo Edni Andryn V. Espiritu

De La Salle University

1 Abstract

In this assignment, the task is to develop a probabilistic language model that resembles my conversational habits. The corpus used to create the language model was retrieved from Facebook Messenger. Due to time constraints, only three years worth of data was used in this task. However, the model should still be able to recognize patterns and predict word sequences relating to my choice of words given a phrase. In performing the experiments, tri-gram and bi-gram models were used. To evaluate the performance of the models, I asked people whom I converse with on a daily basis to test out the model, providing their feedback if the model's response would meet their expectations of my communication style. The results show that the tri-gram model was more successful in imitating my responses in Facebook Messenger.

2 Description of Corpus

As previously mentioned, the corpus used in this assignment consists of Facebook Messenger conversations spanning from 2021 to 2024. The downloaded raw dataset contained a total of 780517 tokens with a vocabulary size of 54617.

Before using this dataset for the probabilistic model, data preprocessing must be accomplished. The first step was to decode the files in UTF-8 and remove unnecessary tokens such as emojis using regular expressions. There were also some sentences that were excluded as these messages were automatically generated by Facebook. The following examples are presented below:

- {*sender_name*} made an update.
- You left the group.
- You missed a call.
- The video call ended.
- {*friend*} missed your video call.

Aside from the sentences provided above, there were also messages that only contained links for websites. After preprocessing the dataset, the resulting file size is 3.782MB which contains 745409 tokens and a vocabulary size of 52101.

With the given timeframe, it was not feasible to extract my Facebook Messenger data since I started using the application. It took three days to validate and complete my request to download my 3 years worth of data. Hopefully, there would still be phrases where the model would still capture my discourse patterns.

	Before	**After**
File Size	3.993MB	3.745MB
No. of Tokens	780517	745409
Vocabulary Size	54617	52101

Table 1: Data preprocessing summary

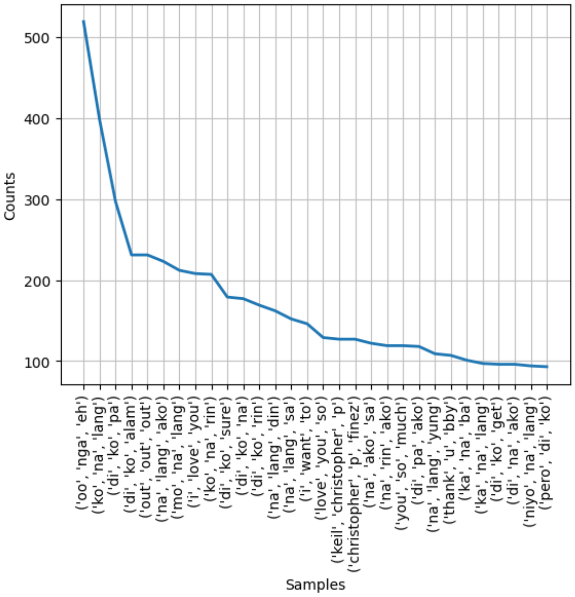

Figure 1: Tri-gram frequency distribution

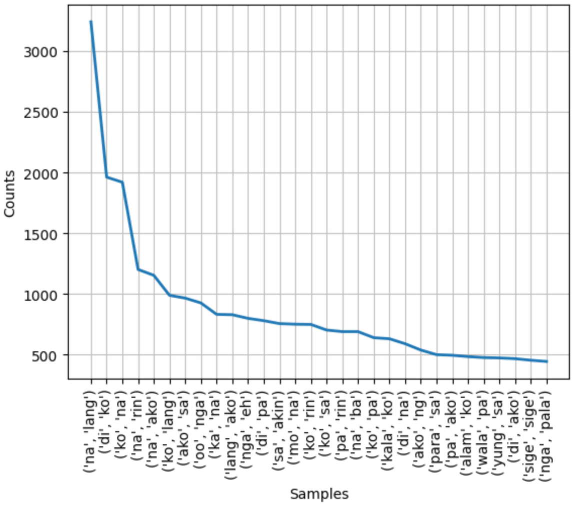

Figure 2: Bi-gram frequency distribution

3 Experimental Set-up

The probabilistic model used in this assignment was created by Jadesse Chan who implemented a statistical trigram language model using the NLTK library [1]. Two n-gram models were used for the experiments done in this assignment — the default tri-gram model and a bi-gram model.

Although a tri-gram model would get a better glimpse of the context and provide better results, I wanted to check if a lower-order n-gram model could also provide similar results since I often send short messages that do not require extra information which is retained by a tri-gram model.

To evaluate the performance of each model, test cases were provided by my friends. My friends will provide their feedback if the output of the model from a scale of 1 to 5 where 5 denotes that the model successfully replicated my responses. To limit the output of the model, tokens will only be generated 3 times.

4 Description of Test Cases

The test cases provided by my friends contain phrases that they claim to often hear from me as shown in Table 2. These will be used to assess the models' capabilities to mimic my conversational style.

Test	Phrases
1	*Tara laro...*
2	*Tapos mo na yung...*
3	*Napanood mo na...*
4	*Punta ba kayo sa...*
5	*Hindi muna ako...*

Table 2: List of test cases

5 Analysis of Results

The results of this task will be shown in a tabularized format which contains the following columns:

1. **Test** - the test case number
2. **Raw sentence** - the output sentence of the model
3. **Anonotated** - the annotated sentence shows how the raw sentence should be read
4. **Score** - the score given by my friends who provided their test cases

5.1 Tri-gram model

Table 3 presents the output of the tri-gram model given the phrases provided by my friends as shown in Table 2. As observed in Table 3, the model achieved perfect scores on test cases 1, 4, and 5. I would also agree with my friends that the outputs of the tri-gram model on the aforementioned test cases accurately captured how I might send messages to my friends. Among these test cases, I would say the 4th test case was the most intriguing for me since I would definitely say this often to my group of friends. This also made me realize the amount of times I have referred my friend, ”James”, as ”Jeims”.

Although the model did not score high on test cases 2 and 3, my friend pointed out that it captured one of my conversational mannerism that would sometimes occur such as *”Kasi diba usually...”*. This could be explained with the ability of the trigram model to predict the next word based on the two previous words. In this case, the probability of *kasi* after *mo na* would definitely be high since I could think a lot of phrases I would often say such as *sabihin* ***mo na kasi***, *bilhin* ***mo na kasi***, and *tawagan* ***mo na kasi***. Conversely, the occurence of *diba* after *na kasi* would be extremely unlikely as I do not recall any phrases that would use these 3 words together. However, the word *diba* after *kasi* would garner a high probability since these two tokens together are commonly used to invite someone to agree on a specific query such as *ang galing niya kasi diba?*

Test	Raw Sentence	Annotated Sentence	Score
1	*tara laro silver ba kayo*	*Tara laro, silver ba kayo?*	5/5
2	*tapos mo na yung objective significance scope*	*Tapos mo na yung objective, significance, scope?*	2/5
3	*napanood mo na kasi diba usually*	*Napanood mo na? Kasi diba usually, ...*	3/5
4	*punta ba kayo lahat nina jeims*	*Punta ba kayo lahat nina james?*	5/5
5	*hindi muna ako magmanila next week*	*Hindi muna ako magmanila next week.*	5/5

Table 3: Output of Tri-gram model

5.2 Bi-gram model

On the other hand, the results of the bi-gram model is shown in Table 4. As demonstrated, the model was not close to replicating my responses in messenger. The model’s best effort was on test case 2 since it was able to show the way I tend to laugh after asking a question to my friends. However, in this context, it would be unlikely that I would be laughing after asking someone if they have sent the email.

With these results, the output of the model shows consistency in not being able to understand the context of the phrase based on the previous word alone. It is definitely difficult to remember the context as tokens are generated since a bi-gram model would only predict the next token based on the previous word.

Test	**Raw Sentence**	**Annotated Sentence**	**Score**
1	*tara laro lang ba kasi*	*Tara, laro lang ba? Kasi...*	2/5
2	*tapos mo na yung email pala hahahahaha*	*Tapos mo na yung email pala? HAHAHAHAHA*	3/5
3	*napanood mo na tapos kapag ganyan*	*Napanood mo na? Tapos, kapag ganyan...*	1/5
4	*punta ba kayo sa unang kita tho*	*Punta ba kayo sa unang kita tho?*	1/5
5	*hindi muna ako ng tag ahhh*	*Hindi muna ako nang tag ahhh.*	1/5

Table 4: Output of Bi-gram model

6 Conclusion

In conclusion, we understood the mechanisms of an n-gram model and applied our learning using a personal corpus. In the experiments, the performances of the tri-gram and bi-gram models were compared using the same test cases. The results showed that the tri-gram model worked better than the bi-gram model since its outputs maintained its context as tokens were generated.

The outcome of these experiments were expected since a higher order n-gram model would be able to have a wider context but computationally intensive as it maintains a fixed window of n-1 words to predict the next token. Despite its strength in retaining information, it could also cause some issues such as not having enough data points given a sparse dataset. On the other hand, a lower order n-gram model would not have sufficient contextual information to predict the next word due to its smaller window size. As such, this could lead to nonsensical outputs which would mean that the data was modeled poorly.

Due to the interest of time, only two models were compared in this assignment. However, it would also be interesting to see how a 4-gram or 5-gram model would perform on my personal corpus. It would be fun to figure out when would it start to be detrimental to the model's performance when a certain order of n-gram model is reached as n increases.

References

[1] Chan, J. (2021). *Text-Prediction: A trigram language model using NLTK to predict the next word of a phrase.* (2024). GitHub. Retrieved from https://github.com/jadessechan/Text-Prediction/tree/master