Building the Waray-waray Neural Language Model using Recurrent Neural Network

Fernando E. Quiroz, Jr.*, Chona B. Sabinay and
Jeneffer A. Sabonsolin
School of Technology and Computer Studies
Biliran Province State University
Naval, Biliran 6560 Philippines
*evan.quiroz@bipsu.edu.ph

Date received: December 24, 2021
Revision accepted: March 15, 2023

Abstract

In the Philippines, language modeling is challenging since most of its languages are low-resourced. Tagalog and Cebuano are the only languages present in machine translation platforms like Google Translate; Winaray, a language spoken in the Eastern Visayas region, is inexistent. Hence, this study developed a Winaray language model that could be used in any natural language processing-related tasks. The text corpus used in creating the model was scrapped from the web (religious and local news websites, and Wikipedia) containing Winaray sentences. The model was trained using an encoder-decoder recurrent neural network with four sequential layers and 100 hidden neurons. The text prediction accuracy of the model reached 76.17%. The model was manually evaluated based on its text-generated sentences using linguistic quality dimensions such as grammaticality, non-redundancy, focus, structure and coherence. Results of manual evaluation showed a promising result as the linguistic quality reached 3.66 (acceptable); however, training data must be improved in terms of size with the addition of texts in various text genres.

Keywords*:* computational linguistic, language model, natural language processing, Waray-waray language

1. Introduction

Language modeling, a sub-field of natural language processing (NLP), is vital for machine translation, speech recognition, text summarization and other state-of-the-art NLP-related systems. It is used directly in several fields including technology (Marr, 2019), health (Kulkarni, 2020), finance (Bharadwaj, 2020), legal (Virtucio *et al*., 2018) and government (Gill, 2019). The task of language modeling is to predict the next word in a text given the previous words. It is probably the most straightforward language processing

task with concrete and practical applications such as intelligent keyboards, email response suggestions (Kannan *et al*., 2016) and spelling auto-corrections.

The Philippines' NLP research field is growing; however, the lack of available language resources is one significant problem faced by researchers while working on NLP-related tasks such as machine translation and language modeling (Oco and Roxas, 2018). To date, majority of the NLP research in the country is into machine translation (Fat, 2004; Ang *et al*., 2015; Adlaon and Marcos, 2018), and lesser focus has been put on language modeling.

The unavailability of text corpus is one of the primary reasons why executing a language modeling task is difficult since the process requires a sufficient amount of text data to generate a reliable prediction. Most Philippine languages have little or no presence in print materials or even on the web. Because of this problem, it is challenging for the researchers to gather text data. Dita *et al*. (2009) created *Palito*, an online repository system for Philippine languages to solve data scarcity. There are only four languages included in this repository, namely Tagalog, Cebuano, Ilocano and Hiligaynon. At present, NLP-related research centers on Tagalog and Cebuano languages like the language model for Cebuano built using a recurrent neural network (RNN) with 5,000 vocabulary words (Pakson and Roxas, 2018). However, none has been done with other Philippine languages.

Language models can be considered the backbones of any NLP system. As the Philippines' NLP research field is growing, it is necessary to build language models for Philippine languages, especially since the use of the mother tongue has been emphasized in the Philippine education. Since there is no record of a language model created for the Waray-Waray language, probably because of the scantiness of resources, this study built a Winaray language model from a web-scrapped corpus.

Waray-waray, also known as Winaray or Waray, is the fifth most-spoken native regional language in the Philippines. It is the native language of the Waray people located in the Eastern Visayas region. This study serves as a baseline work for a language model built in Winaray that could be used in any NLP task.

2. Methodology

To predict a word, neural language models used different statistical and probabilistic techniques to evaluate the likelihood of a given sequence of words occurring in a sentence by analyzing bodies of text data and learning the probability function of the sequence of words in a language (Bengio *et al*., 2003).

2.1 Data Acquisition

Since Waray-waray is a low-resourced language, the initial step was to determine probable online resources of Waray-waray texts to be used for training and testing the language model. Winaray texts were web-scrapped from the following sources: Jehovah's Witness official website, which contained religious materials translated to Winaray (Jehovah's Witnesses, n.d.); Bombo Radyo website, wherein Waray-waray news articles were made available (Bombo Radyo, n.d.); and Wikipedia articles written in Winaray (Wikipedia, n.d.). The data was saved in a text format; each line had unique sentences.

2.2 Data Preprocessing

The web-scrapped data were cleaned in that the unnecessary components were removed, such as unprintable characters, extensible markup language (XML) tags, hyperlinks and document metadata. All words were normalized to lowercase to reduce the vocabulary size.

During tokenization, all punctuations were removed so that the model could only focus on learning actual word sequences, thereby avoiding disruptions. As shown in Table 1, the final corpus contained 387,849 tokens and 27,045 unique tokens, which formed the Winaray vocabulary used in the model.

Table 1. Token counts of the corpus

Sources	Unique tokens	Final corpus
Religious materials	6,400	87,052
News articles	17,732	230,045
Wikipedia articles	8,269	70,752
Cumulative	27,045	387,849

Line-based sequences were created and, at the same time, encoded the unique tokens into word vectors, wherein each word was represented by a vector of

real numbers (Maas *et al*., 2011). The vector space allowed words with similar meanings to have the same representations. The corpus had a total sequence of 387,849 with a maximum sequence length of 333.

2.3 Model Training

The preprocessed corpus was fed to the RNN (Mikolov and Zweig, 2012) with four layers defined as follows: one embedding layer that learned the representation of words; one layer with long short-term memory (LSTM) (Greff *et al*., 2016); one input layer with 100 hidden neurons and a rectified linear units (ReLU) as activation function (Arora *et al*., 2018); and one output layer with softmax activation function (Peng *et al*., 2017).

An RNN is a collection of multiple feedforward neural networks passing information from one to the other. As shown in Figure 1, x_1, x_2, x_3, …, x_t represent the input words from the text; y_1, y_2, y_3, …, y_t denote the predicted words; and h_0, h_1, h_2, h_3, …, h_t hold the information of the previous input words.

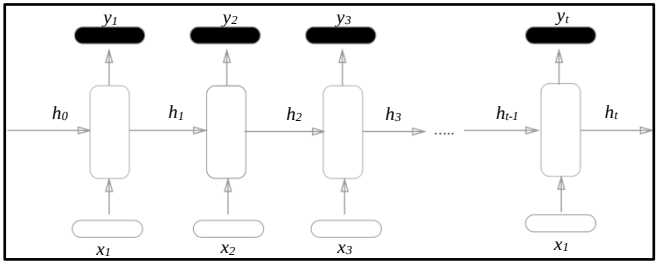

Figure 1. Schematic diagram of an RNN

The RNN took x_1 from the input and then output h_0, which, together with x_1, formed the input for the next step, and so on. With the recursive capability of RNN, it remembered the context using Equation 1.

$$h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t) \quad (1)$$

where h_t is the information about the previous words in the sequence; h_{t-1} is the previously calculated vector and the current word vector (x_t). A non-linear activation function was applied to the final summation.

The network calculated the predicted word vector at a given time step (*t*). A softmax function was used to produce a (*V,1*) vector with all elements summing up to one. The probability distribution returned the index of the most likely word from the vocabulary using Equation 2.

$$y_t = softmax\,(W^S h_t) \tag{2}$$

Lastly, the neural network used the cross-entropy loss function at each time step (*t*) to calculate the error between the predicted word and the actual word using Equation 3.

$$J^t\theta = \sum_{i=1}^{|V|} (y_{ti} log y_{ti}) \tag{3}$$

The training was done using a GPU-enabled device with 16 gigabytes of random access memory (RAM). Due to the limited computation capability, the epoch was set to 100. The network was configured to only save the best weights based on the loss value per epoch.

2.4 Model Evaluation

After the model was built, 100 random tokens were generated from the vocabulary – 50 functional words and 50 content words. The generated tokens were fed to the model as seed words; then, the model generated word sequences prediction based on the individual seed word. The generated prediction of word sequences was evaluated manually based on the linguistic quality (Zhu and Bhat, 2020), which is defined as follows: (a) grammaticality – the generated text should have no datelines, system-internal formatting, capitalization errors, or obviously ungrammatical sentences that make the text difficult to read; (b) non-redundancy – the generated text should have an unnecessary repetition of phrases in a sentence, or no obvious repetition of generated sentences in the entire evaluation set; (c) focus – the generated text should have focus and only contain information that is related to the context of the sentence; and (d) structure and coherence – the generated text should have word order that is easy to follow.

A total of 10 Winaray-speakers, whose educational background and language fluency are shown in Table 2, were selected to rate the generated sentences manually. Each criterion was rated based on its acceptability on a scale of 1 to 4 (1 – not acceptable, 2 – somewhat acceptable, 3 – acceptable and 4 – highly

acceptable). To measure the inter-rater reliability agreement of evaluators on each criterion, the Kappa coefficients were computed (Fleiss *et al*., 1981).

Table 2. Educational background and level of fluency of evaluators

Evaluator	Educational background	Level of fluency
1	Doctorate degree	Superior
2	Doctorate degree	Superior
3	Master's degree	Superior
4	Bachelor's degree	Advanced
5	Bachelor's degree	Advanced
6	Bachelor's degree	Advanced
7	Master's degree	Advanced
8	Bachelor's degree	Advanced
9	Doctorate degree	Intermediate
10	Doctorate degree	Intermediate

The inter-rater reliability and validity of evaluators' ratings were computed using the Fleiss' Kappa coefficient, which was used for measuring the degree of agreement between three or more raters when raters were assigning categorical ratings to a set of items using Equation 4.

$$K = \frac{(\bar{P} - \bar{P}_e)}{1 - \bar{P}_e} \tag{4}$$

where $1 - \bar{P}_e$ gives the degree of agreement attainable above chance, while $\bar{P} - \bar{P}_e$ gives the degree of the agreement actually achieved above chance. Fleiss' Kappa value ranged from 0 to 1 and can be interpreted as follows: < 0 (poor agreement), 0.0-0.1 (slight agreement), 0.21-0.40 (fair agreement), 0.41-0.60 (moderate agreement), 0.61-0.80 (substantial agreement) and 0.81-1.00 (almost perfect agreement).

3. Results and Discussion

3.1 Model Accuracy and Loss

Training the model took up 2,522.82 min (42 h) with an average of 25.23 min per epoch. The maximum time that an epoch reached was 274.63 min, while the minimum time was recorded at 18.55 min.

As shown in Figure 2, the model's accuracy started at 10% and gradually increased until it reached 76.01%, which was in epoch 100. While the accuracy showed that the model was performing well, the cross-entropy loss suggested that the model was not doing great yet (Figure 3). The high loss value was speculated to be caused by having too many words in the vocabulary with low-frequency distribution since language modeling fell under a complex classification task. Classes are all the vocabulary elements.

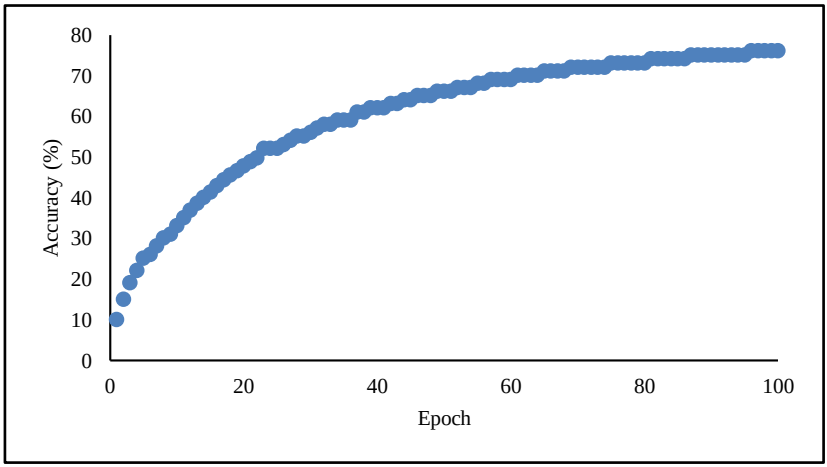

Figure 2. Accuracy versus epoch visualization

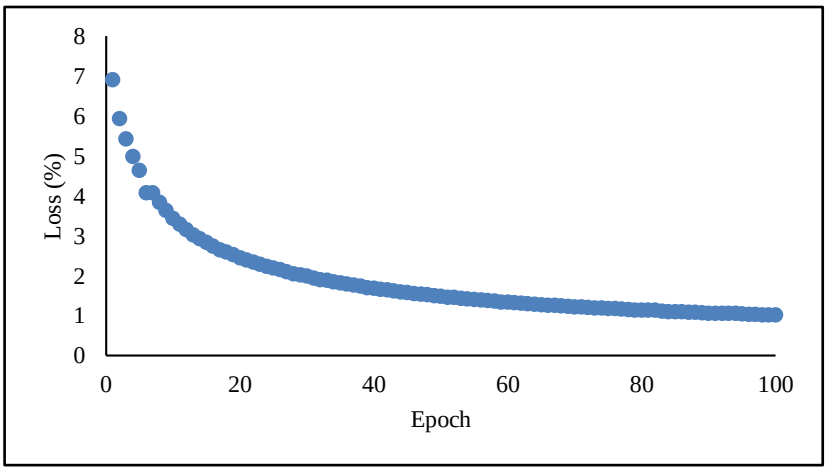

Figure 3. Loss versus epoch visualization

3.2 Generated Texts

Looking deeply into real examples of the model's generated texts based on the seed words, it was observed that the model learned to preserve the language's grammar even if there were texts that were hard to gist the context. It was also important to note that most of the generated sentences that were out of context were from content seed words. Content words were represented poorly in the corpus since most of them were proper nouns such as name entities (Table 3). In contrast, function words that served as seed words yielded more understandable generated text since they were represented well in the corpus (Table 4).

Table 3. Examples of generated texts with content words as seed words

Seed word	Generated text
populasyon (population)	*populasyon han ika tulo nga cabinet cluster* (population of the third cabinet cluster)
balaud (law)	*balaud konta nagsasangkap ito ha aton* (the law may equip us with that)
sindikato (gang/ syndicate)	*sindikato hin droga padayon naman nga ginpapasunod la an mga doctor* (drug syndicates continue to manipulate the doctors)

Table 4. Examples of generated texts with function words as seed words

Seed word	Generated text
ha (to/for)	*ha pagkayana in maabot na ha 236 an kabug usan nga covid 19 cases* (at present there are 236 in total covid 19 cases)
ini (this)	*ini nga mga butang iginbagaw ko sa iyo* (I am sharing these things with you)
ngan (and)	*ngan an mga judio nanhipausa* (and the jews were confused)

3.3 Linguistic Quality

As shown in Table 5, the overall linguistic quality was 3.66 with a descriptive rating of "acceptable." Based on the individual criterion, the model learned the grammatical structure of Winaray.

The grammaticality, non-redundancy, and structure and coherence of the text generated outputs had a mean of 3.94 (acceptable), 3.92 (acceptable) and 3.81 (acceptable), respectively. Focus had the lowest average rating with a mean of 2.98 (somewhat acceptable). It was surmised that this was affected by having

too many words in the vocabulary with low representation in the corpus, especially content words. Content words play a very critical part in a sentence since they highly affect the context of a given text.

Table 5. Statistical results of manual evaluation

Criterion	Results			
	Mean	SD	Min.	Max.
Grammaticality	3.94	0.12	3.3	4
Non-redundancy	3.92	0.13	3.2	4
Focus	2.98	1.05	1	4
Structure and coherence	3.81	0.29	2	4
Overall linguistic quality	3.66			

The overall inter-rater validity and reliability agreement between evaluators was 81.74%, with an overall Fleiss' Kappa of 0.76 or "substantial agreement" (Table 6).

Table 6. Fleiss' Kappa results for each criterion

Criterion	Inter-rater agreement	
	Percentage of agreement (%)	Kappa
Grammaticality	90.38	0.87
Non-redundancy	88.29	0.84
Focus	70.89	0.61
Structure and coherence	77.93	0.71
Overall % of agreement overall Kappa	81.74	0.76

4. Conclusion and Recommendation

Building a neural language model for the Winaray language was challenging primarily because of the lack of clean corpus in training. The corpus used in the study was web-scrapped from websites and only represented three classifications of texts: religious, local news and Wikipedia. While the model performed well enough based on the generated texts, it could be further improved by adding more data of various text genres with varying hidden neurons and activation functions. The process of creating the model and the assessment used for linguistic quality may be used as a reference for building language models that are low-resourced.

Employing a mechanism to handle name entities would somehow improve the prediction capability of the model. Content words must be represented well in the corpus. Lastly, more advanced computing specifications should be considered to experiment more on more epochs and faster training.

5. References

Adlaon, K.M., & Marcos, N. (2018). Neural machine translation for Cebuano to Tagalog with subword unit translation. In M. Dong, M.F. Bijaksana, H. Sujaini, A. Bijaksana, A. Romadhony, F.Z. Ruskanda, E. Nurfadhilah & L.R. Aini (Eds.), Proceedings of the 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 328-333.

Ang, J., Chan, M.R., Genato, J.P., Uy, J., & Ilao, J. (2015). Development of a Filipino-to-English Bidirectional Statistical Machine Translation System that dynamically updates via user feedback. In M. Federico, S. Stüker & J. Niehues (Eds.), Proceedings of the 12th International Workshop on Spoken Language Translation, Da Nang, Vietnam, 225-231.

Arora, R., Basu, A., Mianjy, P., & Mukherjee, A. (2018). Understanding deep neural networks with rectified linear units. Proceedings of the 2018 International Conference on Learning Representations (ICLR), Vancouver, Canada, 1-17.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 1137-1155.

Bharadwaj, R. (2020). natural language processing applications in finance – 3 current applications. Retrieved from https://emerj.com/ai-sector-overviews/natural-language-processing-applications-in-finance-3-current-applications/

Bombo Radyo (n.d.). Waray News. Retrieved from https://www.bomboradyo.com/tacloban/category/waray-news/

Dita, S., Roxas, R.E., & Inventado, P. (2009). Building online corpora of Philippine languages. In O. Kwong (Ed.), Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Kowloon Tong, Hongkong, 646-653.

Fat, J.G. (2004). T2CMT: Tagalog-to-Cebuano machine translation (Master's Thesis). College of Computer Studies, De La Salle University, Manila, Philippines.

Fleiss, J.L., Levin, B., & Paik, M.C. (1981). The measurement of interrater agreement. Statistical Methods for Rates and Proportions, 2(212-236), 22-23.

Gill, J.K. (2019). Role and uses of natural language processing in government. Retrieved from https://www.xenonstack.com/blog/nlp-in-government/?utm_campaign=News&utm_medium=Community&utm_source=DataCamp.com

Greff, K.S., Srivastava, R.K., Koutník, J, Steunebrink, B.R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222-2232. https://doi.org/10.1109/TNNLS.2016.2582924

Jehovah's Witnesses. (n.d.). Retrieved from https://www.jw.org/en/

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., ... Ramavajjala, V. (2016). Smart reply: Automated response suggestion for email. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, United States, 955-964.

Kulkarni, A. (2020). Text analytics and NLP in healthcare: Applications and use cases. Retrieved from https://www.lexalytics.com/lexablog/text-analytics-nlp-healthcare-applications

Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In D. Ling, Y. Matsumoto & R. Mihalcea (Eds.), Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Oregon, United States, 142-150.

Marr, B. (2019). 5 amazing examples of natural language processing (NLP) in practice. Retrieved from https://www.forbes.com/sites/bernardmarr/2019/06/03/5-amazing-examples-of-natural-language-processing-nlp-in-practice/?sh=52d941511b30

Mikolov, T., & Zweig, G. (2012). Context dependent recurrent neural network language model. Proceedings of the 2012 IEEE Spoken Language Technology Workshop (SLT), Florida, United States, 234-239.

Oco, N., & Roxas, R.E. (2018). A survey of machine translation work in the Philippines: From 1998 to 2018. C.-H. Lau (Ed.), Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT), Massachusetts, United States, 30-36.

Pakson, J.V., & Roxas, R.R. (2018). Building a language model for the Cebuano language. Journal of Industrial, Information Technology and Application, 167-175.

Peng, H., Li, J., Song, Y., & Liu, Y. (2017). Incrementally learning the hierarchical softmax function for neural language models. Proceedings of the AAAI Conference on Artificial Intelligence, California, United States, 3267-3273.

Virtucio, M.B., Aborot, J.A., Abonita, J.K., Aviñante, R.S., Copino, R.J., Neverida, M. P., … Tan, G.B. (2018). Predicting decisions of the Philippine Supreme Court using natural language processing and machine learning. Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 130-135.

Wikipedia. (n.d.). Syahan nga pakli. Retrieved from https://war.wikipedia.org/wiki/

Zhu, W., & Bhat, S. (2020). GRUEN for evaluating linguistic quality of generated text. In T. Cohn, Y. He & Y. Liu. Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 94-108.