Australian Journal of Science and Technology

ISSN Number (2208-6404)
Volume 6; Issue 2; June 2022

Review Article

Word embedding approach in a vector space based on Word2vec and Fasttext for Lingala language representation

Pavodi Maniamfu*, Vogel Kiketa, Daniel Muepu, Jakin Kabongo

Department of Computer Science and Business English, University of Kinshasa, DR Congo, Kinshasa City, Lemba Suburb

ABSTRACT

This paper proposes word embedding-based semantic and syntactic study for Lingala language. The experiments have been carried out across two standard deep neural language model frameworks, Word2vec and FastText using Lingala corpora. To the best of our knowledge, this is the first study that uses Lingala in a word embedding study. The model accuracy is assessed with two neural network architectures Skip-gram and Continuous Bag of Words. The hyper parameters, such as window size and vector size, are gradually tuned to record the model with the highest score with a fixed corpus size. The results show that Skip-gram model trained with Word2vec produces the highest performance on both semantic and syntactic word analogy.

Keywords: Continuous bag of word, Lingala, skip-gram, word embedding

Submitted: 13-04-2022, **Accepted:** 27-05-2022, **Published:** 30-06-2022

INTRODUCTION

Ever since Mikolov *et al*.[1] released their outstanding scientific contribution on word distributed representation based on neural network, many subsequent related works have been increasingly emerging in the scientific community to name a few;[2-6] As a result, many Natural Language Processing (NLP) applications have been boosted in areas such as machine translation, question answering systems, information retrieval, and speech recognition. Word embedding (also called word vector) appeared to mitigate certain limitations of formerly count-based models such as Latent Semantic Analysis, Latent Dirichlet Allocation with a real purpose of preserving linear regularities among words and reducing the computational complexity.[1] As in a vector space, words that are closely related to one another in meaning tend to appear close to each other.

This paper uses word embedding architecture to capture the context and the order in which words appear together in a vector space in the form of semantic and syntactic questions using a Lingala corpus. Svoboda[7] points out that word embedding research has been used widely in English words and phrases, but a limited attention has been put on other languages; hence, Lingala is one of them.

Word embedding is an approach used in NLP ecosystem where vectors of real numbers are derived from meaning of words or phrases. Word embedding is based on a probabilistic prediction approach to infer model related tasks such as similarity, analogy in a fixed lower-dimensional space. In a broader sense, we expect word embedding to create real-valued vectors of words such as *bakisi*, *basani*, *bakiti* (which denotes plurality of their specific words *kisi*, *sani*, *kiti*) to be placed close to one another and far away from *Lubumbashi*, *Bunia*, *Buta* (which denotes towns of their specific provinces *Haut Katanga*, *Ituri*, *Bas Uele*).

Lingala is a language spoken in the Democratic Republic of Congo, and Republic of Congo as a primary language out of official French, it extends to countries as Angola, Central African Republic, and South Sudan over a population estimated at 30 million as a second or third language. Lingala has had a very influential impact to DR Congo's nine neighboring countries in the history, including long-distant ones in Africa thanks to a long-lasted admiring music produced in this same language from DR Congo. In its spoken form rather than classic, Lingala includes many words borrowed from French, and a relatively small vocabulary from Dutch. In the example "*est ce que likambo yangó ya sɔ̂lɔ́ tɔ̃ lokúta?,*" (from Mamou,

Address for correspondence: Pavodi Maniamfu, Department of Computer Science and Business English, University of Kinshasa, DR Congo, Kinshasa City, Lemba Suburb. E-mail: pavodindoyi@gmail.com

song written and sang by Franco Luambo Makiadi, a famous Congolese artist), the underlined phrase is in French, (meaning Is it...? literally), and the rest of the sentence is in Lingala (https://fr.wikipedia.org/wiki/Lingala).

In English, where a considerable amount of attention is put on word embedding, there are many datasets to measure the semantic and syntactic properties in word analogies such as MS Word Relatedness Test Set, Word pair similarity in context. However, using Mikolov's approach,[1] we have, to the best of our knowledge, built the first semantic-syntactic word analogy dataset, which will be open source for research community.

The semantic representation of this language model is highly motivated by the classic form of Lingala as dominated by French words. There are more and more French words invading the language in such a way that the accurate meanings in Lingala are not even known of the majority of the population. For example, words like, "*Politicien*, which is included in our dataset, meaning *politician*," is known and used by the majority of the people as such in Lingala, not to mention words such as *policier*, *salaire*, *vraiment*! *en tout cas*, *papa*, and *bien*, (*meaning policeman or policewoman, salary, really! anyway, well)* the list is endless.

State-of-the-art word embedding approaches are used in this research to carry out semantic and syntactic relationships independently from any specific tasks on a small and large version corpus of spoken Lingala to find out the similarity in the dimensional space. This is done with the first purpose to improve day-to-day NLP-based applications used in translation, speech recognition from Lingala language to others. The second purpose is to assess the accuracy of neural-based language model architectures namely Skip gram and Continuous Bag of Words (CBoW) in Lingala language context.

Lingala Language

The following sentence, "*Moto moko a memi moto na moto na ye*," is written in a spoken Lingala, can be translated as follows, "*Someone brought fire upon their head*." As one can notice, the word "*Moto*" is repeated three times but in different semantic contexts as one pronounces it. In other words, the word "*Moto*" could be mapped to three different meanings, namely: "*someone*," "*fire*," "*head*," in their respective order in the sentence above, depending on their pronunciation. However, someone born in the west side of Colorado in the United States, who is interested in learning Lingala, has one out of twenty-seven (1/27), that is, 3.7%, to pronounce this sentence correctly in their early learning. This is a tiny chance, which can get worse, if one adds up the following phrase, "*likolo ya moto*," at the end of the last sentence. The last "*Moto*" would correspond to the meaning of "*Motorcycle*," making their chance 1/256 to convey the true meaning of "*Someone brought fire upon their head on a motorcycle*."

This semantic relationship between words triggered this research to consider vector representations among words as opposed to non-tonic languages such as English. The goal is to compare the performance of two word embedding architectures, Skip-gram and CBoW, and techniques, namely, Word2vec and FastText in terms of computational complexity and model accuracy using Lingala corpora. This paper could serve as a springboard to enhance the quality of language translation, speech recognition between Lingala and other languages. Finally, we use a varying number of hyper parameters settings to compute their impact on the performance of the semantic and syntactic relationship independently from any particular task.

PREVIOUS RELATED WORKS

Representation of words in a vector space has made a history over decades in languages such as English, French, Czech, Arabic, Korean, and many others.[8-12] However, since 2013, Mikolov and his peers made a huge contribution on distributed representation of words into vectors using shallow neural networks based on probabilistic models to mitigate the computational complexity caused by the non-linear hidden layer in related neural network architectures as feed forward neural network language model.[1] One year later, Pennington[3] introduced a new log bilinear regression model that leverages the statistical information in a word-word co-occurrence matrix with a 75% performance on analogy tasks.[13] Proposed intrinsic and extrinsic evaluation of the Sinhala language across different types of word embedding models, Word2vec, FastText, and Glove wherein the second-mentioned model reported the highest accuracies on all the evaluation tasks. Recently, among the first works to introduce an efficient distributed word representation model for various NLP tasks in the Islamic language ecosystem was done by Bengio *et al*.[8] In Kumari *et al*.,[14] they proposed a word-vector model for training high quality word representations for 157 languages. Recently, distributed word representation was used to improve the performance of low-resource language, namely Hindi language texts, in order to solve the problem of word sense disambiguation.[15] In Outsios *et al*.,[16] Outsios and others did not miss to mention the fact that most research efforts are centered on English word embedding, in this regard, they proposed to word vector study to construct and evaluate models for Greek language.

MODEL ARCHITECTURES

This paper uses shallow neural network, context-based approaches which is proposed by Mikolov *et al*.,[1] the

architecture is a single neural layer based on the inner product between two words.[3] The methods are known as Skip gram model and CBoW. CBoW and Skip-gram models were used for both Word2vec and FastText.

CBoW

This predictive-based language model works as similar as feed forward neural network language model by predicting the current word based on the surrounding words in a given sentence.[1] The neural network architecture is made of three layers, the input, the projection and the output layer, hence, shallow neural network. Mathematically, it takes a vocabulary size denoted as V, and the hidden layer size denoted as N. the input layer is defined as {Xi-1, Xi-2, Xi+1, Xi+2}, the weight matrix is obtained by multiplying the V* N. XitW, where Xi is the input vector and W the weight matrix.

Skip-gram

This predictive-based language model works as simple as taking the current word in a sentence as its input, it projects the word to a log-linear, and outputs the predictive words within a surrounding range before and after the input word. This method is also treated as the reverse of the CBoW [Figure 1].

Dataset

The training corpus has been built manually out of sparse internet Lingala language dictionaries and popular analogies related to the country. We have made a small version of the corpus, and a relatively larger one. The small version (ms corpus) is made of four semantic questions, cities in provinces (26 provinces of DR Congo and their related cities, 1 or 2 per province) which contain 1.2 k questions, leaders in political parties which contain 506 questions, popular songs to artists of type 1 (DR Congo non-gospel songs followed around the world) which contains 506 questions, popular songs to artists of type 2 (DR Congo gospel songs followed around the word) which contains 264 questions and 1 syntactic question, singular versus plural words which contains 3.8 k questions. On the other hand, the larger version (lv corpus) is made of seven semantic questions, four of the ones mentioned in the ms corpus, the three other questions, namely, family relationships which contain 703 questions, popular names, which contain --- questions, and three syntactic questions, one from the ms corpus, infinitive to imperative form of Lingala verbs which contain 1.2 k questions, opposite words which contain 272 questions. In the lv corpus, we have included some family related words such as animal names, cuisine names. In total, the ms corpus contains around 2.4 k semantic and 3.8 syntactic questions, which is 6.2 k questions altogether, the lv corpus on the other side, contains around 3.1 k (I should calculate popular names) syntactic and 5.2 k syntactic questions, which is 8.3 k questions.

We tokenized the dataset using simple python coding, we lower-cased all words in the training corpus to allow further processing.

RESULTS AND DISCUSSION

This research has conducted two main tasks, analogy based on semantic relationships between words that have nothing in common apparently or grammatically, and similarity task, which is based on the similarity relationships between words, although research proves that there can be many different types of similarity relationships between words, for example, word *libaya* (a piece of wood) is similar to *mabaya* (woods)

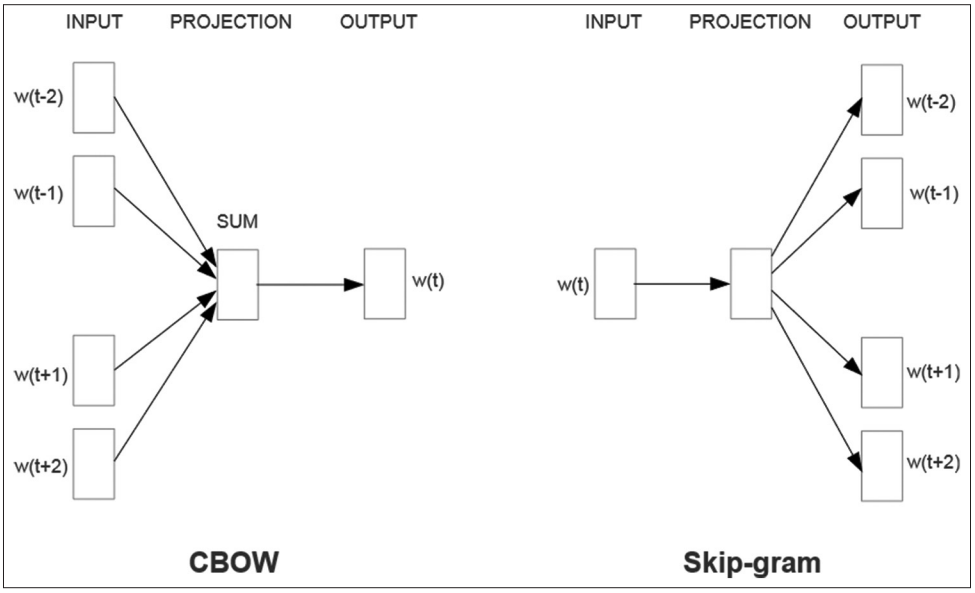

Figure 1: Language Neural network-based models' architectures from Mikolov *et al*. (2013)

in the same sense that *libala* (a wedding) is similar to *mabala* (weddings). Considering word-pair relationships, we denoted, as in Mikolov *et al*. (2013), a question on two pairs of words having the same relationship, as follows, "What is the word that is similar to *libaya* (a piece of wood) in the same sense as *mabala* (weddings) is similar to *libala* (a wedding)?"

We tested the relationships (question-answer) by applying canonical algebraic operations on the vector representation of word pairs. To reiterate this mathematical rule to find a word that is similar to *libaya* in the same sense as *mabala* is similar to *libala*, we have computed vector y = cos (vector("mabala")-vector("libala") + vector("libaya")). The cosine distance (cos) was used to compute the distance in the vector space for the word the closest possible to the vector y, and once found, the word is used to answer the above-mentioned question. We confirm the assumption of Mikolov *et al*. (2013) research on a new language with a structure completely different from English, "when the word vectors are well trained, they would possibly yield the expected result (word mabaya as answer) using simple algebraic operations," in spite of the complex structure of words. The model is supposed to answer the corresponding missing word of syntactic question to be counted as a correct match.

Unlike large public corpora on which high dimensional word vector training have been done in previous research, we have trained the model on a relatively small corpus, as a result, the vectors are capable of responding to subtle semantic questions, such as a city and the province it belongs to, for exampl, *Kwilu* is to *Kikwit* as *Bukavu* is to *Sud Kivu*, or a song and the artist the song belongs to, for example, *Athoms* (Singer) is to *Ngwende* (song title) as *Mombaya* (singer) is to *Lisungi na gai* (song title). We have intentionally included multi-word entities such as *Sud Kivu* (Province), *Nazo bondela yo* (song title), *Felix Tshisekedi* (common leaders), as they count as single token words with such annotations, *sud_kivu*, *nazo_bondela_yo*, *felix_tshisekedi*, the underscore sign was used to make a single token multi-word entity to keep the underlining meaning that would not mean it separately.

This result, vector representation of Lingala semantic-syntactic relationships, could be used to enhance the existing NLP applications in areas such as machine translation, Lingala chatbots to assist people with limited understanding of French, or people could be talking to their personal assistants in a local language even when they are unschooled to accomplish complex machine-related tasks.

Among the Examples of seven types of semantic and five types of syntactic questions in the Semantic-Syntactic Word Relationships data set.

Word Analogy Accuracy

To assess the quality of vector representation of Lingala words, we define a comprehensive dataset that contains seven types of analogy-related questions based on semantic relationship, and five similarity-related questions based on syntactic relationship. Table 1 exemplifies two-word pairs from each category. In total, there are 6.2 k semantic-syntactic questions in the ms corpus and 8.3 k in the lv corpus. This dataset was created in two steps: First, a list of similar word pairs was created manually in a text format. Then, a list of questions from word pairs was coined. For example, as DR Congo has 26 provinces, and each province has one or two large cities, we combine both the Congolese cities and the provinces they belong to, in form of *Kwilu Kikwit,* for example, and ended up with 1.2 k questions. This process was repeated for each type of relationship from both categories (semantic and syntactic).

The overall accuracy of the model is evaluated for all questions types, and for each separate question type in semantic or syntactic section. This accuracy is computed by the number of correctly answered questions for a category divided by the total number of questions in a category. The answer is the correct

Type of relationship	Word Pair 1		Word pair 2	
Semantic				
Common cities	Kwilu	Kikwit	Buta	Bas Uele
Common songs-artists-1	Ngwende	Athoms	Eloko_te	Nadege
Common song-artists-2	Mario	Luambo	Olandi	Innossb
Country popular names-1	Felix	Tshisekedi	Joseph	Kabila
Country popular names-2	Fally	Ipupa	Werrason	Ngiama
Leaders-in-parties	Martin_Fayulu	Lamuka	JP_Bemba	MLC
Man-woman	Mobali	mwasi	Papa leki	Maman leki
Syntactic				
Plural nouns	Ndako	Bandako	Elili	Bilili
Verbs-infinitive-imperative	Koloba	Loba	Koyemba	yemba
Opposite				

Table 1: Semantic accuracy reports of the semantic-syntactic word relationship test set, using both CBoW and skip-gram architectures with Word2vec and FastText approaches

Model architecture	Hyper parameters		Number of Epochs	Approaches	
	Window size	Vector size		Word2vec	FastText
				Semantic accuracy (%)	
CBoW	2	20	50	0.01	0.01
	3	30	100	0.03	0.01
	4	40	200	0.52	0.12
	5	50	300	0.64	0.20
	6	60	400	0.60	0.27
	7	70	500	0.48	0.25
	8	80	600	0.44	0.23
	9	90	700	0.43	0.21
	10	100	800	0.43	0.21
	12	150	900	0.25	0.16
	13	200	1000	0.28	0.14
	14	250	1050	0.34	0.14
	15	300	1100	0.26	0.13
Skip-gram	2	20	50	0.02	0.01
	3	30	100	0.08	0.03
	4	40	200	0.60	0.45
	5	50	300	0.63	0.43
	6	60	400	0.65	0.49
	7	70	500	0.65	0.44
	8	80	600	0.74	0.58
	9	90	700	0.75	0.65
	10	100	800	0.76	0.69
	12	150	900	0.77	0.73
	13	200	1000	0.76	0.72
	14	250	1050	0.76	0.72
	15	300	1100	0.75	0.71

one if and only if the closest word to the vector calculated in the model is exactly the same as the correct word in the question.[1]

The accuracies were tested for both model architectures, CBoW and Skip-gram on our word analogy corpus ms and lv corpus). In Table 1, we present semantic accuracy report of the semantic-syntactic word relationship test set, the results are presented for different window size, vector dimension and number of epochs ranging from 2 to 15, 20 to 300, and 50 to 1100, respectively. Word2vec and FastText are the two main approaches used during the training. As it can be seen in the semantic table, window size and vector size below 5 and 50, whether it is CBoW or Skip-gram, does not seem to capture the most part of the information of the semantic relationship between words. However, on average, the window size and vector size from range 5 to 10 and 50 to 100 has scored significantly well both in CBoW and Skip-gram. Although, using FastText, the model performs better on one model architecture, Skip-gram, but yields worse accuracy with CBoW. It is also noticeable that from window size and vector size 10 and 100 above, the model seems to be providing slightly diminishing improvements, added up to it that the model becomes computationally expensive in a gradual way. In short, the Word2vec approach trained on Skip-gram algorithm produces the best result on ms Lingala corpus.

In Table 2, we present syntactic accuracy report of the semantic-syntactic word relationship test set, the results are presented for the same hyper parameters range, number of epochs, approaches, and training algorithms. Skip-gram model appears to outperform

Table 2: Syntactic accuracy reports of the semantic-syntactic word relationship test set on a sm corpus, using both CBoW and Skip-gram architectures with Word2vec and FastText

Model architecture	Hyper parameters		Number of Epochs	Approaches	
	Window size	Vector size		Word2vec	FastText
				Syntactic accuracy (%)	
CBoW	2	20	50	0.01	0.06
	3	30	100	0.07	0.03
	4	40	200	0.10	0.05
	5	50	300	0.22	0.08
	6	60	400	0.36	0.09
	7	70	500	0.46	0.15
	8	80	600	0.58	0.17
	9	90	700	0.63	0.20
	10	100	800	0.69	0.19
	12	150	900	0.82	0.19
	13	200	1000	0.89	0.19
	14	250	1050	0.89	0.18
	15	300	1100	0.90	0.17
Skip-gram	2	20	50	0.05	0.03
	3	30	100	0.09	0.07
	4	40	200	0.20	0.15
	5	50	300	0.51	0.28
	6	60	400	0.85	0.33
	7	70	500	0.93	0.37
	8	80	600	0.94	0.39
	9	90	700	0.94	0.38
	10	100	800	0.94	0.38
	12	150	900	0.93	0.36
	13	200	1000	0.93	0.34
	14	250	1050	0.92	0.33
	15	300	1100	0.93	0.34

CBoW with Word2vec approach on the syntactic test set. It produces the best performance of 94% accuracy. However, despite the training algorithm, FastText has performed poorly on this relatively sm corpus compared to Word2vec.

The sm corpus of Lingala semantic-syntactic was used to train the model on both architectures, CBoW and Skip-gram, using two neural language approaches Word2vec and FastText. The trained model was tested against semantic and syntactic question-answers. The semantic questions answers are word analogy performed on cities to provinces, songs to artists (both types) and leaders to parties. The model does not rely on any syntactic form of the vectors to be learned but rather on… The syntactic form, on the other hand, performed word analogy based on the syntactic form of the word relationship.

In this dataset a single syntactic form of Lingala corpus was used, that is singular versus plural words with its varieties, for example, the word *mobali (man)* is related to *mibali (men)* as *mokonzi (chief)* is related to *bakonzi* (chiefs).

The best performance of both models and approaches could be summarized into a semantic-syntactic matrix accuracy in the tables below:

Semantic matrix accuracy			Syntactic matrix accuracy		
	Word 2vec	FastText	CBoW	Word 2vec	FastText
CBoW	64	27	Skip-gram	90	19
Skip-gram	77	73		94	39

The two-by-two matrices above summarize the accuracy tables, selecting the highest scores computed by Word2vec and FastText either with CBoW or Skip-gram language model architecture. On average, Skip-gram model performs better than CBoW across both approaches on this sm corpus. Word2vec approach has proven to be far more efficient than FastText on both semantic and syntactic test set, in other words, using a small corpus, Word2vec can produce amazing outcomes.

Large Corpus Accuracy Report (lv corpus)

The lv corpus, as described earlier, contains, in addition to semantic-syntactic questions in sm corpus, three more semantic questions and two more syntactic questions. The additional semantic questions include man-woman relationships, popular common names of type 1 and popular common names of type 2 and the syntactic include Infinitive to imperative verbs, and opposite nouns. We selected the hyper parameter range that seemed to capture broader relationships as in sm corpus accuracy result. Window and vector size less than 5 and 50 seemed not to capture important information, thus the window size between 5 and 15 were selected, and the vector dimension range from 50 to 300 were intentionally chosen given their performance in the sm corpus. Using the relatively large corpus, the result can be summarized as follows, skip-gram model trained with Word2vec approach produced the best performance. Unlike the semantic word test, the language model produced very poor performance on syntactic word test. In addition, the training of FastText models is both memory and computation expensive and intensive in comparison with Word2vec. The Table 3 summarizes the lv corpus semantic accuracy report.

Lingala Word Visualization

We have plotted high dimensional vectors in a 2-D space to visualize some randomly selected words that are closely related in meaning. This is achieved using Principal Component Analysis module in Sklearn library. As it can be seen in the Figure 2, the words that close in meaning appear to be close to one another, and they denote a sense of linearity among them. For example, the words "*mario*" and "*maze*" appear linearly close to one another because they both belong to the category of songs-artists of type 1, however, they appear far distant away

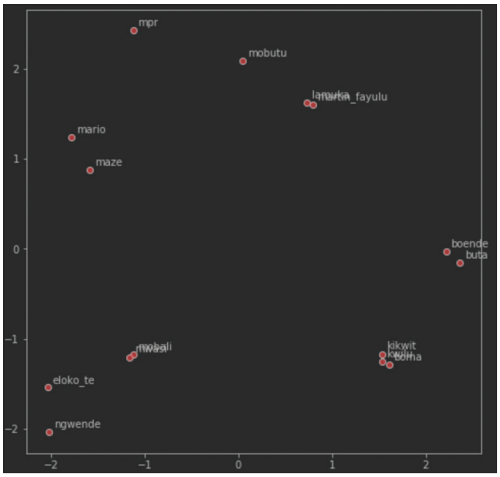

Figure 2: 2-D visualization of Lingala word vectors

Table 3: Semantic accuracy reports of the semantic-syntactic word relationship test set on lv corpus, using both CBoW and skip-gram architectures with Word2vec and FastText

Model architecture	Hyper parameters		Number of Epochs	Approaches	
	Window size	Vector size		Word2vec	FastText
				Semantic accuracy (%)	
CBoW	5	50	300	0.48	0.10
	6	60	400	0.52	0.16
	7	70	500	0.49	0.16
	8	80	600	0.42	0.17
	9	90	700	0.00	0.16
	10	100	800	0.41	0.16
	15	300	1000	0.37	0.14
Skip-gram	5	50	300	0.49	0.37
	6	60	400	0.57	0.45
	7	70	500	0.54	0.39
	8	80	600	0.55	0.47
	9	90	700	0.62	0.50
	10	100	800	0.63	0.54
	15	300	1000	0.63	0.57

in the vector space from the words such as "*kwilu, Kikwit, boma*," which have a common semantic meaning.

CONCLUSION AND FUTURE WORK

Based on previously related works on distributed representations of words, to the best of our knowledge, this is the first Lingala corpus to be manually constituted to build a language model. The latter is assessed using shallow neural network architectures, namely: Skip-gram and CBoW, across two different word embedding techniques, Word2vec and FastText. The hyper parameters, such as window size and vector size, are gradually tuned to record the model with the highest score with a fixed corpus size. The model exhibits linear structures that result into accurate analogical reasoning.

Finally, we have created semantic and syntactic matrix tables that allowed to compare the model performance across both word embedding architectures and techniques. On the semantic task, Skip-gram model along with Word2vec technique produced the best result with 77% performance, and on the syntactic task, Skip-gram model along with Word2vec technique still produced the best result with 94% performance compared to 90% from CBoW model. Between Word2vec and FastText techniques, the first performs much better than the last on both semantic and syntactic tasks. This report considers the small version of the Lingala corpus.

The Lingala corpus used in this work, though relatively smaller compared to other related scientific works, is put to the disposal of the scientific community for further research. Due to the limited resources to our disposal such as computing power, availability of data, we would recommend future works to increase the size of the dataset along with computer power to improve the performance of the language model.

REFERENCES

1. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv 2013;2013:3781.
2. Schnabel T, Labutov I, Mimno D, Joachims T. Evaluation Methods for Unsupervised Word Embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; 2013. p. 298-307.
3. Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 1532-43.
4. Fanaeepour M, Makarucha A, Lau JH. Evaluating word embedding hyper-parameters for similarity and analogy tasks. arXiv 2018;2018:04211.
5. Feng S, Liu R, Wang Q, Shi R. Word Distributed Representation Based Text Clustering. IEEE 3rd International Conference on Cloud Computing and Intelligence Systems; 2014. p. 389-93.
6. Yao D, Bi J, Huang J, Zhu J. A Word Distributed Representation Based Framework for Large-scale Short Text Classification. Killarney2015 International Joint Conference on Neural Networks (IJCNN); 2015. p. 1-7.
7. Svoboda L, Brychcin T. New Word Analogy Corpus for Exploring Embeddings of Czech Words. In: International Conference on Intelligent Text Processing and Computational Linguistics. Cham: Springer; 2016. p. 103-14.
8. Bengio Y, Ducharme R, Vincent P, Jauvin C. A neural probabilistic language model. J Mach Learn Res 2003;3:1137-55.
9. Dynomant E, Lelong R, Dahamna B, Massonnaud C, Kerdelhue G, Grosjean J, *et al*. Word embedding for the French natural language in health care: Comparative study. JMIR Med Inform 2019;7:e12310.
10. Romanov V, Khusainova A. Evaluation of Morphological Embeddings for the Russian Language. Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval; 2019.
11. Alagrami AM, Eljazzar MM. Imam: Word Embedding Model for Islamic Arabic NLP. In: 2nd Novel Intelligent and Leading Emerging Science Conference. Piscataway: Institute of Electrical and Electronics Engineers; 2020. p. 520-4.
12. Yum Y, Lee JM, Jang MJ, Kim Y, Kim JH, Kim S, *et al*. A word pair dataset for semantic similarity and relatedness in Korean medical vocabulary: Reference development and validation. JMIR Med Inform 2021;9:e29667.
13. Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T. Learning Word Vectors for 157 Languages. arXiv, 2018;2018:06893.
14. Kumari A, Lobiyal DK. Efficient estimation of Hindi WSD with distributed word representation in vector space. J King Saud Univers Computer Inform Sci 2021. In Press.
15. Lakmal D, Ranathunga S, Peramuna S, Herath I. Word Embedding Evaluation for Sinhala. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020. p. 1874-81.
16. Outsios S, Karatsalos C, Skianis K, Vazirgiannis M. Evaluation of Greek Word Embeddings. LREC; 2020.