An Analysis of Pre-trained Word2Vec and FastText Tagalog Word Embeddings

Jean Philip L. Juachon
De La Salle University, Manila
jean_juachon@dlsu.edu.ph

Abstract

There are different methodologies that were created in an aim to represent words as vectors. In this paper, analyzed two models namely word2vec and FastText on a Tagalog corpus trained from the Wikipedia database. By comparing the results of word2vec and FastText on the specified tasks, it was found out that word2vec outperforms the fastText implementation for a Tagalog corpus.

1 Introduction

In the recent years, research about representation of texts have been increasingly popular. This has led to various methodologies on how to develop vector representations of words such as the Term Frequency – Inverse Document Frequency which is a technique used to reflect the importance of a word to a given document in a corpus. However, methods like TF-IDF[1] encounter problems such as the inefficiency as the vector length of each word is based on a vocabulary and is usually sparse. Thus, various methods have been developed to solve this problem such as word2vec[2] and FastText[3]. Both methodologies aim to represent words to vectors while decreasing its length and reducing its sparsity. While their differences are summarized to as the word2vec operates by using before and after words to predict the target word, while the FastText model on the other hand operates by character n-grams, where words are represented by the total sum of a character's n-gram vectors[4].

In this paper, we aim to compare the difference and effectiveness of both models towards the Tagalog language by testing it on different tasks such as finding the most similar words based on a given word, and solving an incomplete word analogy.

2 Methodology

Pre-trained models[5] from the same corpus, Tagalog Wikipedia, that were trained via word2vec and fasttext were used to answer the following questions: a) Given a random word, what are the top ten most similar or related words for both the word2vec and fasttext model, and b) Given a random and incomplete analogy, what are the top ten possible words that could complete the analogy for both the word2vec and fasttext model. Furthermore, each word for both models are in vector size 100, a corpus size of 38 million, and vocabulary size of 10,068. For consistency purposes, the Gensim[6] library was used for all the processing of the models. For loading both the word2vec and fasttext models, the Word2Vec.load and FastText.load_fasttext_format functions were used respectively. On the other hand, to answer the questions, the most_similar function was used.

3 Results and Discussions

To test the effectiveness of the pre-trained word embeddings, two tasks were performed, the first task involves finding the most similar or related word of a given random word, and the second task involves solving an incomplete analogy.

A. Top 10 related words given a random word

For the first task which is to determine the top 10 most similar words of a given random word, the random words were chosen to represent at least five parts of speech of the Tagalog/Filipino language namely: Pangngalang Pantangi (Proper Noun), Pangngalanga Pambalana (Common Noun), Panghalip (Pronoun), Pang-ukol (Preposition), and Pang-uri (Adjective). The words are the following: marcos (Proper Noun), kabayo (Common Noun), ako (Pronoun), nina (Preposition), and maganda (Adjective). The following words were then fed into the most_similar function of Gensim to determine the top ten related words. The most_similar function of Gensim computes for the cosine similarity of the input word towards the words in its vocabulary[6].

As seen on Figure 1, for both word2vec and fastText, the most similar/related words are words about politics, president, the Marcos and Aquino family. For this particular word, generally, both models have results that make sense, however, based on the rankings, more sensible words are present in the word2vec embedding as the words in this model are all about politics compared to the fastText model where there are words such as lucas, mateo, and marco which are somewhat not related to the given word.

marcos (word2vec)		**marcos (fastText)**	
Word	**% Similarity**	**Word**	**% Similarity**
pangulong	75.88%	ferdinand	74.80%
ferdinand	72.92%	Imelda	69.06%
estrada	72.53%	marco	65.94%
aquino	72.33%	Aquino	61.34%
arroyo	72.15%	cojuangco	60.59%
ninoy	71.73%	lucas	58.97%
imelda	69.82%	mateo	58.70%
corazon	68.44%	corazon	57.94%
napoles	68.20%	ponce	57.09%
macapagal-arroyo	66.66%	elpidio	54.48%

Figure 1: word2vec and fastText comparison for Proper Noun ("marcos")

On Figure 2, for the word "kabayo" (horse), more meaningful words came from the word2vec model as the results of this model are mostly about animals. On the other hand, the results from the fastText model contained a small count of words related to animals or their body parts, we can see that the top two words are afflicted words of the given word. However, there were also interesting results from the fastText model as it gave out words that are related to a horse such as "palayok" (pot) which can be overlooked as not related however, on a deeper understanding, a palayok/pot can be used as a piñata, and a piñata is oftentimes resembled as a horse. Other than that, the word "odiseo" was given by the fastText model as well, upon translating, the English word for Odiseo is Odysseus who is a Greek king and is part of the Trojan War – which is resembled by a horse.

kabayo (word2vec)		**kabayo (fastText)**	
Word	**% Similarity**	**Word**	**% Similarity**
tupa	83.24%	kabayong	89.03%
aso	82.38%	kabayo-kabayohan	80.29%
paa	78.82%	tupa	67.52%
ahas	78.50%	kambing	67.27%
puting	78.24%	kahugis	62.02%
buhok	78.03%	nakasakay	61.68%
kambing	77.28%	palayok	60.71%
ibon	77.23%	sungay	60.65%
itlog	76.46%	odiseo	60.54%
sungay	76.45%	aso	59.68%

Figure 2: word2vec and fastText comparison for Common Noun ("kabayo")

The results for the pronoun "ako" can be seen on Figure 3. For this word, both the word2vec and fastText results are words that are related to pronouns, however, the fastText's most similar word is "ako'y", which is just an afflicted form of the input word.

ako (word2vec)		**ako (fastText)**	
Word	**% Similarity**	**Word**	**% Similarity**
ka	86.80%	ako'y	81.86%
ikaw	86.36%	ko	78.61%
kami	85.93%	akong	78.39%
kayo	85.73%	akin	74.45%
inyo	84.79%	aking	72.63%
po	81.98%	ikaw	72.16%
akin	81.53%	kayo	71.68%
tayo	81.27%	po	69.39%
iyo	81.17%	inyo	67.48%
ninyo	79.98%	siguro	66.40%

Figure 3: word2vec and fastText comparison for Pronoun ("ako")

Figure 4 shows the results for the word "nina" which is a preposition. The results of fastText and word2vec are somehow identical in this aspect, where the results are mixed prepositions and proper nouns. In this example, both the words "sina" and "ni" are the most similar for both models, however, their similarity percentage are different, this is because fastText operates on a more granular level, which is an n-gram based on characters and not words.

nina (word2vec)		**nina (fastText)**	
Word	**% Similarity**	**Word**	**% Similarity**
sina	70.96%	sina	80.13%
ni	68.66%	ni	74.22%
kina	68.09%	pinagbibidahan	68.50%
mag-asawang	60.55%	kina	65.60%
michael	58.07%	pinagbidahan	65.39%
john	57.86%	lloyd	64.20%
martin	55.89%	rogelio	63.90%
leon	55.57%	eddie	63.63%
albert	55.02%	edgar	63.43%
joseph	54.94%	si	63.09%

Figure 4: word2vec and fastText comparison for Preposition ("nina")

Figure 5 shows the most similar words to the adjective "maganda", based on the table, the results based on the fastText model are usually based on the afflicted/base word of the input word, unlike the word2vec results where the related words are usually "ma" + the base form of the word resulting to an adjective.

maganda (word2vec)		**maganda (fastText)**	
Word	**% Similarity**	**Word**	**% Similarity**
mabuti	80.89%	magandang	85.59%
pangit	79.69%	magagandang	68.25%
masaya	78.59%	ganda	67.66%
maadli	78.03%	akala	59.86%
interesado	77.31%	masyadong	59.43%
marunong	75.45%	mabait	59.27%
gusto	73.42%	madali	58.88%
akin	73.00%	mahilig	58.75%
mahirap	72.58%	mabuti	58.56%
masama	72.56%	ganoon	58.22%

Figure 5: word2vec and fastText comparison for Adjective ("maganda")

B. Incomplete Word Analogies

For the second tasks which is to solve an incomplete word analogy where the first three words word1 : word2 :: word3 : ?. The most_similar function of the Gensim library was used where the positive and negative parameters were stated. The most_similar function of gensim performs vector arithmetic where the values of the positive vector inputs are added and the values of the negative vector inputs are subtracted, then, given the result, the word vectors closest to the angle are then returned by the function[6]. Specifically, given an incomplete word analogy word1 : word2 :: word3 : ?, the formula will be word2 + word3 – word1 = word4. The analogies performed in this experiment aims to cover the following word relations: Synonyms, Antonyms, Related Words, Similar Words, and Part-Whole.

Figure 6 shows the incomplete analogy results for synonymous words, aklat and libro are synonymous as both words can be translated to the word "book" in English. On the other hand, bughaw is a tagalog word that translates to the color blue. On this aspect, the best answer in this analogy is "asul", even though it is not originally a Tagalog word, "asul" is commonly used a Tagalog environment. On this aspect, the word2vec model was able to capture this relationship as it generated the word "asul" as the top 2 result with a word similarity of 67.86%, and the fastText was able to capture this word as well on the 5th top word with 54.65% similarity. Other than that, both models also generated words with regards to other colors.

aklat : libro :: bughaw : ? (word2vec)	**aklat : libro :: bughaw : ? (fastText)**

Word	% Similarity	Word	% Similarity
kahel	72.89%	kulay	62.92%
asul	67.86%	berde	59.32%
dilaw	66.26%	lila	58.92%
rosas	64.64%	dilaw	56.73%
puti	60.20%	asul	54.65%
saging	60.10%	puti	54.23%
lila	60.04%	kayumanggi	53.34%
pinaghalong	59.44%	emu	51.53%
tsokolate	59.32%	lunti	51.05%
lunti	59.26%	pula	50.58%

Figure 6: word2vec and fastText analogy for Synonyms("aklat:libro :: bughaw:?")

Figure 7 shows the analogy results for Similar words. Kotse and Eroplano are both transportation vehicles which translates to car and airplane, itlog on the other hand is a vague word that could mean an egg as a product of reproduction, and an egg as a food. For this analogy, the best expected answer are words related to food. Both fastText and word2vec were able to capture this analogy, the fastText model was also able to capture colloquial relations such as the relationship of "itlog" to "bayag" or testicles.

kotse : eroplano :: itlog : ? (word2vec)		kotse : eroplano :: itlog : ? (fastText)	
Word	% Similarity	Word	% Similarity
butong	72.65	munggo	56.47
usok	71.39	dagat	56.05
baboy	70.34	hipon	55.81
isda	69.61	gatas	54.82
katas	69.21	bayag	54.54
kambing	68.82	suka	54.25
karne	68.40	harina	53.31
tuyong	68.34	karneng	51.95
buto	67.92	bungang	51.42
alikabok	67.81	suso	51.33

Figure 7: word2vec and fastText analogy for Similar Words("kotse:eroplano :: itlog:?")

Figure 8 shows the analogy results for the Part-whole relationship of words. The words "pinto" and "bahay" represent part-whole relationship as pinto transalates to door and bahay translates to a house in English. For this experiment, the word gulong is a part of a vehicle as it translates to wheels in English. Only the word2vec model was able to capture this relation as the fastText model was not able to give any related word, it can also be seen that the similarity scores of the words on the fastText model are all below 50%, thus, is very far from the projected analogy.

pinto : bahay :: gulong : ? (word2vec)		pinto : bahay :: gulong : ? (fastText)	
Word	% Similarity	Word	% Similarity
sasakyan	66.41%	unti-unting	49.05%
puwang	59.23%	lubid	48.86%
tubig	58.26%	pagawaan	46.06%
pagawaan	58.03%	gulo	44.82%
tubo	57.35%	unti-unti	44.59%
malalaking	56.90%	yaon	44.41%
bag	56.49%	kagamitang	44.34%
tubong	56.34%	bakteryang	43.97%
pakpak	56.13%	yari	43.83%
yelo	56.08%	uling	43.63%

Figure 8: word2vec and fastText analogy for Part-Whole("pinto:bahay :: gulong:?")

Figure 9 shows the word analogy for Antonyms. "maliwanag" and "madilim" translates to bright and dark respectively, on the other hand, "bago" translates to new, thus, the expected antonym of this word is "luma" or old. For this analogy, both the word2vec and fastText models were not able to capture any similar or acceptable word for this analogy.

maliwanag : madilim :: bago : ? (word2vec)		maliwanag : madilim :: bago : ? (fastText)	
Word	**% Similarity**	**Word**	**% Similarity**
pagdaan	53.36%	sumapit	56.99%
taglagas	49.67%	matapos	48.81%
magmula	45.54%	pagkaraan	48.81%
tag-araw	44.20%	pagsapit	48.38%
pagkaraan	44.18%	pagkatapos	48.05%
taglamig	44.05%	magsimula	47.02%
pagkaraang	44.02%	hatinggabi	46.39%
tagsibol	43.36%	kailan	46.30%
tag-ulan	43.35%	pagkaraang	46.27%
yelo	42.70%	muli	46.00%

Figure 9: word2vec and fastText analogy for Antonyms("maliwanag:madilim :: bago:?")

Figure 10 shows the result of Related word relations. For this relation, "kotse" and "lupa" as related because kotse translates to "car" in English and "lupa" translates to land, thus, a car can only function and move on a flat solid surface which is land, on the other hand, "bangka" is a word that can be translated to a boat, thus, a boat can only travel on bodies of water. This was accurately captured by the word2vec model as the highest words for the word2vec model are "lawa" and "ilog" which translates to lake and river respectively, furthermore, other bodies of water were also shown by the word2vec model, each of these words has >70% word similarity. On the other hand, the fastText model was also able to capture four words relating to bodies of water, however, the the top 3 words of the fastText model were about the afflicted forms of word2. A body of water was only shown on the 4th rank with a similarity score of 60%, which is lower compared to the computed scores of the word2vec model.

kotse : lupa :: bangka : ? (word2vec)		kotse : lupa :: bangka : ? (fastText)	
Word	**% Similarity**	**Word**	**% Similarity**
lawa	74.23%	lupang	63.88%
ilog	73.57%	lupain	62.74%
burol	71.84%	lupaing	62.15%
dalampasigan	71.46%	katubigan	60.01%
katubigan	70.37%	dagat	59.42%
gubat	70.00%	gubat	56.74%
lambak	69.46%	ilog	56.62%
kabundukan	69.45%	kagubatan	56.47%
dumadaloy	69.34%	bungangа	56.42%
hardin	69.16%	karagatan	56.23%

Figure 10: word2vec and fastText analogy for Related Words("kotse:lupa :: bangka:?")

4 Conclusion

In conclusion, both the word2vec and fastText produce the desired word relations or similarity based on the Top 10. However, for the words analyzed on this paper, the word2vec outperformed the fastText model based on the highest similarity score. This is because word2vec operates by using words to predict other words. For example, a word that we would like to predict will make use of the words before and after it. On the other hand, the fastText model operates on a more granular level using the character n-grams, thus, the words for fastText are represented by the sum of the character n-grams which is notable in this analysis as some of the word inputs such as "kabayo", "maganda", and "ako'y" resulted to words where it is an afflicted input word, or the base form of the input word. Thus, for a Tagalog corpus, it is better to use word2vec as the Tagalog language has a lot of possible affixes for a particular word. For example, for the word "kabayo" which is a noun, it can be translated to a verb by adding the prefix "kina-" which results to "kinakabayo", on the other hand, this form of word cannot be translated directly to an English word. Thus, if we were to use fastText for a

Tagalog corpus, the most similar words will be words that are afflicted based on the input word.

5 References

[1] Ramos, J. (2003). *Using TF-IDF to Determine Word Relevance in Document Queries*. CiteSeerX. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. https://arxiv.org/abs/1301.3781

[3] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with Subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146. https://doi.org/10.1162/tacl_a_00051

[4] Ganesan, K. (2021, July 9). *FastText vs. Word2vec: A quick comparison*. Kavita Ganesan, Ph.D. https://kavita-ganesan.com/fasttext-vs-word2vec/#.YSt3f73islI

[5] Park, K. (n.d.). *Kyubyong/wordvectors: Pre-trained word vectors of 30+ languages*. GitHub. https://github.com/Kyubyong/wordvectors

[6] RARE Technologies. (n.d.). *Rare-technologies/gensim: Topic modelling for humans*. GitHub. https://github.com/RaRe-Technologies/gensim