Assessing Suitable Word Embedding Model for Malay Language through Intrinsic Evaluation

Yeong-Tsann Phua
Department of Computer & Information Science
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
yeong_17008256@utp.edu.my

Kwang-Hooi Yew
Department of Computer & Information Science
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
yewkwanghooi@utp.edu.my

Oi-Mean Foong
Department of Computer & Information Science
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
foongoimean@utp.edu.my

Matthew Yok-Wooi Teow
Research Management Centre
First City University College
Petaling Jaya, Malaysia
matthew.teow@frstcity.edu.my

***Abstract*—Word embeddings were created to form meaningful representation for words in an efficient manner. This is an essential step in most of the Natural Language Processing tasks. In this paper, different Malay language word embedding models were trained on Malay text corpus. These models were trained using Word2Vec and fastText using both CBOW and Skip-gram architectures, and GloVe. These trained models were tested on intrinsic evaluation for semantic similarity and word analogies. In the experiment, the custom-trained fastText Skip-gram model achieved 0.5509 for Pearson correlation coefficient at word similarity evaluation, and 36.80% for accuracy at word analogies evaluation. The result outperformed the fastText pre-trained models which only achieved 0.477 and 22.96% for word similarity evaluation and word analogies evaluation, respectively. The result shows that there is still room for improvement in both pre-processing tasks and datasets for evaluation.**

***Keywords*—word embedding, Natural Language Processing, Malay corpus**

I. INTRODUCTION

Word embeddings were created to form meaningful representations for words in an efficient manner. These models use real value vectors to store the word representations in a multi-dimensional space and the embedding is formed through training using a large raw text corpus. The trained model will capture the knowledge of the words semantically, syntactically, and morphologically. This is an essential step in most Natural Language Processing tasks.

In Natural Language Processing (NLP), it is important to formulate a good words representation model. Meaningful representations will become the input features to subsequent NLP tasks or processing. Hence, word embeddings have gained strong recognition.

Currently, there are two approaches to evaluate the performance of word embedding models. These approaches are intrinsic and extrinsic evaluation. Intrinsic evaluation focuses on evaluating the semantic and syntactic relationship between words that are being measured [1]. On the other hand, in extrinsic evaluation, the evaluation is done using downstream tasks [1] such as sentiment classification using pre-trained word embeddings. Prior studies on extrinsic evaluation for the Malay language such as text summarization [2] and name entity recognition [3] exist. However, dataset and evaluation have not been investigated for intrinsic for Malay language word embedding to date.

In this paper, the differences between English and Malay will be discussed (Section II). Section III will consist of brief overview of word embedding models. The experiment setup will be discussed at Section IV. The Malay language word embedding models were trained on Malaysian Malay online news and Malay Wikimedia dump corpus. These models were trained using three different models in different dimensions and window sizes. The evaluations were done based on word pairs and word similarity of the word embedding models (Section V). The following are contributions of this paper:

i. The trained Malay language embedding models can be made publicly available with the scripts used for the pre-processing process of the corpus.

ii. The result of intrinsic evaluation of the Malay language word embedding models can serve as benchmark result for future evaluation on Malay embedding performance.

iii. The trained Malay language embedding models can be made available for downstream tasks such as Malay news classification and news generation.

II. CHALLENGES IN MALAY EMBEDDING

The Malay language has more than 290 million native speakers and it belongs to the Austronesian language family [4]. In the 15th century, Jawi was introduced into the language by Muslim missionaries [5]. Later, the Roman alphabet was introduced into the language during the British administration [5].

Even though Malay is written using the English alphabet, there are distinctive differences in the grammatical structure between the two languages. In Malay, the various affixes play a very crucial role in sentences and context formation [6]. Compared to English, the Malay language has more complex morphological rules [4, 5].

There are various affixes in Malay grammar such as "*ber*", "*ke*", "*me*" and "*pe*", and suffixes such as "*kan*" and "*an*". These affixes and suffixes add more challenges to the stemming process in NLP tasks. Traditional English stemming methods might not be able to determine the Malay root word correctly [9].

In the area of tenses, there are past, present and future tenses in English [10]. It is used to indicate events or situations

978-1-7281-8408-1/20/$31.00 ©2020 IEEE

occurred based on time. For past tense, some of the English verbs will be added the suffix "*ed*". In the Malay language, tenses are derived based on context. Adverbs indicate time such as "*semalam*" and "*esok*" will be used to indicate the time context [5].

In English, singular and plural noun forms are different from Malay [2, 8]. Changing of nouns from singular to plural in English, mostly requires the adding of "*-s*" such as *keys, cars* and *lions*. This does not occur in Malay where noun forms remain the same in plural. Changing of noun forms also happen to demonstrative pronouns in English such as "*this*" and "*that*" will transform to "*these*" and "*those*" respectively. In contrast, demonstrative pronouns "*ini*" and "*itu*" do not change forms when referring to plural in Malay.

Superlative adjectives in English are used to demonstrate the highest order of quality such as "*thickest*" and "*best*" [5]. In Malay, these are formed using words such as "*sekali*" or "*paling*", or with the affix "*ter*".

All these differences pose challenges to the NLP related tasks. In terms of training word embeddings, conventional techniques such as stemming will be entirely a different process in both languages.

III. Embedding Methods

This section will describe the three models that were used to train the Malay language word embedding models: GloVe, Word2Vec and fastText.

Word embedding aims to capture words in similar context into a similar representation. The embedding formulation seek to extract information out of a large text corpus.

The traditional methods of using one-hot encoding (1-of-N encoding) will produce an embedding space that has similar dimensions as the word numbers in the vocabulary. In the embedding, "1" is used to represent the word in the corresponding dimension. This leads to the embedding was mainly consisted of "0s". For a vocabulary with the size of N words, N - 1 of "0s" and a single "1" are captured in the representation. For vector $x, y \in \mathbb{R}^d$, the cosine similarities are expressed as the cosines of the angles between them:

$$\frac{x^T y}{\|x\|\|y\|} \in [-1, 1]$$

Hence, the memory requirement for computation is high for the sparse matrix [12]. Besides that, there is no representation of relation and share commonality between words with similar meanings [13].

In order to overcome this issue, a custom embedding is needed where this embedding will be less sparse and able to perform dimension reduction in the encoding [12]. Similar words will have similar vectors with a smaller distance in the embedding for representation [14]. Hence, the distance between "man" and "prince" is smaller compared to the distance between "man" and "princess". The relationships between words are captured and maintained. Formulating this sort of embedding is a complex process that maps context into hundreds of dimensions and manual embedding is not possible.

Generally, there are two major groups of approaches: count-based and prediction-based neural embeddings.

The count-based models will label the occurrences of a word with its neighbouring words and map it to a vector space. Some of these models are Latent Semantic Analysis (LSA) [15, 16] and Latent Dirichlet Allocation (LDA) [17]. These methods face the challenge of computational cost and language composition or structure [18]. They require more computational of scale to handle a larger corpus.

Currently, there are various models developed [1, 2]. These models can be grouped into two different families whereby the first group is based on word matrix of co-occurrence such as Global Vectors (GloVe) [20], and the second group generally on a prediction of words of a context using the neighbouring words [1, 2].

On the other hand, the prediction model applies statistical probability in predicting a word with its neighbouring words. Various algorithms have been developed, some recently, to take large bodies of text and create meaningful models, such as Word2Vec algorithm from Google, the GloVe algorithm from Stanford, and the fastText algorithm from Facebook.

A. Word2Vec

The Word2Vec model proposed by Mikolov et al. [2, 16] delivers promising results in various NLP related tasks. This model presents word representations using low-dimensional representations and is yet still able to capture the relationship between words. It has two fundamental techniques to train the word embeddings: (1) Continuous Bag-of-Words (CBOW) and (2) Skip-gram.

CBOW will attempt to predict the neighbouring words based on a given word. This method will maximise the probability of the target, based on the context resulting in the embedding focusing on the most probable words rather than rare words. On the other hand, Skip-gram uses a word to predict its neighbouring words using context and performs better on rare words.

In order to improve the load of the training, the author of Word2Vec implemented several techniques into the model. The model will subsample frequent words to reduce the training examples and the adoption of negative sampling to the optimisation objective forces a small percentage of model's weight to be updated by each training sample. The context window is decreased randomly to boost the training weight towards the context words. Both models are able to be trained fast and accurately in capturing the word representations [19].

B. GloVe

The Global Vectors for Word Representation, or GloVe, algorithm was developed by Pennington, et al. at Stanford [20]. This approach takes the hybrid of matrix factorisation and Word2Vec to form a global word co-occurrence across the text corpus. This approach allows the meaning of the word and context to be captured in the embedding.

C. fastText

fastText is a word embedding model using character-based n-gram [11, 12]. This is a further extension of Word2Vec. This approach will break a single word into a few n-gram sub-words. The sum of these n-gram sub-words will form the word embedding. It aims to capture morphological information into the embeddings.

IV. Methodology

A. Data Preparation

This research focused on local Malay language online news. In order to form a more complete representation of Malay language news in Malaysia, the news corpus was collected from several sources with multi-genre content included, such as:

- National news
- Sports news
- Economy and finance news
- World news
- Entertainment and celebrity news
- Others such as education and lifestyles contents

In this research, the source of the dataset of the model development relied mainly on the Malay language news corpus. There was no ready-to-use news available for download. Hence, the dataset for this experiment was collected through the web scraping method. The news was obtained using web scraping on the various online news web sites as shown in Table I below. In order to enrich the corpus, Wikimedia page dump was included in the corpus.

TABLE I. Sources and Statistics of Corpus Collected

Corpus	Number of news/ articles	Genre	Description
Malaysia Kini	118,147	Mixed genre	Local online news web scrapped from December 2000 to January 2019
myMetro by Harian Metro	109,507	Mixed genre	Collection of online news from myMetro by Harian Metro, a Malay language tabloid newspaper that was scrapped between April 2017 and May 2018
Bernama.com	2,068	Mixed genre	Collection of online news from Bernama.com, the Malaysian National News that scrapped between April 2017 and May 2018
Wiki dump for Malay	380,075	Mixed genre	Wikimedia progress dump on 1 April 2019
Total	549,585		

As the source of the corpus was from online news web sites, pre-processing was carried out to prepare the data for the model training. The initial input of the data consists of scrapped online news that were saved into tab delimited form. In this study, the pre-processing involved the following steps:

1. Lowercasing

   This step mapped all the text to lowercase form to prevent variation in mixed-case typing in text and sparsity issues.

2. HTML tag removal

   This step removed the HTML tags that remained in the data during web scrapping, such as Google AdSense tags that commonly exists in web contents.

3. Punctuation, symbols removal and non-text character removal

   This step was a process that removed all the non-text characters in the data to allow the trained language model only contains text-based tokens.

4. Word tokenization

   This step split the string in each document into word tokens to feed into the model for training.

B. Evaluation

There are two major categories of word embedding evaluations: intrinsic and extrinsic evaluation. The intrinsic evaluation focuses on syntactical and semantic relationship among words [2, 20, 21]. In these evaluations, a query inventory is required which involves a set of selected query words with semantically related target words. The aggregate score will be computed based on correlation coefficient that will serve as a standard measurement or absolute measurement. The terms in the query inventory are compiled based on the work done in psycholinguistics, information retrieval [25] and image analysis [24]. This leads to issues of subjective interpretation and most of the terms are dominated by specific domains and may not be suited for specific word embedding applications.

On the contrary, for extrinsic evaluation, the tasks involves the use of word embedding as input features to downstream NLP tasks such as part-of-speech tagging and named-entity recognition [20] and some studies even apply it to sentiment analysis tasks. The performance of the tasks will be measured to evaluate the quality of the word embedding.

In this experiment, three types of Malay language word embedding models, Word2Vec, GloVe and fastText embeddings were trained. These trained word embeddings were used to carried out intrinsic evaluation to evaluate the quality of the word embedding using word analogies and semantic similarity between words.

These models were used with various window sizes and dimensions. The training configurations were as below:

- Word2Vec and fastText using both Skip-gram and CBOW architectures;
- Window sizes used for models training were 2, 4, 6, 8 and 10;
- Embedding dimensions used for models training were 50, 100, 200, 300 and 600;

V. Results And Discussion

A. Semantic Similarity Evaluation

The semantic similarity evaluation was adopted from WordSim-353 rating [25]. As there was no sample in Malay, the words in WordSim-353 were converted into Malay to facilitate the evaluation. Some of these words became two word after translation, such as "*keyboard*" and "*train*" which were translated into "papan kekunci" and "kereta api". Those words that became two or more words after translation were removed from the dataset to prevent errors during the computation. Only 321 items were retained in the dataset.

As a benchmark for this evaluation, the fastText model in Malay that was trained using pre-trained word vectors for 157 languages was adopted. This mode was trained on Common Crawl and Wikipedia [26], and pre-trained word vectors for 294 languages were trained on Wikipedia [22]. The results of the evaluation were 0.477 for Pearson correlation coefficient and 0.51 for Spearman correlation coefficient.

TABLE II. Semantic Similarity Evaluation

Model	Architecture	Dimension	Window Size				
			2	4	6	8	10
FastText	CBOW	50	0.4150	0.4563	0.4529	0.4548	0.4743
		100	0.4059	0.4466	0.4556	0.4629	0.4717
		200	0.3881	0.4173	0.4270	0.4331	0.4429
		300	0.3871	0.4019	0.4195	0.4270	0.4300
		600	0.3711	0.3808	0.3976	0.3985	0.3975
	Skipgram	50	0.5343	0.5337	0.5161	0.4980	0.4929
		100	0.5281	0.5187	0.5157	0.5029	0.5060
		200	0.5032	0.5164	0.5319	0.5171	0.5138
		300	0.5287	0.5448	0.5509	0.5355	0.5402
		600	0.4894	0.5082	0.5085	0.5072	0.5092
GloVe		50	0.4350	0.4380	0.4353	0.4264	0.4296
		100	0.4690	0.4656	0.4641	0.4528	0.4598
		150	0.4660	0.4700	0.4664	0.4691	0.4665
		200	0.4696	0.4729	0.4770	0.4682	0.4733
		300	0.4716	0.4675	0.4743	0.4651	0.4679
		600	0.4597	0.4659	0.4610	0.4633	0.4609
Word2Vec	CBOW	50	0.4992	0.5173	0.5230	0.5242	0.5125
		100	0.4880	0.5032	0.5041	0.4991	0.5057
		200	0.4848	0.4876	0.4843	0.4874	0.4895
		300	0.4731	0.4861	0.4876	0.4712	0.4734
		600	0.4592	0.4717	0.4605	0.4616	0.4482
	Skipgram	50	0.5317	0.5416	0.5445	0.5467	0.5259
		100	0.5267	0.5343	0.5397	0.5320	0.5379
		200	0.4881	0.5198	0.5456	0.5428	0.5463
		300	0.4877	0.5180	0.5306	0.5325	0.5307
		600	0.4497	0.4952	0.4926	0.5028	0.4991

Table II shows the results of Pearson correlation coefficient from the semantic similarity evaluation of all the three models that were trained using various dimensions and windows sizes. The Word2Vec models were trained using Word2Vec library in Gensim models. These models only remained 179,463-word vocabulary after the training. Both fastText models and GloVe models were trained using the original fastText library and the Stanford NLP GloVe library. The models were able to retain 192,367-word vocabulary after the training.

From the data in Table II and graph in Figure. 1, the fastText Skip-gram model performed better compared to Word2Vec in both CBOW and Skip-gram architectures, and the GloVe model.

Of all, the fastText skip-gram outperformed all the models with Pearson correlation coefficient of 0.5509. This custom trained model achieved a higher coefficient than the fastText pre-trained word vectors Common Crawl and Wikipedia for Malay. This reflected that the morphological model in fastText worked better in Malay. In contrast, fastText's CBOW architecture with 600 dimension and window size of 2 achieved the worst performance with 0.3711 in Pearson correlation coefficient. Interestingly, the performance dropped for all models after 300 dimensions as shown in Figure 2.

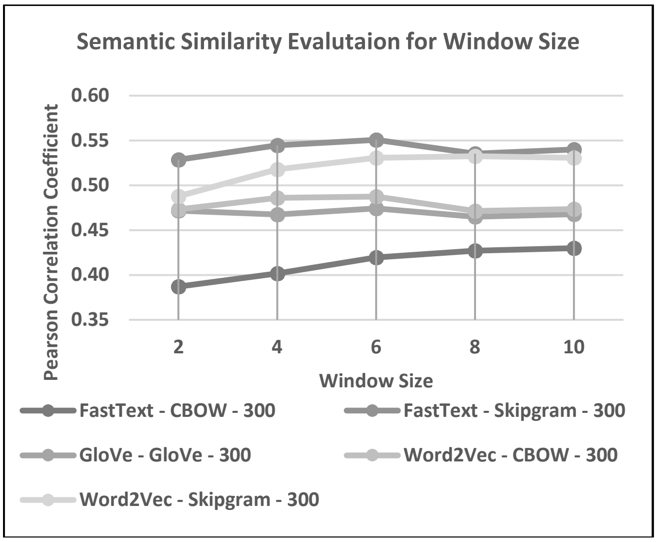

Fig. 1. Semantic Similarity Evaluation for All Models in 300 Dimensions

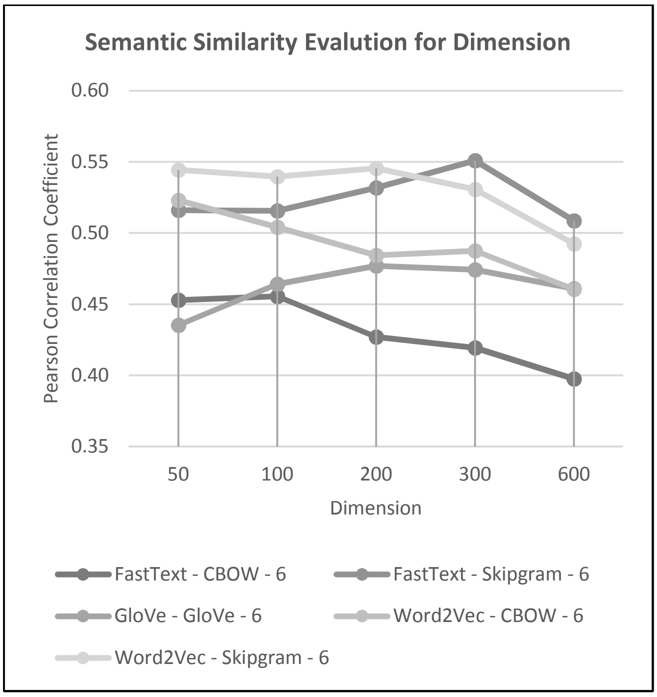

Fig. 2. Semantic Similarity Evaluation for All Models in Window Size of 6

In terms of internal architecture of the models, Skip-gram performed better than CBOW for both fastText and Word2Vec as shown in Figure 3 and 4. The Skip-gram architecture emphasises a higher weight for similar context words compared to CBOW [19]. This emphasis led to a slow model training to compute the weights for nearby context words. Hence, the Skip-gram architecture models achieved a better performance compare to CBOW architecture models. This is aligned with the strength of Skip-gram which should perform better on small datasets [19].

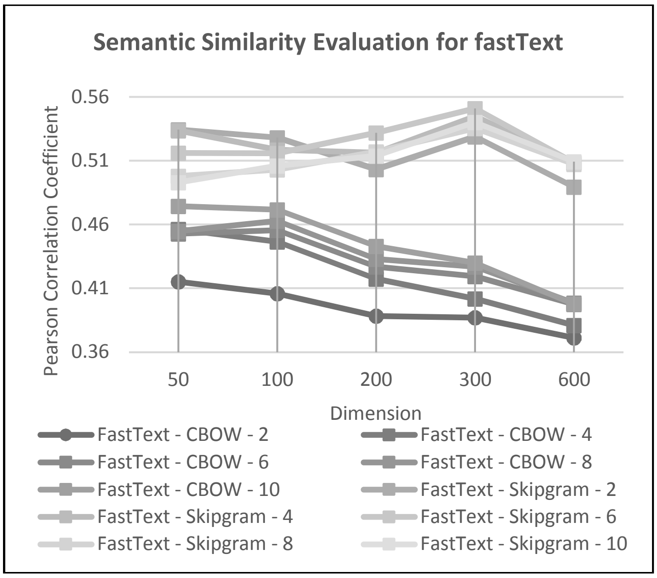

Fig. 3. Semantic Similarity Evaluation for fastText based on Dimension

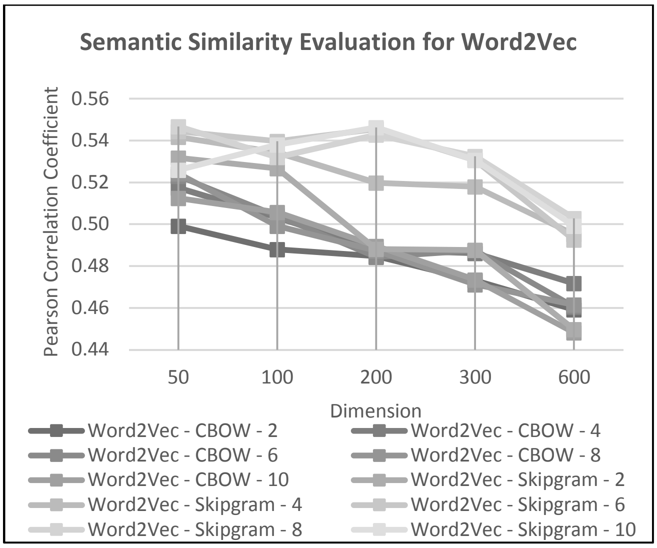

Fig. 4. Semantic Similarity Evaluation for Word2Vec based on Dimension

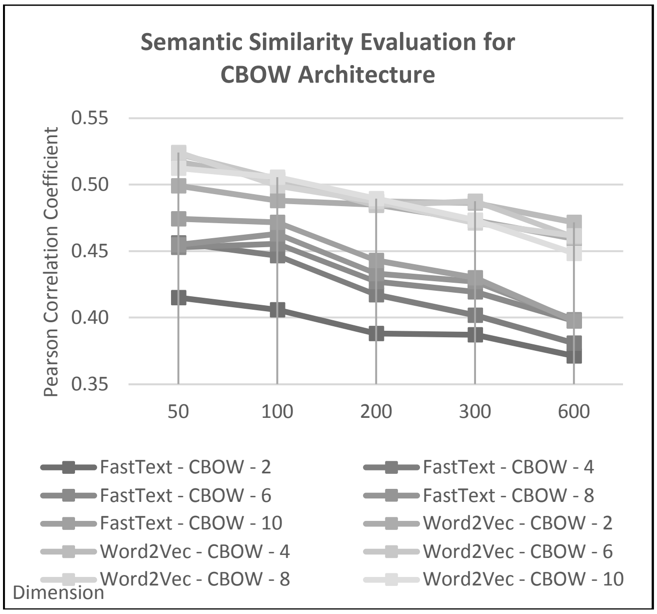

Fig. 5. Semantic Similarity Evaluation for CBOW Architecture based on dimension

At the same time, both fastText and Word2Vec models' CBOW architecture shown a declining trend in terms of the Pearson correlation coefficient as the dimension increased as shown in Figure 5. There was a negative relationship between embedding dimension and the correlation coefficient.

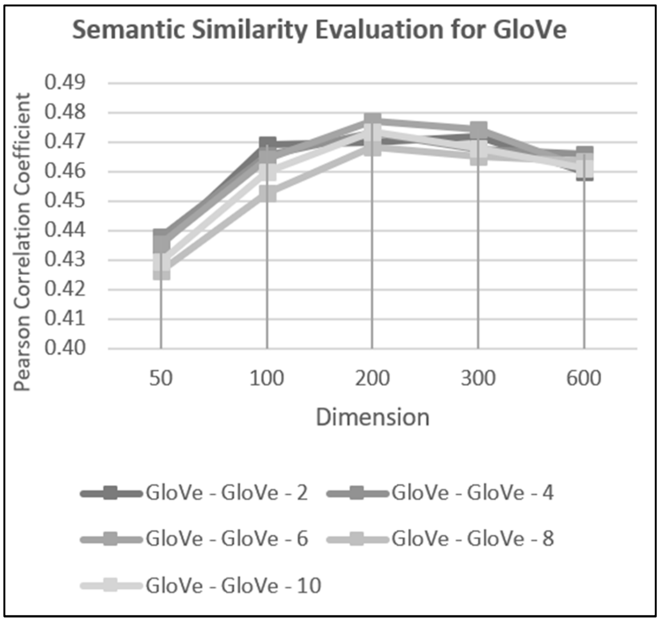

Fig. 6. Semantic Similarity Evaluation for GloVe Based on Dimension

In Figure 6, the GloVe models displayed an increasing trend of Pearson correlation coefficients beginning at 50 dimensions for all the window sizes. The trend gradually declined after 200 dimensions.

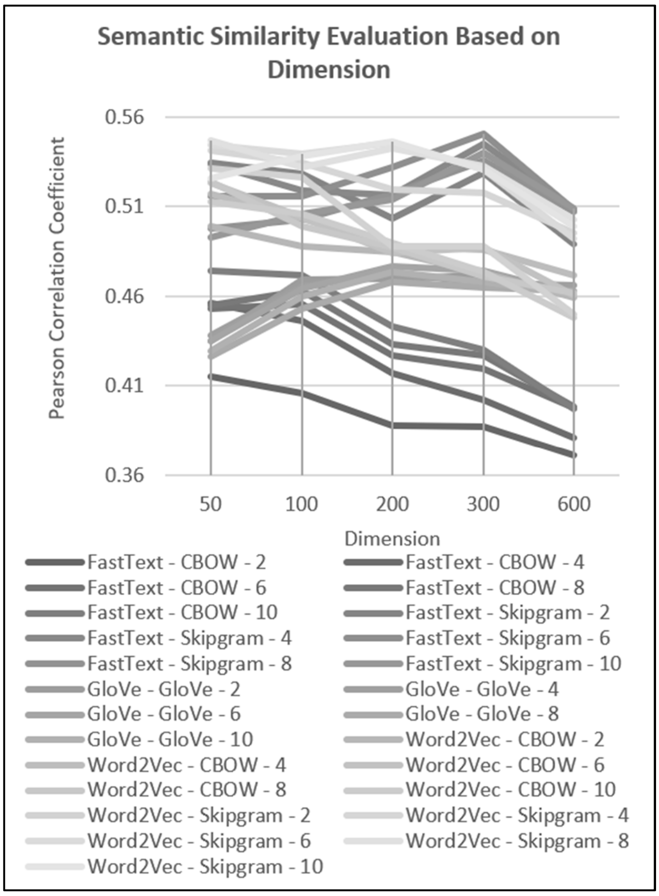

Fig. 7. Semantic Similarity Evaluation Based on Dimension

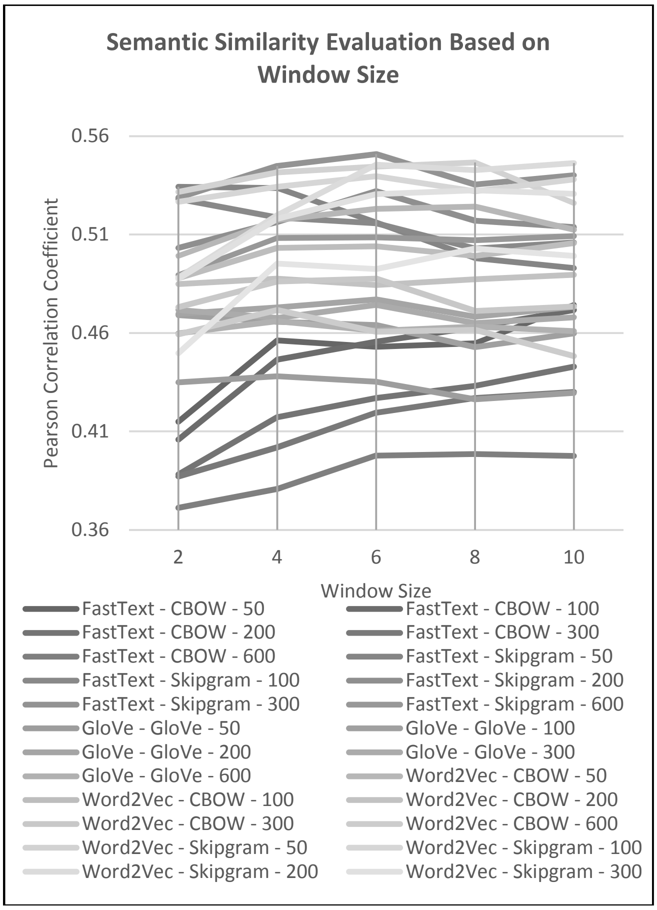

Fig. 8. Semantic Similarity Evaluation Based on Window Size

From all the results of semantic evaluation compiled (Figure 7), increasing the dimensionality of the models did not show any improvement toward the Pearson correlation efficient. To a certain extent, most of the models achieved the highest coefficient at 300 dimensions. The performance of the models declined after 300 dimensions.

In Figure 8, the graph shows the plot of the coefficient over window size. From the graph, most of the models showed an increase of the coefficient which started from window size of 2. Most of the models achieved the maximum coefficient around window size of 6 to 8. After window size 8, most of the models' coefficient decreased. Only some of the fastText CBOW models achieved the maximum coefficient at window size of 10.

B. *Word Analogies Evaluation*

The word analogies evaluation was designed in English using the question-words.txt from Google (http:// https://code.google.com/archive/p/word2vec/source/default/source) [21]. There are 14 sections in the question-words.txt as shown in Table IV.

Sections 1 to 4, 8, 9 and 11 were not taken into consideration as the corpus collected does not contain a list of these items or not applicable to the Malay language. Besides that, due to the morphological differences in both English and Malay as discussed in Section II, those sections that involved tenses and nouns were replaced with affixes in Malay. There were 9 sections with 753 items created for the evaluation as shown in Table III.

TABLE III. SECTIONS IN ORIGINAL QUESTION-WORDS.TXT

No	Section	Sample
1	capital-common-countries	Athens Greece Baghdad Iraq
2	capital-world	Abuja Nigeria Accra Ghana
3	currency	Algeria dinar Angola kwanza
4	city-in-state	Chicago Illinois Houston Texas
5	family	boy girl brother sister
6	gram1-adjective-to-adverb	amazing amazingly apparent apparently
7	gram2-opposite	acceptable unacceptable aware unaware
8	gram3-comparative	bad worse big bigger
9	gram4-superlative	bad worst big biggest
10	gram5-present-participle	code coding dance dancing
11	gram6-nationality-adjective	Albania Albanian Argentina Argentinean
12	gram7-past-tense	dancing danced decreasing decreased
13	gram8-plural	banana bananas bird birds
14	gram9-plural-verbs	decrease decreases describe describes

This dataset was used to evaluate fastText models in Malay that were trained using pre-trained word vectors for 157 languages, trained on Common Crawl and Wikipedia [26], and pre-trained word vectors for 294 languages trained on Wikipedia [22]. The word analogies evaluations achieved 22.96%.

Table IV shows the results of the trained word embedding's performance on Malay word test set with the default Gensim's pre-processing library. As expected, the percentage of correct semantic analogies was poor. The fastText CBOW model with window size of 4 and 10 achieved the best performance which was 36.80%. This performance is far better than the pre-trained fastText model trained [26].

TABLE IV. RESULT OF WORD ANALOGIES EVALUATION

Model	Architecture	Dimension	Window Size				
			2	4	6	8	10
FastText	CBOW	50	27.44	30.15	28.90	27.86	27.03
		100	33.06	34.72	32.43	34.72	34.10
		200	34.93	36.80	36.17	35.76	36.80
		300	33.06	36.17	36.38	35.97	34.72
		600	32.43	33.89	35.14	35.97	33.68
	Skipgram	50	13.72	8.94	7.90	5.82	4.78
		100	21.00	16.84	14.97	12.47	11.85
		200	28.27	22.66	17.46	15.59	15.18
		300	30.77	22.45	20.37	16.01	14.97
		600	24.74	22.66	21.41	18.09	18.71
GloVe		50	4.78	6.24	6.44	5.20	6.65
		100	8.73	8.73	8.52	8.73	8.73
		150	9.15	10.81	10.60	9.36	11.02
		200	11.43	11.23	10.19	11.02	11.23
		300	11.02	10.40	11.02	11.23	13.10
		600	9.98	10.81	10.19	11.85	11.85
		1000	9.36	11.43	9.15	9.98	11.02
Word2Vec	CBOW	50	13.98	11.37	8.29	9.72	10.43
		100	19.19	19.19	14.22	13.27	11.37
		200	22.99	19.43	18.48	17.30	17.30
		300	22.51	21.56	20.38	21.80	18.48
		600	23.70	20.85	21.09	18.96	17.77
		1000	22.51	20.38	20.62	19.91	19.67
	Skipgram	50	6.64	7.58	5.92	6.64	6.64
		100	13.27	12.80	11.37	11.14	12.56
		200	18.48	15.17	15.40	15.17	16.35
		300	18.48	17.30	19.19	15.17	15.17
		600	20.85	18.25	16.82	17.77	16.82
		1000	19.19	17.77	18.48	16.59	16.82

By design, fastText can capture morphological structures, and evaluation of word analogies are heavily evaluated on word morphology. This is due to the nature of this model which is based on the sum of their n-gram embeddings [22]. For a morphological rich language like Malay, the fastText model showed significant advantages.

The performance of the word analogies evaluation increased gradually for various models as shown in Figure 9, 10 and 11. Most of the models' accuracies decreased after 300 dimensions.

For both fastText and Word2Vec, the CBOW architectures achieved better accuracy compare to Skip-gram architectures. Most of the fastText models and GloVe models showed a decrease in accuracy after 200 dimensions. On the other hand, Word2Vec models' decreasing trend only occurred after 300 dimensions.

Based on the performance shown in Figure 9, 10 and 11, increasing the dimensions in all the models did not show any further improvement on evaluation results. This is a sign of overfitting as the dimension increase.

Figure 12 shows the word analogies evaluation for fastText models. Overall, the CBOW architectures outperformed the skip-gram architectures. The fastText skip-gram models showed a decline in accuracy as the window size increased. In contrast, the CBOW architecture models showed a slight increase in accuracy as the window size increased.

On the other hand, the evaluation for Word2Vec models generally showed the decrease of accuracy as the window size increased, as seen Figure 13. For the GloVe models the accuracy increased as the window size increased as shown in Figure 14.

Out of the three models, fastText generally achieved better accuracy compared to Word2Vec and GloVe. This seems to align with the claim by [22] where the fastText models are more capable of capturing the analogies in morphological rich languages such as Malay.

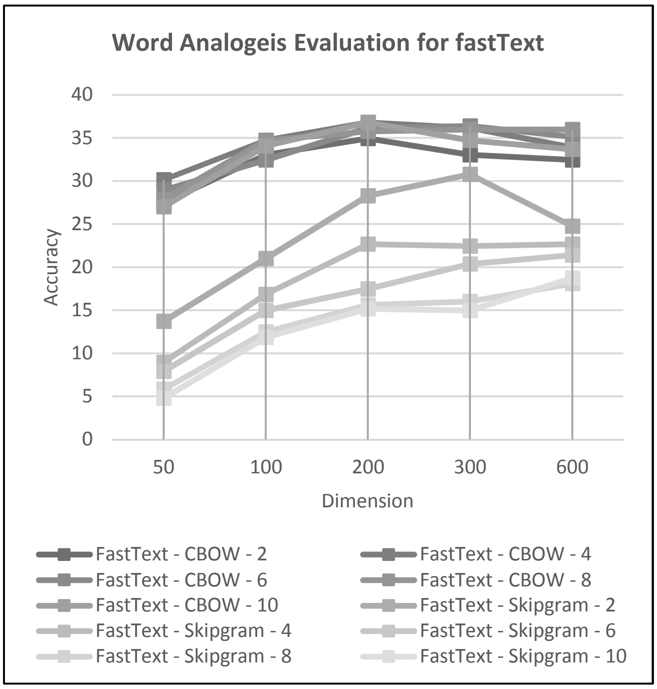

Fig. 9. Word Analogies Evaluation for fastText based on Dimension

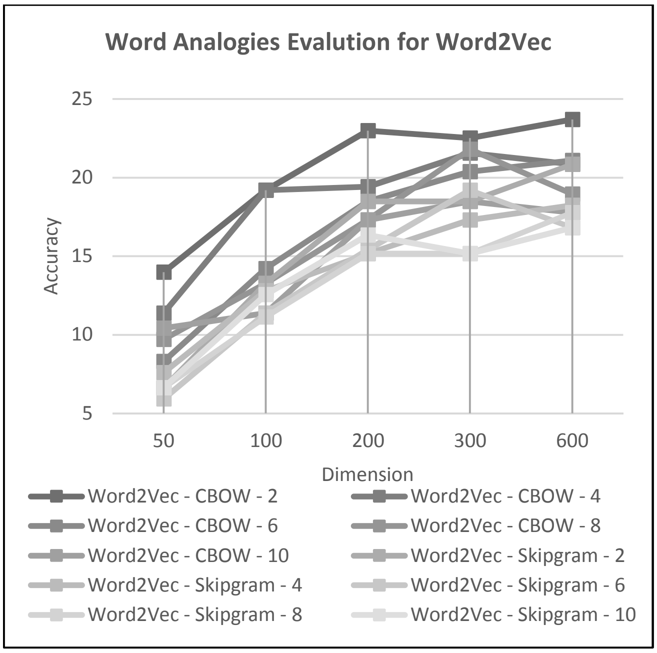

Fig. 10. Word Analogies Evaluation for Word2Vec Based on Dimension

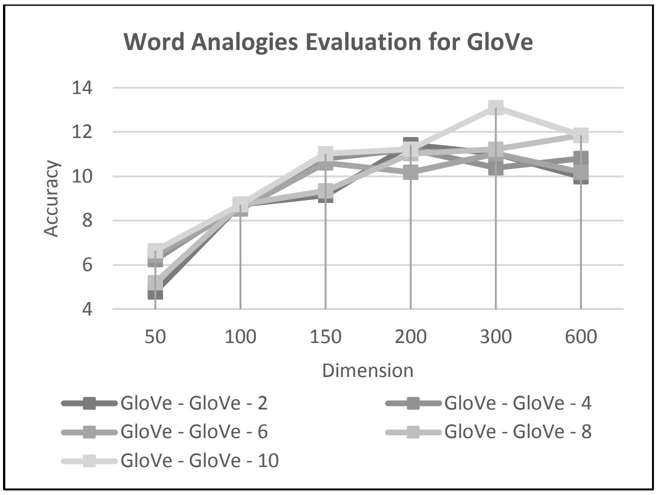

Fig. 11. Word Analogies Evaluation for Glove Based on Dimension

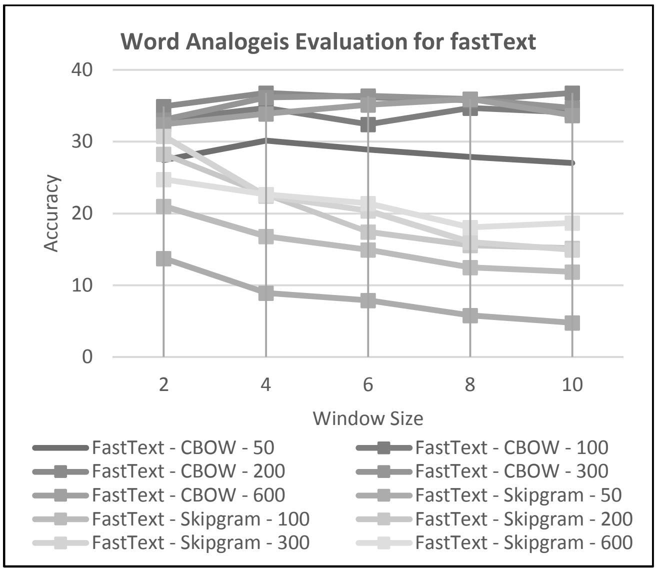

Fig. 12. Word Analogies Evaluation for fastText Based on Window Size

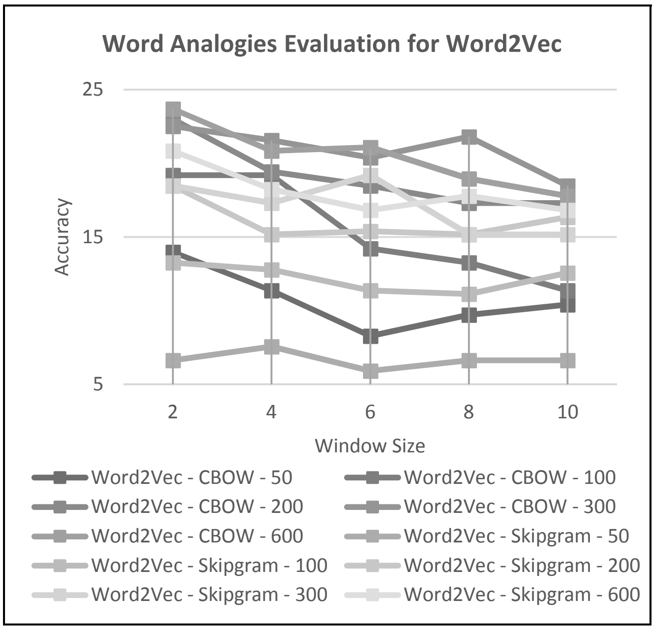

Fig. 13. Word Analogies Evaluation for Word2Vec Based on Window Size

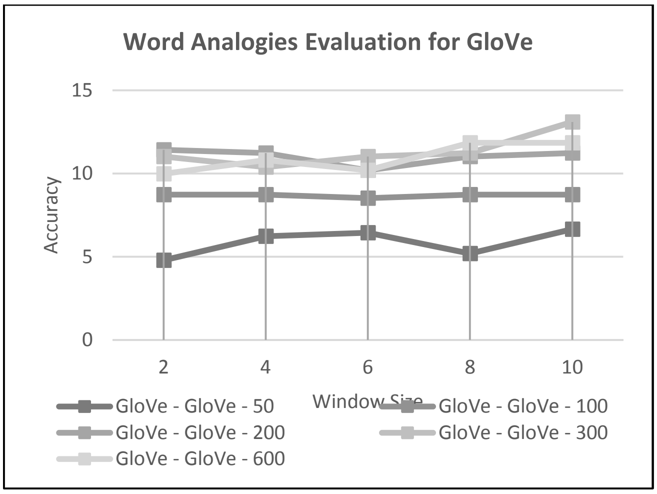

Fig. 14. Word Analogies Evaluation for GloVe Based on Window Size

VI. CONCLUSION AND FUTURE WORK

In this paper, three different techniques of Malay language word embeddings were compiled and evaluated. The results obtained from the evaluation did align with the basic of word embedding as discussed by the previous researchers. The dimensionality of the trained models impacted the performance of the evaluations [27]. Small dimension embedding may train the fastest, but it is not able to capture all the possible relationship between words. The very large dimension model tends to suffer from over-fitting problems.

Overall, fastText achieved better results compared to the Word2Vec and GloVe models. At the same time, the Skip-gram architecture also yielded a better performance compare to CBOW. This shows that the Malay language, a morphological rich language requires word embedding that supports subword features. Hence, because this experiment was carried out with a small dataset, the performance at Skip-gram showed better results.

Nevertheless, the performances of the models were way below an acceptable level. In terms of word analogies, the evaluation results showed that these methods were not suitable to evaluate word embeddings [28]. There should be further investigations required in the subword settings that could better fit the morphological structure of the Malay language.

Besides that, there is still much work to be done on both pre-processing of the model training and evaluation techniques. In terms of pre-processing, different data cleaning strategies, tokenisation and stop-word removal are required to improve the processing efficiency. In addition, the evaluation datasets require a more localised design that fits the Malay language context including word distribution and the quality of the corpus. Lastly, further evaluation work is required in extrinsic evaluations.

VII. ACKNOWLEDGEMENT

The research is partially sponsored by Universiti Teknologi PETRONAS YUTP Grant (Cost Center 015LC0-173). Special thanks to staffs who have contributed either directly or indirectly to the research.

VIII. References

[1] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, no. September, pp. 298–307, doi: 10.18653/v1/D15-1036.

[2] N. Zamin and A. Ghani, "Summarizing Malay Text Documens," in *WSEAS International Conference on COMPUTER*, 2011, vol. 12, pp. 39–46, doi: 10.5829.

[3] F. Morsidi, S. Sarkawi, S. Sulaiman, and R. A. Wahid, "Malay named entity recognition: a review," *J. ICT Educ.*, vol. 2, no. 1, pp. 1–14, 2015, doi: 10.13140/RG.2.1.1043.4960.

[4] N. Mat Awal, K. Abu Bakar, N. Z. Abdul Hamid, and N. H. Jalaludin, "Morphological Differences Between Bahasa Melayu and English: Constraints In Students' Understanding," *North Univ. Malaysia*, vol. 1, no. 1, 2007.

[5] O. Sulaiman, *Malay for Everyone*. Petaling Jaya, Malaysia: Pelanduk Publications (M) Sdn. Bhd., 2019.

[6] M. Yasukawa, H. T. Lim, and H. Yokoo, "Stemming Malay Text and Its Application in Automatic Text Categorization," *IEICE Trans. Inf. Syst.*, vol. E92-D, no. 12, pp. 2351–2359, Jun. 2009, doi: 10.1587/transinf.E92.D.2351.

[7] H. Abdullah, *The Morphology of Malay*. Kuala Lumpur: Dewan Bahasa Dan Pustaka, 1974.

[8] T. Baldwin and S. Awab, "Open Source Corpus Analysis Tools for Malay," *5th Int. Conf. Lang. Resour. Eval.*, pp. 2212–2215, 2006.

[9] M. N. Kassim, M. A. Maarof, A. Zainal, and A. A. Wahab, "Word stemming challenges in Malay texts: A literature review," in *2016 4th International Conference on Information and Communication Technology (ICoICT)*, 2016, pp. 1–6, doi: 10.1109/ICoICT.2016.7571887.

[10] M. Baker, *In Other Words*, 2nd ed. Routledge, 2011.

[11] M. N. L. Azmi *et al.*, "The Comparisons and Contrasts Between English and Malay Languages," *English Rev. J. English Educ.*, vol. 4, no. 2, p. 209, 2016, doi: 10.25134/erjee.v4i2.335.

[12] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A Neural Probabilistic Language Model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003, doi: 10.1162/153244303322533223.

[13] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive Into Deep Learning*. n.d., 2020.

[14] Y. Goldberg, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017.

[15] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, Sep. 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[16] O. M. Foong and S. P. Yong, "Swarm LSA-PSO clustering model in text summarization," *Int. J. Adv. Soft Comput. its Appl.*, vol. 8, no. 3, pp. 88–99, 2016.

[17] D. M. Blei, B. B. Edu, A. Y. Ng, A. S. Edu, M. I. Jordan, and J. B. Edu, "technique...Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003, doi: 10.1162/jmlr.2003.3.4-5.993.

[18] R. D. Paradis, J. K. Guo, J. Moulton, D. Cameron, and P. Kanerva, "Finding semantic equivalence of text using random index vectors," *Procedia Comput. Sci.*, vol. 20, pp. 454–459, 2013, doi: 10.1016/j.procs.2013.09.302.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *NIPS*, vol. 1, no. 1, pp. 3111–3119, Oct. 2013, doi: 10.1.1.741.434.

[20] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *CoRR*, vol. 1, pp. 1–12, 2013, doi: 10.1162/153244303322533223.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, Dec. 2017, doi: 10.1162/tacl_a_00051.

[23] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, vol. 67, no. 9, pp. 427–431, doi: 10.18653/v1/E17-2068.

[24] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 238–247, doi: 10.3115/v1/P14-1023.

[25] L. Finkelstein *et al.*, "Placing search in context: the concept revisited," *ACM Trans. Inf. Syst.*, vol. 20, no. 1, pp. 116–131, 2002, doi: 10.1145/503104.503110.

[26] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," pp. 3483–3487, Feb. 2018, doi: https://doi.org/10.1162.

[27] Z. Yin and Y. Shen, "On the Dimensionality of Word Embedding," in *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018, no. NeurIPS, pp. 887--898.

[28] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems With Evaluation of Word Embeddings Using Word Similarity Tasks," in *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 2016, pp. 30–35, doi: 10.18653/v1/W16-2506.