Word Embeddings

The data that I used is taken from Ortiz, B. (2020). The dataset is called Open Super-large Crawled Aggregated coRpus (OSCAR). It is a massive multilingual corpus created by using the goclassy architecture to classify and filter the Common Crawl corpus. Although we are tasked to find a pre-trained embedding, I chose to train in similar dataset in order to compare the two algorithms - fastText (Joulin et al., 2016), word2vec (Tomas et al., 2013), I trained a model from scratch with the following parameters (vector_size=300, window=5, min_count=5, workers=16, sg=0, total_examples=147844, epochs=10) with gensim library (Rehurek & Sojka, 2010).

In generating the results, I followed the tutorial given by Dive Into Deep Learning (n.d.) and used the methods in gensim. The results of word similarity in word2vec shows similarities. Along with the similarities, it also shows word antonyms for the word `malamig` (root word init), `mainit` (root word lamig). For the word `araw`, `gabi`, and `kinabukasan`, the embedding space shows somewhat in the context of time. In fastText, the word `malamig` is surrounded by words that are related to lamig and have a similar root word. The word `mainit` is also the same. The word `araw` is surrounded by words with the root word `araw` as well. The words `gabi` and `kinabukasan` are also in the context of time but with most of the words coming from the same root word. In terms of analogy, fastText shows a close similarity with the token c or the support word in the analogy e.g. `bukas` (`pabukas`, `buksan`) and `suklay` (`susuklay`, `sinusuklay`). Word2vec shows better results in this case. An example of this is `bukas` - `sarado` and `ospital` - `doctor`

The algorithms have advantages over the other but the results show that fastText have an advantage over word2vec with the words with similar meaning and root word clustered together.

For a word embedding to have a better representation, I think that the words or the dataset should be better as well. It is possible that when an algorithm represents a word, it does not follow the context a user wanted because the dataset has a different context or insufficient words. If there is no other dataset, I think that fastText can perform better as I think it is based on the segments of the word. Assuming this is true, I think that it can outperform other algorithms on rare word occurrences as it will find the related segments in a similar context.

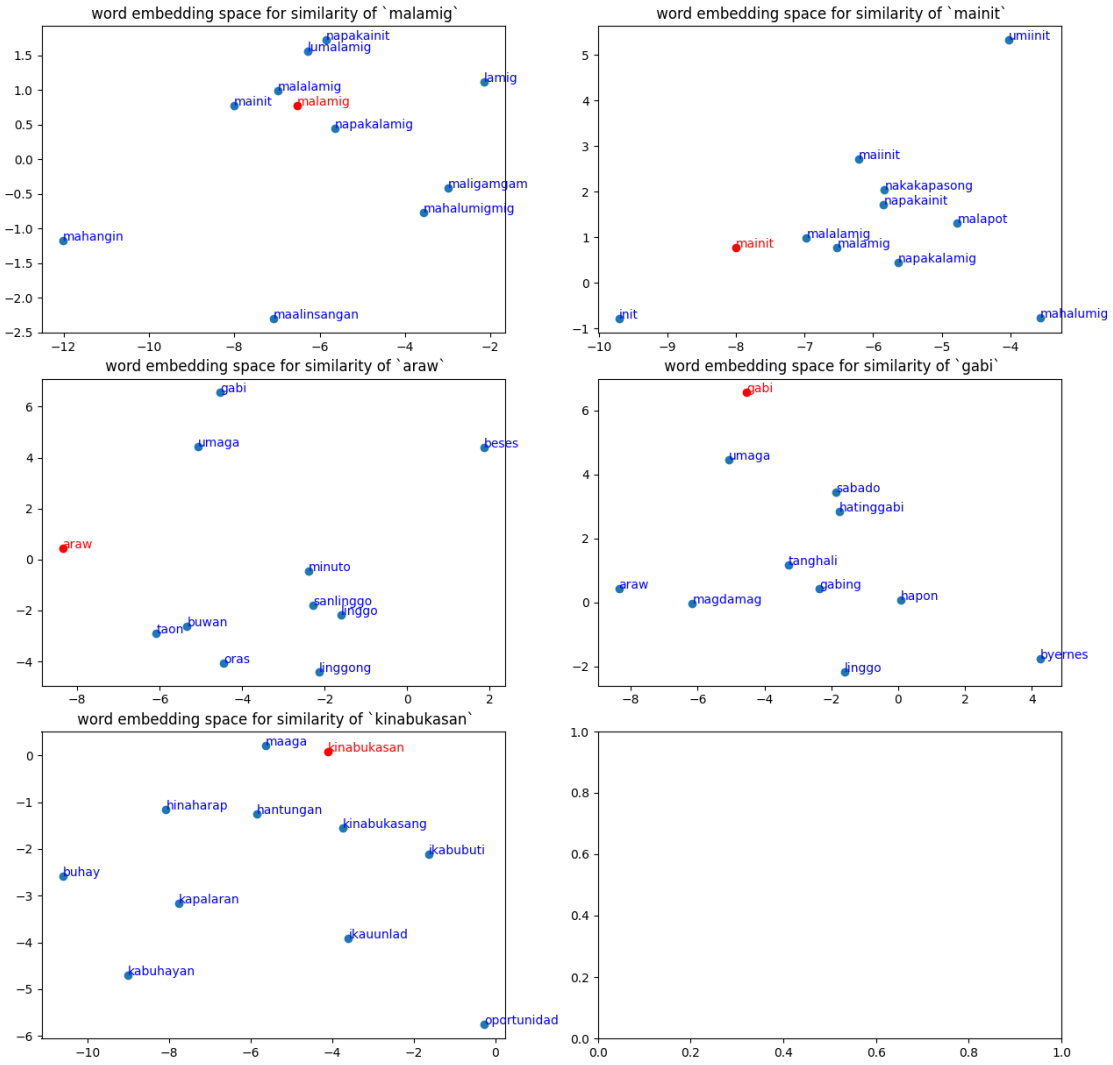
Results

Word2vec

Similarity

Words from left to right = [
'malamig', # website
'mainit', # carpenter
'araw', # browser
'gabi', # germ
'kinabukasan' # cardiologist
]

Color representation: **red** for the word, **blue** for word similarities



Analogy

Word analogy from left to right = [

('malapit', 'malayo', 'bukas', 'sarado'), # near is to far, as open is to close

('paaralan', 'guro', 'ospital', 'doktor'), # desert is to ocean, as dry is to wet

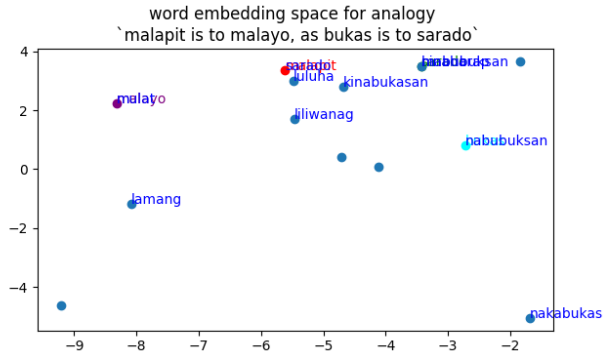
('martilyo', 'pako', 'suklay', 'buhok'), # hammer is to nail, as comb is to hair

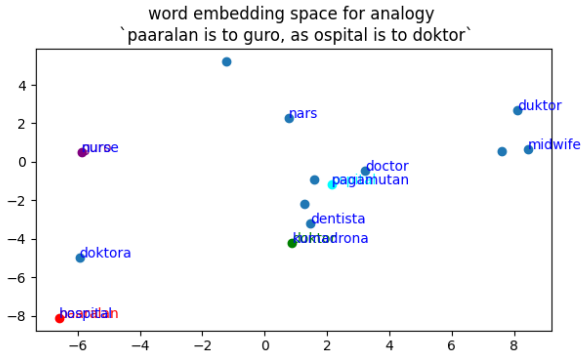
('kamay', 'sulat', 'bibig', 'basa'), # hand write mouth read

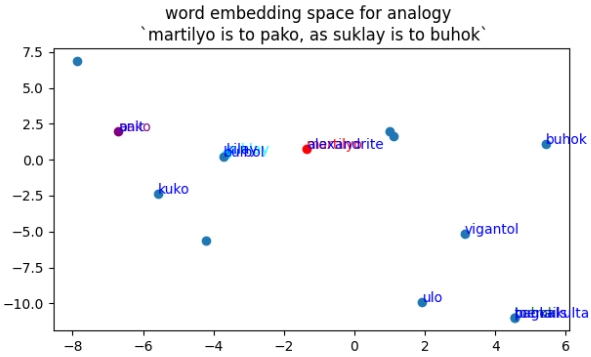
('aso', 'tuta', 'pusa', 'kuting'), # dog is to puppy, as cat is to kitten

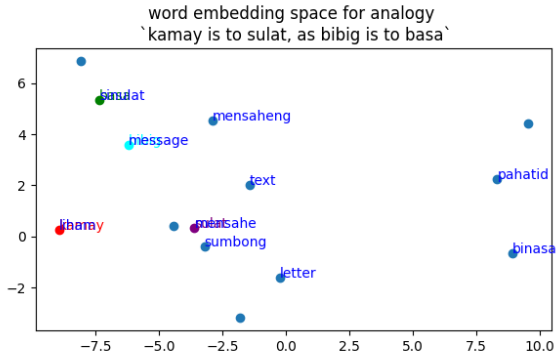
]

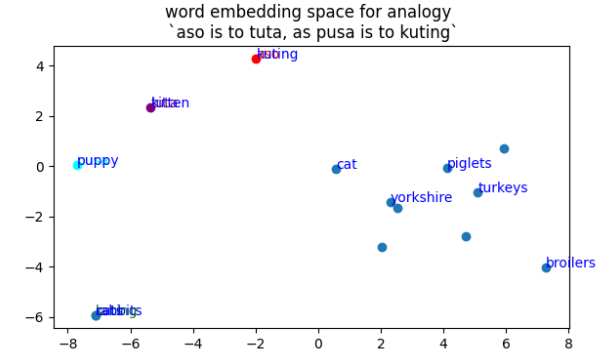
Color representation: **red** for the token_a, **purple** for the token_b, **cyan** for token_c, **green** for token_d, and **blue** for word similarities

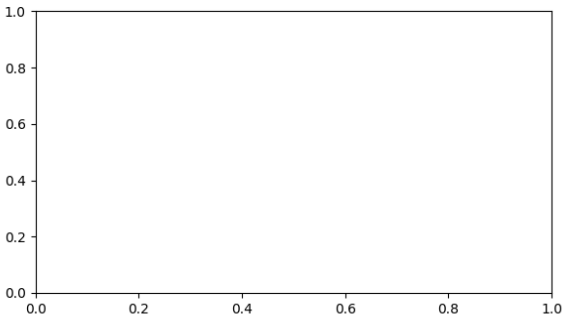










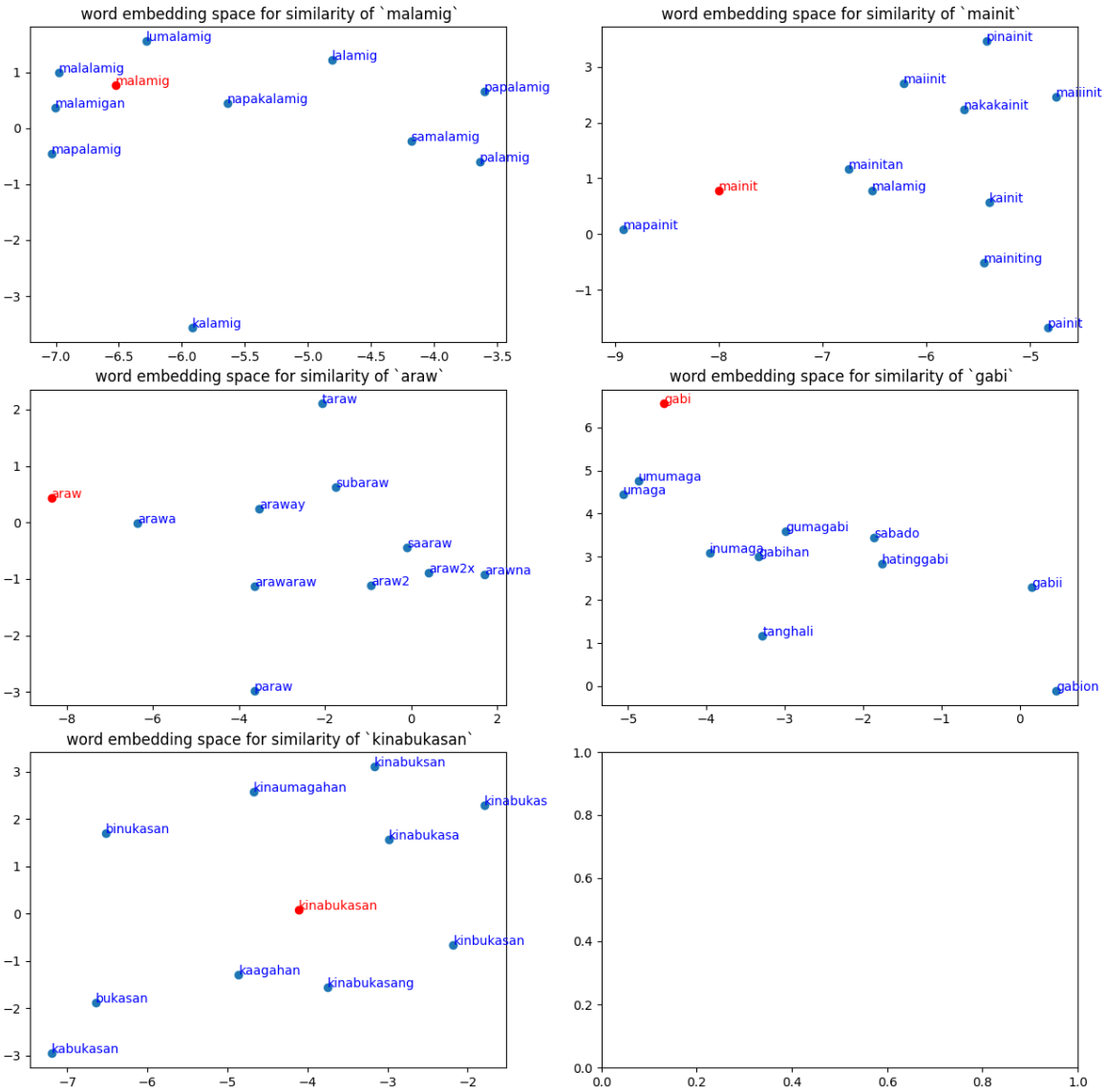


FastText

Similarity

Words from left to right = [
'malamig', # website
'mainit', # carpenter
'araw', # browser
'gabi', # germ
'kinabukasan' # cardiologist
]

Color representation: **red** for the word, **blue** for word similarities



Analogy

Word analogy from left to right = [

('malapit', 'malayo', 'bukas', 'sarado'), # near is to far, as open is to close

('paaralan', 'guro', 'ospital', 'doktor'), # desert is to ocean, as dry is to wet

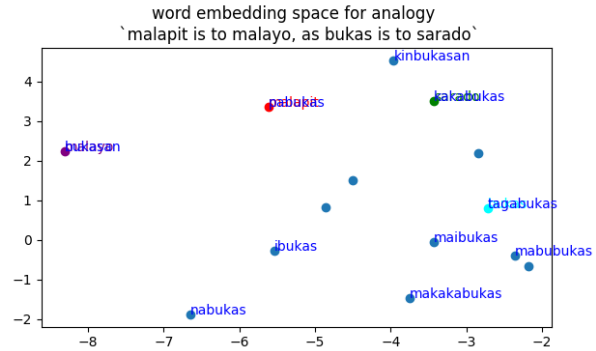
('martilyo', 'pako', 'suklay', 'buhok'), # hammer is to nail, as comb is to hair

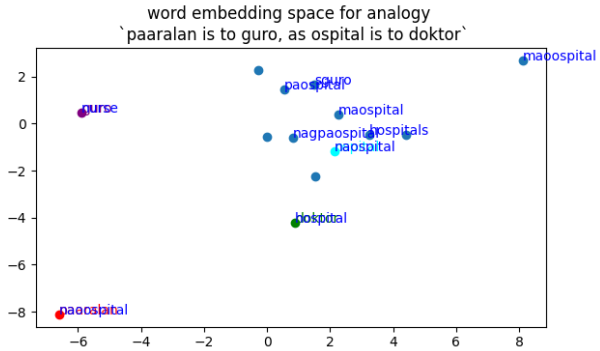
('kamay', 'sulat', 'bibig', 'basa'), # hand write mouth read

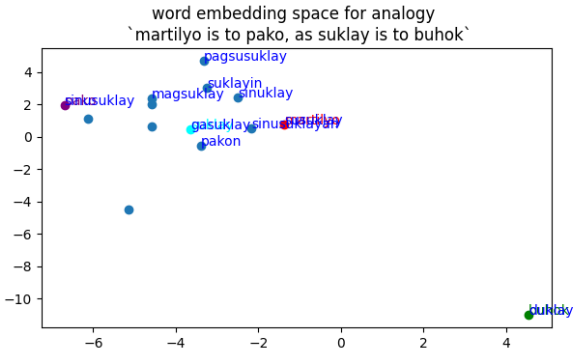
('aso', 'tuta', 'pusa', 'kuting'), # dog is to puppy, as cat is to kitten

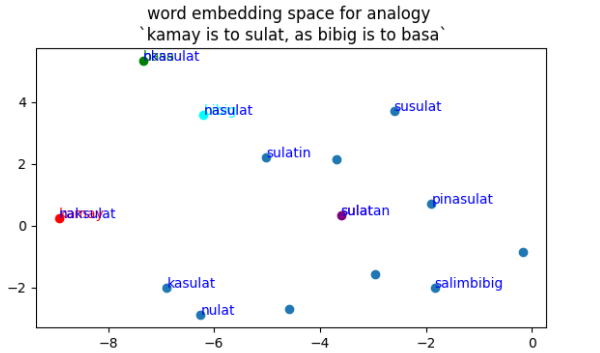
]

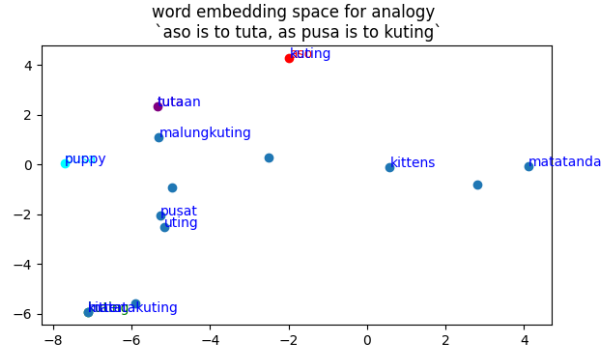
Color representation: **red** for the token_a, **purple** for the token_b, **cyan** for token_c, **green** for token_d, and **blue** for word similarities

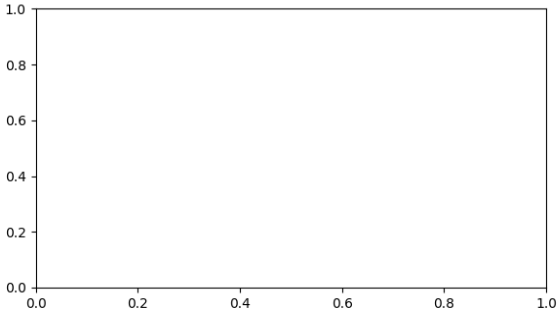












References

Dive Into Deep Learning. Word Similarity and Analogy. (n.d.). https://d2l.ai/chapter_natural-language-processing-pretraining/similarity-analogy.html.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Tomas Mikolov, Kai Chen, Greg Corrado, & Jeffrey Dean. (2013). Efficient Estimation of Word Representations in Vector Space.

Ortiz Suárez, B. (2020). A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1703–1714). Association for Computational Linguistics.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks.