Sinhala Sentence Embedding: A Two-Tiered Structure for Low-Resource Languages

Gihan Weeraprameshwara[*,1], **Vihanga Jayawickrama**[*], **Nisansa de Silva**[*], and **Yudhanjaya Wijeratne**[**]

[*]Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka
[**]LIRNEasia, Sri Lanka
[1]gihanravindu.17@cse.mrt.ac.lk

Abstract

In the process of numerically modeling natural languages, developing language embeddings is a vital step. However, it is challenging to develop functional embeddings for resource-poor languages such as Sinhala, for which sufficiently large corpora, effective language parsers, and any other required resources are difficult to find. In such conditions, the exploitation of existing models to come up with an efficacious embedding methodology to numerically represent text could be quite fruitful. This paper explores the effectivity of several one-tiered and two-tiered embedding architectures in representing Sinhala text in the sentiment analysis domain. With our findings, the two-tiered embedding architecture where the lower-tier consists of a word embedding and the upper-tier consists of a sentence embedding has been proven to perform better than one-tier word embeddings, by achieving a maximum F1 score of 88.04% in contrast to the 83.76% achieved by word embedding models. Furthermore, embeddings in the hyperbolic space are also developed and compared with Euclidean embeddings in terms of performance. A sentiment data set consisting of Facebook posts and associated reactions have been used for this research. To effectively compare the performance of different embedding systems, the same deep neural network structure has been trained on sentiment data with each of the embedding systems used to encode the text associated.

1 Introduction

An effective numerical representation of the textual content is crucial for natural language processing models, in order to understand the underlying relational patterns among words and discover patterns in natural languages. For resource-rich languages like English, numerous pre-trained models as well as the required materials to develop an embedding system are readily available. On the contrary, for resource-poor languages such as Sinhala, neither of those options could be easily found (de Silva, 2019). Even the data sets that are available for training often fail to meet adequate standards (Caswell et al., 2021). Thus, discovering a convenient methodology to develop embeddings for text would be a great step forward in the NLP domain for the Sinhala language.

Sinhala, also known as Sinhalese, is an Indo-Aryan language that is used within Sri Lanka (Kanduboda, 2011). The primary user base of this language is the Sinhalese ethnic group of the country. In total, 17 million people use Sinhala as their first language while 2 million people use it as a second language (de Silva, 2019). Furthermore, Sinhala is structurally different from English, which uses a subject-verb-object structure as opposed to the subject-object-verb structure used by Sinhala as shown in the figure 1 thus most of the pre-trained embedding models for English may not be effective with Sinhala.

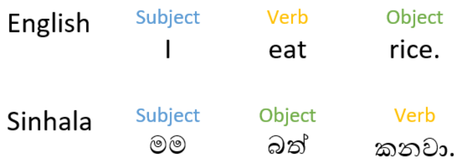

Figure 1: SVO grammar structure of English and SOV grammar structure of Sinhala

This study therefore is focused on discovering an effective embedding system for Sinhala text that provides reasonable results when used in training deep learning models. Sentiment analysis with Facebook data is utilized as the use case for the study.

Upon considering common forms of vector presentations of textual content, bag of words, word embedding, and sentence embedding are three of the leading methodologies in the present. Word embeddings have been observed to surpass the performance of bag of words for large enough data sets (Rudkowsky et al., 2018) because bag of words often met with various problems such as disregarding the grammatical structure of the text, large vocabulary dimension and sparse representation (Le and Mikolov, 2014; El-Din, 2016). In order to tackle the above challenges, word embeddings can be used. Since word embeddings capture the similarities among ingrained sentiments in words and represent them in the vector space, word embeddings tend to increase the accuracy of classification models (Goldberg, 2016).

However, one of the major weaknesses of word embedding models is that they fail to capture syntax and polysemy; i.e. the presence of multiple possible meanings for a certain word or a phrase (Mu et al., 2016). In order to overcome these obstacles and also to achieve fine granularity in the embedding, sentence embeddings are used. The idea is to test common Euclidean space word embedding techniques such as fastText (Bojanowski et al., 2017; Joulin et al., 2016), Word2vec (Mikolov et al., 2013), and GloVe (Pennington et al., 2014) with sentence embedding techniques. The pooling methods (i.e. max pooling, min pooling and avg pooling) will be considered as the baseline methods for the test. More advanced models such as sequence to sequence model (i.e. seq2seq model) (Sutskever et al., 2014) and the modified version of the sequence to sequence model introduced by the work of Cho et al. (2014) with GRU (Chung et al., 2014) and LSTM (Hochreiter and Schmidhuber, 1997) recurrent neural network units will be tested against the pooling means. Furthermore, the addition of attention mechanism (Vaswani et al., 2017) into the sequence to sequence model will also be tested.

Most models created using word and sentence embeddings are based on the Euclidean space. Though this vector space is commonly used, it poses significant limitations when representing complex structures (Nickel and Kiela, 2017). Using the hyperbolic space provides a plausible solution for such instances. The hyperbolic space is a negatively-curved, non-Euclidean space. It is advantageous for embedding trees as the circumference of a circle grows exponentially with the radius. The usage of hyperbolic embedding is still a novel research area as it was only introduced recently, through the work of Nickel and Kiela (2017); Chamberlain et al. (2017); Sala et al. (2018). The work of Lu et al. (2019, 2020) highlight the importance of using the hyperbolic space to improve the quality of embeddings in a practical context within the medical domain. However, research done on the applicability of hyperbolic embeddings in different arenas is highly limited. Thus, the full potential of the hyperbolic space is yet to be fully uncovered.

Through this paper, we are testing the effectiveness of a set of two-tiered word representation models that include various word embeddings as the lower tier and sentence embeddings as the upper tier will be compared.

2 Related Work

The sequence to sequence model introduced by the work of Sutskever et al. (2014) is vital in this research as it is one of the core models in developing sentence embedding. Though originally developed for translation purposes the model has gone under multiple modifications depending on the context such as description generation for images (Karpathy and Fei-Fei, 2015), phrase representation (Cho et al., 2014), attention models (Vaswani et al., 2017) and BERT models (Devlin et al., 2018) thus proving the potential it holds in the machine learning area.

The work of Nickel and Kiela (2017) introduces and explores the potential of hyperbolic embedding by using an n-dimension Poincaré ball. The research work compares the hyperbolic and Euclidean embeddings for a complex latent data structure and comes to the conclusion that hyperbolic embedding surpasses the Euclidean embedding in effectivity. Inspired by the above results, both Leimeister and Wilson (2018) and Dhingra et al. (2018) have extended the methodology introduced by Nickel and Kiela (2017). Leimeister and Wilson (2018) have developed a hyperbolic word embedding using the skip-ngram negative sampling architecture taken from Word2vec. In lower embedding dimensions, the developed model performs better in comparison to its Euclidean counterpart. The work of Dhingra et al. (2018) uses re-parameterization to extend the Poincaré embedding, in order to learn the embedding of arbitrarily parameterized objects. The framework thus created is used to develop word and sentence embeddings. In our research, we will be following the footsteps of the above papers.

When considering the usage of hyperbolic embeddings in a practical context, the work of Lu et al. (2019, 2020) can be examined. The research by Lu et al. (2019) improves the state-of-the-art model used to predict ICU (intensive care unit) re-admissions and surpasses the accepted benchmark used to predict in-hospital mortality using hyperbolic embedding of Electronic Health Records, while the work of Lu et al. (2020) introduces a novel network embedding method which is capable of maintaining the consistency of the node representation across two views of networks, thus emphasizing the capabilities of hyperbolic embeddings. To the best of our knowledge, hyperbolic embeddings have not been previously applied to Sinhala content. Therefore, this research may reveal novel insight regarding hyperbolic embedding and its effectivity in sentiment analysis.

In the research work of Senevirathne et al. (2020), capsule-B model (Zhao et al., 2018) is crowned as the state-of-the-art model for the Sinhala sentiment analysis. In this work, a set of deep learning models are tested for the ability to predict the sentiment of Sinhala news comments. The GRU (Chung et al., 2014) model with a CNN (Wang et al., 2016) layer which is used for the testing of each embedding in this work is taken from the aforementioned research. Furthermore, the work of Weeraprameshwara et al. (2022) has extended the idea and tested the same set of deep learning models with the addition of sentiment analysis models introduced in the work of Jayawickrama et al. (2021) using the Facebook data set which is used in this research work. According to their results, the 3 layer stacked BiLSTM model (Zhou et al., 2019) outshines as the state-of-the-art model.

3 Methodology

In order to test the feasibility of two-tiered word representation as a means of representing Sinhala text in the sentiment analysis domain, a series of experiments were conducted as described in the following subsections.

3.1 Data Set

The data set used for the project is extracted from the work of Wijeratne and de Silva (2020), which contains 1,820,930 Facebook posts from 533 Facebook pages popular in Sri Lanka over the time window of 2010 to 2020. The research work has produced two cleaned corpora and a set of stop words for the given context. The larger corpus among them consists of a total of 28 to 29 million words. The data set covers a wide range of subjects such as politics, media, and celebrities. Table 1 illustrates the fields taken from the data set for the embedding development, model training and testing phases.

Field Name	Total Count	Percentage(%)
Likes	312,282,979	93.58
Loves	10,637,722	3.19
Wow	1,633,255	0.49
Haha	5,377,815	1.61
Sad	2,611,908	0.78
Angry	1,158,182	0.35
Thankful	12,933	0.00

Table 1: The counts and percentages of the reactions in the Facebook data set

3.2 Preprocessing

Even though there are two preprocessed corpora introduced through the work of Wijeratne and de Silva (2020), the raw data set was used for this research with the objective of preprocessing it to suit our requirements. As such, numerical content, URLs, email addresses, hashtags, words in other languages except for Sinhala and English, and excessive spaces were removed from the text. While the focus of this study is colloquial Sinhala, English is included in the data set as the two languages are often codemixed in colloquial use. Codemixing of Sinhala with other languages is much less in comparison. Furthermore, stop words were removed from the text as well, as recommended by Wijeratne and de Silva (2020). Posts with no textual content after thus preprocessing as well as posts with no reaction annotations were also removed as they yield no value in the annotation stage. The final preprocessed data set consists of Sinhala, English, and Sinhala-English code mixed content, adding up to a total of 542,871 Facebook posts consisting of 8,605,849 words.

3.3 Annotation

Since the procedure followed in the model development is supervised learning, the data set needed to be annotated (Schapire and Freund, 2012). It is quite a considerable challenge to obtain sufficiently large annotated data sets for resource-poor languages like Sinhala thus Facebook data set is ideal for the given scenario as the Facebook posts are pre-annotated by Facebook users using Facebook reactions. Though this is not an expert annotation, it can be considered as an effective means of community annotation as the collective opinion of a large number of Facebook users is represented by the reaction annotation (Pool and Nissim, 2016; Freeman et al., 2020; Graziani et al., 2019; Jayawickrama et al., 2021).

A binary classification method which was introduced through the work of Senevirathne et al. (2020) and further improved for Facebook data by Weerapremeshwara et al. (2022) is used in this research as the annotation schema which is illustrated in the figure 2. Here, the Facebook reactions are divided into two classes; positive reactions and negative reactions. The reactions *love* and *wow* are considered as positive reactions while *sad* and *angry* are classified as negative reactions. The reactions *like* and *thankful* have been excluded as they are outliers in the data set with respect to the other reactions. The *like* is the de facto reaction given by the users and it does not yield a valid sentiment. The *thankful* reaction has appeared in a small time period making the presence insignificant compared to other reactions (only 0.00003% of the total reaction count). The *haha* reaction is also excluded due to the contradicting nature of its use cases (Jayawickrama et al., 2021). The *care* reaction is not included in this data set as it was first introduced to the platform in 2020 (Lyles, 2020), after the creation of the data set.

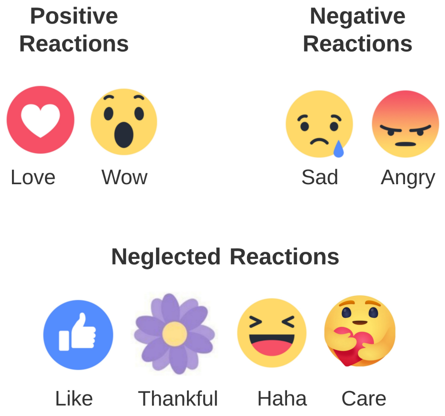

Figure 2: Reaction categorization for the annotation

3.4 Word Embeddings

The final vector representation of Facebook posts consists of two major elements: word embeddings and sentence embeddings.

Word embeddings are used both as the first tier of the two-tiered embedding systems and as the basic one-tiered embedding systems used in the form of a benchmark against which the performance of two-tiered embedding systems would be compared. The performance of both Euclidean and hyperbolic word embeddings has been thus evaluated in this research.

3.4.1 Euclidean Word Embeddings

For the purpose of representing words in the Euclidean space; fastText, Word2vec, and GloVe word embedding techniques were utilized. Word vectors consisting of 200 dimensions were created using each of the aforementioned models and a window size of 40 was picked based on the work of Senevirathne et al. (2020); Weerapremeshwara et al. (2022) which precedes this research.

3.4.2 Hyperbolic Embeddings

The hyperbolic space exhibits different mathematical properties in comparison to the Euclidean space. Due to its inherent properties, the Euclidean space struggles to model a latent hierarchy. This issue could be addressed by mapping the embedding into a higher dimension (Nickel and Kiela, 2017). However, this may lead to sparse data mapping, causing the curse of dimensionality to affect the performance. This may induce adverse effects such as causing the machine learning model to overfit by the data and using a high memory capacity for computations and storage.

The hyperbolic space has caught the attention of researchers as a plausible solution to such issues encountered in using the Euclidean space for modeling complex structures. The unique feature of this mathematical model is that the space covered by an n-ball in an n-dimensional hyperbolic space increases exponentially with the radius. In contrast to the Euclidean space where the space covered by an n-ball remains restricted by the n^{th} power of the radius, the hyperbolic space could easily handle complex models such as tree-like structures within a limited dimensionality.

The distance (D) between two vectors (i and j) in the hyperbolic space can be calculated as shown in equation 1.

$$D_{(i,j)} = \text{arcCosh}\,(1 + \frac{2||i-j||^2}{(1-||i||^2)(1-||j||^2)}) \tag{1}$$

Gaussian curvature is denoted by K. The circumference (C) of a hyperbolic circle with radius r is calculated as displayed in equation 2 while the area (A) can be calculated using equation 3.

$$C = 2\pi R \sinh{(r/R)} \tag{2}$$

$$A = 2\pi R^2 \cosh{(r/R - 1)} \tag{3}$$

Here, R is a constant of which the value is depicted by equation 4.

$$R = \frac{1}{\sqrt{-K}} \tag{4}$$

Since both the circumference and the area of a hyperbolic circle grow exponentially with the radius, the hyperbolic space has the capability to effectively store a complex latent hierarchy of data using a much lower number of dimensions than the Euclidean space would require to store the exact same structure.

In order to create hyperbolic word embeddings, the data set should be reformed in such a manner that the syntactic structure of data is highlighted. However, an adequate language parser for Sinhala does not currently exist (de Silva, 2019). Using parsers dedicated to the English language is also unfitting since the underlying grammatical structure of Sinhala is significantly different from that of English. Furthermore, for codemixed colloquial data present in this data set, grammatical structures of both Sinhala and English languages would have to be taken into consideration. Therefore, the parsing mechanism shown in figure3 is used to generate word tokens. A total of 8605849 tokens have been thus generated.

The two-dimensional illustration of the Poincaré ball after training with the Facebook data set is shown in the figure 4. Each node represents a word in the figure and each edge represents the connection between words. Here for the illustration purposes, only a thousand nodes are shown and the dimension is projected from 200 to 2.

The clustering of semantically related words in the Poincaré embedding is shown in the figure 5.

Sentence :- මම bus එකේ ගෙදර යනවා

Relations :- {මම: මම bus එකේ ගෙදර යනවා}
{bus: මම bus එකේ ගෙදර යනවා}
{එකේ: මම bus එකේ ගෙදර යනවා}
{ගෙදර: මම bus එකේ ගෙදර යනවා}
{යනවා: මම bus එකේ ගෙදර යනවා}

Figure 3: Examples of parsing mechanism used for the hyperbolic embeddings where each word is matched to the sentence

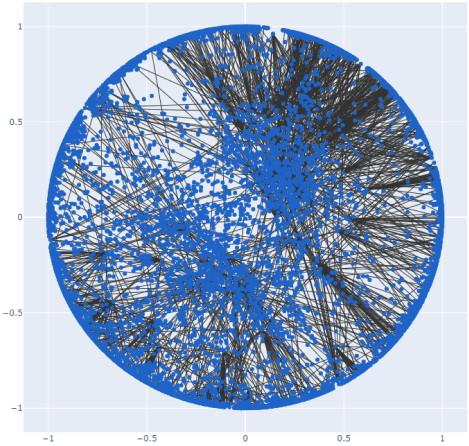

Figure 4: Poincaré word embedding done on Facebook data set

A set of words related to cricket sport is clustered in the top left corner while a set of Sinhala words related to Christianity is clustered in the bottom left. A cluster which represents news-related terms is formed in the bottom right corner. With this evidence, we can safely assume that the hyperbolic space has the capability to store a complex latent hierarchy such as the semantic relation of words.

3.5 Sentence Embeddings

Sentence embeddings are used as the second tier of the two-tiered embedding models. Basic pooling methods as well as the sequence to sequence model are used to generate sentence embeddings by using the word embedding of each word in a sentence. For both Euclidean space and hyperbolic space embeddings, the sentence embeddings are generated in a similar fashion as described in section 3.5.1.

With the parsing method mentioned in the section 3.4.2 the sentence embeddings of the Poincaré embedding is formed as illustrated in the figure 6. Both sentences and words included in them are added to the Poincare ball in the form of nodes, with nodes representing words surrounding those

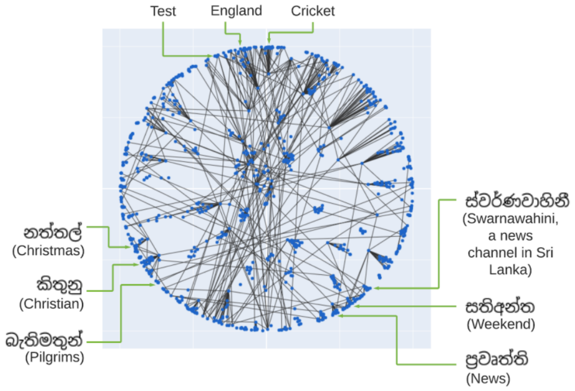

Figure 5: Word clustering in the Poincaré embedding. The meaning of the Sinhala words are given in the brackets

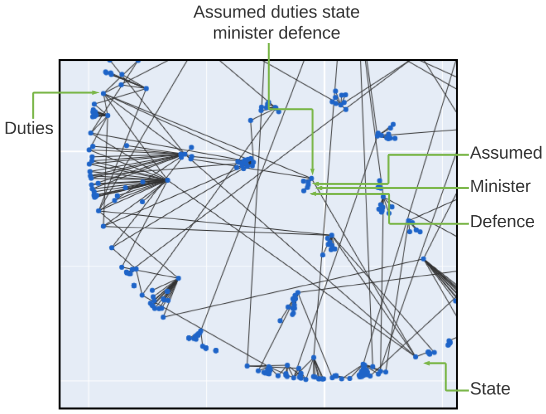

Figure 6: The sentence vector representation in the Poincaré embedding

representing the sentences they are used in. Words that are used more frequently in the data set are pushed further towards the edge of the ball while less frequent words reside closer to the centre. The vector representing a sentence stripped of stop words would be placed closer to the centre of the ball as well, since a sentence would be repeated much less frequently in the data set in comparison to a word. As figure 6 portrays, the word *defence* which is not frequently used in the data set is located further away from the edge of the ball than the word *state*, which is used more frequently.

3.5.1 Pooling

Sentence embeddings have been created with three different pooling mechanisms for each of fastText, Word2vec, GloVe, and hyperbolic word embeddings; namely, max pooling, min pooling, and avg pooling. Pooling embeddings will be considered as baseline sentence embeddings against which the performance of the sequence to sequence model is compared.

Since the hyperbolic vector space has different mathematical properties, a set of equations different from those used for the Euclidean space is required for the pooling methods.

$$ASE_{(i)} = \sum_{j=1}^{n} \sum_{k=1}^{300} WE_k \tag{5}$$

$$ASE_{(i)} = \sum_{j=1}^{n} \sum_{k=1}^{300} WE_k \tag{6}$$

$$ASE_{(i)} = \sum_{j=1}^{n} \sum_{k=1}^{300} WE_k \tag{7}$$

3.5.2 Sequence to Sequence Model

This sentence embedding mechanism follows the sequence to sequence model introduced through the work of Sutskever et al. (2014), referred to as the seq2seq model from here onwards.

The data set is randomly shuffled and a subset consisting of 400,000 data rows is used for training the encoder, decoder units.

In the original model, the encoder accepts a set of vectors which consists of the word embedding of each word in a sentence followed by the $< EOS >$ token as input and returns a context vector as the output. In order to train the model, the decoder is fed with the context vector from the encoder, with the objective of getting the $< SOS >$ token followed by the translated sentence as the final output. For our research, the output expected from the decoder is the same sentence that has been inputted into the encoder. For a given sentence, the word embedding of each word in the sentence is inputted into the Recurrent Neural Network encoder, which has a hidden layer similar in dimensions to the word embedding. Since the expected output from the RNN decoder is also the same sentence, the context vector (output of the encoder) can be considered as the sentence embedding that we are seeking.

Different sentence embeddings are thus generated using both Euclidean and hyperbolic word embeddings as inputs to the seq2seq model. For each type of word embeddings, the RNNs inside the encoder and decoder are also modified to generate different sentence embeddings. Here, GRU (Chung et al., 2014), LSTM (Hochreiter and Schmidhuber, 1997), and simple RNN models are used. The architecture of the GRU seq2seq model has been inspired by the model introduced through the work of Cho et al. (2014).

Furthermore, two different decoder structures have been used to train the seq2seq model. A simple decoder which functions as explained above and a decoder with the attention mechanism introduced in the work of Vaswani et al. (2017) are thus utilized. Both models use a teacher forcing value of 0.5 with the objective of performing better at the prediction task (Lamb et al., 2016).

The squared L2 norm between the predicted word embedding and the actual word embedding is used as the loss function for Euclidean embeddings. Equations 8 shows embedding value of the ith sentence which is calculated by summing up all the word embeddings in the predicted sequence of word embeddings. The symbol n is the length of the longest sentence which may vary for the selected data set. WE_k is the value of each dimension in the 200 dimension word embedding. Equation 9 calculates the value of ith true word embedding sequence (TV_i) which is the summation of the word embeddings of True word sequence. Then in the equation 10, the squared L2 norm (Err) is calculated. n denotes the number of data items used. The procedure follows for both Euclidean and hyperbolic space embeddings.

$$PV_i = \sum_{j=1}^{n} \sum_{k=1}^{200} WE_K \tag{8}$$

$$TV_i = \sum_{j=1}^{n} \sum_{k=1}^{200} WE_K \tag{9}$$

$$Err = (1/n) \sum_{i=1}^{n} (PV_i - TV_i)^2 \tag{10}$$

3.6 Testing

To the extent of our knowledge, there does not exist a well known or effective benchmark to test the performance of Sinhala sentence embeddings. Therefore, the GRU RNN model with a CNN layer introduced by the work of Chung et al. (2014); Senevirathne et al. (2020) is used to test each embedding. The function of this model is to understand the sentimental reactions of Facebook users to Facebook posts and thus classify each post as either positive or negative based on its prediction of the sentimental reaction of users to that post. The classification of the Facebook posts was done as explained in the section 3.3.

As mentioned above since the scarcity of a large enough data set for Sinhala language to train deep learning models, the same Facebook data set is used for the model training purpose. However, a different set of Facebook posts are used in order to avoid repetition of the data set and a total of 200,000 posts were used for the training purpose. The holdout method was used with data set splits into the 8:1:1 ratio for train, validate, and test sets. Tests were run multiple times and the average performance measures were recorded.

4 Results

The results obtained by training the models only using word embeddings are displayed in table 2. Here, the row fastText(Sinhala News Comments) taken from the work of Weeraprameshwara et al. (2022) is used as a benchmark against which the performance measures of the other word embeddings are compared. There, the Facebook data set was embedded using the fastText word embeddings trained with the Sinhala News Comments data set introduced through the work of Senevirathne et al. (2020), while the latter rows display the results of embedding the Facebook data set with word embeddings trained with the Facebook data set itself.

As the table portrays, using the Facebook data set containing 542,871 preprocessed Facebook posts, which is much larger in size than the Sinhala News Comments data set with 15,000 Sinhala News comments, to develop the word embeddings has resulted in a comparatively higher F1 score.

The results of each embedding in the two-tiered structure are shown in table 3. The first column presents the word embedding method used while the second column depicts the sentence embedding method utilized and the rest of the columns are used to present the performance measures. The best performance measures from each word embedding category are highlighted.

The best F1 score is produced by the two-tiered embedding which uses fastText as the word embedding and the seq2seq model with GRU RNNs and attention layer as the sentence embedding while the second-best F1 is scored by the fastText embedding with average pooling. For each of the sentence embedding methods, the highest F1 score is produced by pairing with fastText word embeddings. fast-

Word Embedding	Performance Measures			
	Accuracy	Precision	Recall	F1
fastText (Sinhala News comments) (Weerapremeshwara et al., 2022)	81.17	81.17	81.57	81.37
Word2vec	83.47	83.65	83.47	83.56
GloVe	82.09	81.91	82.65	82.28
fastText	**83.76**	**83.76**	**83.76**	**83.76**
Hyperbolic	82.78	82.11	83.58	82.84

Table 2: Word Embedding results

Text embeddings have resulted in a better F1 score in the one-tiered embeddings as well. Thus, we can conclude that fastText is the word embedding schema which provides the best performance in this context.

Upon taking the word embedding categories into consideration, Word2vec embeddings provide the second-best results, with performance scores slightly lower than those of fastText. The ranking of F1 scores achieved by hyperbolic and GloVe embeddings seem to be highly dependent on the type of sentence embedding used. However, the best F1 score obtained by hyperbolic embeddings, which was by pairing with the seq2seq model with GRU encoder and decoder units including an attention layer, is higher than the best F1 score GloVe embeddings have achieved upon pairing with max pooling sentence embeddings. It should be noted that the structure of data utilized here may not be optimal for hyperbolic embeddings.

In sentence embeddings, the performance of seq2seq model with GRU encoder, decoder units and an attention layer tend to surpass other sentence embedding models except when the word embedding utilized is GloVe. Nonetheless, stripping off the attention layer brings the performance of the seq2seq model with LSTM encoders and decoders to a level higher than that obtained with GRU encoder and decoder units, with the exception of the case where hyperbolic word embeddings are utilized.

Furthermore, there is a clear improvement in the performance scores of the seq2seq model when the attention layer is applied to the decoder. However, when the attention layer is not applied, pooling embeddings manage to perform better than seq2seq models except when hyperbolic word embeddings are utilized. The reason for this exception could be that the Euclidean pooling mechanisms used may not be the best fit for hyperbolic embeddings.

5 Conclusion

Comparing tables 2 and 3 makes it evident that there is a clear improvement in performance when two-tiered embedding systems are used, in contrast to simply using a single tier of word embeddings. The possibility of sentence embeddings used in two-tiered embedding systems to enable the models to consider the syntax of sentences could be the reason for this improvement. When word embeddings of Sinhala Facebook posts are directly fed to a sentiment analysis model, the model is likely to see the Facebook posts as merely an unorganized set of words instead of an organized set of sentences.

In addition, the results displayed in table 3 exhibit the use of the two-tiered embedding system that combines fastText word embeddings and seq2seq sentence embeddings with GRU encoder and decoder units as well as an attention layer has given rise to the best performance measures. Although Word2vec embeddings follow closely behind in performance, they have failed to surpass fastText, possibly due to the inability of the embedding system to consider the internal structure of words, which the fastText embedding system by nature is capable of (Bojanowski et al., 2017; Joulin et al., 2016).

Though the hyperbolic space has an advantage over the Euclidean space due to its ability to effectively represent complex hierarchical data structures (Nickel and Kiela, 2017), fastText and Word2vec have outperformed hyperbolic embed-

Embedding level		**Performance Measures**			
Word	**Sentence**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Word2vec	Max Pooling	77.23	80.06	94.59	86.72
	Min Pooling	77.29	**81.55**	92.41	86.64
	Avg Pooling	77.44	81.43	93.41	87.01
	Seq2seq GRU	75.86	76.74	97.14	85.75
	Seq2seq GRU with attention	**79.12**	79.72	96.45	**87.29**
	Seq2seq LSTM	75.97	76.17	**98.76**	86.01
	Seq2seq LSTM with attention	77.42	77.86	97.36	86.53
GloVe	Max Pooling	75.63	77.74	96.79	**86.22**
	Min Pooling	75.34	**78.68**	94.81	85.99
	Avg Pooling	**76.11**	76.90	97.38	85.93
	Seq2seq GRU	74.23	74.15	**100.00**	85.16
	Seq2seq GRU with attention	74.23	74.09	**100.00**	85.12
	Seq2seq LSTM	74.23	74.15	**100.00**	85.16
	Seq2seq LSTM with attention	74.23	74.09	**100.00**	85.12
fastText	Max Pooling	79.93	81.23	94.78	87.49
	Min Pooling	79.80	82.49	93.22	87.52
	Avg Pooling	**80.86**	**82.55**	94.07	87.93
	Seq2seq GRU	78.12	80.90	92.33	86.23
	Seq2seq GRU with attention	80.61	81.31	96.00	**88.04**
	Seq2seq LSTM	79.00	82.12	91.59	86.60
	Seq2seq LSTM with attention	80.31	80.06	**96.98**	87.72
Hyperbolic	Max Pooling	76.71	77.54	95.95	85.77
	Min Pooling	76.11	77.68	94.11	85.11
	Avg Pooling	77.00	77.31	95.56	85.47
	Seq2seq GRU	76.38	77.09	**97.57**	86.13
	Seq2seq GRU with attention	**77.31**	**78.31**	96.70	**86.54**
	Seq2seq LSTM	76.48	77.91	95.49	85.81
	Seq2seq LSTM with attention	77.19	78.22	96.24	86.30

Table 3: Performance measures of each embedding

dings in this research. The reason for this could be the lack of potent parsing tools for the Sinhala language (de Silva, 2019). To obtain the optimum performance from hyperbolic embeddings, an effective hierarchical structure such as sentence structures identified via parsing is required. The simple

$[word, sentence]$ relation structure used in this research may not be sufficient for this. Furthermore, the pooling techniques also fail to be on par with the seq2seq model, possibly due to the fact that the vectors generated by applying Euclidean pooling mechanisms on hyperbolic embeddings do not always fall within the space of the Poincaré ball.

Another noteworthy fact is that the GloVe embeddings tend to underperform in comparison to the other word embeddings models used in this research. Unlike resource-rich languages such as English, no pre-trained GloVe models exist for the Sinhala language. This could hinder the ability of GloVe embeddings to achieve their full potential.

Thus, it can be concluded that though a robust embedding model for Sinhala that is applicable across all domains may not be currently available, it could be possible to develop an effective embedding system that would at least be potent within the domain of the training data set by applying a two-tiered embedding model such as the seq2seq sentence embeddings with GRU encoders and decoders stacked on top of fastText word embeddings on a sufficiently large data set.

6 Future Work

This research is related to the work of Jayawickrama et al. (2021) and as the final goal, a Facebook reaction prediction tool for colloquial Sinhala text will be developed and the word representations developed in this project will be used for that tool.

The data set contains both Sinhala and English text since our aim is to develop a word representation for colloquial Sinhala text which consists of English and Sinhala code-mixed content. However, a pure Sinhala embedding can be generated in the future.

Furthermore, Poincaré embeddings could be developed for Sinhala text with the use of a proper parser to identify sentence structures though developing a reasonable parser for colloquial text will be a challenge.

Though this research only considers sentiment analysis for the Sinhala language, the applicability of the two-tiered embedding systems discussed in other areas of natural language processing as well as for other resource-poor languages could be tested as well.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *arXiv preprint arXiv:2103.12028*.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. 2017. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Nisansa de Silva. 2019. Survey on publicly available sinhala natural language processing tools and research. *arXiv preprint arXiv:1906.02358*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Christopher J Shallue, Mohammad Norouzi, Andrew M Dai, and George E Dahl. 2018. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*.

Doaa Mohey El-Din. 2016. Enhancement bag-of-words model for solving the challenges of sentiment analysis. *International Journal of Advanced Computer Science and Applications*, 7(1).

Cole Freeman, Hamed Alhoori, and Murtuza Shahzad. 2020. Measuring the diversity of facebook reactions to research. *Proceedings of the ACM on Human-Computer Interaction*, 4(GROUP):1–17.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Lisa Graziani, Stefano Melacci, and Marco Gori. 2019. Jointly learning to detect emotions and predict facebook reactions. In *International Conference on Artificial Neural Networks*, pages 185–197. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Vihanga Jayawickrama, Gihan Weeraprameshwara, Nisansa de Silva, and Yudhanjaya Wijeratne. 2021. Seeking sinhala sentiment: Predicting facebook reactions of sinhala posts. In *2021 21st International Conference on Advances in ICT for Emerging Regions (ICter)*, pages 177–182. IEEE.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

AB Kanduboda. 2011. The role of animacy in determining noun phrase cases in the sinhalese and japanese languages. *Science of words*, 24:5–20.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in neural information processing systems*, pages 4601–4609.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Matthias Leimeister and Benjamin J Wilson. 2018. Skip-gram word embeddings in hyperbolic space. *arXiv preprint arXiv:1809.01498*.

Qiuhao Lu, Nisansa de Silva, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Berthold Reinwald, and Yunyao Li. 2020. Exploiting node content for multiview graph convolutional network and adversarial regularization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 545–555.

Qiuhao Lu, Nisansa de Silva, Sabin Kafle, Jiazhen Cao, Dejing Dou, Thien Huu Nguyen, Prithviraj Sen, Brent Hailpern, Berthold Reinwald, and Yunyao Li. 2019. Learning electronic health records through hyperbolic embedding of medical ontologies. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 338–346.

Taylor Lyles. 2020. Facebook adds a 'care' reaction to the like button.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2016. Geometry of polysemy. *arXiv preprint arXiv:1610.07569*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30:6338–6347.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chris Pool and Malvina Nissim. 2016. Distant supervision for emotion detection using facebook reactions. *arXiv preprint arXiv:1611.02988*.

Elena Rudkowsky, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Štefan Emrich, and Michael Sedlmair. 2018. More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, 12(2-3):140–157.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR.

Robert E Schapire and Yoav Freund. 2012. Foundations of machine learning. *Mit Press*.

Lahiru Senevirathne, Piyumal Demotte, Binod Karunanayake, Udyogi Munasinghe, and Surangika Ranathunga. 2020. Sentiment analysis for sinhala language using deep learning techniques. *arXiv preprint arXiv:2011.07280*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2428–2437.

Gihan Weeraprameshwara, Vihanga Jayawickrama, Nisansa de Silva, and Yudhanjaya Wijeratne. 2022. Sentiment analysis with deep learning models: a comparative study on a decade of sinhala language facebook data. In *2022 The 3rd International Conference on Artificial Intelligence in Electronics Engineering*, pages 16–22.

Yudhanjaya Wijeratne and Nisansa de Silva. 2020. Sinhala language corpora and stopwords from a decade of sri lankan facebook. *arXiv preprint arXiv:2007.07884*.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

Junhao Zhou, Yue Lu, Hong-Ning Dai, Hao Wang, and Hong Xiao. 2019. Sentiment analysis of chinese microblog based on stacked bidirectional lstm. *IEEE Access*, 7:38856–38866.