Word2vec(Skip-Gram)- The effect of using a wider context window on the learned embeddings

Livhuwani Mutshafa, (1717376) and Gloria Pucoe, (1477437)

***Abstract*—In this paper, we explore the implementation of skip-gram for Word2vec from a Harry Potter book. The vectors were used to train skip-gram models with different context window sizes. The model was implemented using Tensorflow/Keras, and the impact of different window sizes on the quality of learned word embeddings was analysed. From the results, it was observed that a wider context window shows word embeddings with a more dispersed distribution.**

***Index Terms*—word2vec, skip-gram , word embeddings, context/window size .**

I. Introduction

WORD2VEC is a foundational technique in natural language processing (NLP) that transforms words into vector representations by capturing their contextual meanings. By embedding information about the context in which words appear, Word2Vec has been influential in the development of advanced models like GPT. This model's ability to generate meaningful representations from large text datasets has been a significant breakthrough in NLP. The Skip-gram model, a variant of Word2Vec introduced by [1], focuses on predicting the surrounding words for a given central word, emphasizing the importance of context in word meaning. Unlike methods such as Singular Value Decomposition (SVD), Word2Vec directly trains vectors to encode contextual details, facilitating more accurate word predictions.

This paper explores the impact of different window sizes on word embeddings, using visualization and evaluation to analyze their effectiveness in capturing word contexts.

II. Methodology

A. Data Collection

The *HP1.txt* file containing the text from a Harry Potter book was utilized. The initial size of the dataset was 436,711 characters before preprocessing.

B. Preprocessing

The preprocessing phase involved several key steps:

- **Lowercasing and Punctuation Removal:** All words were converted to lowercase, punctuation marks were removed, hexadecimal escape sequences were eliminated, and newline characters were replaced with spaces. The first 5,000 words were then extracted for the analysis.
- **Unique Words Extraction:** Unique words were identified to avoid redundancy and overfitting in the training data. Each unique word was mapped to a unique index and vice versa, facilitating efficient processing.

C. One-Hot Encoding and Dataset Creation

Each unique word was mapped to a one-hot vector, representing words in a high-dimensional space with most components as zero except for a single one at the index corresponding to the word.Word-context pairs were created using sliding windows of sizes 2 and 4. For each target word, its context words were treated as labels, generating input-label pairs for the training phase.

D. Neural Network Training

The neural network was built using Keras, with weights initialized using a random normal distribution. The Adam optimizer was used for training, with a learning rate of 0.01. The models were trained with different context window sizes to evaluate the impact of context width on embedding quality. A neural network with two dense layers and a linear hidden layer was constructed.

E. Inference Function

An inference function was implemented to obtain word embeddings by performing matrix multiplication between one-hot vectors and the embedding weights.

F. Visualization and Vector Evaluation

The t-SNE technique was employed to reduce the dimensionality of the embeddings for visualization. The cosine similarity metric was used to evaluate the semantic relationships captured by the word vectors.

III. Results and Discussion

The models were evaluated by tracking the loss over epochs, illustrated in Figure 1. Both models start with a high loss around 5.96 in the first epoch and gradually reduce to around 4.49 by the tenth epoch. This indicates that both models are learning and adjusting their weights effectively during training. This suggests that increasing the context window size did not have a noticeable impact on the training performance for this task. However, the results indicate that the choice of context window size may slightly influence the model's ability to learn effective embeddings in this specific scenario. Further analysis would be necessary to evaluate the impact on the quality of the learned embeddings.

Word embeddings, which provide dense representations of words in a continuous vector space, are crucial in natural language processing. This section examines the word embeddings produced by the Skip-gram model with different context window sizes. By comparing vectors for the word ”harry” and

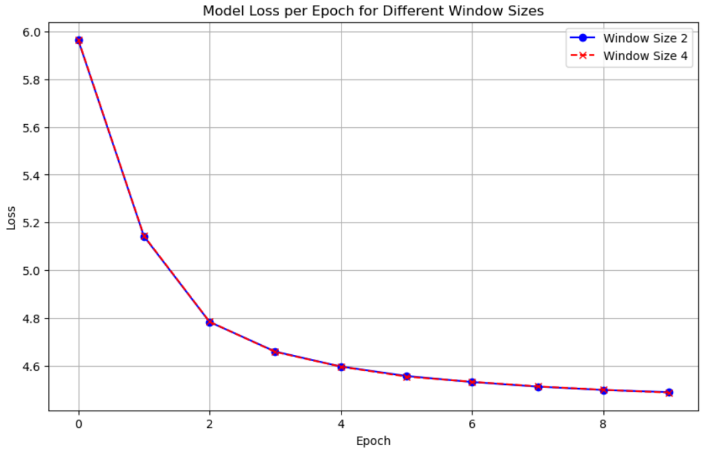

Fig. 1. Model Loss per epoch for different Window Sizes.

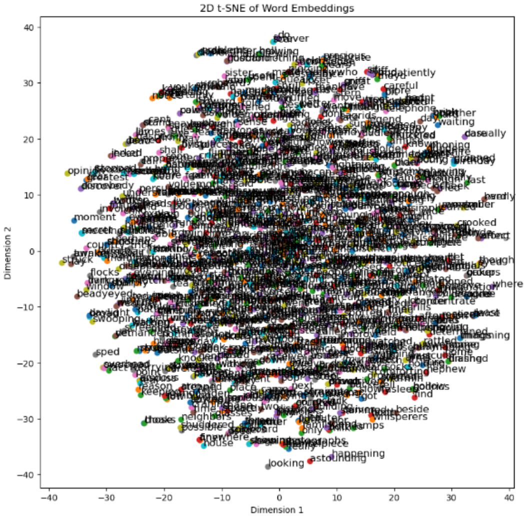

Fig. 2. Word embeddings when window size is two.

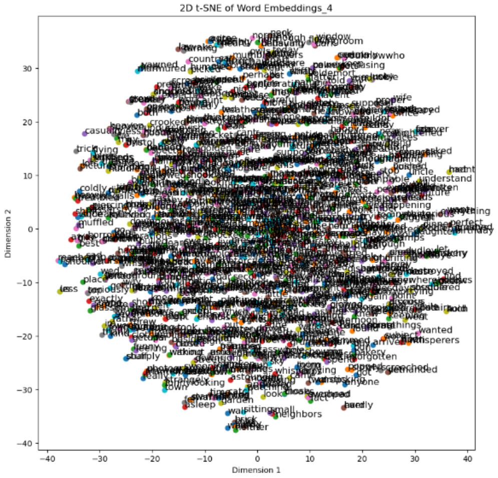

Fig. 3. Word embeddings when Window size is four.

analyzing related terms through cosine similarity, we assess the embeddings' qualities. Cosine similarity, which measures the cosine of the angle between two non-zero vectors in an inner product space, yields a value between -1 and 1.

A. Word embeddings at window size of two

In Model 1, with a window size of two, the embedding vector for the word "harry" exhibits a distribution of positive and negative values, indicating a balanced influence from nearby words. The embedding vector has relatively low magnitude values, suggesting a compact representation in the vector space. The top similar words to "harry" included "demanded," "potter," "ter," "yet," and "murmured," suggesting a focus on direct, verb-based terms and a strong semantic link to the main character, "potter". The use of t-SNE for dimensionality reduction in Figure 2 revealed a dense, overlapping distribution of embeddings without clear clusters, suggesting the model captured immediate word relationships but struggled with broader semantic themes.

B. Word embeddings at window size of four

In Model 2, with a window size of four, the "harry" embedding displays a wider range of values, including larger positive and negative magnitudes. This broader window size allows for richer and more varied contextual information. The top similar words included "stove," "hiding," "tonight," "weather," and "luck," reflecting a mix of contextual and situational terms, as well as broader thematic elements. The model begins to understand more complex and varied semantic relationships. Figure 3 showed a denser distribution of word embeddings with more clustering, suggesting the model began to capture slightly broader word relationships beyond immediate neighbors, although the embeddings were less specific than those with a context size of 2.

IV. Contribution

We split the labour equally between the two of us. Gloria was responsible for gathering the findings of the research into a document that could be used for creating reports, while Livhuwani concentrated mostly on code compilations.

V. Conclusion

This study demonstrates that the window size selection affects the Skip-gram model's ability to produce meaningful word embeddings. While bigger window widths enrich the embeddings with broader contextual information and improve the model's capacity to capture more complicated semantic links, smaller window sizes offer stability and accurate local context. Due to technical limitations, the study's usage of the Harry Potter corpus and evaluation techniques was restricted. Potential avenues for further research include investigating the effects of bigger window sizes on embedding quality and investigating other hyperparameters that could enhance the quality of semantic representations within a given corpus.

References

[1] *Efficient estimation of word representations in vector space*, Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey, 2013.