

Fixed Embeddings for Small Transformer Language Models

Jet Hughes

1 Introduction

This report investigates alternate methods of embedding input words for a transformer language model. We first describe a method for vector embedding of vocabulary using a Skip-gram model. Using this vocabulary embedding, we train a transformer model and compare it to a similar model trained on one-hot encoded vocabulary. We find that the onehot model is better at predicting text verbatim from the dataset and that the Skip-gram model is better at producing good novel text.

The datasets used for this project are two classic novels: Jane Austen’s *Pride and Prejudice* and Leo Tolstoy’s *War and Peace*. The text was cleaned by removing unwanted characters and standardising spaces between alphanumeric characters.

2 Tokenisation

War and Peace contains 34k unique words and 600k total words, while *Pride and Prejudice* contains approximately 12k unique words and 120k total words. Both texts were converted into N-gram tokens using byte pair encoding (BPE) [4]. Although most published work on language models uses a vocabulary size of about 32K, our dataset is significantly smaller, and our resources are limited. While a smaller vocabulary size is more efficient, anything less than 1000 tokens is so small that almost no tokens are complete words. This can be seen in Table 1. Since sub-word tokens contain little semantic information, it would be difficult for the Tok2vec model to create meaningful embeddings, and the language model would perform poorly. For us, a vocabulary size of 5000 or higher requires a prohibitive amount of memory and computation. Figure 1 shows the number of occurrences of the most common 1000 words in *War and Peace*. For this project, we think a vocabulary size that captures at least the most

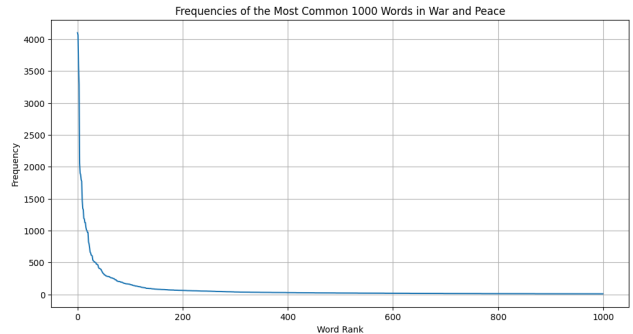


Figure 1: Frequencies of the Most Common 1000 Words in *War and Peace*

common 200-400 words is sufficient. Table 1 shows that a vocabulary size of 2000 tokens captures 711 complete words. This is a good compromise between efficiency and performance and sufficient for the transformer model to learn meaningful embeddings.

Vocab Size	No. Words	No. Tokens in Text
500	51	1416586
1000	261	1295098
2000	711	1151325
5000	2011	887094

Table 1: Vocabulary size vs. number of English words and total tokens in text

3 Vector Embedding

Good information about the semantics and syntax of words is crucial for a language model to perform well. Mikolov et al. [2] introduced two new methods of word representation. They represent words as dense vectors in a continuous vector space, where closely related words have vector embeddings that are closer together. Both methods use a simple neural network with one hidden

layer to learn these vector representations. Given a text encoded as a sequence of one-hot encoded words, the first method, Continuous Bag of Words (CBOW), is trained to predict a target word given a number of context words before and after it. The second method, Skip-gram, is trained to predict context words given a target word. The rows of the weight matrix between the input and hidden layer become the word embeddings.

The results from [2] show that the CBOW model is faster to train and has better accuracy for frequent words, while the Skip-gram model is better for small datasets and has better accuracy for infrequent words. Since each training example for CBOW is the entire context window, it needs a larger dataset for good generalisation. Conversely, the Skip-gram model predicts many context words from a single target word, so there are more training examples, and it can learn better from small datasets. Since our dataset is relatively small, we use the Skip-gram model.

In [2], the accuracy of CBOW on the semantic-syntactic dataset with vector dimensions of 50, 100, 300, and 600 was tested. The results showed that increasing the dimension from 50 to 100 gave a large improvement, but between 100 to 300, there was only moderate improvement, and from 300 to 600, there was little improvement except for in very large datasets. Since we have a very small dataset and limited resources, we used a vector dimension of 150. In [2], the parameter C is the maximum distance of the context words from the target word. Mikolov et al. chose $C = 10$. Since words further from the context word tend to be less relevant, they reduced the contribution of words further from the target word by sampling less from those words. To simplify our system and reduce computational complexity, we chose a smaller $C = 4$ and sampled from all words equally¹.

This Skip-gram model was trained on the entirety of War and Peace encoded as tokens using a vocabulary size of 2000. As in [2], we use an initial learning rate of 0.025 and decrease it linearly to reach zero at the end of the final epoch. We use the Adam optimiser [1] and a batch size of

¹In hindsight, the window size of the vector embedding had a minimal effect on resource use. In future, it may be better to use $C = 10$ and sample more distant words less.

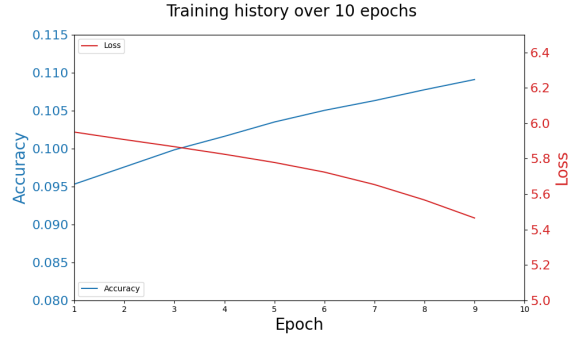


Figure 2: Training history over 10 epochs

32. The model was trained for 10 epochs. Figure 2 shows the accuracy and loss while training. As shown in 2, increasing the number of epochs past 20 did not improve the accuracy of the model².

Table 2: Accuracy for different numbers of epochs

No. Epochs	Accuracy
1	0.09939
3	0.10748
5	0.10840
10	0.10969
20	0.10968

3.1 Results and Discussion

A t-SNE plot of 100 tokens from the embedding is shown in Figure ???. The embedding appears to have some structure, with similar tokens being grouped together. For example, the names "Catherine", "Bennet", "Lydia", etc. are grouped together. The name "Elizabeth" is separated from the others. There are groups of similar words such as "himself" and "herself", "know" and "thought", and "brother" and "sister". The embedding clearly encodes some amount of meaning within the embeddings. This is sufficient for our purposes.

²This does not necessarily indicate that training past 10 epochs does not improve the quality of the embeddings. We did not have time to analyse the quality of embeddings over an exhaustive search of training parameters, so we used accuracy as an estimate

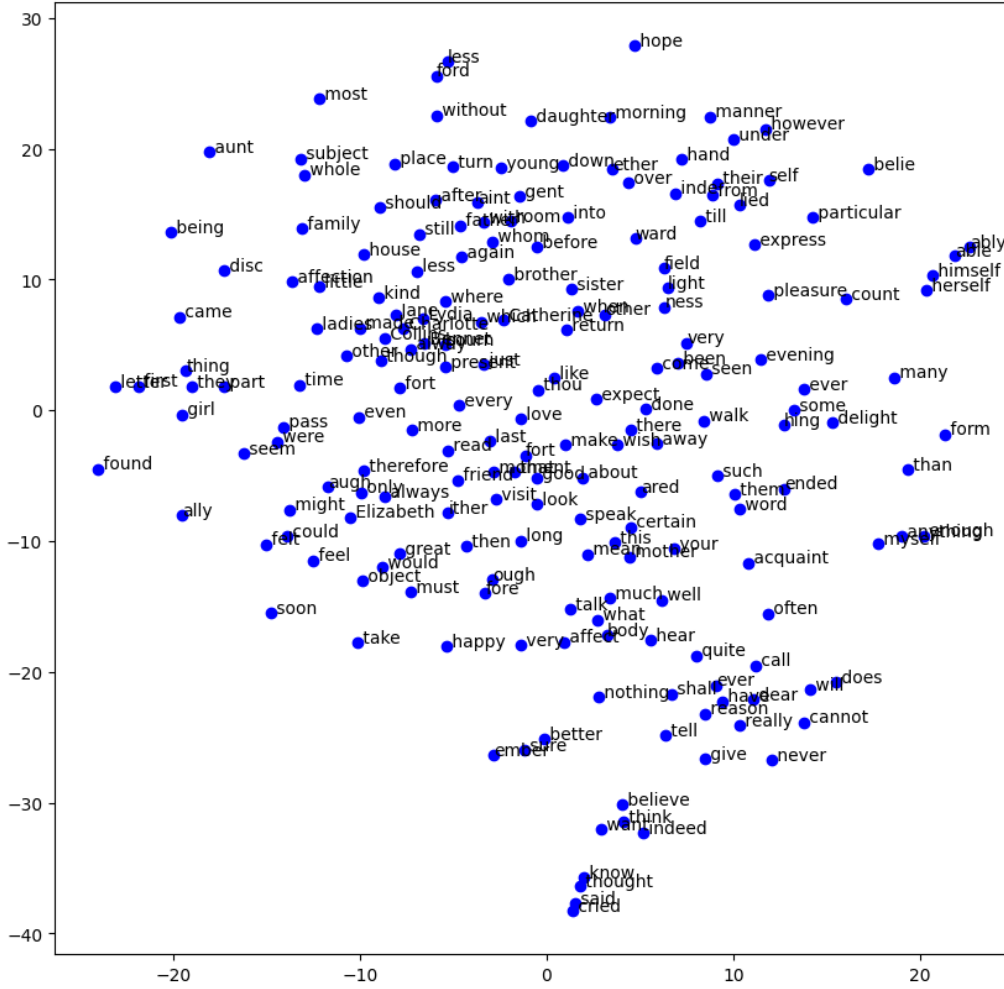


Figure 3: A t-SNE plot of 200 words from the Skip-gram vector embedding trained on War and Peace

4 Transformer Architecture

Vaswani et al. [5] introduced self-attention, where each token in a sequence is modified to reflect the context and the relationships between the words surrounding it. They used attention for their transformer model. This is a sequence transduction model, which takes a sequence as input and predicts another sequence. We are interested in a language model, which takes a sequence of tokens as input and predicts the next token in the sequence.

Our model is based on the model used in GPT-1 [3]. This model is a generative decoder-only transformer with no cross-attention. Our model differs from GPT-1 in that the vector embeddings are fixed and not learned during training.

The model comprises a fixed embedding layer which converts tokens into dense vectors, followed by a positional encoding layer, a stack of n trans-

former layers, and finally, a linear dense layer and a softmax output layer.

Each transformer layer has masked multi-head self-attention, followed by a dense feed-forward network. Layer normalisation is applied before and after the feed-forward network.

4.1 Model specifications

Due to resource limitations and the size of our dataset, our transformer model is small. We use half the number of transformer layers used in [3] and two-thirds the number of attention heads, and the dimension of the dense feed-forward layers is reduced. The model has six transformer layers. Each transformer layer has 8 attention heads, with a key dimension of 32, and a dense feed-forward layer with 256 neurons. The vocabulary size is 2000, and the vector embedding dimension is 150,

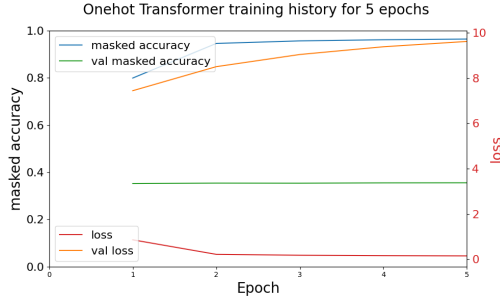


Figure 4: Training and validation loss and accuracy during training of the onehot transformer model

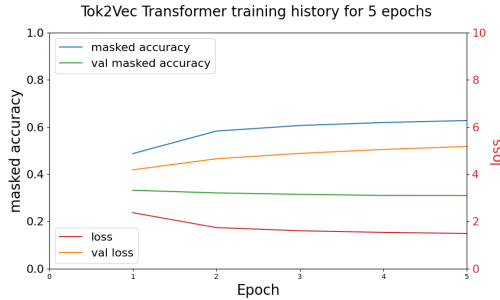


Figure 5: Training and validation loss and accuracy during training of the tok2vec transformer model

as defined earlier.

The average sentence length of War and Peace is 20 words³. We think a sequence length of 128 is sufficient to learn inter-sentence context while remaining within our resource budget.

4.2 Training

The model was trained using a custom Adam optimiser to match the original transformer paper [5]. The loss function is a masked version of sparse categorical cross-entropy, which ignores padding tokens. The model was trained on a single NVIDIA RTX 4080 GPU. We trained for 10 epochs with a batch size of 32 on the 598,184 words in War and Peace.

We also trained a similar model where the embeddings are replaced with simple one-hot encoded BPE tokens. The length of the onehot encoded

³ 600,000 words / 30,000 sentences \approx 20 words per sentence

tokens is much larger than the Skip-gram, so the transformer model is much larger. This makes it more expensive to train and improves its ability to memorise text. To counteract this issue, the vocabulary size for the simple model was reduced to 1000. The model trained with vector embeddings we will refer to as the tok2vec model, and the model with simple one-hot tokens we will refer to as the onehot model.

Graphs of training and validation accuracy and loss are shown in Figures 4 and 5. The accuracy of both models does not increase significantly after the second epoch. The validation accuracy for both models is essentially constant. The validation loss for the onehot model is significantly higher than for the tok2vec model. This is likely due to differences in the model and the embeddings, but it may suggest the onehot model generalises worse.

4.3 Results and Discussion

We expected the tok2vec model to perform much better at generating novel text because the inputs encode semantic and syntactic information about the tokens. Since the onehot model inputs have none of this information, we expected it to rely on learning the text verbatim and perform poorly at generating novel text. Given a sentence taken from the book, we expected that both models could predict the following words exactly, but the onehot model might perform better here. Conversely, given a prompt not taken from the text, we expect the onehot model to tend to generate text from the book and the tok2vec model to generate novel text.

Examples of generated text are included in Appendix A. The models were prompted by the first phrases of War and Peace and of Pride and Prejudice. When given an unfamiliar prompt, the tok2vec model produces novel text that is mostly syntactically correct but not wholly meaningful. When given a familiar prompt, the tok2vec model behaves similarly. The tok2vec model is able to link small sequences of about 3-7 words coherently. However, it is unable to maintain ideas, between sentences. This is likely due to the small context size used for training the Skip-gram vector embedding. A context window with only 4 words on either side of the target word cannot encode

meaning from a wider context and may limit the transformer’s ability to maintain ideas longer than a few words.

The onehot model showed good evidence of remembering text from the training data. When given a familiar prompt, the model didn’t continue the prompt perfectly but did produce large sequences of text from the book. When given an unfamiliar prompt the onehot model still produced large sequences of text verbatim from the book. Text produced by the model that isn’t verbatim from the book is less coherent than the text produced by the tok2vec model.

References

- [1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [4] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Use of AI tools

ChatGPT was used to assist with some simple code. Links to relevant conversations are provided where it was used. ChatGPT was also used to assist with \LaTeX formatting. Grammarly was used for grammar suggestions. Github Copilot (<https://github.com/features/copilot>) was used for suggestions while coding. Blocks of code generated entirely by Copilot are marked by comments.

Appendix A Examples of generated text

Embedding	Result
tok2vec	<p>Prompt: It is a truth universally acknowledged</p> <p>It is a truth universally acknowledged that it is only necessary to retain the whole position. ” “ And there is, your Highness, has not a heart for you been famous, ” said the prince. “ I have not considered it, ” said the prince, with a smile. “ You don’ t know what a terrible good thing is may be unable to forget. ” “ And I am not afraid, ” said Prince Andrew, “ but I am very glad to see you. ” “ Well, thank God, go and throwing over the wound to capture a great sorrowful in this exception of the whole time. ” “ Yes, I will go to the battlefield to the bridge, ” said the tall</p>
tok2vec	<p>Prompt: “Well, Prince, so Genoa and Lucca are</p> <p>“Well, Prince, so Genoa and Lucca are welcomed. ” “ Won’ t you notice that from the arbitrary? ” inquired Dolgor. “ It’ s all the same. ” “ But what is she so charming that? ” asked Prince Andrew. “ Oh, yes, yes, bringing her! ” Pierre continued. “ I am so sorry for her mother. ” “ So little pray I. ” “ And so. . . ” said Pierre, and he went out of the room. Prince Andrew was sitting on a table with a cheerful expression on his face. “ No, I can’ t understand. I can’ t bear him! ” “ Well, thank God,</p>
onehot	<p>Prompt: It is a truth universally acknowledged</p> <p>It is a truth universally acknowledged him and becoming maneuvers, and that as yet untrained, the peasants had come in a large crowd to the barn and sent word that they would not let her leave the village: that there was an order not to move, and that they would unharness the horses. Alpain, was sent out greeting the largeage at the troops, who had paused. The men in one of the crowded closer together. The attack directed by showing the men. The army looked in its retreat from Vereshch, describing the army’ s army incorrect boards of</p>
onehot	<p>Prompt: “Well, Prince, so Genoa and Lucca are</p> <p>“Well, Prince, so Genoa and Lucca are now just family estates of the Buonapartes. But I warn you, if you don’t tell me that among yourself, but it’ s enjoying myself! ” Natalie smiled just as he gazed silently and became still more. Anatole glanced round at him, and screening the candle with its red halo was still burning, and the same shirtlike sphinx lay near the door; but besides all this something creaked, there was a whiff of fresh air, and a new white sphinx appeared, standing at the door. And that sphinx had the pale face and shining eyes of the very Nataly, she was playing dead, but felt sorry for</p>