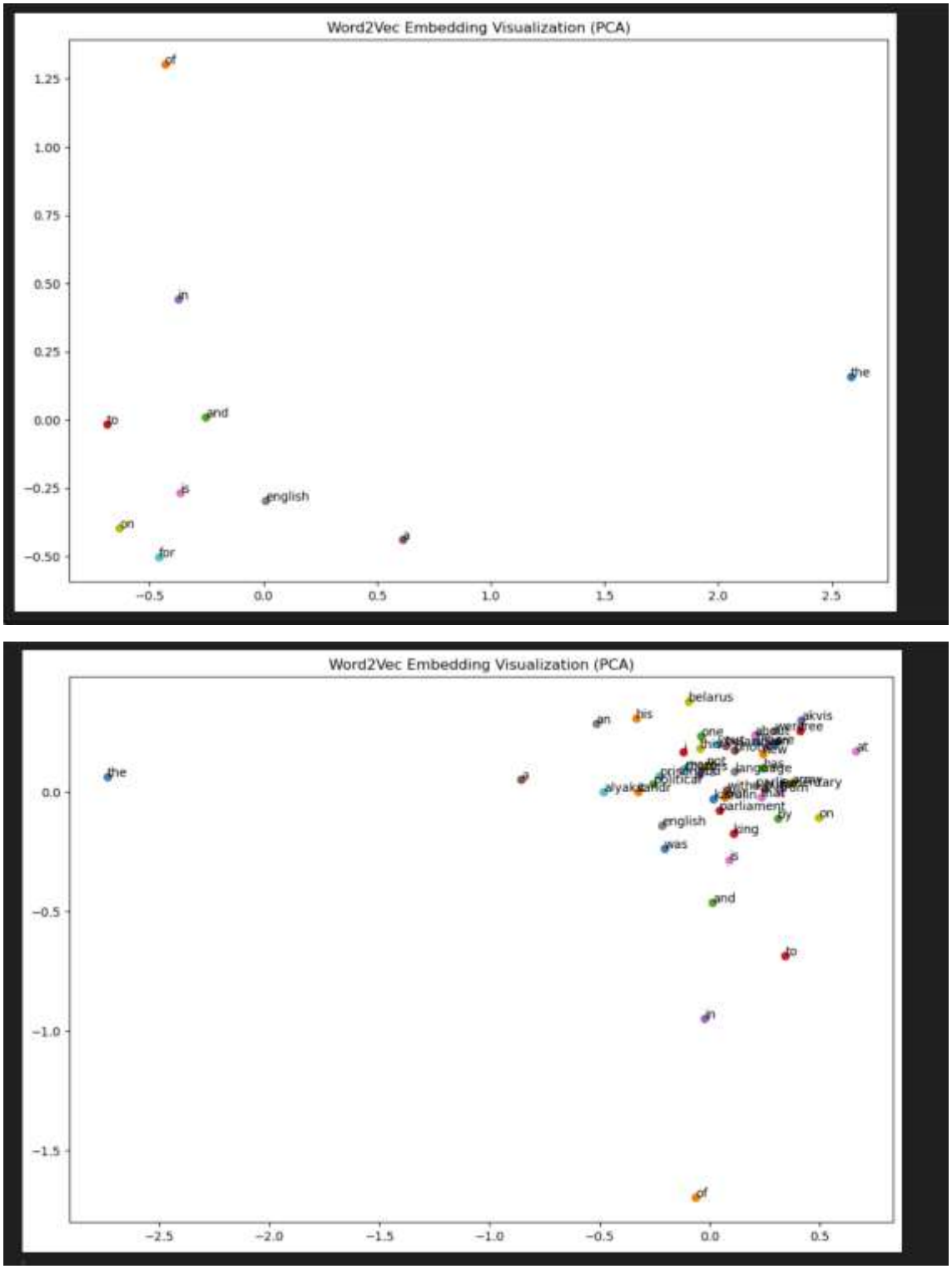
Task 1

```
Epoch 1, Loss: 149403.1121
Epoch 1, Avg Loss: 7.4724
Epoch 2, Loss: 148552.9165
Epoch 2, Avg Loss: 7.4299
Epoch 3, Loss: 144714.7644
Epoch 3, Avg Loss: 7.2379
Epoch 4, Loss: 141370.2986
Epoch 4, Avg Loss: 7.0706
Epoch 5, Loss: 137929.3117
Epoch 5, Avg Loss: 6.8985
Epoch 6, Loss: 134663.9288
Epoch 6, Avg Loss: 6.7352
Epoch 7, Loss: 131940.5729
Epoch 7, Avg Loss: 6.5990
Epoch 8, Loss: 129811.1203
Epoch 8, Avg Loss: 6.4925
Epoch 9, Loss: 128142.6497
Epoch 9, Avg Loss: 6.4091
Epoch 10, Loss: 126837.5695
Epoch 10, Avg Loss: 6.3438
```

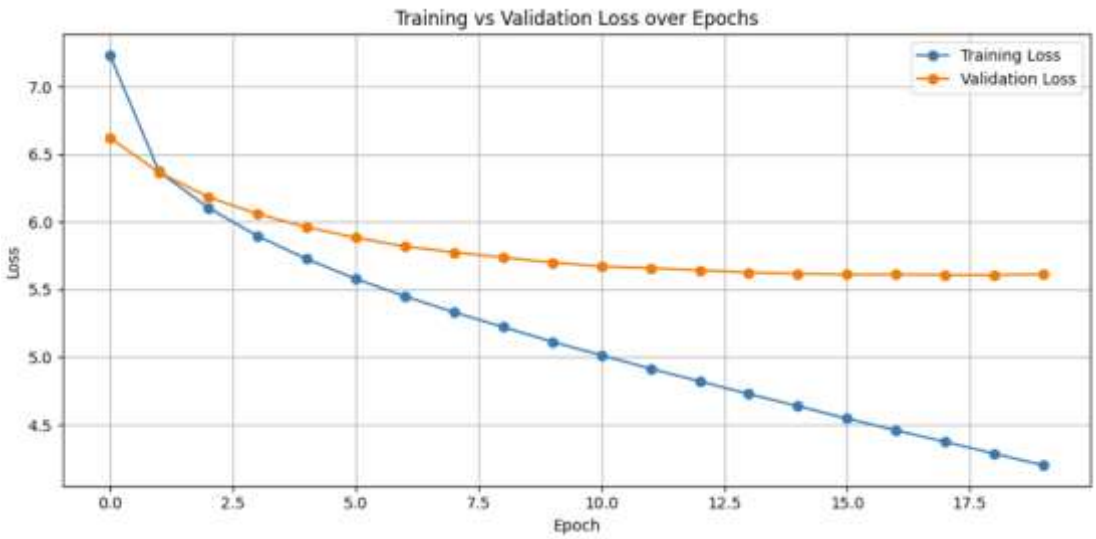
Loss is decreasing -> demonstrates that the model is improving its predictions of context words over time.

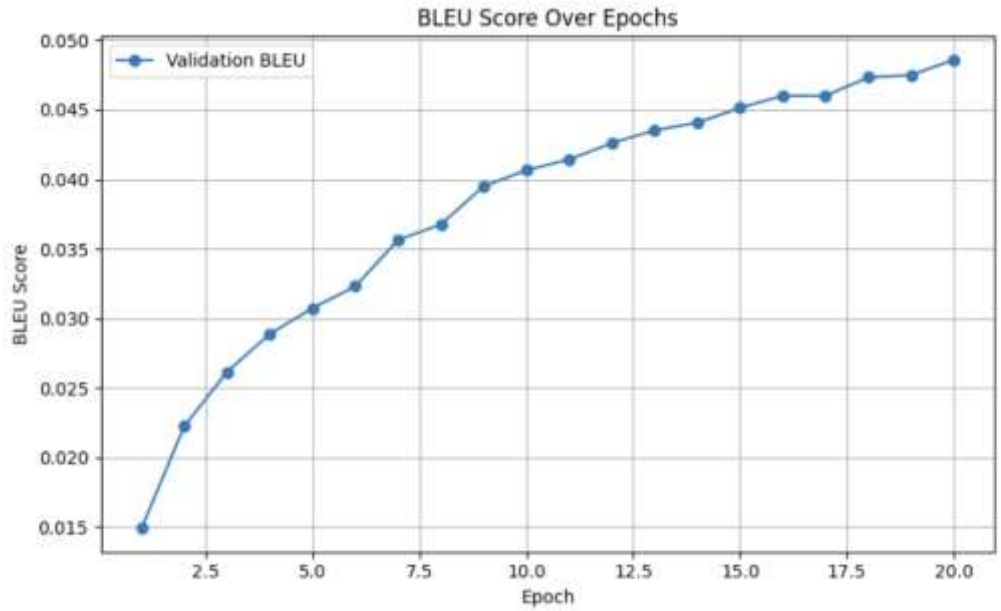


First one is the result of top 10 words. They are really common and non-meaningful words. Second picture is the result of top 50 words. Words like “president”, “political”, “language”, and “parliament” are grouped together, which makes sense. Function words which are in the first picture are far from the cluster.

Task 2

```
Epoch 16/20 | Train Loss: 4.5466 | Val Loss: 5.6108 | BLEU: 0.0460
Training
100% 563/563 [00:20<00:00, 24.02it/s]
Validating
100% 63/63 [02:19<00:00, 2.02s/it]
Epoch 17/20 | Train Loss: 4.4591 | Val Loss: 5.6127 | BLEU: 0.0460
Training
100% 563/563 [00:21<00:00, 27.47it/s]
Validating
100% 63/63 [02:19<00:00, 1.98s/it]
Epoch 18/20 | Train Loss: 4.3740 | Val Loss: 5.6088 | BLEU: 0.0473
Training
100% 563/563 [00:21<00:00, 28.24it/s]
Validating
100% 63/63 [02:17<00:00, 1.89s/it]
Epoch 19/20 | Train Loss: 4.2874 | Val Loss: 5.6091 | BLEU: 0.0475
Training
100% 563/563 [00:21<00:00, 28.26it/s]
Validating
100% 63/63 [02:18<00:00, 2.08s/it]
Epoch 20/20 | Train Loss: 4.2008 | Val Loss: 5.6140 | BLEU: 0.0486
```





Transformer architecture model

I implemented a sequence-to-sequence Transformer model for machine translation from English to German using PyTorch.

- Encoder-Decoder Architecture:
  - Based on PyTorch's nn.Transformer.
  - Shared positional encoding and token embeddings for source and target sequences.
- Core Components:
  - TokenEmbedding: Converts tokens into embeddings.
  - PositionalEncoding: Adds position-aware signals to embeddings.
  - Transformer: The standard multi-head self-attention encoder-decoder module.
  - Linear Generator: Projects decoder outputs to the target vocabulary size.

Hyper parameters

SRC_VOCAB_SIZE = len(vocab_transform['en'])
TGT_VOCAB_SIZE = len(vocab_transform['de'])
EMB_SIZE = 192
NHEAD = 6
FFN_HID_DIM = 192
BATCH_SIZE = 16

NUM_ENCODER_LAYERS = 3
NUM_DECODER_LAYERS = 3
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
NUM_EPOCHS = 20

Training

Dataset: Europarl v7 English-German corpus
Tokenization: spaCy (en_core_web_sm and de_core_news_sm)
Vocabulary: Built using build_vocab_from_iterator with min freq = 1
Data Split: 90% train, 10% validation (from 10,000 samples)
Batch Processing: Sequences padded to max length in each batch

Analysis & Findings

Throughout the training process, the BLEU score steadily increased across epochs, indicating that the model progressively learned to generate more fluent and accurate translations. This improvement suggests that the model was successfully capturing the underlying structure of the English-German language pairs and learning to generalize meaningful patterns for translation.

Both training and validation loss showed a downward trend in the initial epochs, which reflects effective learning and convergence. However, a closer look reveals that while the training loss continued to decrease, the validation loss plateaued in the later epochs. This divergence between BLEU score and validation loss suggests that, although the model's token-level accuracy (measured by loss) did not improve significantly, it was still producing higher-quality translations as measured by BLEU. This could point to potential overfitting, where the model begins to memorize training patterns rather than generalizing well to unseen validation data. Alternatively, it may indicate a saturation point where further improvements in loss do not translate into significant sequence-level improvements.

One contributing factor to this plateau could be the limited dataset size. Only 9,000 sentence pairs were used for training and 1,000 for validation, which is relatively small for training a Transformer-based architecture. A larger dataset would likely help the model better generalize, reduce overfitting, and improve both validation loss and BLEU score over time.

To address these observations, several improvements could be considered. Increasing the amount of training data would be the most straightforward and potentially impactful step. Additionally, incorporating regularization techniques such as label smoothing, dropout tuning, or weight decay could help mitigate overfitting. It may also be beneficial to implement early stopping based on validation BLEU score to avoid over-training the model once performance

plateaus. Finally, further hyperparameter tuning—such as increasing the embedding size or experimenting with different learning rate schedules—could enhance the model's ability to learn more nuanced linguistic patterns and improve overall performance.

BLUE score is increasing in each epoch, which means learning successfully.

Training and Validation loss is decreasing through the epochs.

Despite the increase in BLEU score, the validation loss remained relatively flat, indicating potential overfitting or saturation of the model's learning capacity on the validation set.

Further improvements could involve:

Increasing the number of data