HW3 – The One with the Transformer

Grigore Filoftei-Andrei

1 Data Processing and Organization

The training data was organized and processed using the following pipeline:

- For character-level tokenization:
  - Direct character-by-character encoding
  - Training set size: 1,484 batches
  - Validation set size: 183 batches
- For subword tokenization:
  - Implemented using byte-pair encoding (BPE)
  - Training set size: 1,396 batches
  - Validation set size: 175 batches

2 Model Architectures

Two model architectures were implemented and compared:

2.1 Small Model Architecture

- Hidden dimension: 256
- Number of layers: 4
- Number of attention heads: 4
- Feed-forward dimension: 1024
- Dropout rate: 0.1

2.2 Large Model Architecture

- Hidden dimension: 512
- Number of layers: 8
- Number of attention heads: 8
- Feed-forward dimension: 2048
- Dropout rate: 0.1

3 Training Parameters

The following parameters were used during training:

- Batch size: 64
- Number of epochs: 4
- Initial learning rate: 3e-4
- Warmup steps: 1,000
- Maximum steps: 50,000
- Gradient clipping value: 1.0
- Context window size: 128 tokens
- Early stopping patience: 3 epochs

3.1 Training Schedule

A linear learning rate warmup followed by a cosine decay schedule was implemented:

- Linear warmup for the first 1,000 steps
- Cosine decay from peak learning rate to near zero
- Final learning rate: 1.23e-9

4 Performance Analysis

4.1 Training Dynamics

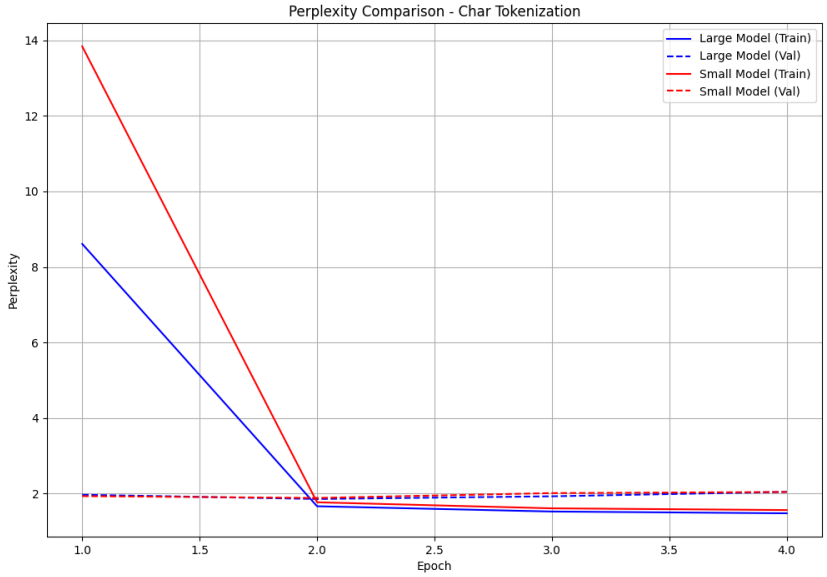

Figure 1: Training and validation perplexity over epochs for character tokenization models. The small model shows higher initial perplexity but converges similarly to the large model by the end of training.

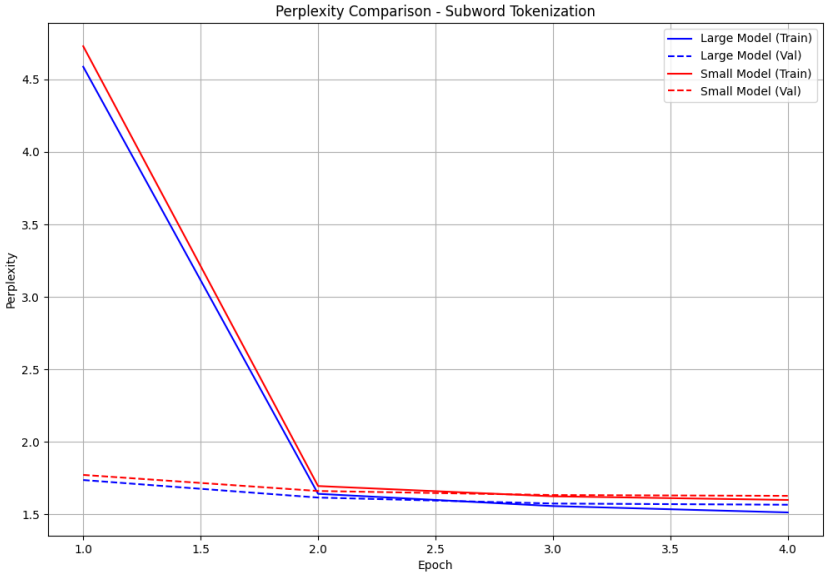

Figure 2: Training and validation perplexity over epochs for subword tokenization models. Both models show similar convergence patterns with lower overall perplexity compared to character tokenization.

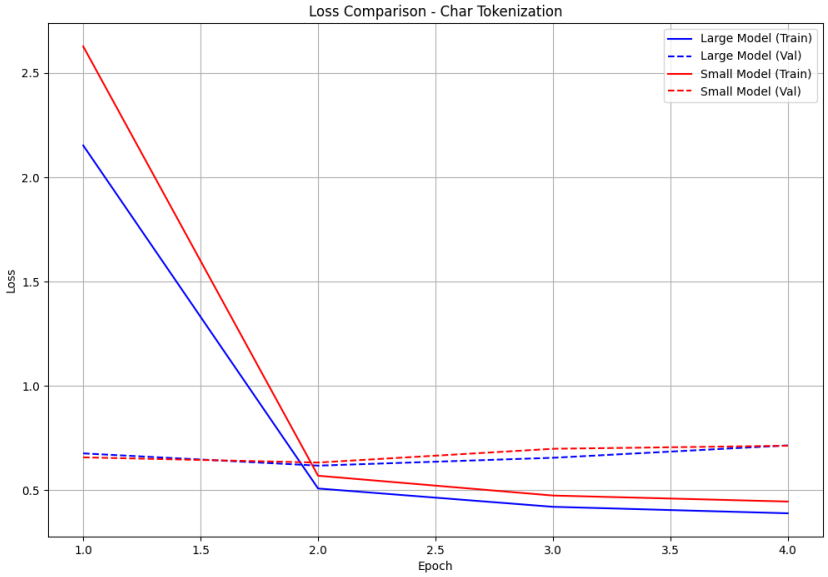

Figure 3: Training and validation loss curves for character tokenization models. The large model demonstrates more stable training with consistently lower loss values.

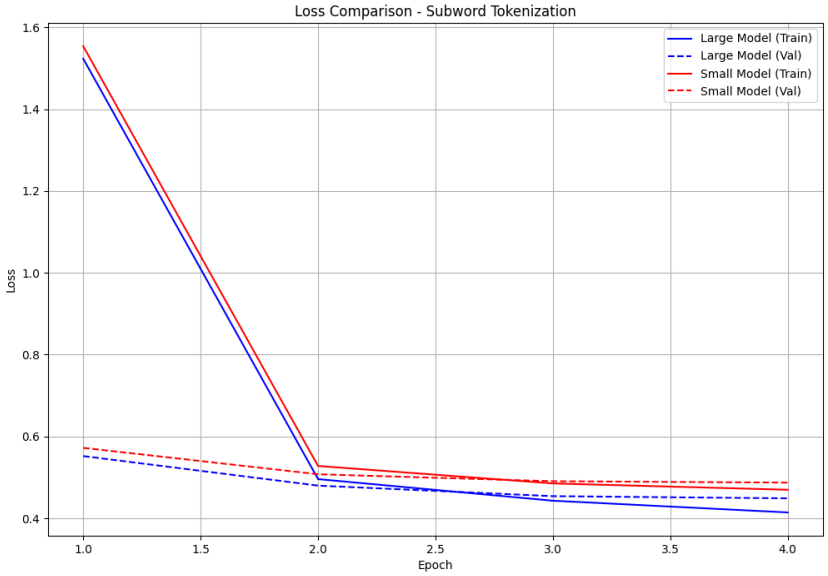

Figure 4: Training and validation loss curves for subword tokenization models. Both models show smooth convergence with minimal gap between training and validation loss.

4.2 Final Performance Metrics

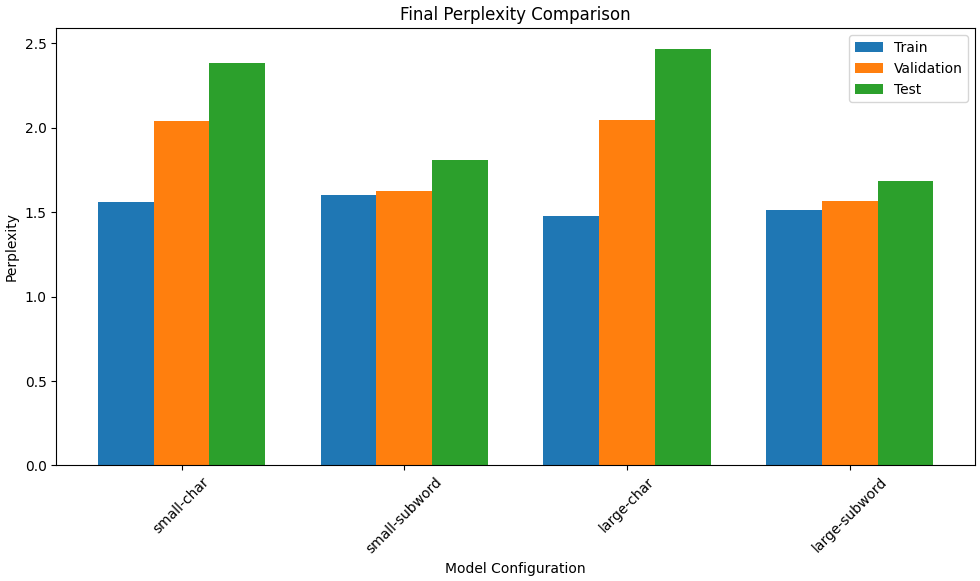

Figure 5: Comparison of final perplexity scores across all model configurations for train, validation, and test sets. Subword tokenization models consistently achieve lower perplexity.

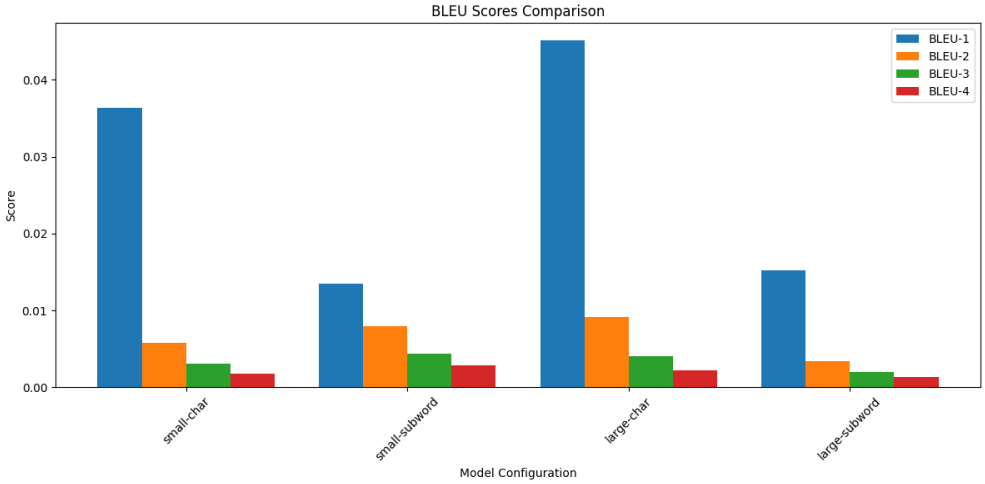

Figure 6: BLEU scores comparison across all model configurations. Character tokenization models achieve higher BLEU-1 scores despite higher perplexity.

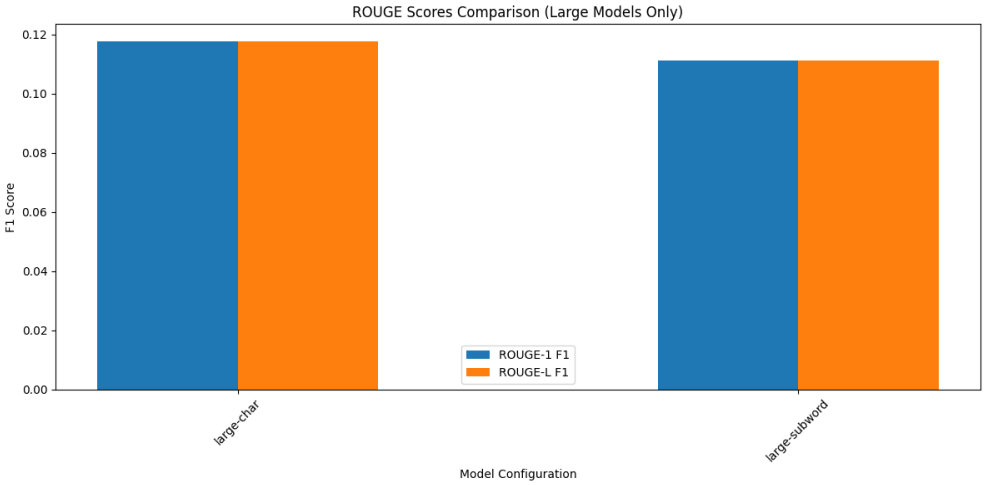

Figure 7: ROUGE scores comparison for large models. Both tokenization schemes achieve similar ROUGE-1 and ROUGE-L F1 scores.

4.3 Perplexity Comparison

Analysis of final perplexity scores across models:

Model Configuration	Train	Validation	Test
Small (Char)	1.56	2.04	2.38
Small (Subword)	1.60	1.63	1.81
Large (Char)	1.48	2.04	2.47
Large (Subword)	1.51	1.57	1.69

Table 1: Final perplexity scores across different model configurations

Key observations:

- Subword tokenization consistently outperformed character-level tokenization in terms of test perplexity
- The large model with subword tokenization achieved the best test perplexity (1.69)
- Character-level models showed higher variance between train and test performance

4.4 BLEU and ROUGE Scores

BLEU-1 scores across models:

- Small (Char): 0.036
- Small (Subword): 0.014
- Large (Char): 0.045
- Large (Subword): 0.015

ROUGE-1 F1 scores for large models:

- Large (Char): 0.118
- Large (Subword): 0.111

5 Generation Parameters

Text generation was performed using the following parameters:

- Sampling strategy: Top-k sampling
- k value: 50
- Maximum generation length: 100 tokens
- Context window size: 256 tokens

6 Qualitative Analysis

Sample generations from different models:

6.1 Character-level Models

6.1.1 Small Model

```
Prompt: Cymbeline
Target: : She's an adulteress.
Generated: : The will be the will be the present the world,
```

6.1.2 Large Model

```
Prompt: Cymbeline
Target: : wringing her hands, and all our house in a great
Generated: : The such a pride of the world of the world,
```

6.2 Subword Models

6.2.1 Small Model

```
Prompt: Cymbeline: Sir, I may tell it
Target: you, I think I have
Generated: .!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

6.2.2 Large Model

```
Prompt: Cymbeline: With Lady Margery, your
Target: midwife there,
Generated: grace's daughter,!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

7 Discussion and Conclusions

The experimental results, as visualized in Figures 1 through 7, reveal several key patterns:

- Training Dynamics (Figures 1, 2):
  - Character tokenization models show higher initial perplexity but demonstrate rapid improvement in early epochs
  - Subword tokenization leads to more stable training curves with lower perplexity throughout training
  - Large models consistently achieve lower training perplexity compared to their smaller counterparts
- Loss Patterns (Figures 3, 4):
  - Both tokenization schemes show good convergence with minimal overfitting
  - Large models maintain lower loss values throughout training
  - Subword models exhibit smoother loss curves compared to character models
- Evaluation Metrics (Figures 5, 6, 7):
  - Subword tokenization consistently achieves lower perplexity across all splits
  - Character tokenization models show surprisingly stronger BLEU-1 scores
  - ROUGE scores are comparable between tokenization schemes for large models

The experimental results reveal several key findings:

- Subword tokenization generally led to better perplexity scores but lower BLEU scores compared to character-level tokenization

- The large model architecture showed improved performance in terms of perplexity and generation quality
- Both tokenization schemes showed limitations in maintaining coherent long-form generation
- Character-level models produced more natural-looking but potentially less contextually relevant continuations
- Subword models showed a tendency to default to repetitive patterns (exclamation marks)

Future improvements could include:

- Implementation of better sampling strategies (e.g., nucleus sampling)
- Increased model size and training data
- Enhanced tokenization methods
- Improved handling of special tokens and punctuation