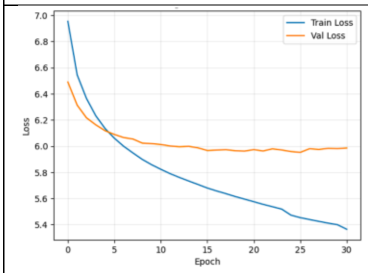
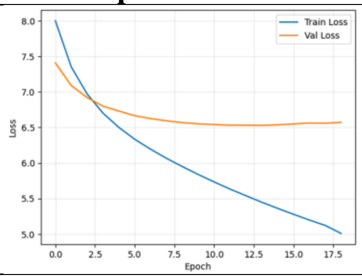
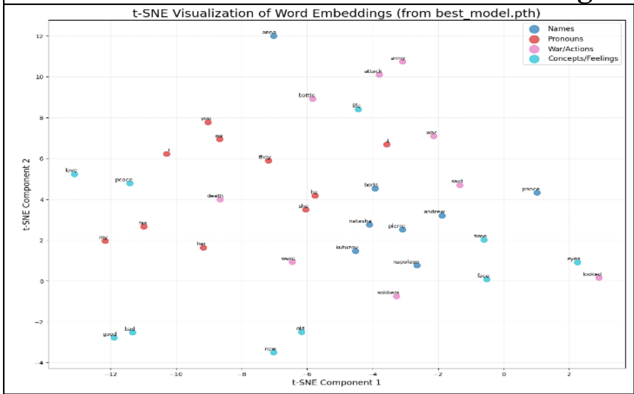
Next – Word Prediction

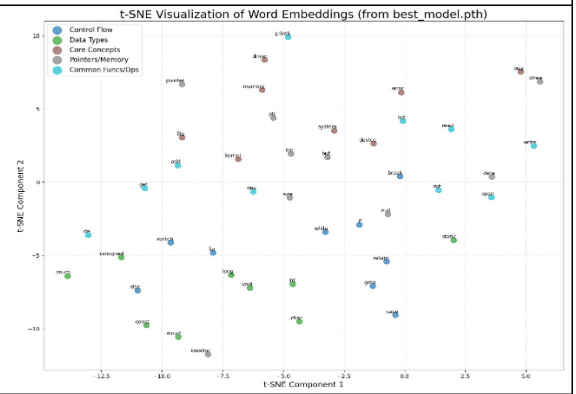
Category - 1	Category - 2
Vocab Size	
17515	109239
10 Most Frequent	
{'the': 34544, 'and': 22221, 'to': 16667, 'of': 14885, 'a': 10544, 'he': 9998, 'in': 8975, 'that': 8169, 'his': 7983, 'was': 7357}	{'*': 33504, '=': 28003, 'if': 18989, '{': 18915, 'the': 17488, '}': 16965, '*/': 13445, '/*': 12190, 'struct': 10997, 'return': 10274}
10 Least Frequent	
{'buonapartes': 1, 'infamies': 1, 'elite': 1, 'grandfathers': 1, 'canceled': 1, ' tease': 1, 'stale': 1, 'impulsiveness': 1, 'enthusiast': 1, 'noblest': 1}	{'linux/kernel/irq/autoprobe.c': 1, 'apis.': 1, "'irqs_waiting'": 1, 'define_mutex(probing_active);': 1, 'probe_irq_on': 1, 'commence': 1, 'probe_irq_on(void)': 1, 'mutex_lock(&probing_active);': 1, '(desc->irq_data.chip->irq_set_type)': 1, 'desc->irq_data.chip- >irq_set_type(&desc->irq_data,': 1}
Dataset Size (Context Size = 3)	
(592128,2)	(795699,2)
Training vs validation loss plot	
	
Final validation loss/accuracy	
5.9521	6.5292
Example	

```
Input: 'he said to'  ->  Predicted: 'the'
Full sentence: 'he said to the'

Input: 'it was the'  ->  Predicted: 'same'
Full sentence: 'it was the same'

Input: 'they went to the'  ->  Predicted: 'door'
Full sentence: 'they went to the door'
```

```
Successfully loaded 'best_model.pth'
Input: 'if the pointer'  ->  Predicted: 'is'
Full sentence: 'if the pointer is'

Input: 'the kernel must'  ->  Predicted: 'be'
Full sentence: 'the kernel must be'

Input: 'this is a bug'  ->  Predicted: '*'
Full sentence: 'this is a bug *'
```

Embedding Visualisation





2) Commentary on learning behaviour.

Category I: "War and Peace" (Natural Language)

» **Learning Behavior:** The training and validation loss curves in the notebook show a classic overfitting pattern.

- The **Training Loss** (blue line) consistently decreases, indicating the model is getting better at memorizing the training data.
- The **Validation Loss** (orange line) decreases until **Epoch 25**, where it hits its minimum (best) value of **5.9521**.
- After this point, the validation loss begins to rise, showing the model is losing its ability to generalize to new data. The early stopping mechanism correctly halted training at Epoch 30.

» **Example Predictions :**

- `"he said to"` -> `"the"`
- `"it was the"` -> `"same"`
- `"they went to the"` -> `"door"`
- **Commentary:** These predictions are excellent. They are all grammatically correct and semantically plausible within the context of a novel. They demonstrate that the model successfully learned common English sentence structures and word associations.

Category II: Linux C Code (Structured Language)

- **Learning Behavior:** This model shows an almost identical learning dynamic to the "War and Peace" model.

    - The "Training vs Validation Loss" plot clearly shows overfitting, where the training loss steadily drops while the validation loss begins to increase after its minimum.
    - The model achieved its best validation loss of **6.5292** at **Epoch 13**.
    - Early stopping was triggered at Epoch 18, correctly preventing further overfitting.

- **Example Predictions :**

    - "if the pointer" -> "is"
    - "the kernel must" -> "be"
    - "this is a bug" -> "*"
    - **Commentary:** These predictions are also excellent and demonstrate a strong grasp of C code syntax. The predictions "is" and "be" are logical and common. The third prediction, "*", is the most revealing. Instead of predicting an English word, the model predicted a symbol that is syntactically very likely to follow "bug" in C (e.g., *bug, as in a pointer declaration or dereference). This proves the model learned the *specific grammar of C code*, not just general English.

3) Discuss your observations on clustering patterns and semantic relationships.

Category I (War and Peace): The clusters are **semantic** and **grammatical**.

- Words with similar meanings or roles are grouped. For example, Names (prince, anna, pierre, kutuzov) form a cluster.
- Pronouns (he, she, it, his, her) form another tight, distinct cluster because they are used in similar grammatical contexts.
- War/Actions (war, battle, army, soldiers) are grouped, and Concepts/Feelings (peace, love, life) are grouped.
- This demonstrates the model learned the **semantic meaning** of words.

Category II (Linux C Code): The clusters are purely **functional** and **syntactic**.

- Control Flow keywords (if, else, for, return) form a very tight cluster, as they serve the same function of directing the program's logic.
- Data Types (int, char, struct, void) are grouped because they are all used in variable declarations.
- Pointers (pointer, ptr, null) are clustered, showing the model learned this specific C programming concept.

- This demonstrates the model learned the **functional role** of tokens within the C language.

5) Compare your two trained models (Category I vs Category II): - Dataset size, vocabulary, context predictability - Model performance (loss curves, qualitative generations) - Embedding visualizations. Summarize insights on how natural vs structured language differs in learnability

Metric	**Category I: "War and Peace" (Natural)**	**Category II: Linux C Code (Structured)**
Dataset Size	(592128,2)	(795699,2)
Vocabulary (stoi)	**17,515**	**109,239**
Best Validation Loss	**~5.95** (Better)	**~6.53** (Worse)
Context Predictability	**Low.** Language is flexible and creative.	**High.** Language is rigid and syntactic.
Qualitative Generation	Good semantic plausibility ("it was the same").	Excellent syntactic accuracy ("the kernel must be").
Embedding Clusters	**Semantic** (e.g., "Names", "Pronouns").	**Functional** (e.g., "Control Flow", "Data Types").