


# Next Word Predictor using Deep Learning and NLP

NLP-powered Text Prediction with LSTM by [Your Name] - [Date]

 by Adon banker



Made with GAMMA



# Problem Statement: The Power of Prediction

Next-word prediction anticipates the next word in a sequence, revolutionizing how we interact with technology. It's crucial for enhancing user experience and efficiency across various applications.

## Chat Applications

Speeds up messaging with intelligent suggestions.

## Google Search

Autocompletes queries, making information retrieval faster.

## Text Editors

Improves writing flow and reduces typing effort.

# Dataset Used: Sherlock Holmes Corpus

For this project, we leveraged the rich narrative of the Sherlock Holmes corpus from Kaggle, providing a robust foundation for our model's linguistic understanding.

## Dataset Details

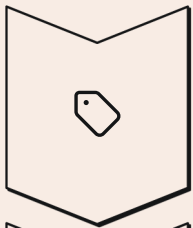
- Source: Sherlock Holmes novels and stories.
- Size: Approximately 5.4 MB of raw text.
- Type: Rich narrative English prose.

## Initial Processing

- **Cleaning:** Removed special characters and normalized text.
- **Tokenization:** Broke text into individual words.
- **Sequence Creation:** Formed sequences for training.

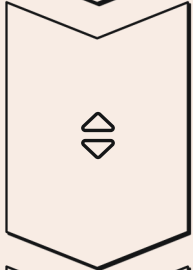
# Text Preprocessing: Shaping the Data

Effective text preprocessing transforms raw text into a format suitable for deep learning models, ensuring the model receives clean and structured input.



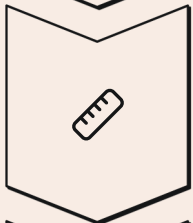
## Keras Tokenizer

Mapped words to unique integer IDs.



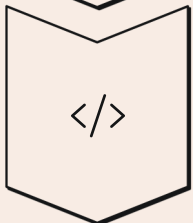
## N-Gram Sequences

Created input-output pairs (e.g., "word1 word2" -> "word3").



## Sequence Padding

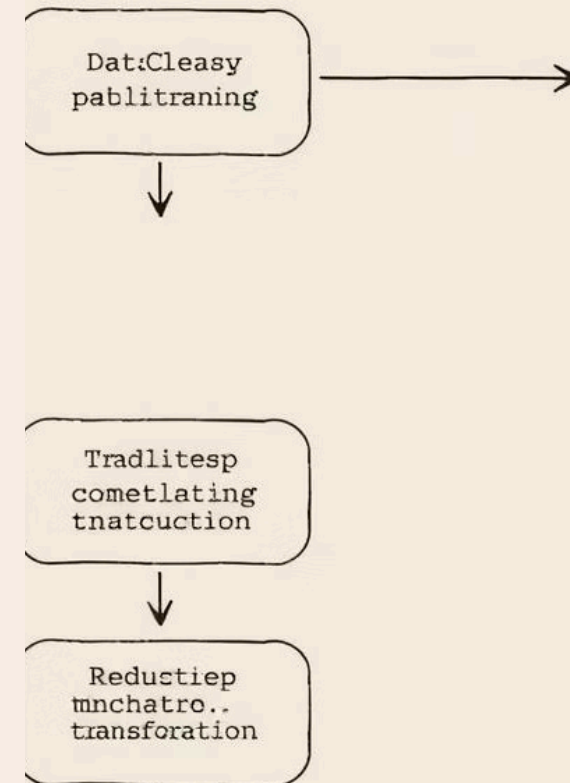
Ensured all input sequences had uniform length.



## One-Hot Encoding

Converted target words into binary vectors.

## Data Preprocessing



# Model Architecture: TensorFlow and LSTM

Our next-word predictor is built using a powerful TensorFlow/Keras architecture, specifically designed to handle sequential data with an LSTM layer.

## Embedding Layer

Converts word integers into dense vectors.

## LSTM Layer (150 Units)

Learns long-term dependencies in the text sequences.

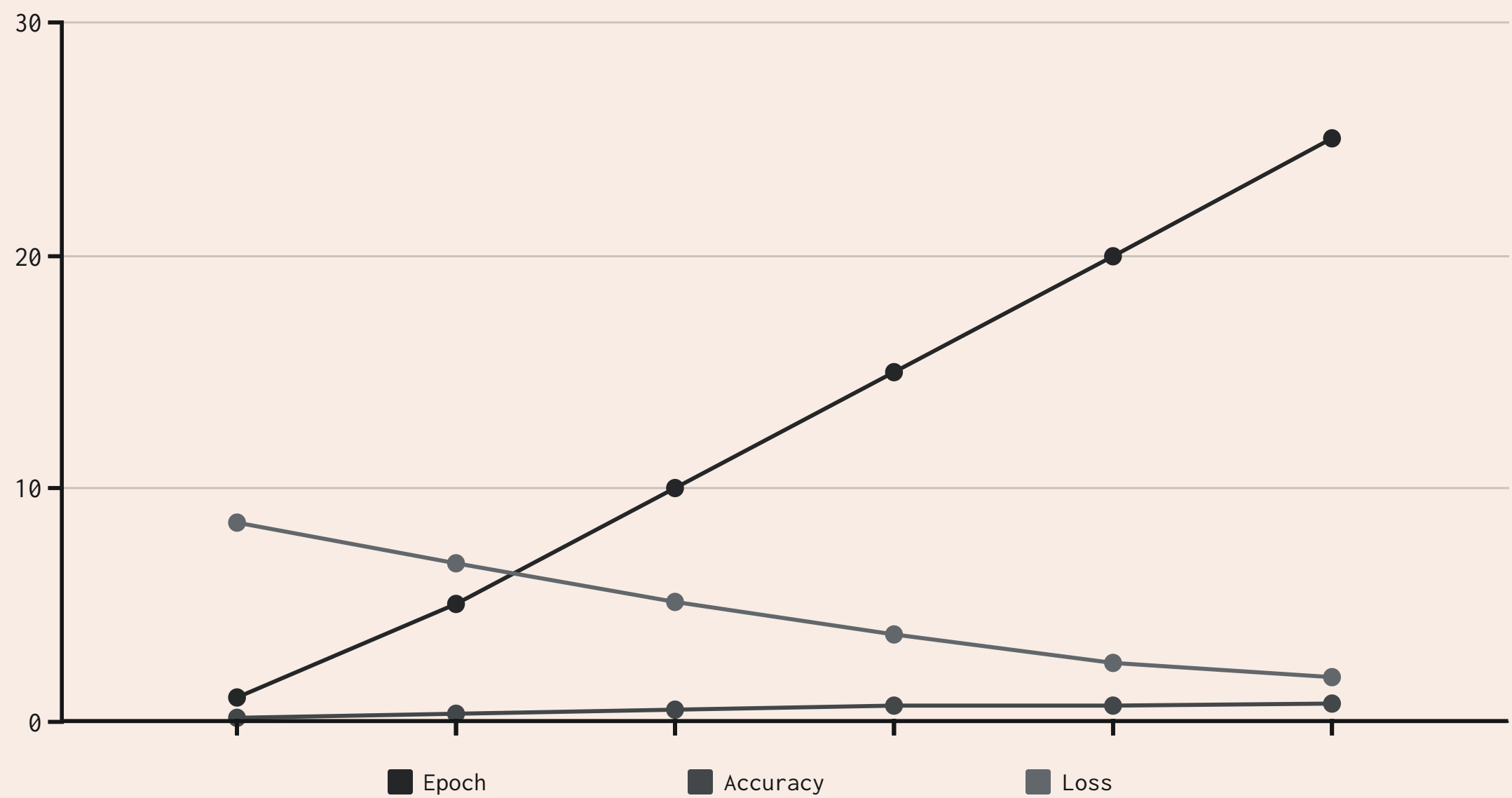
## Dense Output Layer

Predicts the probability distribution of the next word using Softmax activation.

The model was compiled with Categorical Crossentropy loss and optimized using the Adam optimizer.

# Training Summary: Performance Metrics

The model underwent 25 epochs of training, demonstrating a consistent improvement in accuracy and a reduction in loss, indicating effective learning.



Final Accuracy: Approximately 71.6%



# Text Generation Output: A Glimpse of Intelligence

When given the seed phrase "I saw a door," our model produced the continuation: "I saw a door when but taken one of them..."

"I saw a door when but taken one of them..."

This output demonstrates the model's ability to learn sentence structure and generate coherent, though sometimes nonsensical, sequences. It captures grammatical patterns, even if the semantic meaning isn't always perfect.



# Key Learnings: Insights from Experimentation

This project yielded valuable insights into the intricacies of deep learning for natural language processing.



## LSTM Power

Long Short-Term Memory networks are exceptionally effective for sequence learning tasks.



## Preprocessing is Key

Thorough text preprocessing is fundamental for model performance and accuracy.



## Data-Driven Improvement

Model performance significantly benefits from larger datasets and extended training epochs.





# Future Improvements & Next Steps

This project serves as a strong foundation, and several avenues exist for further enhancement and application.

## Enhancements

- **Larger Corpora:** Incorporate Wikipedia or vast book datasets for richer vocabulary.
- **Advanced Models:** Experiment with Transformer or GPT architectures for state-of-the-art results.

## Applications

- **Web Application:** Develop a user-friendly interface using Flask or Streamlit.
- **Real-time Integration:** Explore integrating the model into existing text-based applications.

Thank You! Questions or Feedback?

Contact me

GitHub Repository