# Next Word Prediction Using Multi-Turn Dialogue Dataset

•••

IE 7500 - Natural Language Processing

Team: Kirthana Shri Chandra Sekar, Srinidhi Aduri, Suja Ganesh Murugan

# Objective

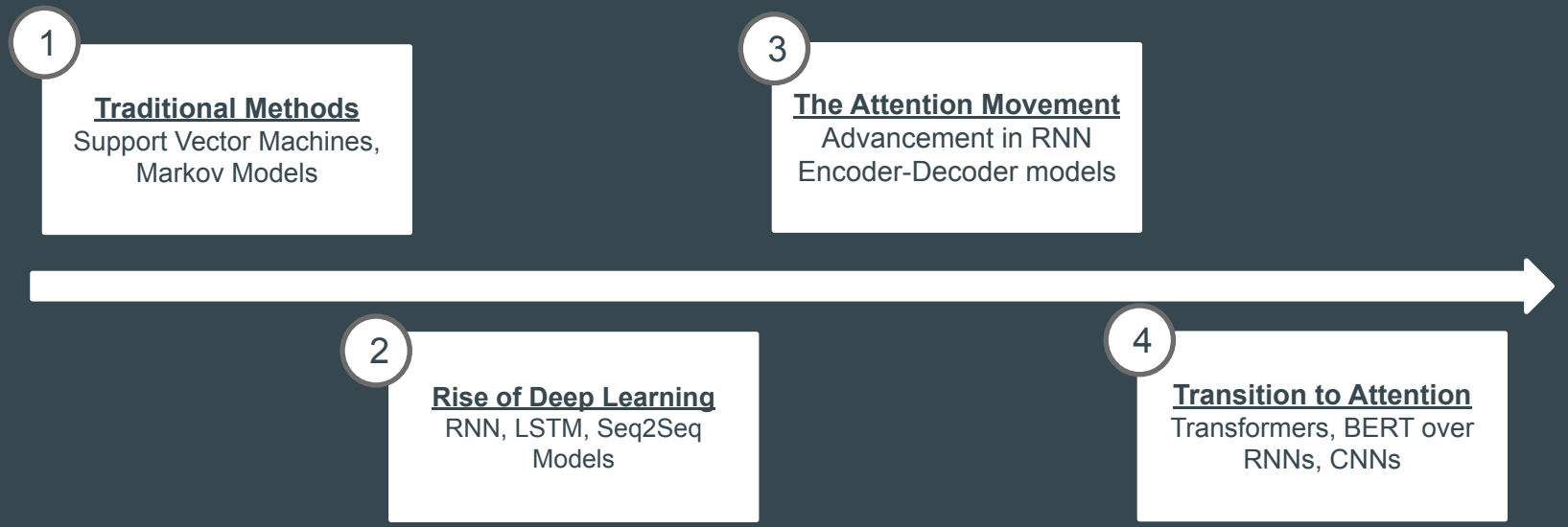
The objective of this project is to build a next word prediction model suitable for auto-completion systems using the multi-turn dialogue dataset

Enhance user interaction through immediate suggestions

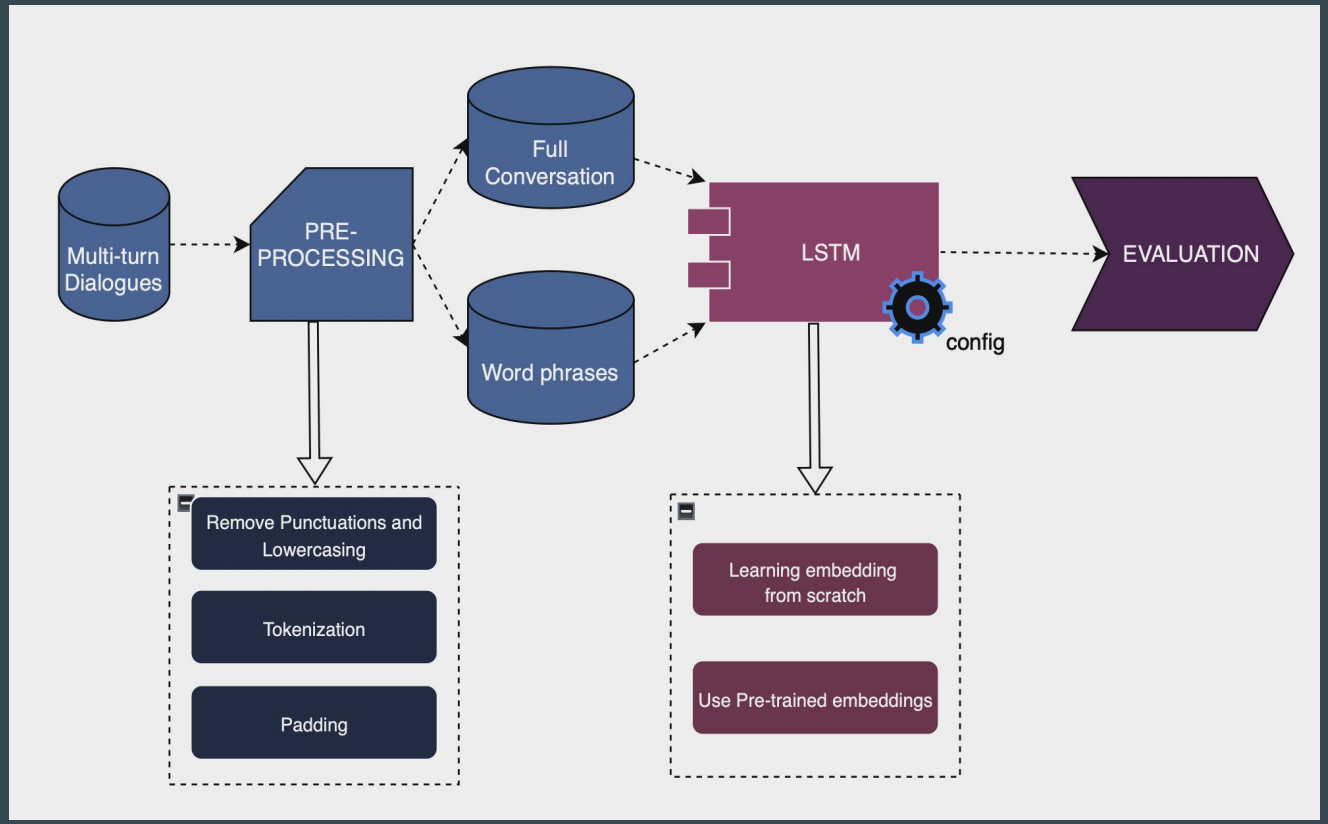
Minimize typing error through improved typing accuracy

Leverage State of the Art NLP techniques

# Outline of Previous Work Done

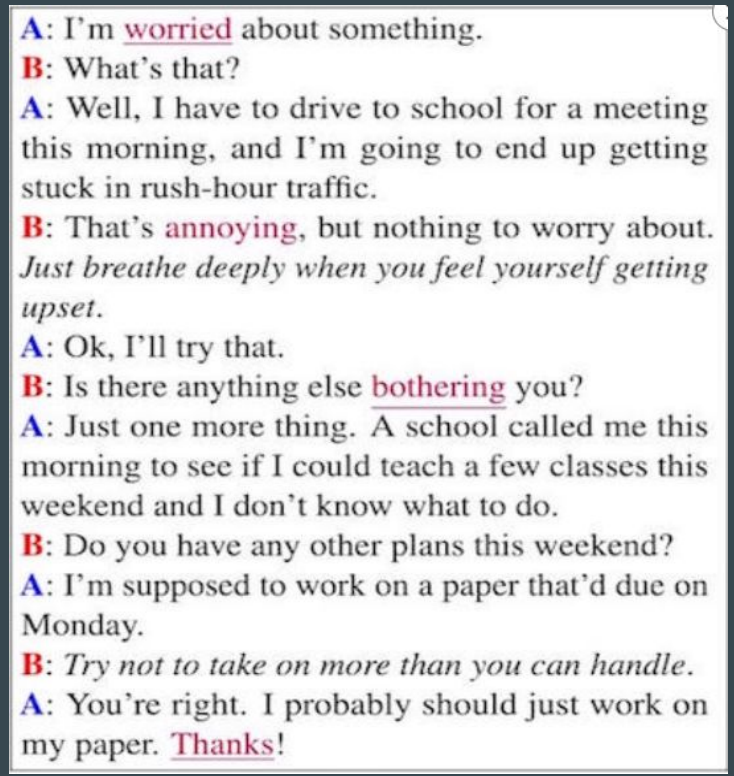


# Methodology

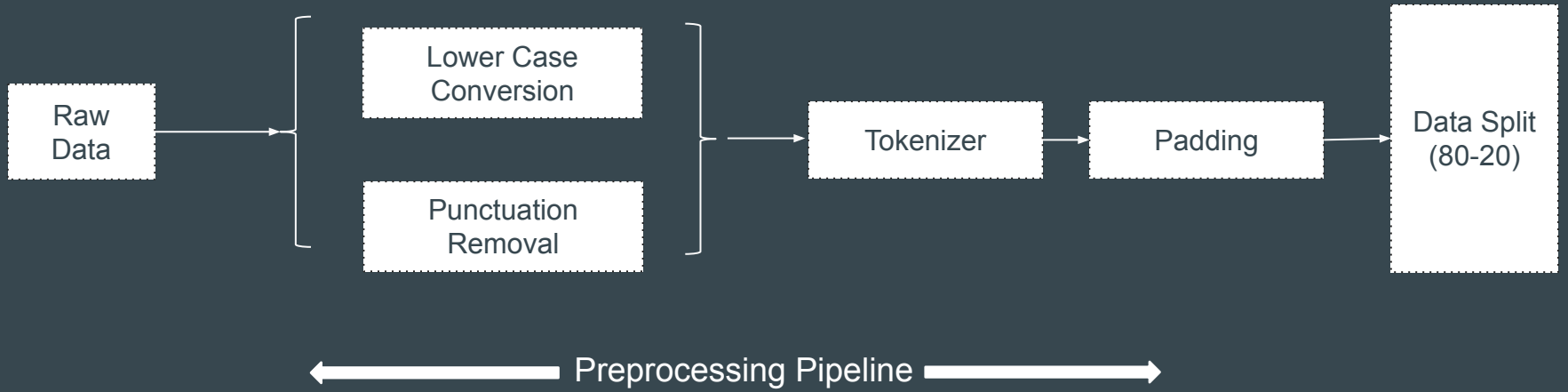


# About the Dataset

- This is a daily dialogue dataset, that is human-written, and contains daily conversations on various topics
- It has around 13,000 transcribed dialogues
- Specifics on dataset:
  - average of 15 tokens per utterance
  - Dialogue separation by <eou>



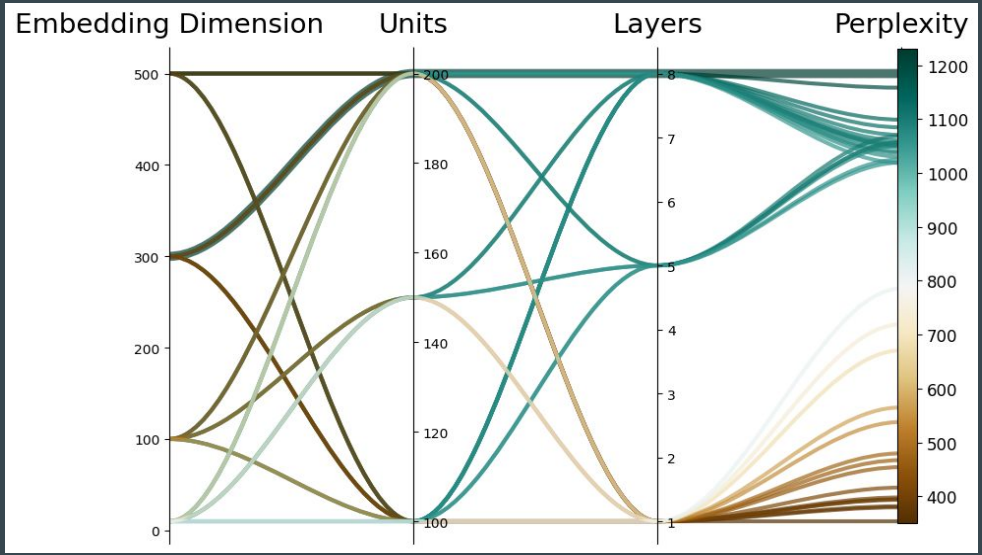
# Data Preprocessing Flow Diagram



# Model Development and Evaluation Framework

| Training Development | Model Evaluation | |
|---|---|---|
| **Training: Tensorflow** | **Loss** | Train, Validate and test |
| **Epochs: 30** | **Accuracy** | |
| **Overfitting mitigation: Early Stopping** | **Perplexity** | Test |
| **Metric Tracking: MLFlow** | **Top k accuracy** | |

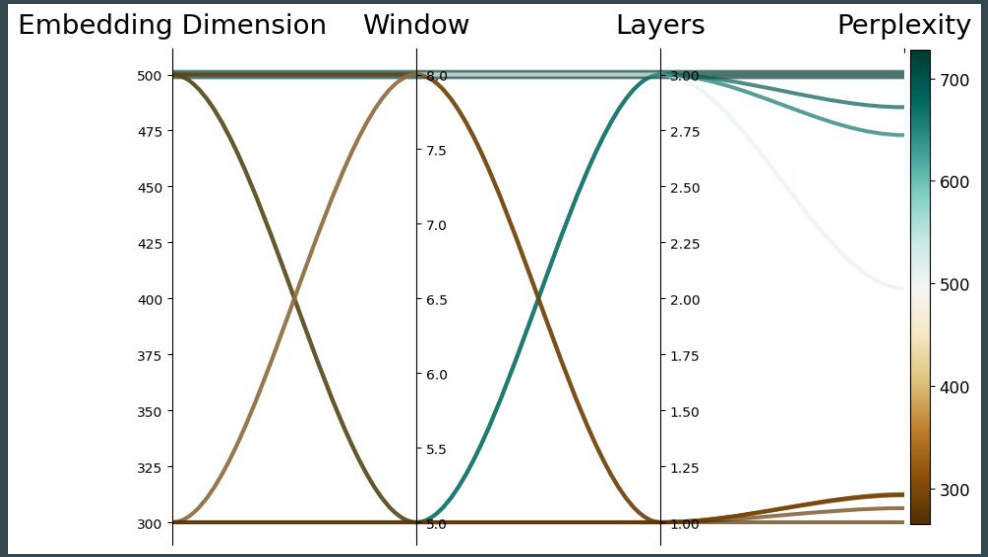
# LSTM – Each Utterance + embedding from scratch




Observations:

- Struggles to remember long term dependencies due to high input sequence length
- Vanishing gradient problem is more pronounced in deep networks

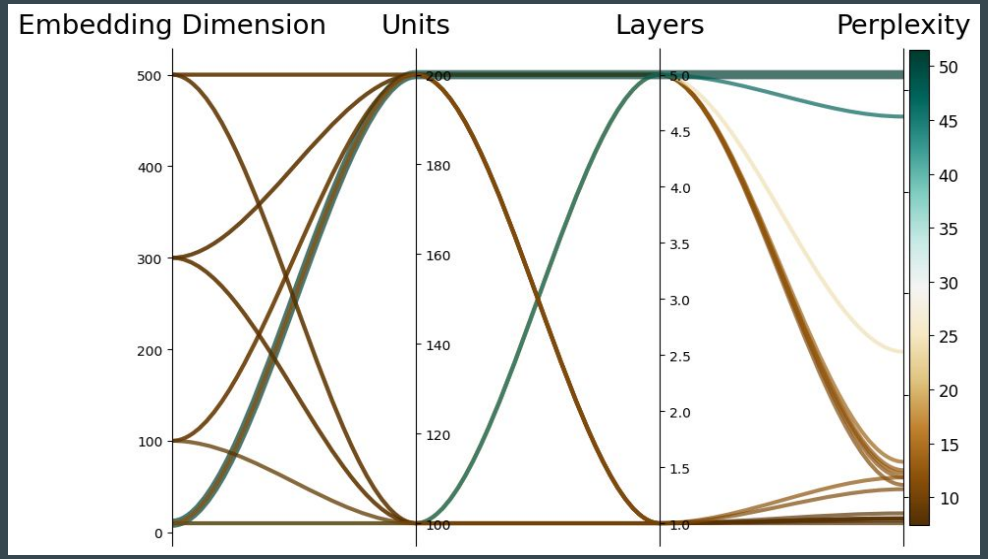
# LSTM – Each Utterance + word2vec




Observations:

- Adopting word2vec yielded better results but the model still struggled to learn effectively

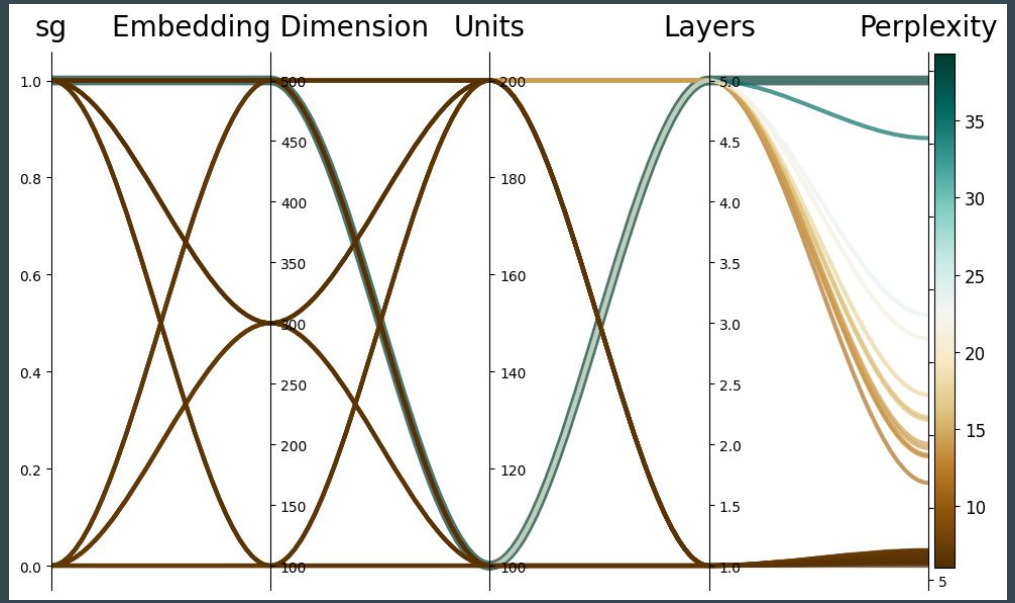
# LSTM Model – Five-in, one-out sequence + embedding from scratch




Observations:

- The model is given five words from a sentence and is trained to predict the sixth word
- Notable performance in model's performance, perplexity reduced substantially
- Demonstrates the effectiveness of using shorter input sequences to enhance the LSTM model

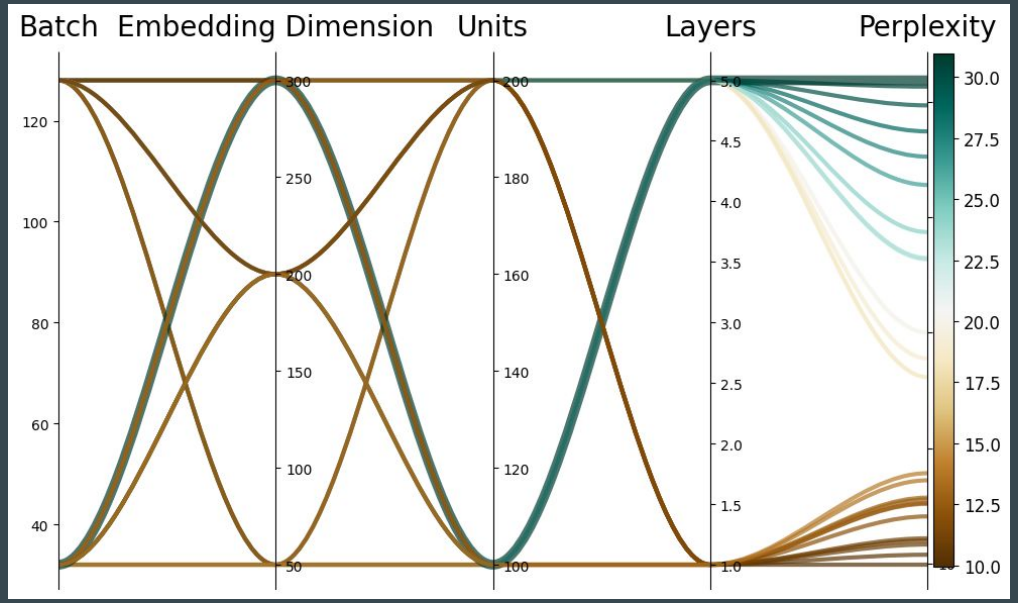
# LSTM Model – Five-in, one-out sequence + Word2Vec embedding




Observation:

- The learning process enhanced due to word2vec embedding, resulting in a further reduction in perplexity.

# LSTM Model - Five-in, one-out sequence + GloVe embedding




Observations:

- The usage of GloVe embedding displayed similar performance as Word2Vec

# Summary of best-performing Models

| Model | Embed_ dim | Layers | Units | Perplexity | Test Acc. | Top-5 Acc. |
|---|---|---|---|---|---|---|
| Word2Vec (sg=1) | 500 | 1 | 100 | **6.00** | 76.78 | **83.22** |
| Word2Vec (sg=1) | 300 | 1 | 200 | 6.14 | 76.30 | 83.10 |
| Word2Vec (sg=1) | 300 | 1 | 100 | 6.20 | **77.00** | 82.84 |
| Word2Vec (sg=1) | 500 | 1 | 200 | 6.34 | 75.90 | 82.88 |
| Word2Vec (sg=1) | 300 | 1 | 200 | 6.41 | 74.84 | 83.08 |

# Conclusion

- Models with Shallow layers consistently outperformed those with dense layers
- Five-Word-In and One-Word-Out sequence format helped achieve improved results
- Pre-trained embeddings improved the performance of the model
- Word2Vec embeddings gave highest accuracy among the tested methods.

# Thank You