Andrea Lolli, Ana Suarez, Valentina Bitetto

Student Essay Dataset

Text mining analysis of IELTS Essay dataset

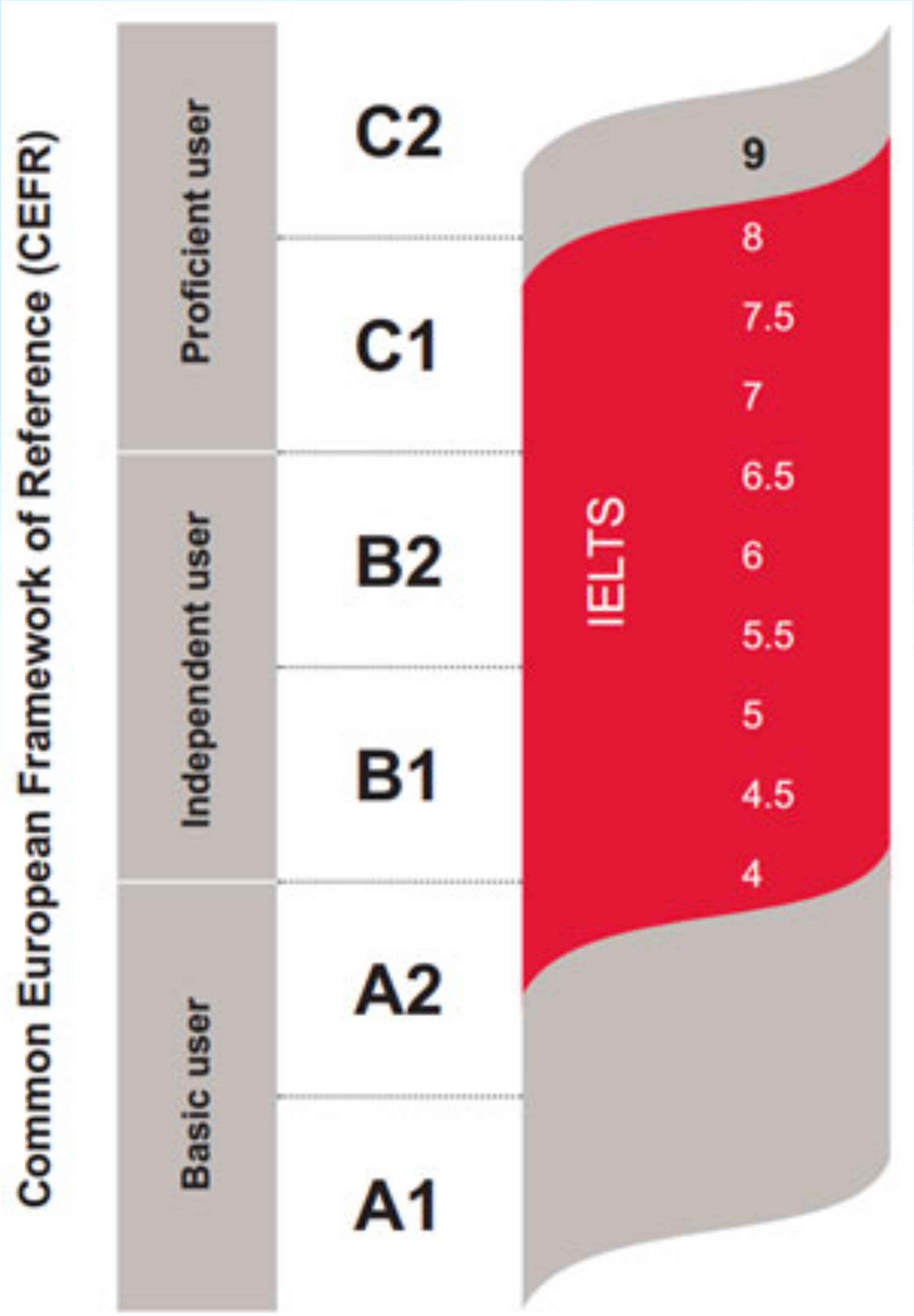
Table of Contents

- Project Objectives
- Introduction to the dataset
- Code explanation
    - Data Preprocessing
    - Vector representation
    - Model
    - Evaluation
- Conclusion

Project Objectives

Advancing Language Proficiency Assessment

- Goal: predict CEFR scores
- Approach: use text mining models, feature analysis, and NN architecture
- Uncover language patterns
- Automated essay grading needs further improvement



IELTS writing dataset

Initial Insights

- Evaluation of writing skills and responses
- Key evaluation criteria of essays
- Size: 1434
- Many missing values —> drop
- Remaining data: task type, question, essay, overall score

	Task_Type	Question	Essay	Examiner_Comment	Task_Response	Coherer
0	1	The bar chart below describes some changes abo...	Between 1995 and 2010, a study was conducted r...	NaN	NaN	
1	2	Rich countries often give money to poorer coun...	Poverty represents a worldwide crisis. It is t...	NaN	NaN	
2	1	The bar chart below describes some changes abo...	The left chart shows the population change hap...	NaN	NaN	
3	2	Rich countries often give money to poorer coun...	Human beings are facing many challenges nowada...	NaN	NaN	
4	1	The graph below shows the number of overseas v...	Information about the thousands of visits from...	NaN	NaN	

ıcy	Overall
aN	5.5
aN	6.5
aN	5.0
aN	5.5
aN	7.0

IELTS writing dataset

Initial Insights

- Removed columns with null values
- Focused dataset for subsequent analysis
- Examined the refined dataset for insights

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1435 entries, 0 to 1434
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Task_Type            1435 non-null   int64  
 1   Question             1435 non-null   object 
 2   Essay                1435 non-null   object 
 3   Examiner_Comment     62 non-null     object 
 4   Task_Response        0 non-null      float64
 5   Coherence_Cohesion   0 non-null      float64
 6   Lexical_Resource     0 non-null      float64
 7   Range_Accuracy       0 non-null      float64
 8   Overall              1435 non-null   float64
dtypes: float64(5), int64(1), object(3)
memory usage: 101.0+ KB
```

IELTS writing dataset

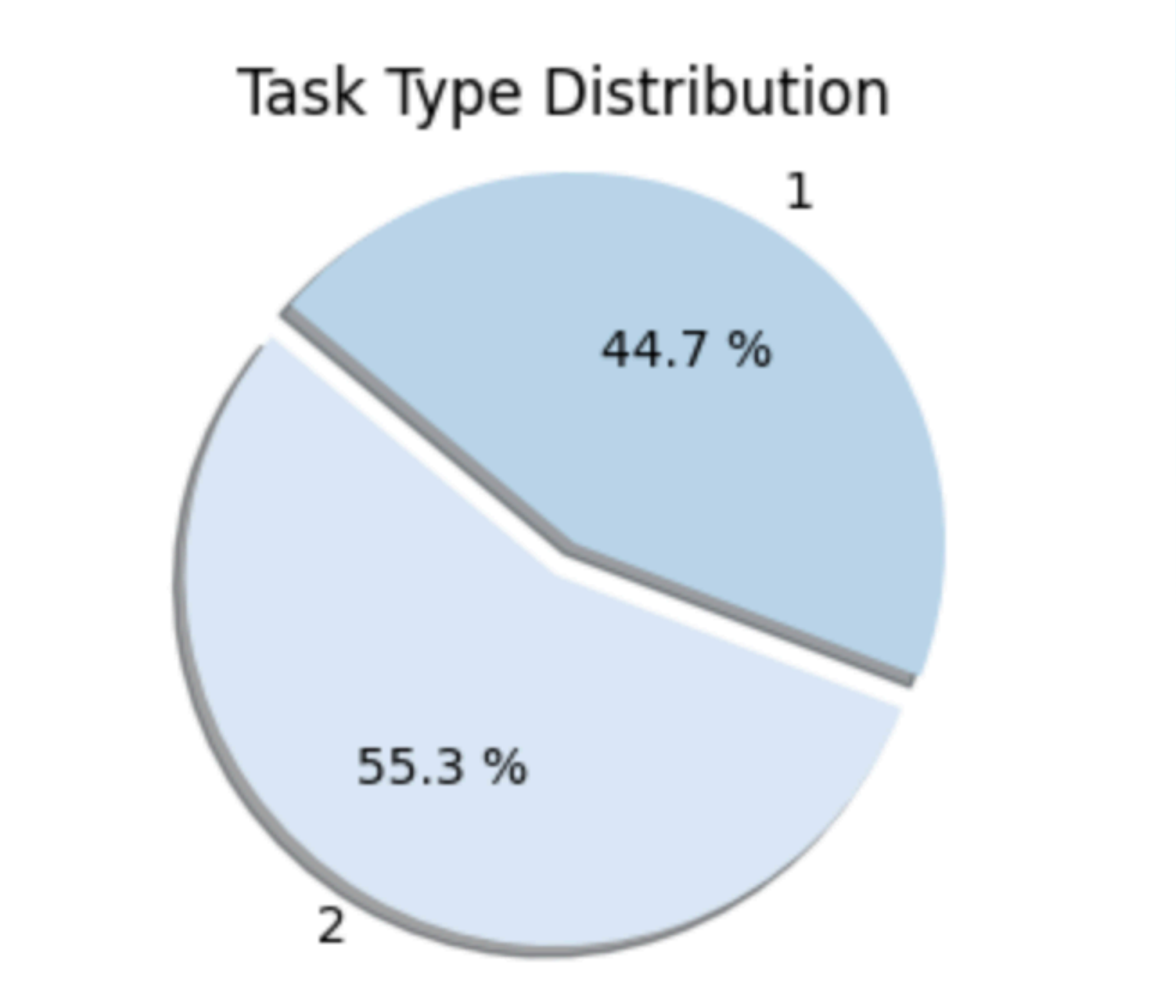
Essential Features

- Task_type: IELTS writing tasks, Task1/Task2
- Question: writing prompts
- Essay: written response of candidates
- Overall: final score of each essay converted into CEFR scores

	Task_Type	Question	Essay	Overall
0	1	The bar chart below describes some changes abo...	Between 1995 and 2010, a study was conducted r...	5.5
1	2	Rich countries often give money to poorer coun...	Poverty represents a worldwide crisis. It is t...	6.5
2	1	The bar chart below describes some changes abo...	The left chart shows the population change hap...	5.0
3	2	Rich countries often give money to poorer coun...	Human beings are facing many challenges nowada...	5.5
4	1	The graph below shows the number of overseas v...	Information about the thousands of visits from...	7.0

IELTS writing dataset

Understanding Task Type and Scores



IELTS writing dataset

Additional Features: Enhancing Dataset Value

- Missing words count
- Mean sentence length and vocabulary richness
- Readability scores (Flesch-Kincaid and Gunning Fog)
- Usage of transitional words and grammar/spelling errors
- Result: Deepening dataset for comprehensive analysis

IELTS writing dataset

Analysis of Results

- Missing words => 250 words minimum
- Variations in mean sentence lengths => differences in complexity and structure
- Wide range of unique word counts => diversity in vocabulary usage
- Reading difficulty levels (Flesch-Kincaid scores from 0 to 20) => range 6.8 to 11.9
- Readability diversity (Gunning Fog Index scores from 0 to 20) => range 8.02 to 12.9
- Patterns in transitional word usage

Preparing the dataset: Data Preprocessing

Data preprocessing pipeline

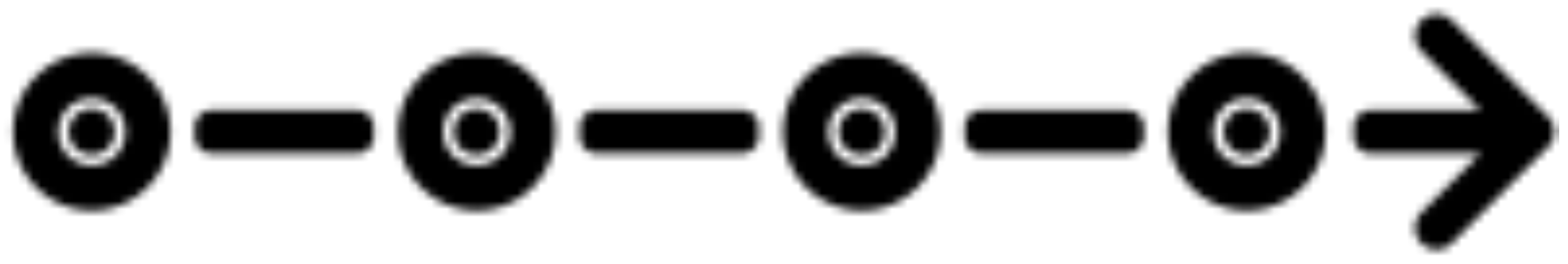
STEP 1: remove special characters and punctuation marks

STEP 2: convert text to lowercase

STEP 3: tokenize the text

STEP 4: remove stop words from the tokenized text

STEP 5: lemmatize each word (reduce to dictionary form)



Preparing the dataset: Data Preprocessing

Function for text preprocessing

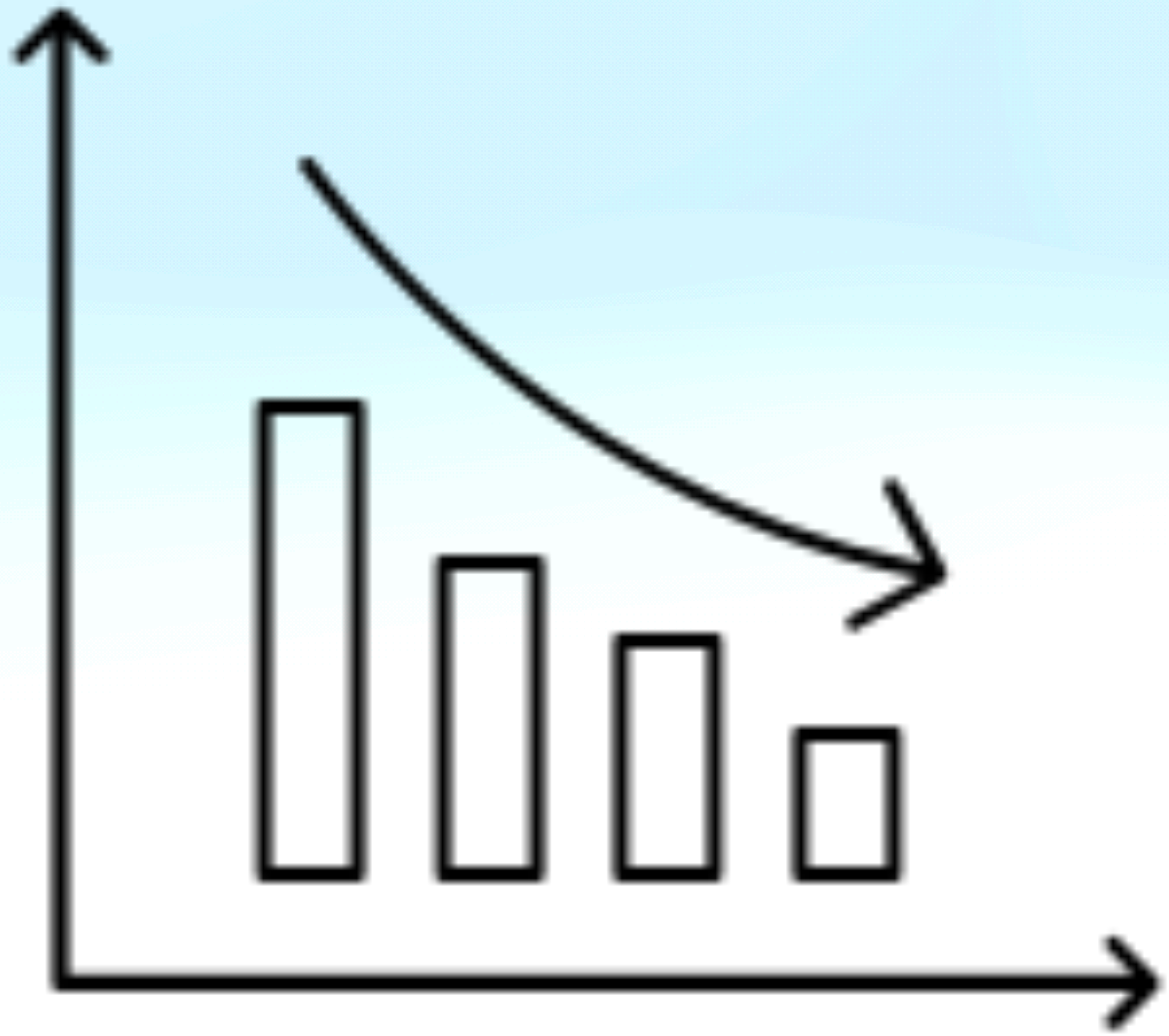
- Function for text preprocessing
- Input: List of text data
- Output: Preprocessed data
- Result: Cleaned and standardized text data ready for analysis

```
def text_preprocessing(data):

    preprocessed_data = []

    for TEXT in data:
        # Remove special characters and punctuation ma
        CLEANED_TEXT = re.sub(r'[^a-zA-Z0-9\s]', ' ',
        # Convert text to lowercase
        LOWERCASE_TEXT = CLEANED_TEXT.lower()
        # Tokenize the text
        TOKENS = word_tokenize(LOWERCASE_TEXT)
        # Remove stopwords from the list of tokens
        FILTERED_TOKENS = [word for word in TOKENS if
        # Lemmatize each word
        LEMMATIZED_TOKENS = [lemmatizer.lemmatize(wor

        preprocessed_data.append(LEMMATIZED_TOKENS)
```

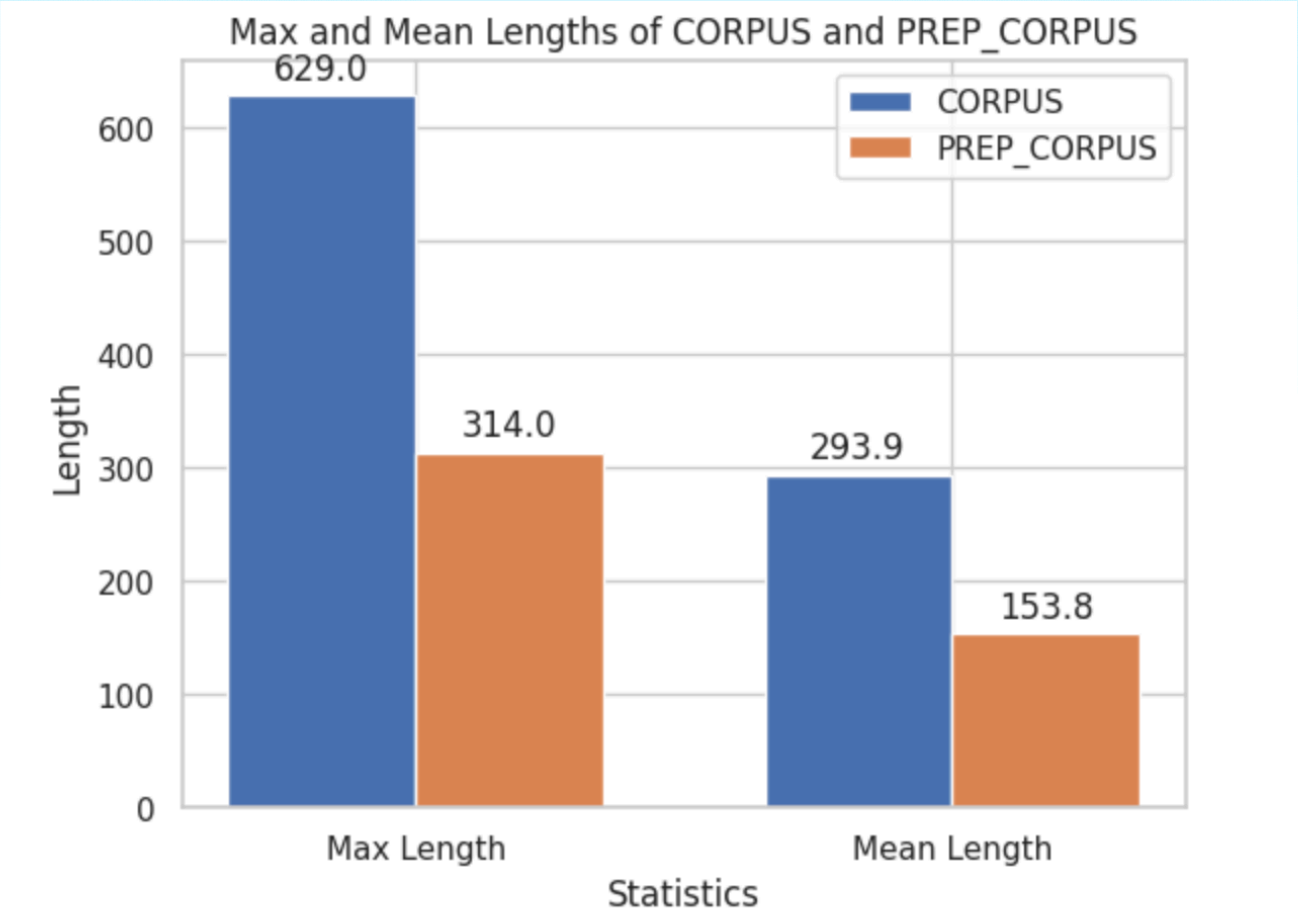
```
BEFORE PREPROCESSING.

CORPUS:
The bar chart below describes some changes about the percentage of people were born in Australia and who were born outside Australia living in urb

_ _ _ _ _ _ _ _ _

AFTER PREPROCESSING.

CORPUS:
['bar', 'chart', 'describes', 'change', 'percentage', 'people', 'born', 'australia', 'born', 'outside', 'australia', 'living', 'urban', 'rural', '
```

Text Length Statistics: Preprocessing Impact

- Original question and essay lengths word range: 46-125
- After preprocessing word range: 26-82
- Shorter and standardized text data



Text Length Statistics: Preprocessing Impact



Dataset Standardization & Vocabulary Construction

- Vocabulary construction
- Vocabulary => NumPy array and remove duplicates
- Mapping word => index
- Report max vocabulary size using <UNK> token
- Result: dataset standardized

Vector Representation

Converting Textual Data into Numerical Vectors

- Techniques for vector representation:
  - Positive Pointwise Mutual Information (PPMI)
  - Term Frequency-Inverse Document Frequency (TF-IDF)
- Enable quantitative analysis and modeling
- Introduction to GloVe Embeddings

PPMI Matrix

Constructing the Positive Pointwise Mutual Information Matrix

- Use co-occurrence counts with a specified window to build a co-occurrence matrix
- Memory efficiency: convert the co-occurrence matrix into a Compressed Sparse (CS) matrix
- Calculate PMI scores to identify meaningful word associations.
- Result: matrix facilitates analysis (word embeddings, semantic relationships)

$$\mathrm{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

TF-IDF Analysis

Understanding the TF-IDF Matrix and Vocabulary

- Matrix shape is (1435, 11750): 1435 essays (rows) and 11750 words (columns)
- Score = word importance within essays
- Feature names: displays a selection of unique words (language richness)

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

GloVe Embeddings

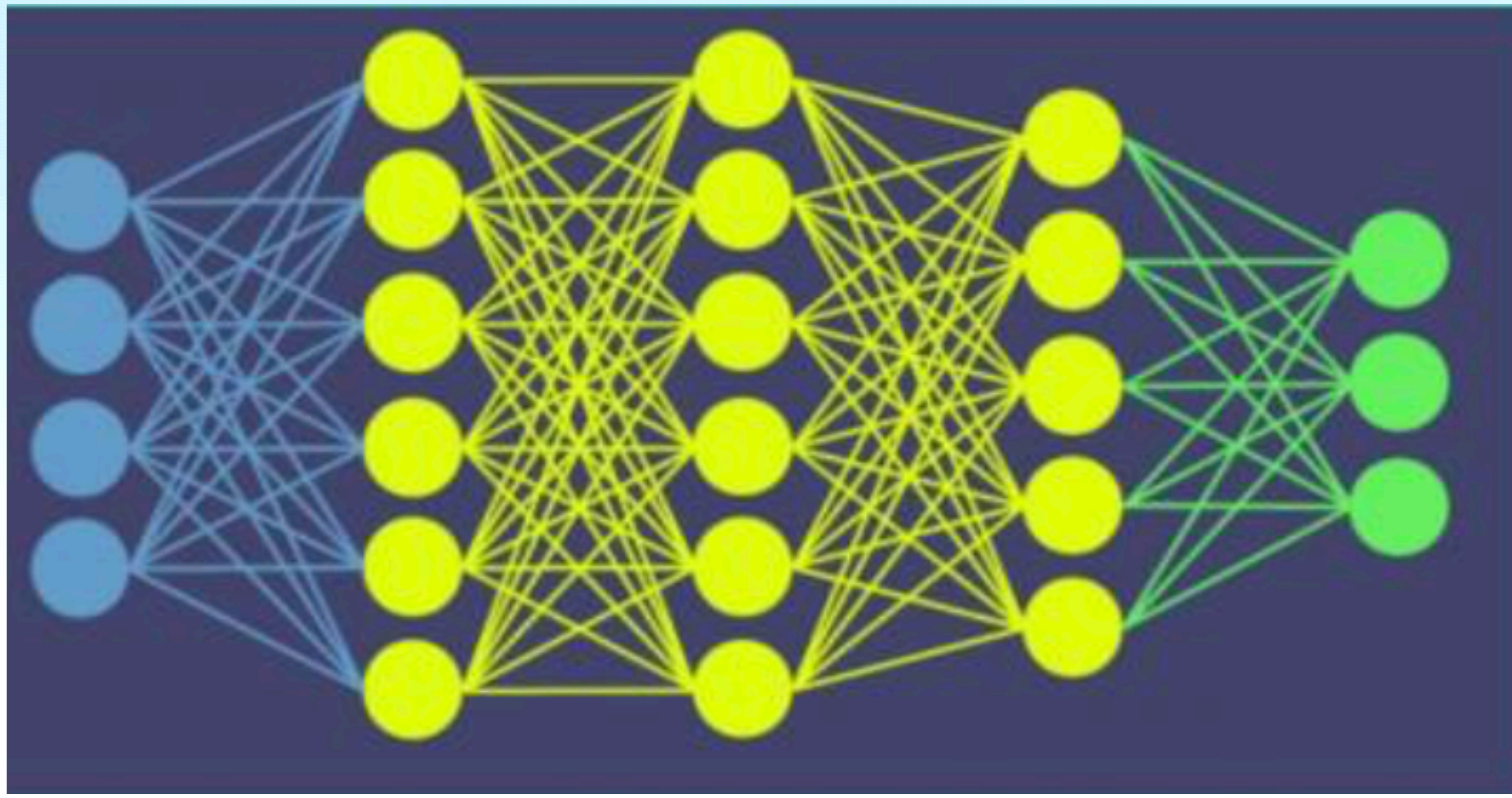
Loading and Utilizing Pre-Trained Word Vectors

- Load pre-trained word embeddings capturing word semantics and relationships

- Create dictionary: words = keys; vectors = values

- Embedding matrix: GloVe embeddings, embedding dimensions (100, 200, 300), vocabulary size as input

- Output: 3 embedded matrices with different dimensions



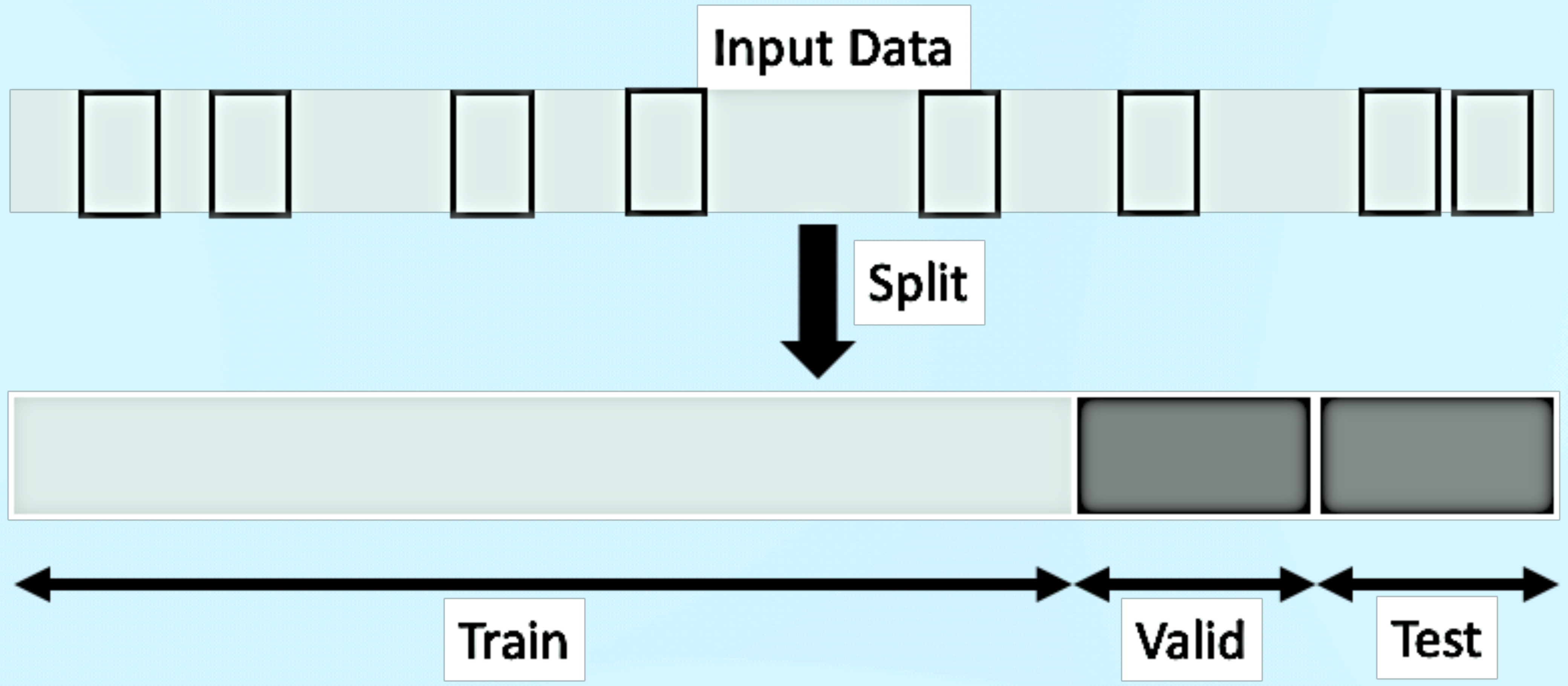
Data Transformation & Label Encoding

Preparing the Dataset for Machine Learning

- Numerical transformation: text-based —> numerical format
- Assignment of unique integers to words
- Features aggregation: combination of various features (task type, linguistic features…)
- Use label encoder to encode scores into categorical features
- Resulting dataset: 1435 samples and 342 features for machine learning analysis

Data Splitting Strategy

Ensuring Robust Model Evaluation

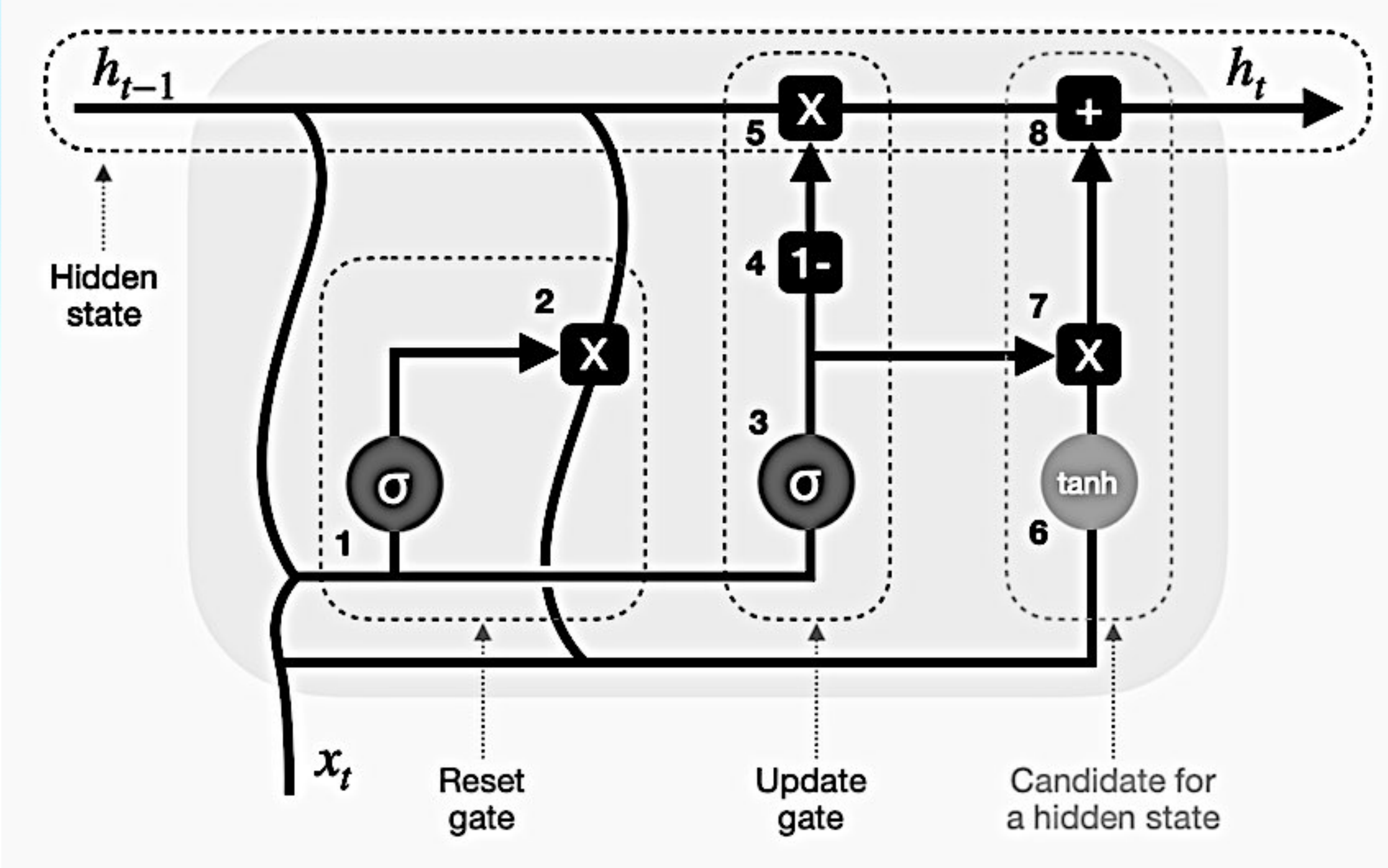


- Divide dataset: 90% training; 10% testing
    - Facilitate model training and initial performance assessment
- Further split training set: 81% training subset; 9% validation subset
    - Essential for fine-tuning and optimizing hyperparameters
- Reliable and unbiased assessment of models applied to this dataset

Neural Network Architecture

Designing a Model for Essay Rating

- Input layer accepts essay representation
- Bidirectional GRU layers analyze in two directions essays with activation functions and kernel initializers
- Convert GRU outputs and concatenate task features
- Dense layers to add non-linearity and transform features



Neural Network Architecture

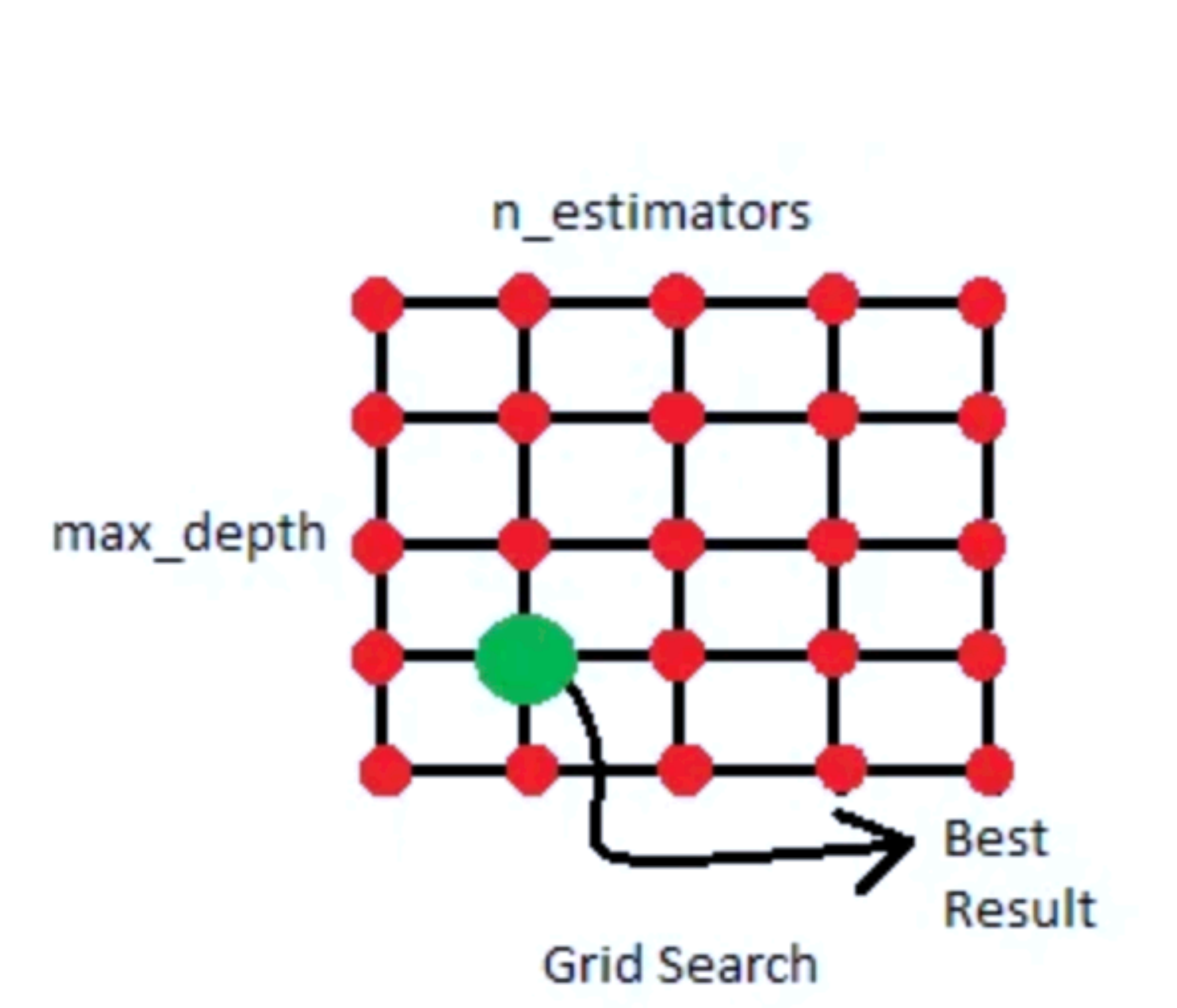
Designing a Model for Essay Rating

- Batch normalization to stabilize training and reduce overfitting
- Output layer: 5 units for rating predictions using softmax
- Minimize loss (loss function) with optimization
- Assemble the architecture into a trainable model

Hyperparameter Tuning

Optimizing neural network with GridSearchCV

- Initialize model with Keras created, hyperparameters, embedding matrix and number of epochs
- GridSearchCV : explores hyperparameter combinations to enhance model performance and tests different functions (relu, sigmoid, tanh) for a GRU layer.
- Goal: find the best hyperparameter combination for maximum training accuracy



Hyperparameter Tuning

Optimizing the neural network

- Lista hyperparameter
- I migliori
- I range

Essay Classification Models

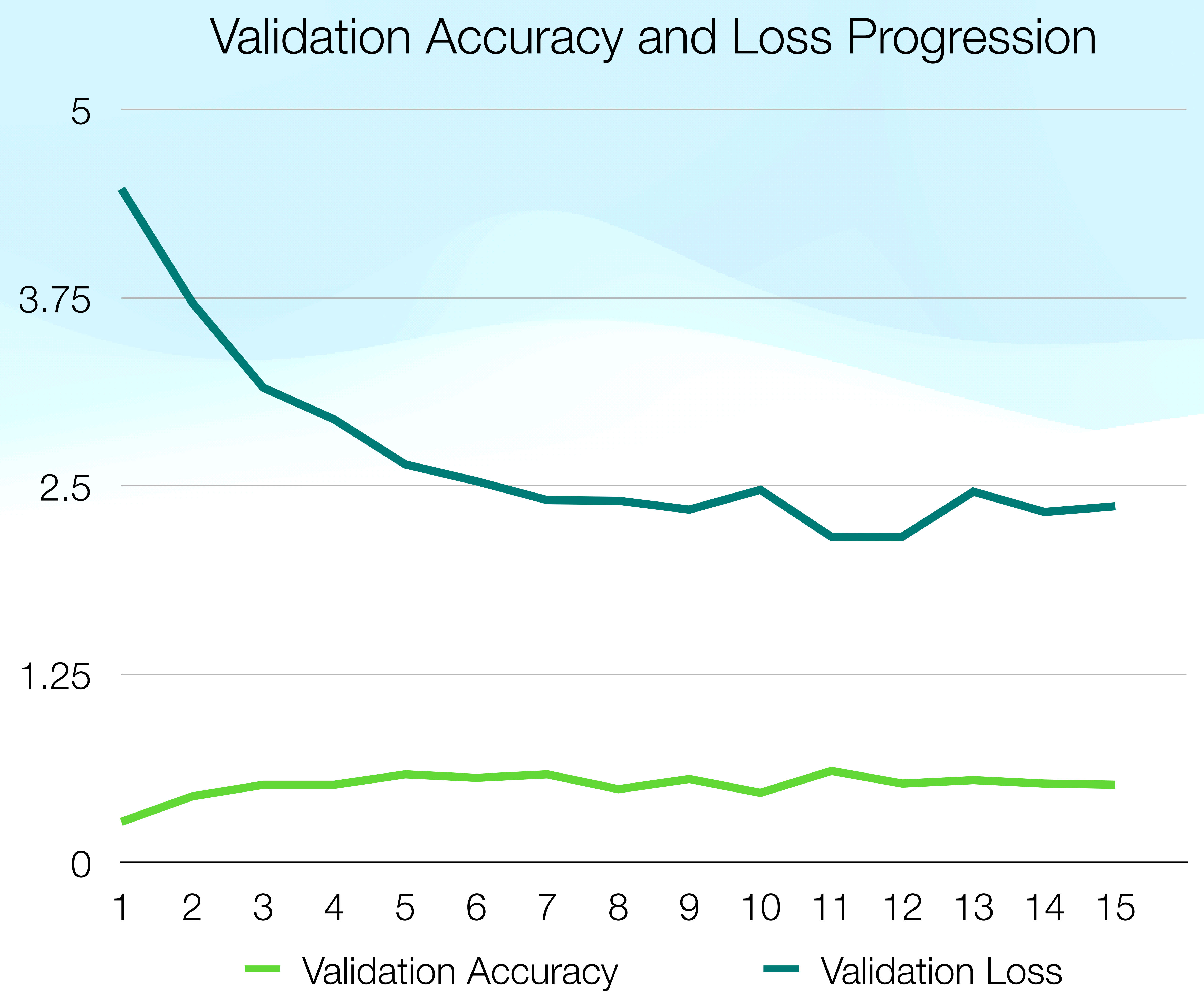
Comparing NLP Models

- Three distinct models compared: GloVe, TF-IDF, PPMI, and BERT
- Comprehensive analysis to determine their effectiveness

Essay Classification Models

The GloVe Model

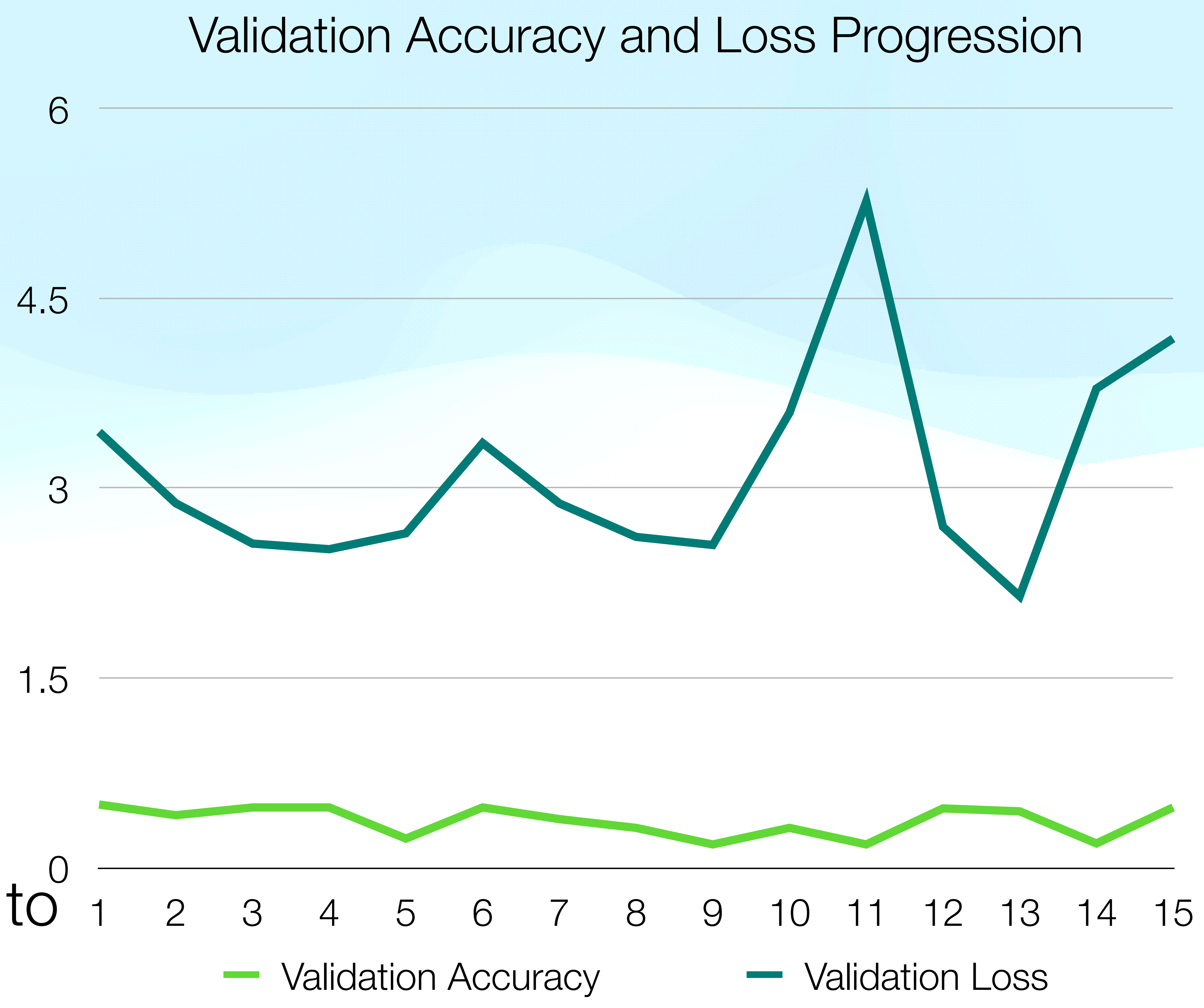
- Extensive training: 15 Epochs
- Decrease in validation loss
- Stabilized validation accuracy
  => consistently predicting data correctly
- Result: strong learning capability,
  moderate generalization on validation set



Essay Classification Models

The TF-IDF Model

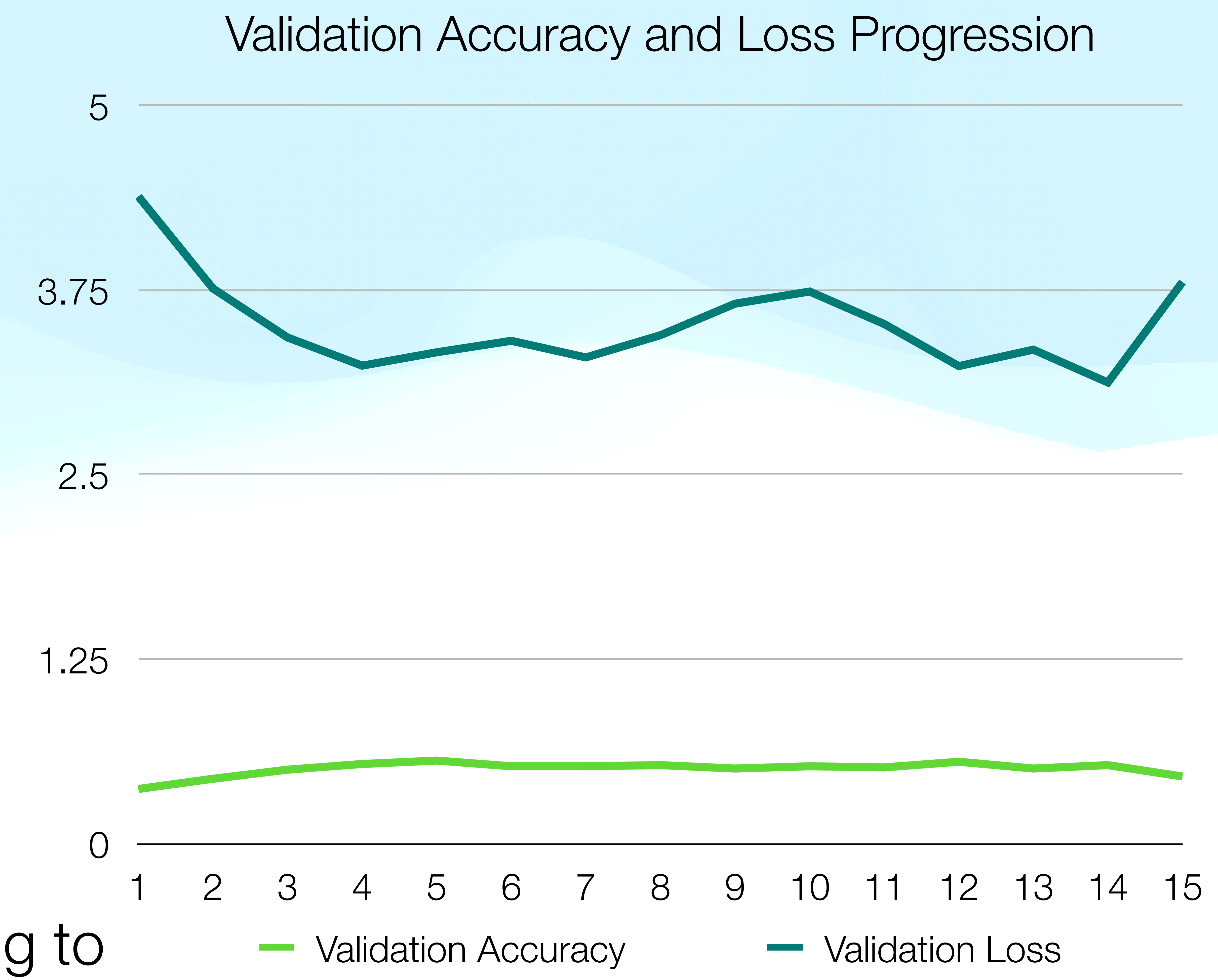
- Extensive training: 15 Epochs
- Training consistent improvement
- Validation accuracy stabilized at 55.38%
- Fluctuating validation loss
- Overfitting
- Result: strong learning capability, struggling to generalize on unseen data



Essay Classification Models

The PPMI Model

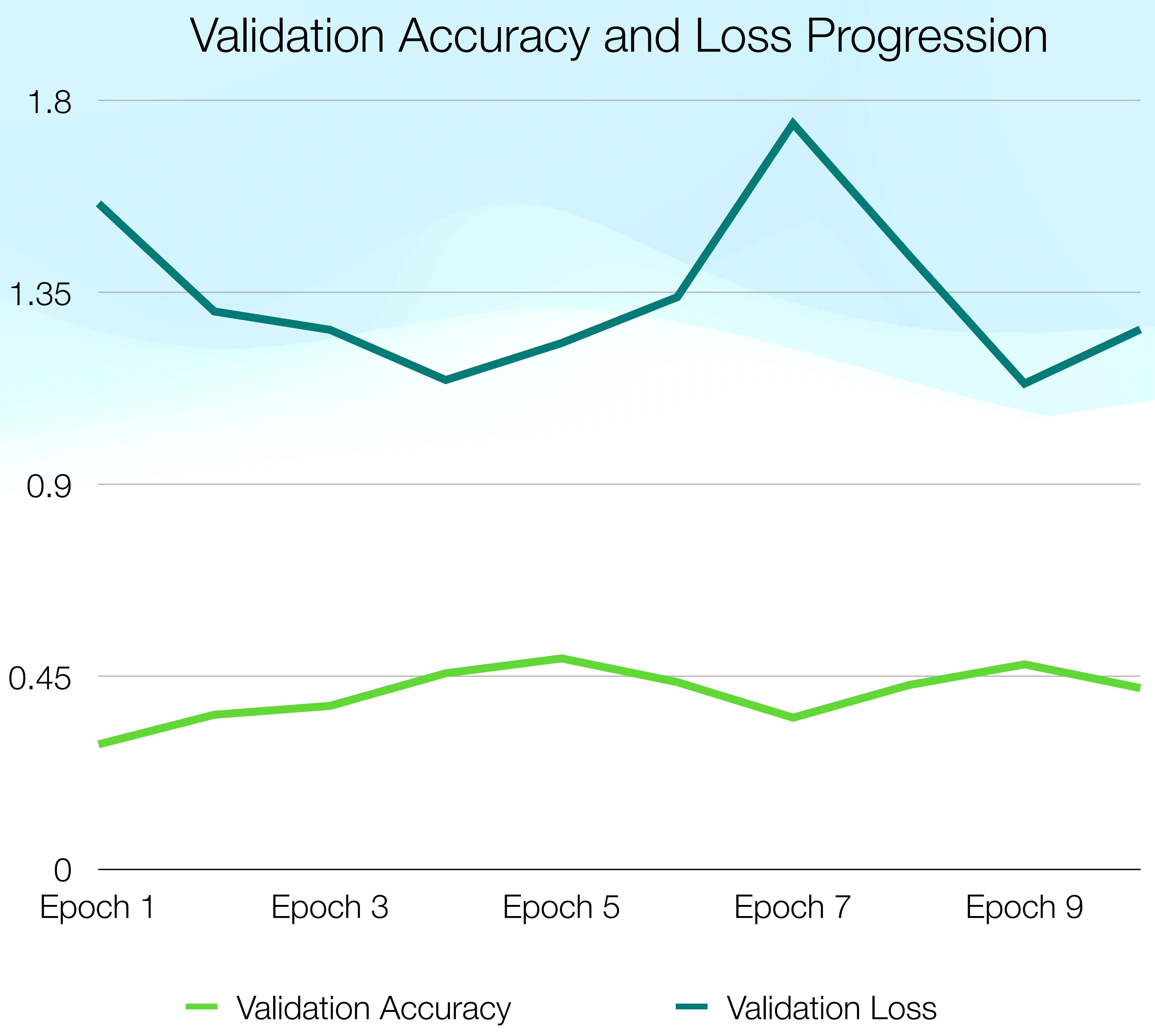
- Extensive training: 15 Epochs
- Training promising trend: accuracy increasing, loss decreasing
- Validation accuracy stabilized at ≈ 53%
- Fluctuating validation loss
- Overfitting
- Result: strong learning capability, struggling to generalize on unseen data



Essay Classification Models

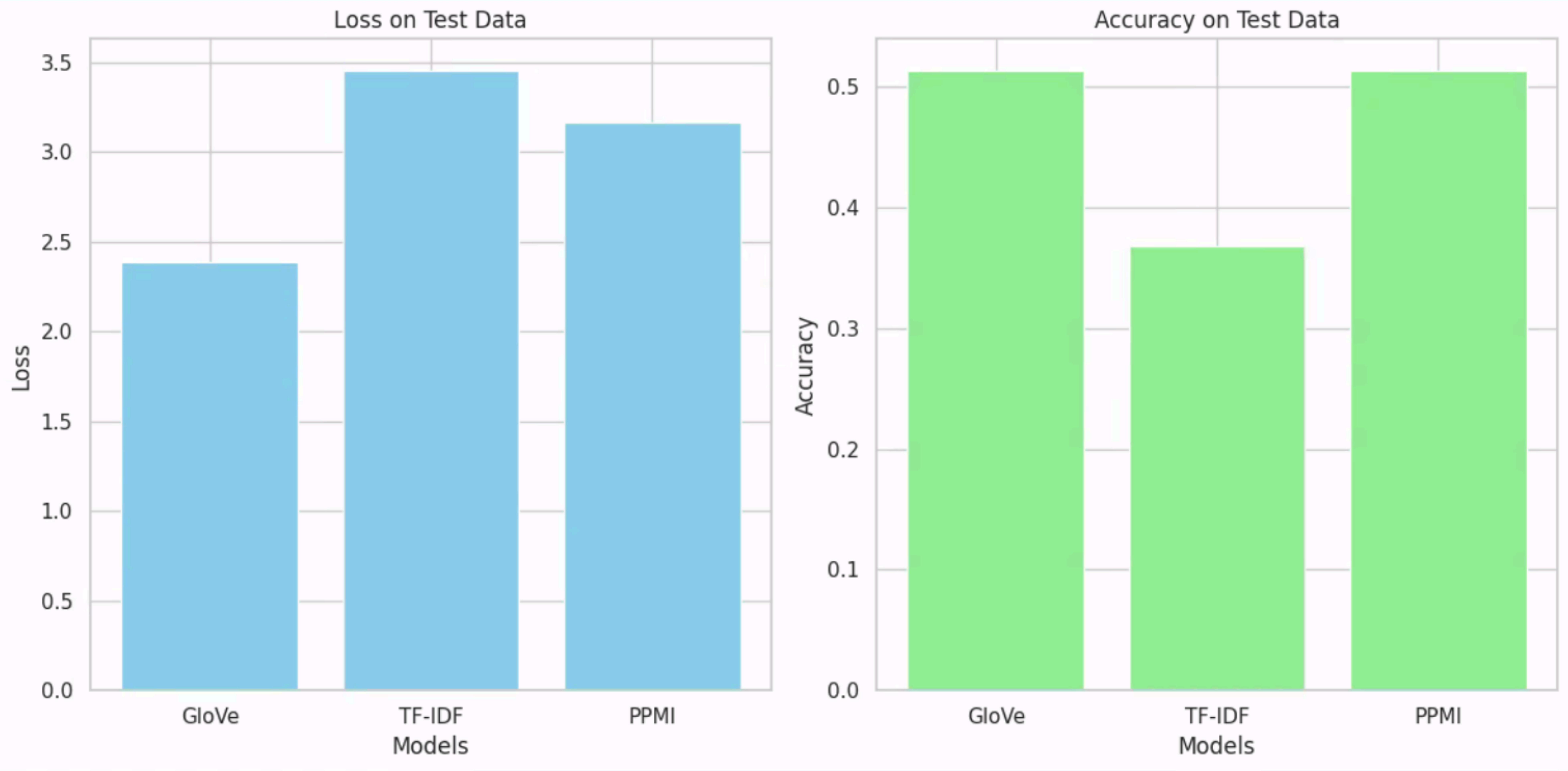
The BERT Model

- Powerful transformer-based model
- Executive training: 10 epochs
- Increasing accuracy
- Decreasing loss
- Training process converges over time => effective learning
- Result: learning and adaptation



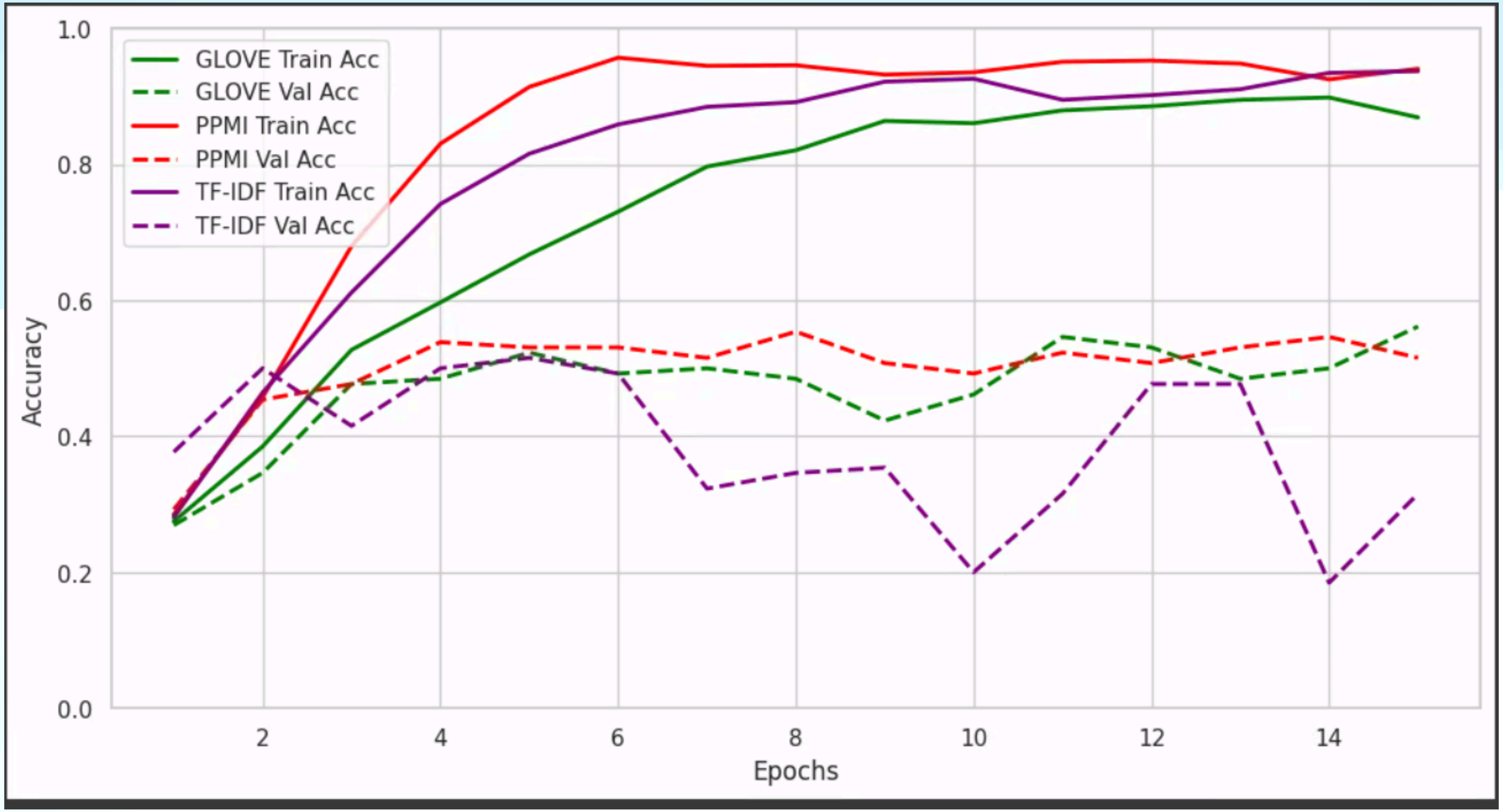
Model Evaluation

Analyzing Essay Classification Models



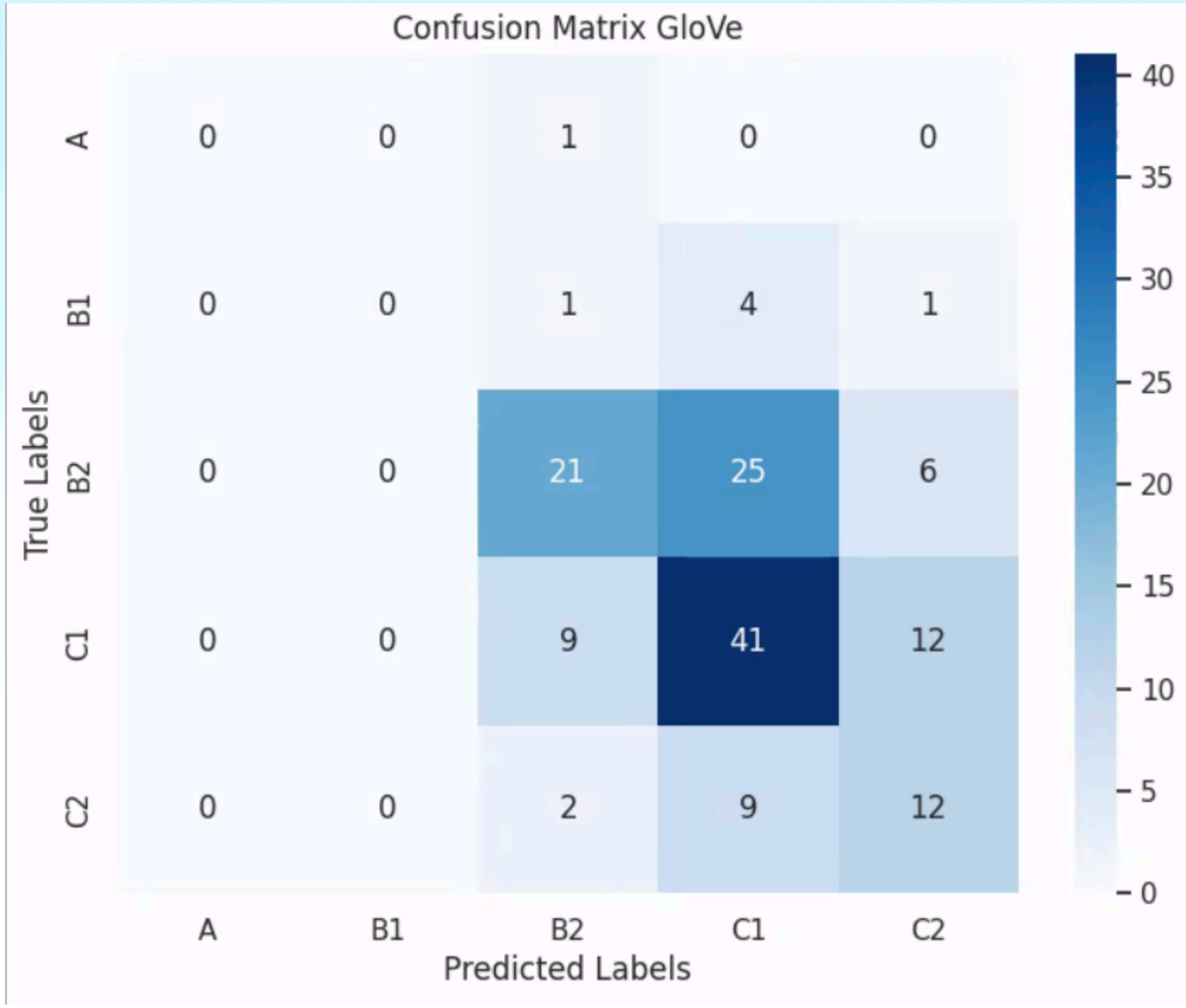
Model Evaluation

Analyzing Essay Classification Models



Model Evaluation

Analyzing Essay Classification Models



Conclusion & Key Takeaways

Evaluating Essays using NLP and ML

- GloVe, TF-IDF, PPMI, and BERT models: architectures, optimization, training progress, validation accuracy and loss
    - Strong performance in assessing essay quality and CEFR score prediction
    - Improved accuracy and reduced loss during training and validation
    - Effectively capture linguistic differences, valuable insights
    - Potential to enhance the efficiency of essay evaluation and language assessment
    - Best suitable model: GloVe