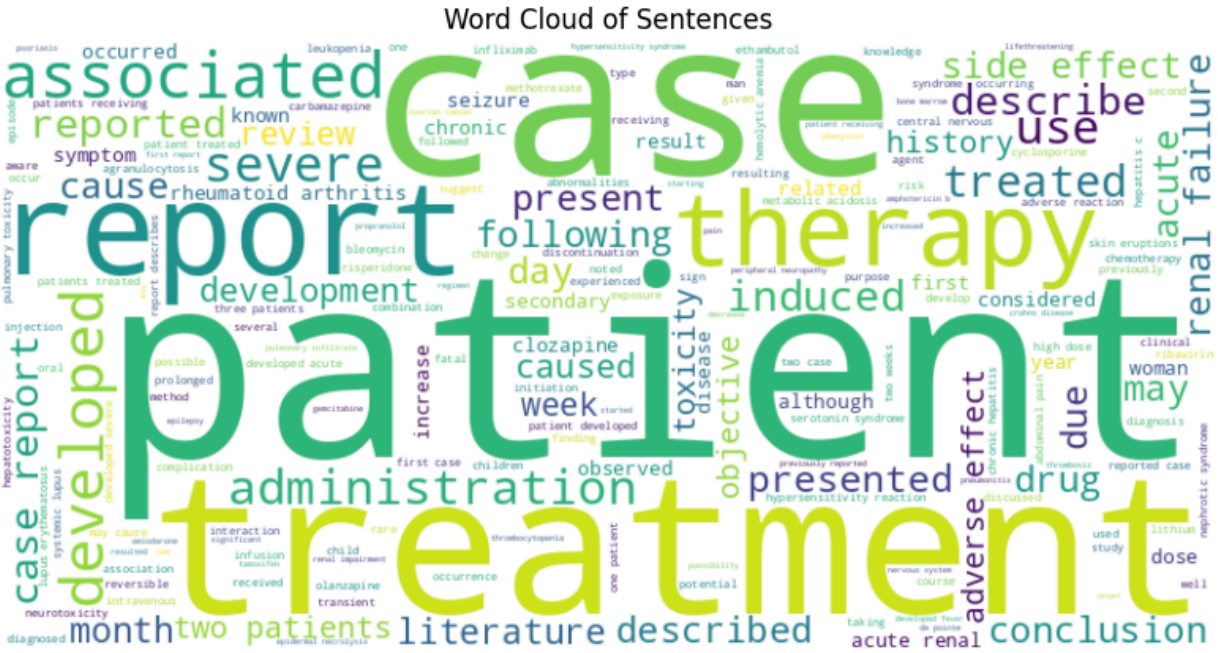
Natural Language Understanding

Assignment- II

REPORT- *Detecting Adverse Drug Events using Long Short-Term Memory (LSTM)*

Marri Bharadwaj	B21CS045



Adverse Drug Events (ADEs) are a major global public health problem that raise healthcare expenditures and cause morbidity and death in patients. Improving medication regulatory procedures and patient safety need early detection and mitigation of ADEs. Deep learning models like Long Short-Term Memory (LSTM) have shown encouraging performance in ADE identification and other natural language processing tasks in recent years. The objective of my project is to investigate and use deep learning architectures for the purpose of identifying drugs and their side effects using clinical text data.

Comprehensive Literature reviews

Adverse drug event detection using natural language processing: A scoping review of supervised learning methods

The academic paper presents a scoping review of supervised learning methods for detecting Adverse Drug Events (ADEs) using Natural Language Processing (NLP) within electronic health records of hospitalised patients. The study aims to fill the knowledge gap in qualitative assessment and critical appraisal of NLP methods for ADE monitoring in hospitals, providing directions for future research and practice. It includes a critical examination of 29 articles that met the inclusion criteria, focusing on tasks like named entity recognition and relation extraction/classification, and highlights the importance of clinical involvement in the studies.

The paper emphasises the need for a system that can routinely, rapidly, and at scale detect ADEs in hospitalised patients to improve medication safety. It discusses the challenges faced in ADE detection, such as the complexity of medical treatment and multimorbidity, and the limitations of current methods like voluntary reporting and pharmacist-led patient chart reviews. The review suggests that multiple NLP modelling approaches are suitable, with Long Short Term Memory and Conditional Random Field methods being the most commonly used, and calls for future research to focus on semi-automated methods to reduce manual annotation effort and examine the implementation of NLP methods in practice.

Delving deeper into the scoping review on adverse drug event (ADE) detection using natural language processing (NLP), I get valuable insights into the application of NLP methods for ADE detection within the context of pharmacovigilance and hospital monitoring.

1. Introduction

- The primary goal is to reduce ADEs by establishing a system that enables hospitals to routinely, rapidly, and at scale monitor ADE occurrences.
- NLP, a computerised approach for analysing text data, holds promise for ADE detection within the framework of electronic health records (EHRs).

2. Methods

- The scoping review included articles where NLP was applied to detect ADEs in clinical narratives within EHRs of inpatients.
- Both quantitative and qualitative data items related to NLP methods were extracted and critically appraised.
- Eligibility screening: Out of 1,065 articles, 29 met the inclusion criteria.

3. Key Findings

i) Tasks and Approaches:

- Named Entity Recognition (NER): The most frequent task involved identifying entities (e.g., drugs, symptoms) related to ADEs (17 out of 29 studies).
- Relation Extraction/Classification: Another common task was extracting or classifying relations between entities (15 out of 29 studies).
- Clinical Involvement: Only nine studies (31%) reported clinical involvement in the NLP process.

ii) NLP Modelling Approaches:

- *Long Short Term Memory (LSTM)* and *Conditional Random Field (CRF)* methods were commonly used.
- While overall system performance was high, there was a significant drop in performance when predicting the *ADE entity* or *ADE relation class*.

iii) Best Practices:

- When annotating corpora, treating an ADE as a *relation between a drug and non-drug entity* seems to be the best practice.
- Future research should focus on *semi-automated methods* to reduce manual annotation efforts and explore the implementation of NLP methods in practice.

4. Conclusion and Future Directions

- The scoping review highlights the feasibility of creating NLP-based systems for ADE detection in hospitalised patients.
- However, challenges remain, especially related to *data preparation* (annotation and pre-processing of text).

- Future research should address these challenges and explore ways to enhance the practical implementation of NLP methods in clinical settings.

Thus, the academic paper provides a comprehensive review of the use of Natural Language Processing (NLP) for detecting Adverse Drug Events (ADEs) in clinical narratives within Electronic Health Records (EHRs). Summarising; the key insights are:

➢ *NLP for ADE Detection*- The review highlights the potential of NLP to support hospitals in monitoring ADEs by analysing text data from EHRs.

➢ *Supervised Learning Methods*- It emphasises the role of supervised learning methods in NLP and suggests that multiple NLP modelling approaches, including Long Short Term Memory and Conditional Random Field methods, are suitable for ADE detection.

➢ *Challenges and Recommendations*- The paper identifies challenges such as the need for detailed qualitative assessment and critical appraisal of NLP methods. It recommends focusing on semi-automated methods to reduce manual annotation effort and examining the implementation of NLP methods in practice.

➢ *Clinical Involvement*- The review notes the importance of involving clinicians in the NLP process to ensure high-quality annotated data and to align NLP models with clinical practice.

MADEx: A System for Detecting Medications, Adverse Drug Events, and their Relations from Clinical Notes

Adverse drug events (ADEs) pose a significant burden on healthcare systems worldwide, contributing to increased costs, longer hospital stays, and higher mortality rates. With a majority of ADEs being preventable through early detection and intervention, there is a pressing need for effective methods to identify

medications, adverse effects, and their relationships from the wealth of unstructured clinical data in electronic health records (EHRs). Natural language processing (NLP) techniques are crucial for extracting structured information from the narrative clinical text where much of the relevant detail is recorded. However, developing high-performing clinical NLP systems for tasks such as named entity recognition of medications and ADEs, as well as extraction of their semantic relationships, remains a challenge. In this paper, we present MADEx, an integrated machine learning system that combines state-of-the-art deep learning and traditional methods to accurately detect medications, adverse drug events, and their relations from clinical notes. Through rigorous evaluation on a benchmark dataset from the MADE 1.0 challenge, I'm demonstrated the effectiveness of our approach and provided insights into the key design choices. Their work advances clinical NLP toward enabling large-scale pharmacovigilance and drug safety monitoring from unstructured EHR data to improve patient outcomes.

The paper begins by highlighting the importance of early detection of adverse drug events (ADEs) from electronic health records (EHRs) to support pharmacovigilance and drug safety surveillance. It cites studies showing the high incidence rates of ADEs ranging from around 3-8% across different countries, and the associated increase in healthcare costs, length of hospital stay, and in-hospital mortality rates. A large majority of ADEs are stated to be preventable, underscoring the importance of early ADE detection.

The key challenge in using narrative clinical text for ADE detection is that much of the detailed information is documented in an unstructured manner, necessitating natural language processing (NLP) techniques to extract structured information.

The authors review prior NLP challenges focused on clinical named entity recognition (NER) and relation extraction:

- The *i2b2 challenges* in 2010 and 2012 examined extracting problems, treatments, tests, and their relations like treatment relations, temporal relations between medical events.

- The 2014 SemEval challenge focused on analysis of clinical text.

- The ShARE/CLEF eHealth challenges looked at detecting disorder names from notes.

- The 2015 BioCreative V challenge had a task on extracting chemical-disease relations from biomedical literature, related to ADE extraction.

They note that most top clinical NER systems use machine learning models like *conditional random fields (CRFs)*, *support vector machines (SVMs)*, *structured SVMs* and ensembles. Recently, deep learning with *recurrent neural networks (RNNs)* and *convolutional neural networks (CNNs)* are emerging as state-of-the-art.

For relation extraction from medical texts, supervised methods like SVMs, kernel methods, tree kernels as well as semi-supervised approaches have been employed, with machine learning again being top performers.

The University of Massachusetts MADE 1.0 challenge in 2018 was the first focused on extracting medications, ADEs and their relations from EHR data, motivating this study. It had 3 sub-tasks:

1) Clinical NER to extract medications, ADEs, attributes
2) Relation extraction between concepts
3) An integrated task combining 1 and 2

I have found the following key findings:
The authors developed MADEx, a machine learning system with two modules:

1) Clinical NER module using a recurrent neural network (RNN) with long short-term memory (LSTM) and conditional random field (CRF) layer.

2) Relation extraction module using support vector machines (SVMs)

For NER, their LSTM-CRF model achieved a *top 3 F1-score of 0.8233* in the MADE challenge, outperforming a baseline CRF model (0.7250 F1). Here, the key aspects were:

- Used bi-directional LSTMs with character and word embeddings
- Novel 2-stage training procedure combining validation set to retrain
- Improved precision and recall over CRF

For relation extraction, their SVM model slightly *outperformed a random forest (RF) model on the test set (0.8466 vs 0.8337 F1)* with the help of:

- Used attributes like local context, distance, n-grams, entity types as features
- Developed separate classifiers for single-sentence and cross-sentence relations

Their integrated MADEx system combining LSTM-CRF NER and SVM relation extraction achieved comparable performance to the best systems (0.6125 F1 on integrated task).

The following methodologies were used:

i) NER Module:

- Bi-directional LSTM layers for character and word representations
- Character embeddings randomly initialised and updated during training
- Pre-trained word embeddings provided by challenge organisers
- Novel 2-stage training:
    - Optimise on validation set
    - Retrain on full set using Stage-1 parameters
- CRF layer on top for sequence labelling

ii) Relation Extraction Module:

- Heuristic rules to generate candidate concept pairs from sentences/consecutive sentences
- Features: local context, concept distances, n-grams, entity types
- SVM and RF classifiers, optimised via cross-validation
- Separate models for single-sentence and cross-sentence relations

iii) Integrated Pipeline: Combined LSTM-CRF NER and SVM Relation Extraction modules

Datasets used were:

The MADE1.0 dataset provided by UMass contained *1,089 de-identified clinical notes* with manual annotations:

- 79,114 annotated entities (medications, ADEs, attributes)
- 27,175 annotated relations
- Training set: 876 notes
- Test set: 213 notes

Performance Metrics in Detail:

- Named Entity Recognition:
  - Micro-averaged F1, Precision, Recall (strict matching of entity span and type)
  - LSTM-CRF (RNN-2) F1: 0.8233, Prec: 0.8149, Rec: 0.8318
  - Baseline CRF F1: 0.7250
  - Performance broken down by entity type (e.g. lower for ADE, Indication)

- Relation Extraction:
  - Micro-averaged F1, Precision, Recall over all relation types
  - SVM F1: 0.8466, Prec: 0.8491, Rec: 0.8441
  - RF F1: 0.8337
  - Performance by relation type (e.g. lower for adverse, reason, duration)

- Integrated Task:
  - Micro-averaged F1
  - MADEx (RNN-2 + SVM) F1: 0.6125
  - RNN-2 + RF F1: 0.6033

The authors' MADEx system achieved top 3 performance for NER and was comparable to the best systems for relation extraction and integrated tasks in the MADE1.0 challenge.

Thus, my insights gained from the literature review presented in this paper are:

- ➢ *Challenges in using clinical text for ADE detection*- The narrative and unstructured nature of clinical notes poses a significant challenge for extracting structured information required for ADE detection. Natural language processing (NLP) is critical for this task, but developing high-performing clinical NLP systems remains difficult.

- ➢ *Prior work and challenge*- The paper reviews prior NLP challenges and research efforts focused on clinical named entity recognition (NER) and relation extraction, which are key tasks for ADE detection. Traditional machine learning models like conditional random fields (CRFs), support vector machines (SVMs), and ensembles have been widely used, with some of the top performances.

- ➢ *Emergence of deep learning methods*- Recent years have seen the emergence of deep learning methods like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) achieving state-of-the-art results for clinical NER. Their ability to automatically learn representations from data shows promise over hand-crafted features.

- ➢ *First challenge focused on ADE extraction*- The 2018 MADE 1.0 challenge organised by UMass was the first open challenge focused specifically on extracting medications, adverse drug events, and their relations from EHR data. This provided a benchmark dataset and motivated this study.

- ➢ *Novel system and training approach*- The authors developed MADEx, a novel machine learning system that combined an LSTM-CRF model for NER and an SVM model for relation extraction. They proposed a two-stage training procedure for the LSTM-CRF that improved performance over the typical approach.

- ➢ *Top performance and insights*- MADEx achieved top 3 performance for NER and was comparable to the best systems for relation extraction in the

MADE 1.0 challenge. The study provides insights into the effectiveness of deep learning for NER, comparing it to traditional CRFs. It also analyses the performance differences across entity/relation types.

- ***Remaining challenges***- While MADEx performed well, the integrated task of extracting entities and relations together remains challenging (best F1 ~0.61). The study identifies areas like encoding external knowledge, handling complex entities/relations, and exploring other deep learning models as important future directions.

Summarising, this literature review highlights the importance of clinical NLP and ADE detection, prior work in the field, the emerging use of deep learning, the novelties of the MADEx system, its performance on a benchmark dataset, and potential areas for further research in this critical healthcare application.

Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records

Adverse drug events (ADEs) pose a significant risk to patient safety and are a leading cause of deaths in the United States. Early and accurate detection of ADEs is therefore crucial for timely assessment, mitigation, and prevention of severe, potentially fatal incidents related to drug complications. Electronic health records (EHRs) contain valuable unstructured clinical narratives that can provide critical information regarding ADEs experienced by patients. However, extracting this information from EHR text is a challenging natural language processing (NLP) task due to the inherent complexity and noise present in clinical notes. EHRs are replete with domain-specific abbreviations, alphanumeric patterns, multi-word medical entities spanning varying lengths, as well as ambiguity in semantics where the same phrase can represent different entity types like ADEs, indications, or signs/symptoms depending on the context. Consequently, there has been growing research interest in developing advanced NLP techniques, especially leveraging deep learning, to automatically identify mentions of ADEs and related medical concepts directly from the unstructured EHR data. This literature review analyses

one such study that presents a high-performing deep learning system called DLADE (Dual-Level Embeddings for Adverse Drug Event Detection) tailored for named entity recognition of ADEs and associated entities from EHR text.

The paper starts by highlighting the importance of early detection of adverse drug events (ADEs) from electronic health records (EHRs) for effective pharmacovigilance and prevention of severe, potentially fatal incidents. It outlines the major challenges in processing unstructured EHR text like presence of abbreviations, numbers, multi-word entities spanning 1-7 words, ambiguity between entity types like ADEs vs indications vs signs/symptoms.

MADE 1.0 NLP Challenge

The authors developed their system DLADE for the Named Entity Recognition (NER) task of the MADE 1.0 NLP challenge on detecting medications and ADEs from EHRs. The challenge provided 1089 de-identified annotated EHR notes, with 876 notes released for training systems. Entities annotated included medication names, dosages, routes, frequencies, durations, ADEs, indications and signs/symptoms. Evaluation was done on a hidden test set using micro-averaged phrase-level F1 score.

DLADE System Architecture

- Preprocessing:
  - Rule-based tokenization of notes into sentences and words while recording character offsets to handle text noise
  - IOB tagging scheme used to represent multi-word entities

- Model:
  - Three layer deep learning architecture:
    1) Bi-LSTM on character sequences to obtain character-level word representations
    2) Bi-LSTM using concatenated character and pre-trained word embeddings for context representations
    3) Feed-forward and linear-chain CRF output layer

- Dual embeddings by concatenating learned 100D character vectors with 200D pre-trained word vectors
- 100D and 300D hidden layers for character and word-level bi-LSTMs respectively
- Dropout of 0.5 for regularisation

- Training:
  - 10% of training sentences used for validation and early stopping
  - Batch size of 20 sentences, variable lengths with masking
  - Adam optimizer with learning rate 0.001

- Experimental Results
  - Test set: 213 EHR notes from challenge
  - Achieved 0.8413 micro F1, with 0.8373 precision and 0.8454 recall
  - F1 scores per entity type ranged from 0.6351 (ADE) to 0.9239 (Route)
  - Statistically significant improvement over word embedding-only baseline
  - Largest 11.4% F1 gain for "Duration" entities like "four cycles", "14 days" using dual embeddings

- Error Analysis
  - Missed multi-word entity boundaries
  - Partially extracted phrases with medical+non-medical text
  - Failed on rare abbreviations like "SIL cytology"
  - Confusion between ambiguous ADE vs indication vs sign/symptom phrases

The authors provide a comprehensive description of their innovative deep learning solution integrating complementary word/character representations and sequence labelling models. Detailed analysis sheds light on remaining challenges around multi-word medical entities, ambiguous semantics, and rare abbreviations. Overall, this work demonstrates a highly performant NER system tailored for the complex unstructured EHR domain.

Analysis of the insights gained from the literature review presented in the paper:

➢ *ADE Detection Importance and Challenges*-

- The review underscores the criticality of early ADE detection in mitigating potential harm to patients and enhancing pharmacovigilance.
- Challenges in ADE detection from Electronic Health Records (EHRs) are highlighted, including unstructured text, medical abbreviations, misspellings, and contextual ambiguity.

➢ *MADE 1.0 NLP Challenge Overview*-

- The MADE1.0 NLP Challenge is introduced as a platform aimed at advancing ADE detection techniques using NLP applied to EHRs.
- Tasks within the challenge, such as Named Entity Recognition (NER), Relation Identification (RI), and Integrated Task (IT), are outlined.

➢ *DLADE System Architecture*-

- The paper introduces the DLADE system, designed specifically for Task 1 (NER) of the MADE 1.0 Challenge.
- Detailed methodologies regarding tokenization techniques, model architecture (combining bi-LSTM and CRF), and data preprocessing are provided.
- The integration of character-level word representation, context representation, and output prediction using CRF is highlighted.

➢ *Experimental Setup and Results*-

- The experimental setup, including hyperparameter settings, data set used for training and testing, and model evaluation criteria, is meticulously described.
- Performance metrics such as Precision, Recall, and F1-score are presented, demonstrating the efficacy of the DLADE system in ADE detection.
- Detailed insights into the performance of individual entities (Drug, Indication, Frequency, Severity, etc.) are provided, offering a granular understanding of system capabilities.

➢ *Error Analysis of DLADE System*-

- An in-depth error analysis is conducted to identify the sources of errors in the NER system.
- Challenges such as entity span across multiple words, mixture of medical and non-medical text, rare occurrence of medical abbreviations, and ambiguity in entity labels (Indication, ADE, and SSLIF) are thoroughly examined.
- Specific instances of misclassifications are analysed, providing valuable insights into the system's limitations and areas for improvement.

➢ ***Conclusion and Implications***-

- The review concludes by affirming the effectiveness of the DLADE system in addressing the challenges of ADE detection from unstructured EHR text.
- Implications for advancing ADE detection techniques, leveraging deep learning architectures, and incorporating dual-level embeddings are discussed.

Overall, the literature review offers a comprehensive understanding of the landscape of ADE detection from academic papers, providing detailed insights into methodologies, challenges, performance metrics, and implications for future research.

—----------

Designing LSTM model

To begin with, I convert the given ADE (Adverse Effects) dataset from a .txt file to a much more organised csv file and displayed it in the Dataframe format as described in detail below-

	PubMed-ID	Sentence	Adverse-Effect	Begin Offset AE	End Offset AE	Drug	Begin Offset Drug	End Offset Drug
0	10030778	Intravenous azithromycin-induced ototoxicity.	ototoxicity	43	54	azithromycin	22	34
1	10048291	Immobilization, while Paget's bone disease was...	increased calcium-release	960	985	dihydrotachysterol	908	926
2	10048291	Unaccountable severe hypercalcemia in a patien...	hypercalcemia	31	44	dihydrotachysterol	94	112
3	10082597	METHODS: We report two cases of pseudoporphyri...	pseudoporphyria	620	635	naproxen	646	654
4	10082597	METHODS: We report two cases of pseudoporphyri...	pseudoporphyria	620	635	oxaprozin	659	668
...	...	...	...	...	...	...	...	...
6816	998323	Lithium treatment was terminated in 1975 becau...	lithium intoxication	531	551	Lithium	479	486
6817	998323	Lithium treatment was terminated in 1975 becau...	lithium intoxication	531	551	lithium	531	538
6818	9988365	Eosinophilia caused by clozapine was observed ...	Eosinophilia	795	807	clozapine	818	827
6819	9988365	Eosinophilia has been encountered from 0.2 to ...	Eosinophilia	76	88	clozapine	131	140
6820	9988365	Successful challenge with clozapine in a histo...	eosinophilia	61	73	clozapine	35	44

6821 rows × 8 columns

```
data.head()
```

	PubMed-ID	Sentence	Adverse-Effect	Begin Offset AE	End Offset AE	Drug	Begin Offset Drug	End Offset Drug
0	10030778	Intravenous azithromycin-induced ototoxicity.	ototoxicity	43	54	azithromycin	22	34
1	10048291	Immobilization, while Paget's bone disease was...	increased calcium-release	960	985	dihydrotachysterol	908	926
2	10048291	Unaccountable severe hypercalcemia in a patien...	hypercalcemia	31	44	dihydrotachysterol	94	112
3	10082597	METHODS: We report two cases of pseudoporphyri...	pseudoporphyria	620	635	naproxen	646	654
4	10082597	METHODS: We report two cases of pseudoporphyri...	pseudoporphyria	620	635	oxaprozin	659	668

```
Shape of the dataset: (6821, 8)

Number of unique values in each column:
 PubMed-ID            1644
Sentence              4271
Adverse-Effect        3341
Begin Offset AE       1366
End Offset AE         1357
Drug                  1319
Begin Offset Drug     1293
End Offset Drug       1286
dtype: int64
```

```
Top Drug-Adverse Effect Associations:
                Drug      Adverse-Effect  Frequency
2819625      heparin    thrombocytopenia         24
2729971   gemcitabine                 HUS          8
1783933    amiodarone      thyrotoxicosis          8
2901502    indapamide            diabetes          7
2387467  desmopressin        hyponatremia          7
3140030    lovastatin      rhabdomyolysis          6
3116262       lithium       neurotoxicity          6
2639400   fludarabine                AIHA          6
1782757    amiodarone      hypothyroidism          6
2615542      fentanyl  serotonin syndrome          6
```

Correlation Matrix-

	PubMed-ID	Begin Offset AE	End Offset AE	Begin Offset Drug	End Offset Drug
PubMed-ID	1.000000	0.146141	0.146065	0.145700	0.145453
Begin Offset AE	0.146141	1.000000	0.999709	0.985356	0.985235
End Offset AE	0.146065	0.999709	1.000000	0.985461	0.985348
Begin Offset Drug	0.145700	0.985356	0.985461	1.000000	0.999940
End Offset Drug	0.145453	0.985235	0.985348	0.999940	1.000000

Coming to the mandatory step of Preprocessing-

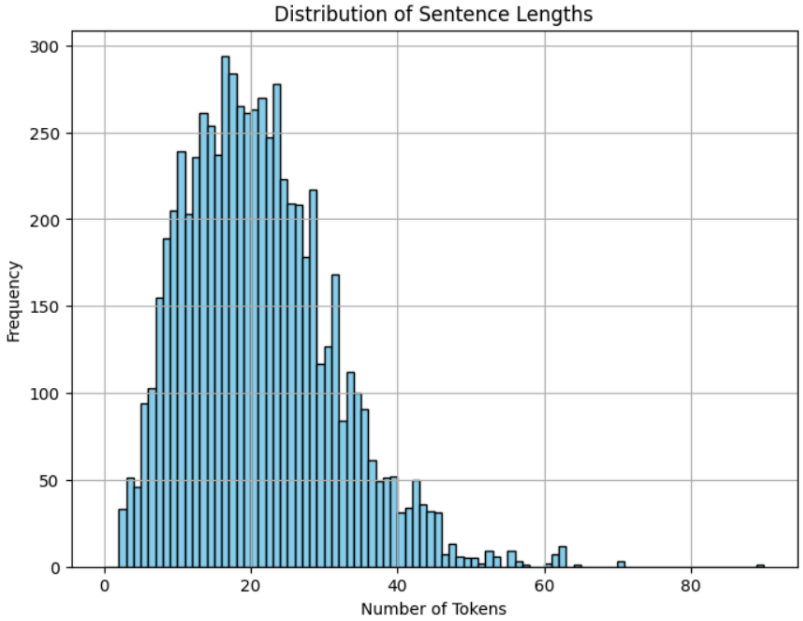
I have performed preprocessing and tokenisation on a text dataset, specifically focusing on the 'Sentence' column. Firstly, a preprocessing function named *preprocess_text* is defined to lowercase the text and remove punctuation and special characters using regular expressions. This function is then applied to each entry in the 'Sentence' column of the dataset, ensuring uniformity and cleanliness of the text data.

Following the preprocessing step, tokenisation is carried out using the Tokeniser class from the Keras library. This tokeniser is fitted on the preprocessed text data to generate a vocabulary of words. The *texts_to_sequences* method of the tokenizer converts each sentence into sequences of integers based on the vocabulary, representing each word with a unique integer index.

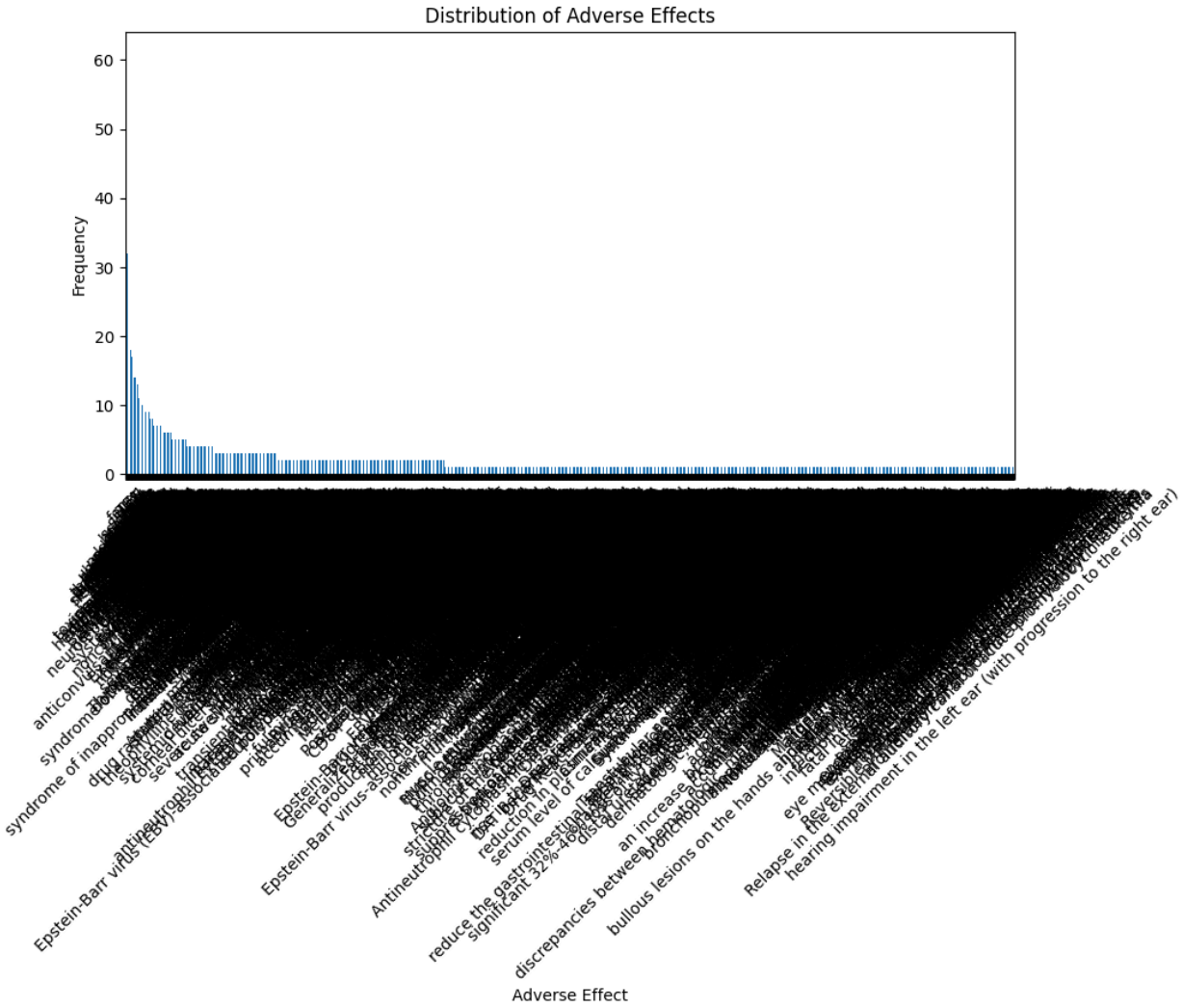
Additionally, the code snippet tokenises each sentence into words separately to count the number of words in each sentence. The average sentence length is then calculated by summing up the word counts across all sentences and dividing by the total number of sentences. This provides insight into the typical length of sentences in the dataset of 20 characters, which was used to later set it as the maximum character length during padding.

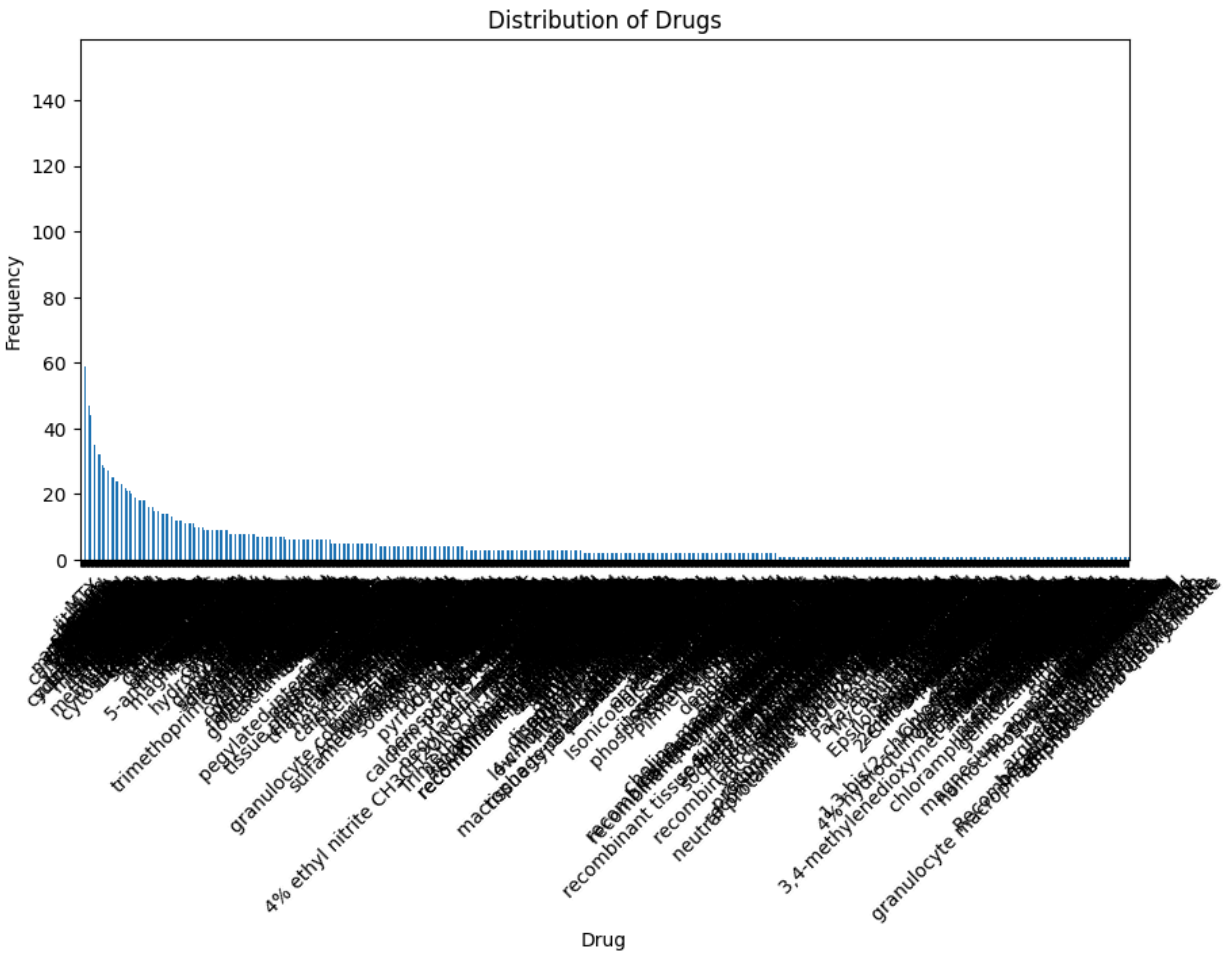
I have then performed categorical encoding- I converted the 'Adverse-Effect' and 'Drug' columns in the dataset to categorical data types using the pd.Categorical function, which assigns a unique numerical code to each category. Subsequently,

the categorical data is encoded into numerical values using the cat.codes attribute, resulting in new columns 'AE_Encoded' and 'Drug_Encoded' containing the encoded representations of adverse effects and drugs, respectively.

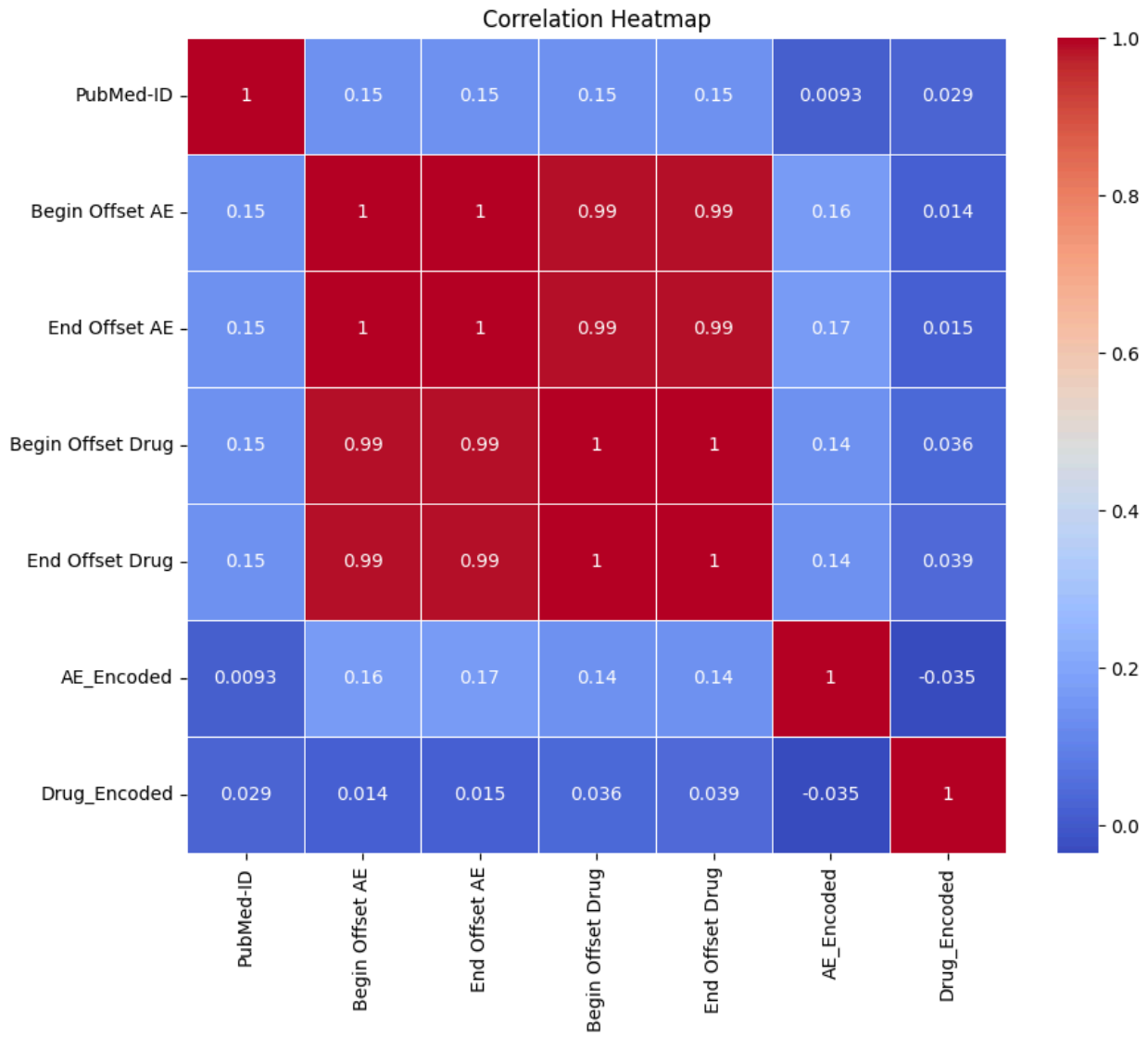


I have then split the dataset for training, validating and testing the due model. Prior to this, I generated 3341 classes of adverse effects and 1319 classes of drugs.




Distribution of Drugs
Frequency
0
20
40
60
80
100
120
140
Drug

I have also generated a heatmap to analyse the interaction between different columns-



Coming to the design of the LSTM model-

I defined the a neural network model architecture using TensorFlow's Keras API. The model consists of an input layer, an embedding layer, an LSTM layer, and two output layers for predicting adverse effects (AE) and drugs. Firstly, the input shape is defined based on the maximum sequence length of the input data. An embedding layer is then instantiated to learn dense representations of words in the input

sentences. The vocabulary size and embedding dimension are determined based on the tokenizer's word index and a predefined embedding dimension (e.g., 100).

Following the embedding layer, an LSTM layer is defined with a specified number of units (64 in this case) to capture sequential patterns in the text data. The input layer is connected to the embedding layer, and the embedded sequences are passed through the LSTM layer to generate LSTM outputs. Two separate output layers are added on top of the LSTM output, each with a softmax activation function to predict the probability distribution over the classes of adverse effects and drugs. The model is compiled with the Adam optimiser and categorical cross-entropy loss functions for each output, and accuracy is chosen as the evaluation metric. Finally, the model summary was printed to provide an overview of the model architecture and parameters.

LSTM ARCHITECTURE and MODEL IMPLEMENTATION

The model architecture consists of several key components, starting with the input layer, followed by the embedding layer, recurrent layers (LSTM), and output layers.

- Input Layer: The input layer defines the shape of the input data, which is based on the maximum sequence length of the input sentences. It serves as the entry point for the input data into the neural network.

- Embedding Layer: The embedding layer is responsible for learning dense representations of words in the input sentences. It transforms each word into a high-dimensional vector space, where words with similar meanings are closer together. This layer helps capture semantic relationships between words and improves the model's ability to understand the textual data.

- Recurrent Layers (LSTM): The LSTM (Long Short-Term Memory) layer is a type of recurrent neural network (RNN) layer that is well-suited for processing sequential data such as text. It maintains an internal state that allows it to retain information over long sequences, making it effective for

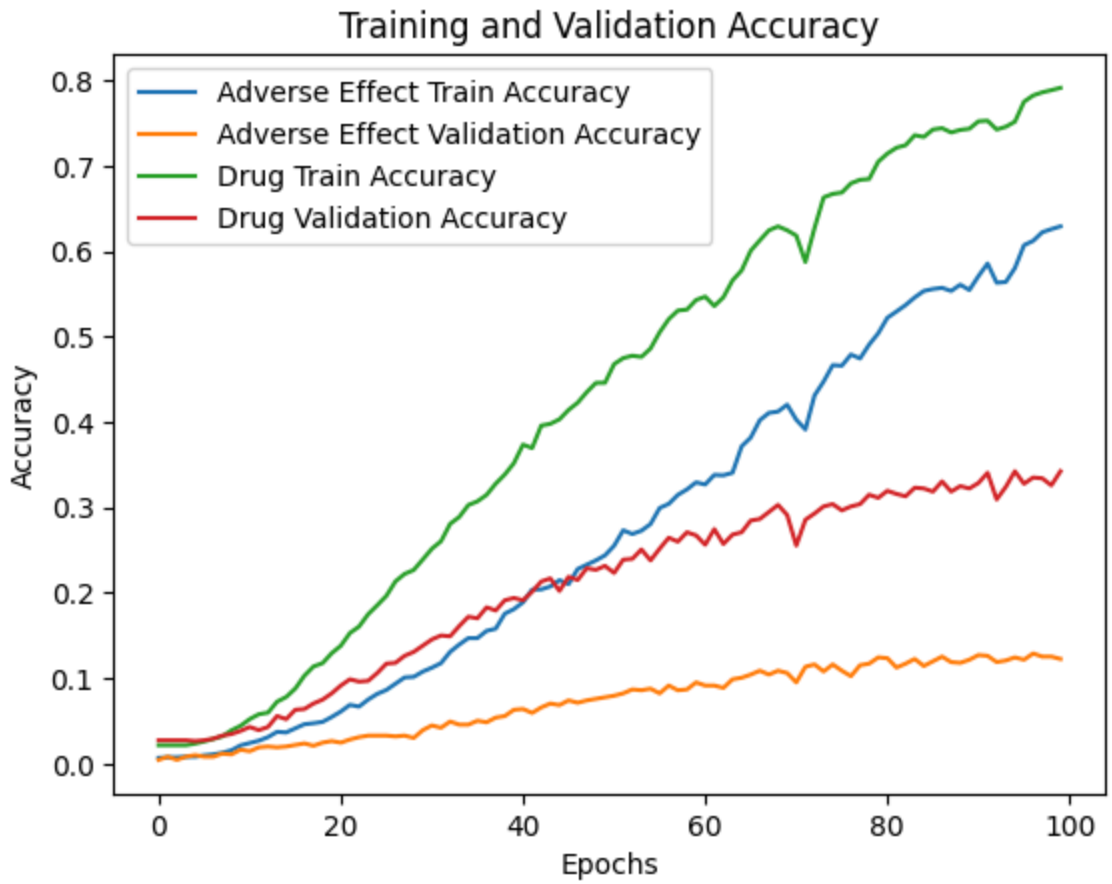
capturing temporal dependencies in the input data. The LSTM layer in this model consists of a specified number of units (64 units in this case), which determine the capacity of the model to learn complex patterns in the input sequences.

- Output Layers: The model has two output layers, one for predicting adverse effects and another for predicting drugs mentioned in the input sentences. Each output layer is a dense layer with a softmax activation function, which converts the raw output scores into probability distributions over the classes. This allows the model to predict the likelihood of each class for a given input sentence.

Thus, the architecture of the model leverages embedding and LSTM layers to process and analyse textual data, followed by output layers to make predictions about adverse effects and drugs.

PERFORMANCE EVALUATION

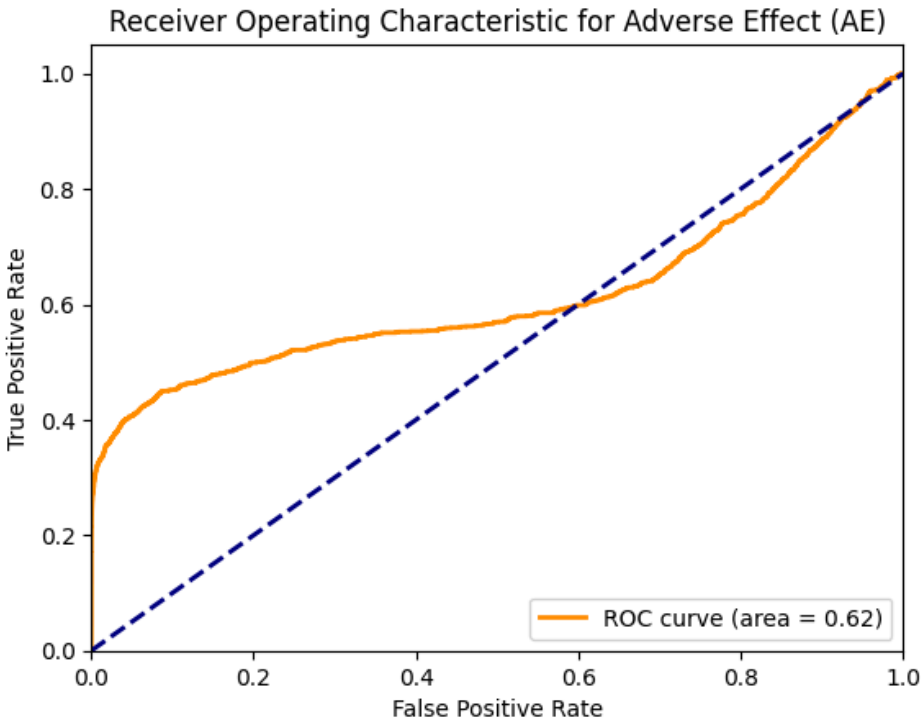
And finally, I have trained the model for 100 epochs-

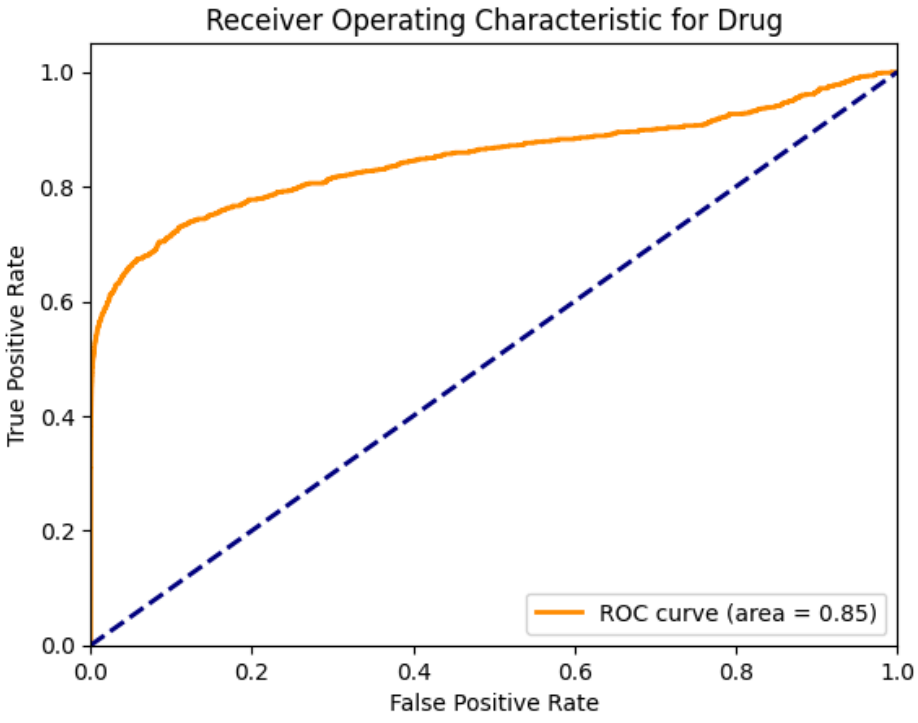


The resulting evaluation metrics I have delivered are-

```
Evaluation Metrics for Adverse Effect (AE):
Accuracy: 0.14725274725274726
Precision: 0.1374369420610022
Recall: 0.14725274725274726
F1-Score: 0.13446339054939152

Evaluation Metrics for Drug:
Accuracy: 0.3413919413919414
Precision: 0.3570889808317087
Recall: 0.3413919413919414
F1-Score: 0.3241030347505538
```





Coming to the embedding- The input shape is defined as (max_seq_length,), indicating the maximum sequence length of input sentences.

An embedding layer is created using the Embedding class from Keras. This layer learns dense representations of words in the input sentences. The vocab_size parameter specifies the size of the vocabulary, which is determined by the number of unique words in the tokenizer's word index plus 1 for the padding token. The embedding_dim parameter defines the dimensionality of the embedding space, which is set to 100 in this example.

Pretrained models doesn't work properly and it gives bad results here because the task we are doing here is related to drugs and their effects so we need to use domain specific vocabulary for better results, not the pretrained models which are used for general case. To still improve vectorization results we could use contextual word embeddings, domain specific
vocabulary

Future Research Directions for ADE Detection using NLP, ML, and DL

1. Expanding Data Sources and Techniques:

- **Electronic Health Records (EHRs):** Develop NLP models that can process unstructured clinical notes, discharge summaries, and radiology reports to identify potential ADEs more comprehensively.
- **Social Media and Online Forums:** Explore the potential of mining user-generated content on social media and online health forums to identify emerging or under-reported ADEs.
- **Real-World Data:** Utilise real-world data sources like insurance claims and pharmacy dispensing records to complement traditional clinical trial data and capture a broader picture of ADE occurrence.
- **Multimodal Learning:** Combine NLP techniques with other AI approaches like computer vision to analyse medical images (e.g., X-rays) for potential drug-related side effects.

2. Improving Model Generalizability and Interpretability:

- **Domain Adaptation Techniques:** Develop NLP models that can adapt to variations in medical language across different healthcare institutions and regions.
- **Explainable AI (XAI):** Integrate XAI techniques into ML/DL models to understand how the models identify ADEs and increase trust and transparency in their predictions.
- **Active Learning:** Develop active learning frameworks where the model can iteratively query healthcare professionals for feedback, improving accuracy in identifying complex or rare ADEs.

3. Early Detection and Prediction:

- **Predictive Modelling:** Utilise machine learning to predict potential ADEs in patients based on their individual medical histories, medications, and genetic factors.
- **Real-time Monitoring:** Develop systems that can continuously monitor a patient's health data (e.g., vital signs, lab results) to detect early signs of ADEs and allow for timely intervention.
- **Chatbots and Virtual Assistants:** Integrate NLP capabilities into chatbots or virtual assistants that can engage patients in conversations to identify potential ADEs and offer initial guidance.

4. Integration with Clinical Workflow:

- **Clinical Decision Support Systems (CDSS):** Integrate ADE detection models with CDSS to provide real-time alerts to healthcare professionals about potential drug interactions or ADE risks for specific patients.
- **Pharmacovigilance Tools:** Develop NLP-powered tools to automate aspects of pharmacovigilance, streamlining the process of identifying, reporting, and analysing ADEs.
- **Personalised Medicine:** Utilise ADE detection models to personalise medication plans and minimise the risk of adverse reactions for individual patients.

5. Addressing Ethical Considerations:

- **Patient Privacy:** Develop robust data anonymization techniques and ensure adherence to data privacy regulations when using patient information for model training and testing.
- **Algorithmic Bias:** Evaluate and mitigate potential biases in NLP and machine learning models used for ADE detection to ensure fairness and inclusivity across different patient demographics.
- **Transparency and Explainability:** Communicate the limitations and potential errors of AI-based ADE detection systems to healthcare professionals and patients.

By exploring these future research directions, NLP, ML, and DL can further revolutionise ADE detection, leading to improved patient safety, early intervention in case of adverse reactions, and personalised medication approaches in healthcare.