Final Project Report: Detecting Adverse Drug Reactions

Justin Na
Jna10

Abstract

Adverse drug reactions (ADRs) are a significant public health concern, as they can cause unintended, harmful responses to medicine that results in allergic reactions, severe skin reactions, or death. Keeping track of and organizing adverse drug reactions can be difficult, especially when a lot of data and reports exists as unstructured data, such as clinical notes, case reports, etc. An automatic system that can accurately extract these (drug, ADR) tuples accurately can greatly improve the patient's safety and reduce the number of preventable errors made when recommending a drug or intervention to a patient. Thus, this project aims to explore automated methods for extracting ADRs accurately from biomedical texts to reduce human effort and improve the scale and reliability of ADR detection. This report contains the methodology, evaluation metrics, and performance of both the baseline approach and machine learning approach.

Introduction

As mentioned in the abstract, adverse drug reactions are a significant problem that is important to address in order to prevent fatal outcomes. However, most data identifying drugs and their corresponding adverse reactions are recorded as unstructured data. Thus, our goal is to develop an automotive system that improves the ADR detection process.

The first part of this project focuses on implementing a baseline approach. For the baseline approach, we use regular expressions and gazetteers to identify drug to adverse reaction pairs from the ADE Corpus. We then use standard evaluation metrics (precision, recall, and f1 score) to measure the performance of our baseline approach.

The second part of the project focuses on implementing a machine learning approach. For the machine learning approach, we use two methods, Named Entity Recognition (NER) and Relation Classification (RC), to automatically identify a drug and its corresponding adverse reaction in unstructured text.

The third part of this project focuses on the evaluation of the machine learning approach. This part will focus on the performance metric scores of the model and classifier and identify areas of improvement.

Related Works

Works in the past have researched and demonstrated the usefulness of integrating both rule based and machine learning based approaches to improve ADR extraction.

Gurulingappa et al. (2012) developed the ADE corpus to support the automatic extraction of drug-related adverse effects from case reports.

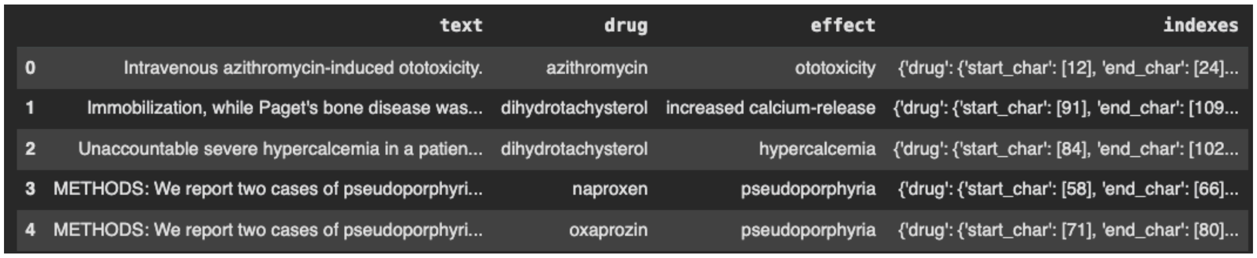
Kim (2022) explored statistical and ML methods for ADR analysis, highlighting how supervised learning played an important role in improving extraction accuracy.

Both these works laid the foundations of automating ADR extractions from biomedical texts and have showcased how rule-based and machine learning based approaches have proven to be effective in extracting drug to adverse reaction pairs.

Baseline Approach

Dataset

For the baseline approach, we used the ADE corpus v2 drug ade relation train split dataset. Each record contained a medical record text and its corresponding drug and adverse reaction extracted from the text.

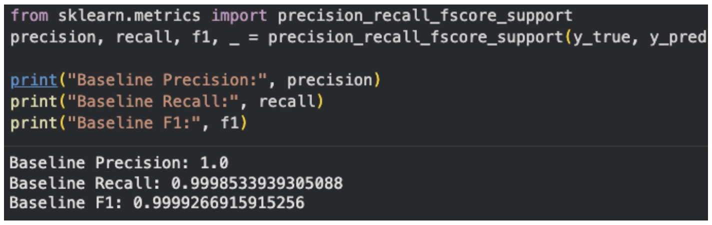


	text	drug	effect	indexes
0	Intravenous azithromycin-induced ototoxicity.	azithromycin	ototoxicity	{'drug': {'start_char': [12], 'end_char': [24]...
1	Immobilization, while Paget's bone disease was...	dihydrotachysterol	increased calcium-release	{'drug': {'start_char': [91], 'end_char': [109...
2	Unaccountable severe hypercalcemia in a patien...	dihydrotachysterol	hypercalcemia	{'drug': {'start_char': [84], 'end_char': [102...
3	METHODS: We report two cases of pseudoporphyri...	naproxen	pseudoporphyria	{'drug': {'start_char': [58], 'end_char': [66]...
4	METHODS: We report two cases of pseudoporphyri...	oxaprozin	pseudoporphyria	{'drug': {'start_char': [71], 'end_char': [80]...

Implementation

For the baseline approach, I first implemented gazetteers to store the unique drugs and adverse effects extracted from the dataset. I then created regular expression patterns to later use to identify drug and adverse reaction terms in each text/sentence. Next, I looked through each sentence in the dataset and used the regular expressions to identify drugs and adverse reactions in the text. Once I had identified a drug and adverse reaction, I used a tuple datatype to pair the drug and adverse reaction together to form the predicted result.

Once I had generated all my predicted drug to ADR pairs for each sentence/text, I then evaluated the performance of the gazetteers and regular expressions I had implemented. For each record, I assigned it a score/label of 1 if both the drug and ADR pair were in their respective gazetteer lists. After assigning the appropriate labels, I calculated the precision, recall, and f1 score performance metrics. Below are the performance metric results.



```
from sklearn.metrics import precision_recall_fscore_support
precision, recall, f1, _ = precision_recall_fscore_support(y_true, y_pred

print("Baseline Precision:", precision)
print("Baseline Recall:", recall)
print("Baseline F1:", f1)

Baseline Precision: 1.0
Baseline Recall: 0.9998533939305088
Baseline F1: 0.9999266915915256
```

Evaluation

Based on these results, the baseline performed very well with all 3 performance metrics above 99%. This is most likely due to the gazetteers identifying all drugs and adverse reactions that appeared in the dataset. However, while regular expressions and gazetteers perform well on already seen data (drugs and ADRs), this method would not perform as well on new data that includes new drugs and ADRs not identified in the training data, spelling errors, or more complex sentence structures. Thus, there is a need for machine learning methods to improve the performance and reliability of automotive ADR extraction.

Machine Learning Approach

Goal

Our baseline approach performed well on known drugs and ADRs but is limited in generalization. Our ML approach aims to compensate for this weakness by automatically identifying these pairs in unstructured biomedical text. We will implement and integrate two machine learning methods. The first is Named Entity Recognition (NER), which detects drugs and adverse reactions in sentences. The second is Relation Classification (RC), which classifies whether a drug to adverse reaction pair identified in text is valid.

Data Preparation

For the machine learning approach, we used the same ADE corpus v2 drug ade relation train split dataset. However, to use this dataset for classification and training, we had to preprocess the dataset.

NER

For our NER model, we had to convert our text data into token level labels in order to use the BioBert token classification model. In order to meet this goal, we first had to create character level tokens by assigning B- (begin) and I- (inside) tags to characters in the text. We chose to use B- and I- tags to improve NER accuracy for multi-token cases. I then used a BioBERT tokenizer to align the character level labels with the token level labels. Tokens not corresponding to any entity were ignored. Lastly, I split the dataset into training (70%), development (10%) and test (20%) subsets.

RC

For our RC classifier, we had to generate (drug, ADR) pairs for each text and label them as 1 (match) or 0 (no match). In order to accomplish this goal, we first used our NER predictions to extract all drugs and ADRs the model could identify in each text. Next, we generated candidate pairs by matching the drugs with ADRs in each text. We then assigned a label to each candidate tuple; 1 if the candidate matched the true tuple for the corresponding text, and 0 otherwise. Lastly, we split the dataset into training (80%) and test (20%) subsets. As a result, we were able to preprocess our dataset to use for sequence classification, where BioBert learns to predict whether a drug to ADR pair is valid.

Model Training

NER

In order to train our NER model, we first used AutoModelForTokenClassification from Hugging Face Transformers with BioBert to generate our NER model. Our model output had one label per token, corresponding to the BIO tagging scheme. B-DRUG, I-DRUG labels corresponded to tokens that are part of a drug entity, B-EFFECT, I-EFFECT labels corresponded to tokens that are part of an ADR, and 0 labels corresponded to tokens not part of any entity. I then used a Data Collator to handle variable length input and batch padding for more consistency. Finally, I utilized Hugging Face's Trainer API to train the NER model.

As a result, I was able to create a NER model that produced token-level predictions for drug and ADR entities. I then used these predictions to generate (drug, ADR) candidate pairs to use for the relation classification stage.

RC

In order to train our relation classification model, we first used AutoModelForSequenceClassification with BioBert to create our binary classifier. We created the inputs by combining the drug, ADR, and surrounding sentence context into a single tokenized sequence. We then used the Hugging Face Trainer API to train our RC model. As a result, our RC model learned how to filter valid (drug, ADR) candidate pairs from sentences. This ensures that even when the NER model detects multiple drug and ADR entities in the same text, only valid relationships are saved for analysis.

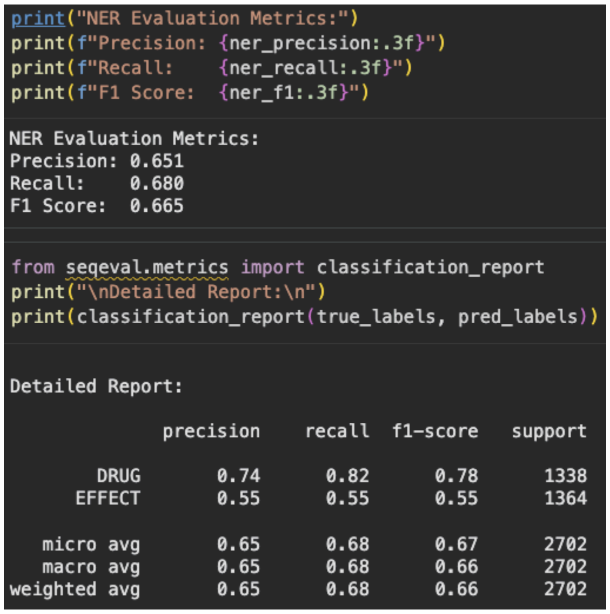
Summary

By separating our automotive ADR extraction process into two parts, NER and Relation Classification, we were able to separate entity detection from relation verification. This creates a more modularized workflow that makes it easier to identify which section to make changes to improve performance, which in turn leads to a more accurate and reliable automotive ADR extraction system.

Results and Evaluation

NER

The results of the NER model's performance on the test subset is shown below.



```
print("NER Evaluation Metrics:")
print(f"Precision: {ner_precision:.3f}")
print(f"Recall:    {ner_recall:.3f}")
print(f"F1 Score:  {ner_f1:.3f}")
```

```
NER Evaluation Metrics:
Precision: 0.651
Recall:    0.680
F1 Score:  0.665
```

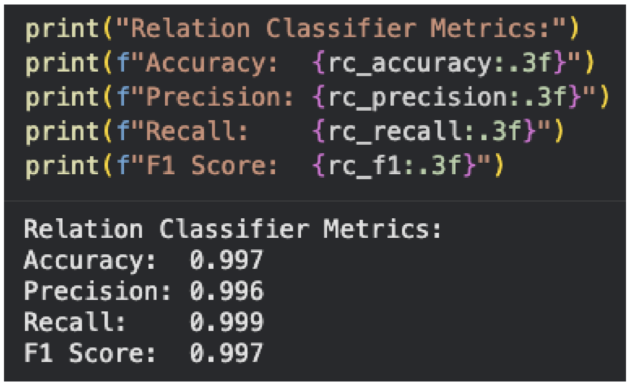
```
from seqeval.metrics import classification_report
print("\nDetailed Report:\n")
print(classification_report(true_labels, pred_labels))
```

```
Detailed Report:

              precision    recall  f1-score   support

        DRUG       0.74      0.82      0.78      1338
      EFFECT       0.55      0.55      0.55      1364

   micro avg       0.65      0.68      0.67      2702
   macro avg       0.65      0.68      0.66      2702
weighted avg       0.65      0.68      0.66      2702
```

Based on these results, the NER model was better at identifying drug entities compared to ADE entities, as highlighted by the higher precision and recall scores for DRUG entities. The lower precision and recall scores for EFFECT entities indicates that some adverse side effects are more difficult to detect. Overall, the performance metric scores of 65% for precision, 68% for recall, and 66% for f1 score shows that the NER model identifies the majority of entities correctly, and established a strong base for relation extraction.

RC

The results of the RC model's performance on the test subset is shown below.



```
print("Relation Classifier Metrics:")
print(f"Accuracy:  {rc_accuracy:.3f}")
print(f"Precision: {rc_precision:.3f}")
print(f"Recall:    {rc_recall:.3f}")
print(f"F1 Score:  {rc_f1:.3f}")
```

```
Relation Classifier Metrics:
Accuracy:  0.997
Precision: 0.996
Recall:    0.999
F1 Score:  0.997
```

Based on these results, the RC model performs very well, with accuracy, precision, recall, and f1 scores all being almost perfect. This indicates that the NER predictions produced provide high-quality inputs, and that our classifier can effectively distinguish valid (drug, ADR) pairs from invalid ones. The high recall score indicates that almost all true (drug, ADR) pairs are correctly identified by our RC model. Recall is a very important metric in healthcare as a high recall score reduces the number of false positives. Overall, very high scores in all 4 performance metrics indicate our RC model did a great job at identifying correct (drug, ADR) pairs from text.

Summary

In this project, we developed an automotive system to detect adverse drug reactions from unstructured biomedical text, using both a baseline and machine learning approach. Our work was broken down into three main parts.

Part 1: Baseline Approach

In the baseline approach, we implemented a rule-based system using gazetteers and regular expressions to detect drug names and adverse drug reactions. The baseline approach achieved above 99% scores for precision, recall, and f1 score performance metrics, highlighting that the baseline approach accurately identifies (drug, ADR) pairs in text. This step serves as a reference on ways we can approve our automotive system to perform well on unseen drugs and ADRs that are not in the training data, alongside other cases such as misspelling.

Part 2: Machine Learning Approach

In the machine learning approach, we trained a BioBert Named Entity Recognition (NER) model to detect drug and ADR entities. We used B- and I- labels to ensure that the model could identify entity spans accurately. Our NER model ended up achieving a F1 score of 68%, highlighting that the model performed well in detecting drug entities and moderately in identifying ADR entities.

We also trained a relation classification (RC) model to determine which (drug, ADR) pairs are valid. The RC model performed almost perfectly, with all four performance metric scores being above 99.5%. These high scores indicate that combining entity detection through our NER model with relation prediction leads to very effective results and performance.

Part 3: Evaluation and Analysis

The NER model precision, recall, and f1 scores of around 67% underscores room for improvement, specifically in the model's ability to detect adverse reactions. One area we could dive into in future research to improve entity detection include word embeddings (Word2Vec) to capture semantics. On the other hand, the RC model's high precision, recall, and f1scores indicate that the system can reliably identify real (drug, adverse reaction) pairs and reduce the number of missed ADRs and false positives. Overall, implementing a ML approach significantly improves the adverse reaction extraction process compared to using simple rule based methods, as a ML approach can handle a wider variety of cases while remaining scalable.

Conclusions

This project showcases how machine learning models can significantly improve the process of extracting adverse reactions from unstructured biomedical text. By combining NER and relation classification techniques, we were able to develop an efficient and organized workflow that can accurately identify valid (drug, adverse reaction) pairs from text. An ML approach can be used to support clinical making decisions and improve patient safety. Future research in this area could focus on a variety of topics to improve the automated process, such as investigating data augmentation methods (synonym replacement, back-translation, etc.) or upgrading the complexity of the models to enable multilingual ADR extraction for languages other than English. Overall, this project provides a strong foundation for automated ADR extraction which others can reference for future research.