Final Year Project Report - Review 02

Drug-Drug Interaction Identification from Biomedical Literature using BioBERT Model

Department of Computer Science, Anna University

Team 22

Juanita J (2018103544) Visshal A (2018103625) Kripashankar T (2018103555)

Abstract:

Drug-Drug Interaction (DDI) prediction is one of the most critical issues in drug development and healthcare. Although multiple DDI resources exist, it is becoming infeasible to maintain these up-to-date manually with the number of biomedical texts growing at a fast pace.Previous neural network based models have achieved good performance in DDIs extraction. However, most of the previous models did not make good use of the information of drug entity names, which can help to judge the relation between drugs. In this project, we present a novel neural network based model using multiple entity-aware attentions with various entity information to predict DDI from biomedical texts.We use an output-modified bidirectional transformer (BioBERT) and a bidirectional gated recurrent unit layer (BiGRU) to obtain the vector representation of sentences. The vectors of drug description documents encoded by Doc2Vec are used as drug description information, which acts as an external knowledge of our model. Then we construct three different kinds of entity-aware attentions to get the sentence representations with entity information weighted, including attentions using the drug description information.The outputs of attention layers are concatenated and fed into a multi-layer perceptron layer. Finally, we get the result by a softmax classifier. We evaluate our

proposed model on the DDIExtraction 2013 corpus benchmark dataset, evaluated using F-score.

Project Guide:

Mrs. M. Saranya

Contents

Problem Statement

Drug-drug interaction (DDI) may lead to a positive or negative impact on expected therapeutic outcomes. A negative consequence may worsen a patient's condition or lead to increased healthcare costs, life-threatening effects or even death. DDI contributes only to 3%–4% of the adverse drug reactions but the fourth leading cause of mortality. Approximately 37–60% of patients admitted to hospital may have one or more potentially interacting drug combinations at admission as per recent report. Drug-drug interactions are known to be responsible for nearly a third of all adverse drug reactions. The identification of drug names is a preliminary and crucial step in pharmacovigilance, drug relation extraction and many text mining tasks such as the detection of the outbreak of diseases, the extraction of medication-related information, the detection of adverse-drug events or the extraction of relationships such as drug-disease, drug-gene interactions, DDIs, among many others. Although multiple resources for DDI identification and extraction exist, they are all unable to keep pace with the roaring rich amount of information available in the fast growing biomedical texts and information, in efficiency, accuracy and complexity. Therefore, to develop a method for automatic extraction of DDIs from biomedical literature is of great significance for current healthcare management and clinical testing research.

Introduction

The simultaneous administration of multiple drugs increases the probability of interaction among them, as one drug may affect the activities of others. Thus, identification of unknown drug-drug interactions (DDIs) is of significant concern for improving the safety and efficacy of drug consumption. Thus, a drug–drug interaction (DDI) is defined as a pharmacokinetic or pharmacodynamic influence of drugs on each other, which may result in desired effects, in reduced efficacy and effectiveness or increased toxicity and lead to adverse drug reactions that can be severe enough to necessitate hospitalization. For example, sildenafil (Viagra) in combination with nitrates can cause a potentially life-threatening decrease in blood pressure. Therefore, DDI extraction, which detects DDIs in unstructured text and classifies them into predefined categories automatically, has become an increasing interest in medical text mining. Usually, human experts manually collect DDI information from various sources such as the FDA's Adverse Event Reporting System. Since there are numerous combinations of drugs available, it is difficult to collect all the DDI events of patients from reports or publications. Also, manually organizing DDI information in natural language into a DDI database is costly and time-consuming. Although

multiple DDI resources exist, it is becoming infeasible to maintain these up-to-date manually with the number of biomedical texts growing at a fast pace. Most existing methods model DDI extraction as a classification problem and rely mainly on handcrafted features, and certain features further depend on domain-specific tools. The biomedical text contains some defining properties of big data such as velocity and variety. For example, PubMed, a well-known database of biomedical articles, comprises more than 30 million citations for biomedical literature from MED-LINE, life science journals, online books, and also adds approximately 800 thousand new articles annually . Manual DDI extraction is time-consuming and could lead to out-of-date information. Moreover, drugs with a narrow therapeutic range or low therapeutic index which are commonly unnoticeable by pharmacists are more likely to be the objects for serious drug interactions. Recently, neural network models using latent features have been demonstrated to yield similar or superior performance compared to existing models.

In this project, we present a novel neural network based model using multiple entity-aware attentions with various entity information to predict DDI. BERT is a recently proposed pre-trained language model. Due to its multi-layer bidirectional transformer structure, BERT can integrate the contextual information of sentences into the word vector both from forward and backward. This makes it contain more context information than traditional word embedding models, such as Word2Vec and GloVe. Besides,the word vector generated by BERT will change according to the context, while the traditional word embedding models are context-free. Wang et al. proposed a BERT based DDIs detection model, and achieved the state-of-the-art result using evidence of supplement-drug interactions from scientific text. Peng et al. made pre-trained BERT models on biomedical literature and the results showed that their models had better performance over other pre-trained language models on some biomedical datasets. BioBERT is another pre-trained BERT model which is trained with large-scale biomedical corpora. Li et al. proposed a BioBERT based model which used GCNN to integrate dependence structure information into the model. However, BioBERT is trained on specific tasks, resulting in the lack of generalisation ability. We use an output-modified bidirectional transformer (BioBERT) and a bidirectional gated recurrent unit layer (BiGRU) to obtain the vector representation of sentences. The vectors of drug description documents encoded by Doc2Vec are used as drug description information, which is an external knowledge of our model. Then we construct three different kinds of entity-aware attentions to get the sentence representations with entity information weighted, including attentions using the drug description information.The outputs of attention layers are concatenated and fed into a multi-layer perceptron layer. Finally, we get the result by a softmax classifier.

We integrate drug descriptions from Wikipedia and DrugBank to our model to enhance the semantic information of drug entities. There are several publicly available databases supporting healthcare professionals to find DDIs. For example, DrugBank, which is an online drug database, consists of 8311

drugs entries. Each drug entry contains more than 200 fields, including a DDI field. However, the databases have a few limitations. It is impossible for healthcare professionals to find DDIs from the overwhelming amount of literature manually and to keep up-to-date with the latest DDI findings. DDI allows to predict mechanistic and static drug-drug interactions (DDIs) among drugs and metabolites. The ability to accurately estimate potential DDIs in silico has several benefits for pharmaceutical companies: Explore possible effects on the pharmacology and toxicology of drugs, Identify species-specific changes to estimate how a drug behaves in animals vs. humans and iInvestigate the safety profile of drugs that are co-administered prior to filing regulatory submissions with the FDA, EMA, and other agencies. It is known that such DDI events may cause preventable drug related harm. Several databases such as DrugBank, PharmGKB, Drugs.com and Stockley's Drug Interactions collect known adverse events caused by DDIs. In our model, we use the DDI 13 dataset.The dataset used is DDI Extraction 2013 corpus, which is the benchmark dataset for the DDIs extraction task. The DDIs corpus consists of 792 texts from the DrugBank database and 233 abstracts from the MEDLINE database.

Related work

Several computational methods have been developed to better understand drug interactions, especially for DDIs. However, these methods do not provide sufficient details beyond the chance of DDI occurrence, or require detailed drug information often unavailable for DDI prediction.

Usually, human experts manually collect DDI information from various sources such as the FDA's Adverse Event Reporting System. Since there are numerous combinations of drugs available, it is difficult to collect all the DDI events of patients from reports or publications. Also, manually organising DDI information in natural language into a DDI database is costly and time-consuming. Several efforts to automatically collect DDI information from the biomedical literature using text mining tools have been made. The DDI Challenges in 2011 and 2013 released gold standard datasets for the task of improving the performance of DDI extraction using a Natural Language Processing (NLP) pipeline. Using support vector machines (SVMs), some of the methods obtained better results on datasets. Unfortunately, the methods that use traditional machine learning classifiers such as SVMs require feature engineering of domain experts, which is also expensive and time consuming. To overcome the problems that the previous methods traditional machine learning techniques like SVM (that requires feature engineering by domain experts) have, a DDI extraction model using an RNN based approach was used. RNN model uses a position feature, a subtree containment feature and an ensemble method to improve the performance of DDI extraction. Many machine learning models have also been proposed in the literature to predict the drug-drug interaction score efficiently. However, these models suffer from the over-fitting

issue. Therefore, these models are not so-effective for predicting the drug-drug interaction score. Among the existing studies that performed well on the DDI '13 corpus, the study by Kim used a linear kernel-based model with a rich set of lexical features. The authors proposed a two-stage method to achieve high performance. FBK-irst utilised the negation scope information. A negation cue (e.g. no) is an important signal that can reverse the meaning of a particular text segment and the negation scope is the text segment that is the subject of negation. The authors of FBK-irst used an SVM classifier with a non-linear kernel.

The following neural network based models were also proposed for the DDI'13 challenge. The Syntax Convolutional Neural Network (SCNN) model uses word embeddings of the shortest dependency paths, position features and POS information to represent the input sentences. The Multi-Channel Convolutional Neural Network (MCCNN) model uses several word embeddings for a CNN. Multiple word embeddings have more coverage than only one word embedding, because they can cover a rare word if it exists in at least one word embedding. The CNN-bioWE model and the CNN-rand model both implemented the Convolutional Neural Network (CNN) model combined with position embedding. The CNN-bioWE model uses word embedding trained on MEDLINE abstracts. The CNN-rand model uses a random initialised word embedding matrix. The Matrix-Vector Recursive Neural Network (MV-RNN) model was re-implemented for the DDI '13 Challenge. The MV-RNN model assigns a vector and a matrix to every node in a parse tree to learn the syntactic and semantic information. The existing methods that are used for DDI rely mainly on manually engineered features, i.e handcrafted features and longer length sentences were not properly classified by these models. To overcome this, Long Short-Term Memory models were used, namely, B-LSTM, AB-LSTM and Joint AB-LSTM. The Joint AB-LSTM used LSTM based architectures with an attention mechanism to achieve high performance.

An author named Socher proposed a Matrix-Vector Recursive Neural Network (MV-RNN) model that assigns a vector and a matrix to every node in a parse tree to classify the relation of two target nouns in a sentence. They showed that their recursive neural network model is effective for finding relations between two entities. Unfortunately, the MV-RNN model's performance on the DDI extraction task was unsatisfactory. Author Zhang used a refined-semantic class annotation method which replaces several important terms related to the PK DDI process with more generic terms. Zhang et al. implemented the all-paths graph kernel method which uses dependency graphs that represent sentence structures. In addition to the semantic class annotation, Zhang also used predicate-argument structures (PASs) in place of the dependency parser result. We denote the dependency parsing version results as DEP_ReSC and the PAS version results as PAS_ReSC, both of which are obtained from the previous study. The PK DDI corpus has only baseline results tested by the authors of the data. We tried to use the baseline results of the DDI'13 corpus for the PK DDI corpus. However, the existing studies that released the code

provide the pre-processing code part only for the DDI '13 corpus or lack details on how to pre-process data other than the DDI'13 corpus. Also, machine learning models that do not go through hyper-parameter adjustments will obtain lower performance; therefore, we note only the baseline results obtained from the previous study.

In fact, several corpora have been built for these purposes in recent years. Here, to review the main corpora annotated with drug entities, giving a special focus on those corpora that also contain DDIs, since each corpus has been developed for a specific task, the definition of the drug entity varies significantly from corpus to corpus. Thus, for example, in Clinical E-Science Framework (CLEF) and BioText corpora, drug names and therapeutic devices or interventions are annotated with the same entity type. Other corpora such as ADE (Adverse Drug Effect), EU-ADR (Exploring and Understanding Adverse Drug Reactions) or ITI TXM (Tissue Expressions and Protein–Protein Interactions)use a single entity type to annotate both drugs and chemicals, while the BioCastercorpus distinguishes between substances for the treatment of diseases and chemicals not intended for therapeutic purposes. Corpora such as PK-DDI (Pharmacokinetic drug–drug interaction) or those developed by Rubrichi and Quaglini propose a more fine-grained classification of pharmacological substances. Despite these advances, the systems did not make good use of the information on drug entity names, which can help to judge the relation between drugs due to their complexities and also the semantic similarity between the two DDI types lead to classification errors. This particular problem was overcome by using a neural network model using BioBERT and multiple entity-aware attentions. The outputs of attention layers were concatenated and fed into a multi-layer perceptron layer. Recently, numerous text mining– and machine learning–based methods have been developed for predicting DDIs. All these methods implicitly utilise the feature of drugs from diverse drug-related properties. However, how to integrate these features more efficiently and improve the accuracy of classification is still a challenge. This was overcome by using five drug-related sources of chemical substructure information, drug-target association, drug-enzyme association, drug-pathway association, and ATC code of drugs were used to form the drug feature along with the Jaccard similarity coefficient. Despite getting this far, the dataset that is most commonly used is imbalanced, i.e, there is uneven distribution of data.

Overall Objectives

- To develop a method for automatic extraction of DDIs from biomedical literature as it is of great significance for current healthcare management and clinical testing research.

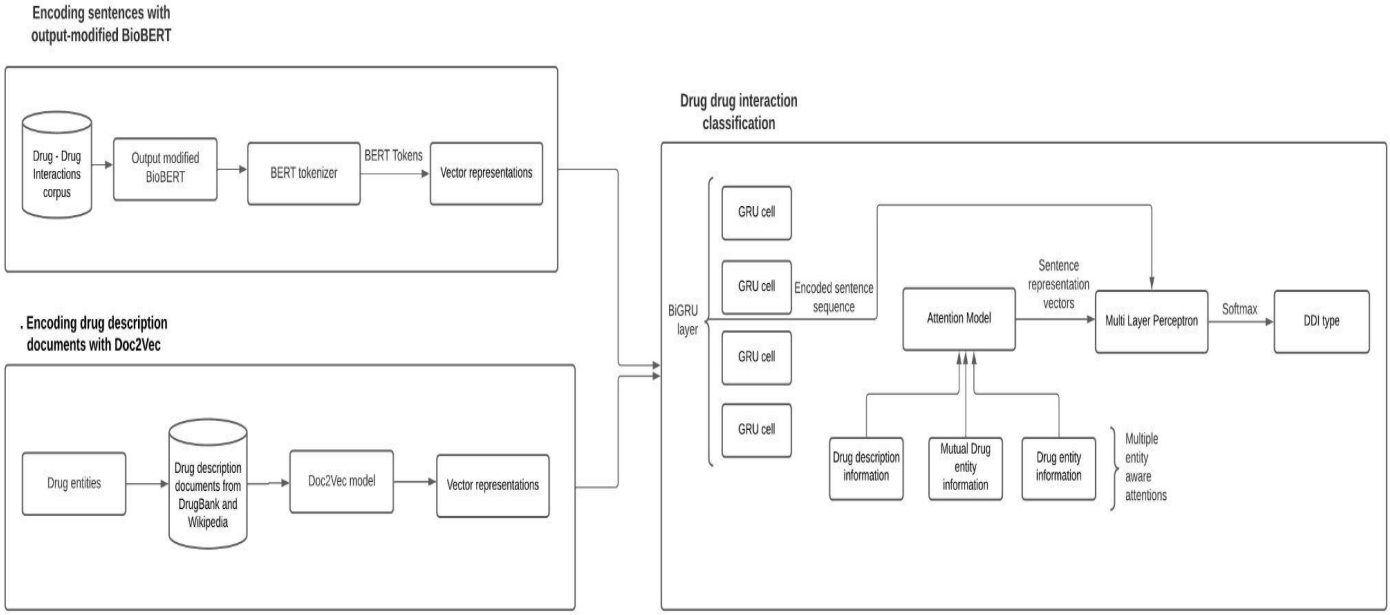
- To propose a novel model using multiple entity-aware attentions with various entity information to extract DDIs from biomedical literature and to strengthen the representations of drug entities in sentences.
- To use an output-modified bidirectional transformer (BioBERT) and a bidirectional gated recurrent unit layer (BiGRU) to obtain the vector representation of sentences
- To integrate drug descriptions from Wikipedia and DrugBank to our model to enhance the semantic information of drug entities.
- To integrate drug description information into the neural network model, the model can better understand the complex drug names in DDIs corpus, and can better extract their relation
- To reduce noise and imbalance by incorporating the GAN/SMOTE algorithm.

Literature Survey

Paper	Author	Publisher & Year	Methodology	Limitations
Extracting drug-drug interactions from texts with BioBERT and multiple entity-aware attentions	Yu Zhu, Lishuang Li, Hongbin Lu, Anqiao Zhou, Xueyang Qin	Elsevier journal, 2020	Proposed a neural network based method using output-modified BioBERT and multiple entity-aware attentions. The outputs of attention layers are concatenated and fed into a multi-layer perceptron.	The semantic similarity between the two DDI types affects the classification model negatively. Tested with an imbalance dataset.
Prediction of drug-drug interaction types with the unified embedding features from drug similarity networks	Xioa-Ying Yan, Peng-Wei Yin, Xiao-Meng Wu, Jia-Xin Han	Frontiers in Pharmacology, 2021	DDI types and drug features were extracted from DrugBank datasets and Jaccard coefficient was used to construct the similarity networks. A	More drug-related sources and suitable similarity measures can be utilized to improve the quality of drug similarity

			multimodal deep autoencoder was adopted to integrate the heterogeneous information on drugs.	matrices. The imbalanced dataset problem hasn't been optimally addressed.
Drug-drug interaction extraction from Biomedical texts using Long Short-Term Memory network	Sunil Kumar Sahu, Ashish Anand	Journal of Biomedical Informatics, 2018	Presented three long short-term memory (LSTM) network models, namely B-LSTM, AB-LSTM, and Joint AB-LSTM.	Imbalance and noise affects the model. Repetitive mention of other drug names affects all models negatively.
Drug-drug interaction extraction from the literature using a recursive neural network	Sangrak Lim, Kyubum Lee, Jaewoo Kang	The Public Library of Science, 2018	Build a DDI extraction model using a RNN based approach. RNN model uses a position feature, a subtree containment feature and an ensemble method to improve the performance of DDI extraction	The model fails when the sentence has a complex structure and the target drugs are positioned far from the primary information. Relations are described using unclear terms and falsely recognized as positive instances.

Architecture Diagram

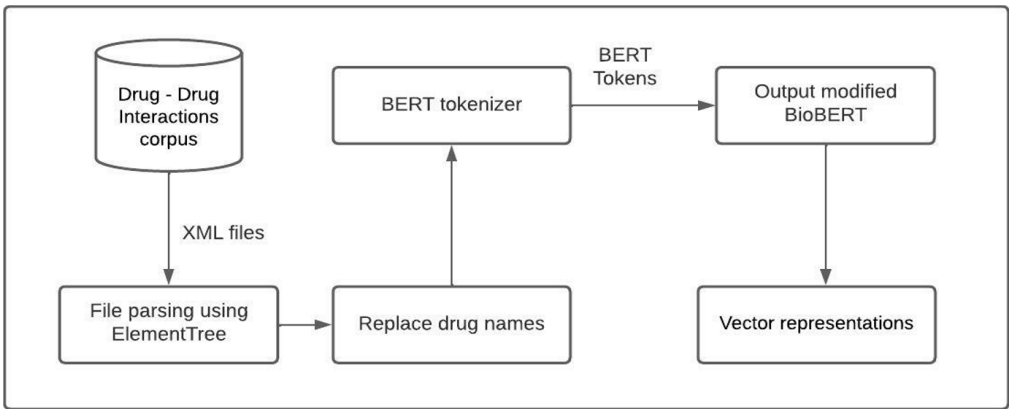


List of Modules

Module 1

Encoding sentences with output-modified BioBERT

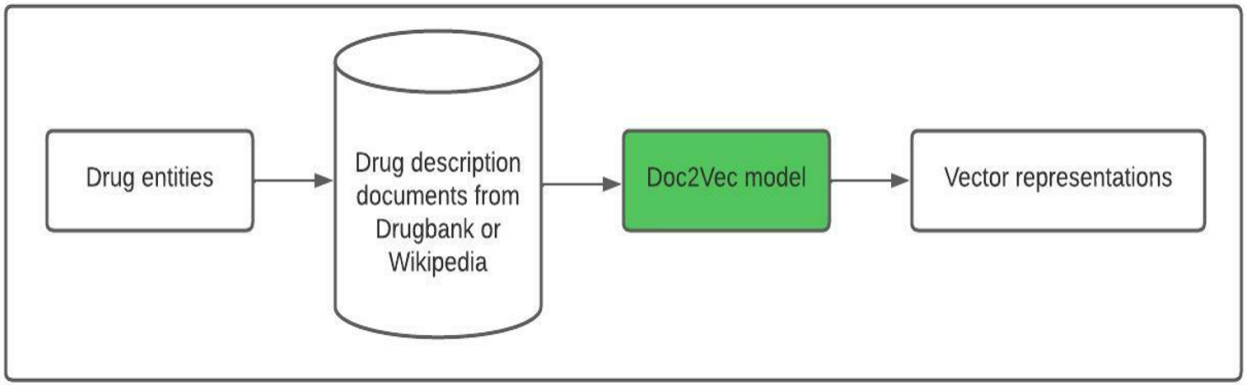
- Retrieve dataset from DDI corpus 2013.
- Given a sentence from the DDIs corpus, split the words in the sentence into BERT tokens
- To eliminate the influence of the drug names on the semantics of sentence, replace the drug entities whose relation need to be extracted with “drug1” and “drug2”
- Use the average output of the last four layers of BioBERT model to get the vectors of the sentence tokens.



Module 2

Encoding drug description documents with Doc2Vec

- For the drug description documents, we use the Doc2Vec tool to get their vector representations.
- Retrieve drug descriptions from DrugBank, and for drugs that can't be found in DrugBank, get their description through Wikipedia with a web crawler.
- Put all drug description documents into Doc2Vec model and get their vector representations.



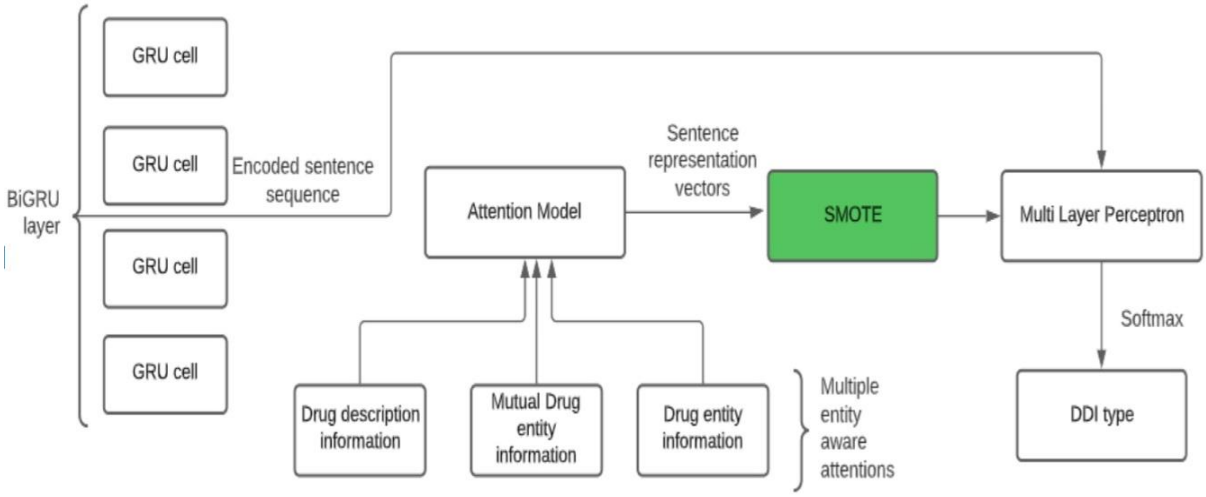
Module 3

Drug drug interaction classification

- Input BioBERT embeddings through the BiGRU layer to encode the output of the BERT layer.
- Input document vectors and BioBERT embeddings through three different types of entity

information into the attention model, which are drug entity information, mutual drug entity information and drug description information.

- Concatenate the outputs of attention layers and the original sentence representation, and put them into the multilayer perceptron layer
- We obtain DDI classification through the softmax layer.



Dataset

DDI '13 corpus is the most widely known and manually annotated corpus among the DDI-related corpora. The dataset used is DDI Extraction 2013 corpus, which is the benchmark dataset for the DDIs extraction task. The DDIs corpus consists of 792 texts from the DrugBank database and 233 abstracts from the MEDLINE database.

A previous version of the DDI corpus was created for the DDIExtraction 2011 challenge. The goal of this task was to promote research and provide a common framework for comparing the latest advances in Information Extraction techniques applied to the extraction of DDIs from biomedical texts. This earlier version of the DDI corpus consisted of 579 documents describing drug interactions that were taken from the DrugBank database. The documents were parsed using the Unified Medical Language System (UMLS) MetaMap Transfer tool (MMTx)to automatically recognize drugs. Then, a pharmacist manually annotated the DDIs in texts. The main limitations of this previous version were:

(1) drugs were automatically annotated without any manual intervention in the process, (2) no guidelines were produced, (3) the annotation was carried out by a single annotator, and (4) the quality of the corpus was not evaluated because the inter-agreement annotator was not measured.

Implementation:

- Dataset download from DDI'13 corpus

(a) The DDI'13 dataset

Data split	Total	NA	ADVICE	EFFECT	MECHANISM	INT
Training	11556	8088	734	1434	1131	169
Validation	1285	899	80	158	129	19
Test	3020	2049	221	357	301	92

(b) The distantly supervised DTI dataset

Data split	Total	NA	substrate	inhibitor	agonist/ antagonist	unknown	other
Training	472k	464k	1710	2612	855	2534	604
Validation	4769	4686	12	20	11	37	3
Test	4817	4734	18	19	10	26	10
Unlabelled	666k	/	/	/	/	/	/

- Parse the XML files using ElementTree (ET) library from python

```
import xml.etree.ElementTree as ET
```

```
#Analyze a .xml and return lists about sentence\entity\pair
def analyze_xml(name):
    """
        Analyze a .xml and return lists about sentence\entity\pair
    """
    with open(name, "rb") as fin:
        # parse xml file
        tree = ET.parse(fin)
        root = tree.getroot()
```

- Retrieve the drug sentence ID, name, drug pairs and type of interaction

```
for child in root:
    if child.tag=="sentence":
    # for sent in child.iter("sentence"):
        # Label names and attributes of
        sent_text = []
        text=child.get("text")
        sent_text.append(text)
        sentence_text.append(sent_text)
        sent_id = []
        id = child.get("id")
        sent_id.append(id)
        sentence_id.append(sent_id)
```

```
for children in child:
    if children.tag=="entity" :
        # Label names and attributes of third-level nodes
        ent_text=[]
        text=children.get("text")
        ent_text.append(text)
        entity_text.append(ent_text)
        ent_id = []
        ent_id.append(children.get("id"))
        entity_id.append(ent_id)
        ent_type = []
        ent_type.append(children.get("type"))
        entity_type.append(ent_type)
        ent_charOffset = []
        ent_charOffset.append(children.get("charOffset"))
        entity_charOffset.append(ent_charOffset)
```

```
if children.tag=="pair":
    pa_id=[]
    pa_id.append(children.get("id"))
    pair_id.append(pa_id)
    pa_e1=[]
    pa_e1.append(children.get("e1"))
    pair_e1.append(pa_e1)
    pa_e2=[]
    pa_e2.append(children.get("e2"))
    pair_e2.append(pa_e2)
    pa_ddi=[]
    pa_ddi.append(children.get("ddi"))
    pair_ddi.append(pa_ddi)
    pa_type=[]
    if children.get("ddi")=="false":
        pa_type.append("none")
    if children.get("ddi")=="true":
        pa_type.append(children.get("type"))
    pair_type.append(pa_type)
```

- To eliminate the influence of the drug names on the semantics of sentence, replace the drug entities whose relation need to be extracted with “drug1” and “drug2”.

Because the DDI '13 corpus is employed for extracting DDIs, a target drug pair is labeled as “False” if its interaction is not represented in a sentence, even though there is an actual interaction between the two drugs. Drug names do not play a significant role in the DDI detection process; therefore, we replaced the drug names with designated names such as “Drug0” for the first drug and “Drug1” for the second drug and so on. Replacing drug entities with designated names also addresses the unusual cases where target entities are composed of two or more non-sequential words.

```
DDI-DrugBank.d716.s2$effect$Mazindol$guanethidine$drug1 may reduce the effects of drug2 (Ismelin).
DDI-DrugBank.d716.s2$effect$Mazindol$Ismelin$drug1 may reduce the effects of drug0 (drug2).
DDI-DrugBank.d716.s2$none$guanethidine$Ismelin$drug0 may reduce the effects of drug1 (drug2).
```

- To make the input readable for the BioBERT model, tokenize the sentences

```
BERT Tokenized output
[[101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108, 108,
1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108,
108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914,
108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```


Results

Input :

```
<sentence id="DDI-DrugBank.d716.s2" text="Mazindol may reduce the effects of guanethidine (Ismelin). ">
  <entity id="DDI-DrugBank.d716.s2.e0" charOffset="0-7" type="drug" text="Mazindol"></entity>
  <entity id="DDI-DrugBank.d716.s2.e1" charOffset="35-46" type="drug" text="guanethidine"></entity>
  <entity id="DDI-DrugBank.d716.s2.e2" charOffset="49-55" type="brand" text="Ismelin"></entity>
  <pair id="DDI-DrugBank.d716.s2.p0" e1="DDI-DrugBank.d716.s2.e0" e2="DDI-DrugBank.d716.s2.e1" ddi="true" type="effect"></pair>
  <pair id="DDI-DrugBank.d716.s2.p1" e1="DDI-DrugBank.d716.s2.e0" e2="DDI-DrugBank.d716.s2.e2" ddi="true" type="effect"></pair>
  <pair id="DDI-DrugBank.d716.s2.p2" e1="DDI-DrugBank.d716.s2.e1" e2="DDI-DrugBank.d716.s2.e2" ddi="false"></pair>
</sentence>
```

Intermediate Output 1 :

```
DDI-DrugBank.d716.s2$effect$Mazindol$guanethidine$Mazindol may reduce the effects of guanethidine (Ismelin)
DDI-DrugBank.d716.s2$effect$Mazindol$Ismelin$Mazindol may reduce the effects of guanethidine (Ismelin).
DDI-DrugBank.d716.s2$none$guanethidine$Ismelin$Mazindol may reduce the effects of guanethidine (Ismelin).
```

Intermediate Output 2 :

```
DDI-DrugBank.d716.s2$effect$Mazindol$guanethidine$drug1 may reduce the effects of drug2 (Ismelin).
DDI-DrugBank.d716.s2$effect$Mazindol$Ismelin$drug1 may reduce the effects of drug0 (drug2).
DDI-DrugBank.d716.s2$none$guanethidine$Ismelin$drug0 may reduce the effects of drug1 (drug2).
```

Final Output


```
BERT Tokenized output
[[101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108, 108,
1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108,
108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914,
108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

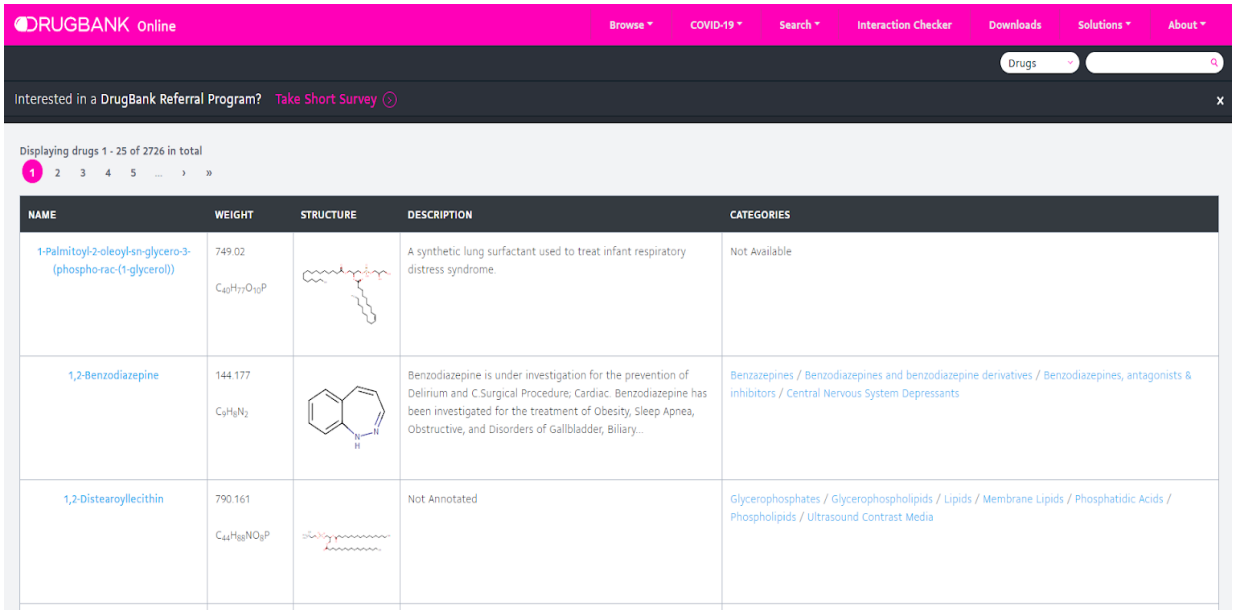

Dataset (Module 2)

The DrugBank database is a comprehensive, freely accessible, online database containing information on drugs and drug targets created and maintained by the University of Alberta and The Metabolomics Innovation Centre located in Alberta, Canada. As both a bioinformatics and a cheminformatics resource, DrugBank combines detailed drug (i.e. chemical, pharmacological and pharmaceutical) data with comprehensive drug target (i.e. sequence, structure, and pathway) information. DrugBank has used content from Wikipedia. Wikipedia also often links to Drugbank, posing potential circular reporting issues.

The DrugBank Online website is available to the public as a free-to-access resource. However, use and re-distribution of content from DrugBank Online or the underlying DrugBank Data, in whole or part, and for any purpose requires a license. Academic users can apply for a free license for certain use cases while all other users require a paid license.

Implementation of Module 2

- The dataset is scraped from https://go.drugbank.com/



- Retrieving drug name and description using BeautifulSoup library (Web Scraping)

Drug Name

```
In [8]: document.find_all( class_ = "name-value text-sm-center drug-name" )
```

```
Out[8]: [<td class="name-value text-sm-center drug-name"><strong><a href="/drugs/DB11331">1-Palmitoyl-2-oleoyl-sn-glycero-3-(phospho-
        rac-(1-glycerol))</a></strong></td>,
         <td class="name-value text-sm-center drug-name"><strong><a href="/drugs/DB12537">1,2-Benzodiazepine</a></strong></td>,
         <td class="name-value text-sm-center drug-name"><strong><a href="/drugs/DB14099">1,2-Distearoyllecithin</a></strong></td>,
```

- Scraped data :

```
['Acetic acid']
['An antimicrobial agent used to treat susceptible infections of the external auditory canal.']


['Acetohexamide']
['Used in the management of diabetes mellitus type 2 (adult-onset).']


['Acetohydroxamic acid']
['A synthetic urea derivative used to treat urea splitting bacterial infections of the urinary tract.']


['Acetophenazine']
['For the treatment of disorganized and psychotic thinking. Also used to help treat false perceptions (e.g. hallucinations or
delusions.)']
```

- Preprocess the scraped data

Tokenizing using gensim.utils.simple_preprocess

```
['drugs', 'and', 'the', 'high', 'cost', 'of', 'health', 'care']
['surgical', 'approach', 'for', 'patients', 'with', 'unstable', 'angina', 'pectoris', 'role', 'of', 'the', 'response', 'to',
'initial', 'medical', 'therapy', 'and', 'intraaortic', 'balloon', 'pumping', 'in', 'perioperative', 'complications', 'after',
'aortocoronary', 'bypass', 'grafting']
['value', 'and', 'limitations', 'of', 'the', 'response', 'to', 'exercise', 'in', 'the', 'assessment', 'of', 'patients', 'wit
h', 'coronary', 'artery', 'disease', 'controversies', 'in', 'cardiology', 'ii']
```

- Tagging Sentences using gensim.models.doc2vec.TaggedDocument

```
[TaggedDocument(words=['drugs', 'and', 'the', 'high', 'cost', 'of', 'health', 'care'], tags=[0]),
 TaggedDocument(words=['surgical', 'approach', 'for', 'patients', 'with', 'unstable', 'angina', 'pectoris', 'role', 'of', 'th
e', 'response', 'to', 'initial', 'medical', 'therapy', 'and', 'intraaortic', 'balloon', 'pumping', 'in', 'perioperative', 'co
mplications', 'after', 'aortocoronary', 'bypass', 'grafting'], tags=[1]),
 TaggedDocument(words=['value', 'and', 'limitations', 'of', 'the', 'response', 'to', 'exercise', 'in', 'the', 'assessment',
'of', 'patients', 'with', 'coronary', 'artery', 'disease', 'controversies', 'in', 'cardiology', 'ii'], tags=[2]),
 TaggedDocument(words=['the', 'results', 'of', 'exercise', 'testing', 'provide', 'probabibity', 'statement', 'rather', 'tha
n', 'definitive', 'answer', 'regarding', 'the', 'existence', 'of', 'coronary', 'disease'], tags=[3]),
```

- Instantiate Doc2Vec model

```
#Instantiating the Doc2Vec model
model = gensim.models.doc2vec.Doc2Vec(data_training, vector_size=4, window=2, min_count=3, workers=4)
```

- Retrieve vectors after passing the data

```
['Acetic acid']
['An antimicrobial agent used to treat susceptible infections of the external auditory canal.']
[ 0.09468366  0.09216903  0.11456724 -0.04980076]


['Acetohexamide']
['Used in the management of diabetes mellitus type 2 (adult-onset).']
[-0.0573249  -0.07823293 -0.12070145  0.024875  ]


['Acetohydroxamic acid']
['A synthetic urea derivative used to treat urea splitting bacterial infections of the urinary tract.']
[ 0.01365925  0.0306412  -0.12466621 -0.0331239 ]
```

Novelty of Module 2:

- Using Doc2Vec model rather than Word2Vec model

Doc2vec will be better for classification since it will aggregate the docs for a particular drug and summarise them in a vector. For an unknown doc you can directly test similarity between that doc's vector and drug vectors or use the vectors as features for other ML algorithms.

Implementation of Module 3:

- BioBERT is a pre-trained biomedical language representation model for biomedical text mining.
- BioBERT pretrained weights (PubMed, PMC + Wikipedia) downloaded from https://github.com/naver/biobert-pretrained

Input 1 :

```
DDI-DrugBank.d716.s2$effect$Mazindol$guanethidine$drug1 may reduce the effects of drug2 (Ismelin).
DDI-DrugBank.d716.s2$effect$Mazindol$Ismelin$drug1 may reduce the effects of drug0 (drug2).
DDI-DrugBank.d716.s2$none$guanethidine$Ismelin$drug0 may reduce the effects of drug1 (drug2).
```

Input 2 :

```
BERT Tokenized output
{[101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108, 108,
1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108,
108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914,
108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Loading Input data function:

```
def loadInstance(fp,fs):
    instance=[]
    instanceResult=[]
    entity1instance=[]
    entity2instance=[]
    result=[]
    with open(fp,'rt',encoding='utf-8') as data_in:
        for line in data_in:
            lines=line.split("$")
            ent1=lines[2].strip((" "))
            ent2=lines[3].strip((" "))
            instanceResult.append(lines[1].strip(" "))
            entity1instance.append(ent1)
            entity2instance.append(ent2)
    data_in.close()
    with open(fs, 'rt', encoding='utf-8') as data_in:
        for line in data_in:
            line=line.strip("\n")
            line=line.strip()
            instance.append(line)
    for line in instanceResult:
        instr=line.strip("\n")
        if instr=="none":
            instresult=0
        if instr=="mechanism":
            instresult=1
        if instr=="effect":
            instresult=2
        if instr=="advise":
            instresult=3
        if instr=="int":
            instresult=4
        result.append(instresult)

    #one hot
    result = array(result)
    # one hot encode
    result = to_categorical(result)

    return instance,result,entity1instance,entity2instance
```

Model function:

```
def bioBERTmodel_GRU_Att():
    e1_kno = Input(shape=(1,), dtype='float32', name='kno_e1')
    e2_kno = Input(shape=(1,), dtype='float32', name='kno_e2')
    bert_token_input = Input(shape=(250,), name='bert_token')
    bert_segment_input = Input(shape=(250,), name='bert_segment')
    bert_m1 = Input(shape=[250], name='bert_m1')
    bert_m2 = Input(shape=[250], name='bert_m2')
    bert_model = load_trained_model_from_checkpoint(config_path,
                                                    checkpoint_path,
                                                    seq_len=250)

    wordVector = bert_model([bert_token_input, bert_segment_input])

    for l in bert_model.layers:
        l.trainable = False
    doc_embedding_layer = Embedding(len(doc_vec_embeding), 200, mask_zero=True, trainable=True,
                                    weights=[doc_vec_embeding])
    e1_doc_vec = doc_embedding_layer(e1_kno)
    e2_doc_vec = doc_embedding_layer(e2_kno)
    e1_doc_vec = Lambda(change_shape, output_shape=out_change_shape)(e1_doc_vec)
    e2_doc_vec = Lambda(change_shape, output_shape=out_change_shape)(e2_doc_vec)
    e1_doc_vec = Dense(768, activation='relu')(e1_doc_vec)
    e2_doc_vec = Dense(768, activation='relu')(e2_doc_vec)
    e1_bert = Lambda(get_entity_vector_zhou, output_shape=get_entity_shape)([wordVector, bert_m1])
    e2_bert = Lambda(get_entity_vector_zhou, output_shape=get_entity_shape)([wordVector, bert_m2])
    entity_dense = Dense(768, activation='relu')
    e1_bert_vec = entity_dense(e1_bert)
    e2_bert_vec = entity_dense(e2_bert)
    # bert_sub = Subtract()([e1_bert, e2_bert])
    e1_all = concatenate([e1_doc_vec, e1_bert_vec], axis=-1)
    e2_all = concatenate([e2_doc_vec, e2_bert_vec], axis=-1)
    all_sub = Subtract()([e1_all, e2_all])
    encoded_seq = Bidirectional(GRU(768, dropout=0.5, recurrent_dropout=0.5, return_sequences=True))(wordVector)
    slice_1 = Lambda(slice, arguments={'h1': 249, 'h2': 250})(encoded_seq)
    slice_1 = Lambda(change_shape, output_shape=out_change_shape)(slice_1)
    att_all_sub = NormalAttention()([all_sub, encoded_seq])
    z = concatenate([slice_1, att_all_sub ])
    z = Dropout(0.3)(z)
    z = Dense(256, activation='tanh')(z)
    main_output = Dense(5, activation='softmax', name='main_output')(z)
    model = Model(inputs=[bert_token_input, bert_segment_input, e1_kno, e2_kno, bert_m1, bert_m2], outputs=main_output)
    model.compile(optimizer="Adam", loss='categorical_crossentropy', metrics=['accuracy'])
    print(model.summary())
    return model
```

Model Summary

Group	Hyper-parameter	Value
Embedding Layer	Doc2Vec embedding size	200
	BioBERT embedding size	768
	Max sentence length	250
	BERT output layer number	4
BiGRU Layer	BiGRU output size	1536
	Drop out	0.5
Attention Layer	Attention output size	1536
	Drop out	0.3
Output layer	MLP output size	256
Training	Learning rate	0.001
	Batch size	128
	Training epoch	100

Layer (type)	Output Shape	Param #	Connected to
bert_token (InputLayer)	[(None, 250)]	0	
bert_segment (InputLayer)	[(None, 250)]	0	
kno_e1 (InputLayer)	[(None, 1)]	0	
kno_e2 (InputLayer)	[(None, 1)]	0	
model_1 (Functional)	(None, 250, 768)	107518464	bert_token[0][0] bert_segment[0][0]
embedding (Embedding)	(None, 1, 200)	1780800	kno_e1[0][0] kno_e2[0][0]
bert_m1 (InputLayer)	[(None, 250)]	0	
bert_m2 (InputLayer)	[(None, 250)]	0	
lambda (Lambda)	(None, 200)	0	embedding[0][0]
lambda_2 (Lambda)	(None, 768)	0	model_1[0][0] bert_m1[0][0]
lambda_1 (Lambda)	(None, 200)	0	embedding[1][0]
lambda_3 (Lambda)	(None, 768)	0	model_1[0][0] bert_m2[0][0]
dense (Dense)	(None, 768)	154368	lambda[0][0]
dense_2 (Dense)	(None, 768)	590592	lambda_2[0][0] lambda_3[0][0]
dense_1 (Dense)	(None, 768)	154368	lambda_1[0][0]
bidirectional (Bidirectional)	(None, 250, 1536)	7087104	model_1[0][0]
concatenate (Concatenate)	(None, 1536)	0	dense[0][0] dense_2[0][0]
concatenate_1 (Concatenate)	(None, 1536)	0	dense_1[0][0] dense_2[1][0]
lambda_4 (Lambda)	(None, 1, 1536)	0	bidirectional[0][0]
subtract (Subtract)	(None, 1536)	0	concatenate[0][0] concatenate_1[0][0]
lambda_5 (Lambda)	(None, 1536)	0	lambda_4[0][0]
normal_attention (NormalAttenti	(None, 1536)	3073	subtract[0][0] bidirectional[0][0]
concatenate_2 (Concatenate)	(None, 3072)	0	lambda_5[0][0] normal_attention[0][0]
dropout (Dropout)	(None, 3072)	0	concatenate_2[0][0]
dense_3 (Dense)	(None, 256)	786688	dropout[0][0]
main_output (Dense)	(None, 5)	1285	dense_3[0][0]

Results

Intermediate Output 1 :

```
traininput shape: (25966, 154)
Traininput[0]: [ 620  217 1127 1925   16    4    3    1    5 1109   88   22  247   36
  606    1  177    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0]
testinput shape: (5264, 154)
Testinput[0]: [217 945   2   4  86  36 274  29   6 461   2 253   1  10   3  10   6 264
   2   1   5   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0]
```

Epochs

```
Epoch 1/10
183/183 [==============================] - 590s 3s/step - loss: 0.4696 - accuracy: 0.8519
Epoch 2/10
183/183 [==============================] - 573s 3s/step - loss: 0.2667 - accuracy: 0.8997
Epoch 3/10
183/183 [==============================] - 575s 3s/step - loss: 0.1957 - accuracy: 0.9258
Epoch 4/10
183/183 [==============================] - 577s 3s/step - loss: 0.1444 - accuracy: 0.9481
Epoch 5/10
183/183 [==============================] - 580s 3s/step - loss: 0.1207 - accuracy: 0.9567
Epoch 6/10
183/183 [==============================] - 577s 3s/step - loss: 0.1029 - accuracy: 0.9619
Epoch 7/10
183/183 [==============================] - 575s 3s/step - loss: 0.0946 - accuracy: 0.9655
Epoch 8/10
183/183 [==============================] - 576s 3s/step - loss: 0.0814 - accuracy: 0.9704
Epoch 9/10
183/183 [==============================] - 579s 3s/step - loss: 0.0746 - accuracy: 0.9724
Epoch 10/10
183/183 [==============================] - 581s 3s/step - loss: 0.0695 - accuracy: 0.9750
Model evaluation
731/731 [==============================] - 364s 491ms/step - loss: 0.0444 - accuracy: 0.9830
```

Results

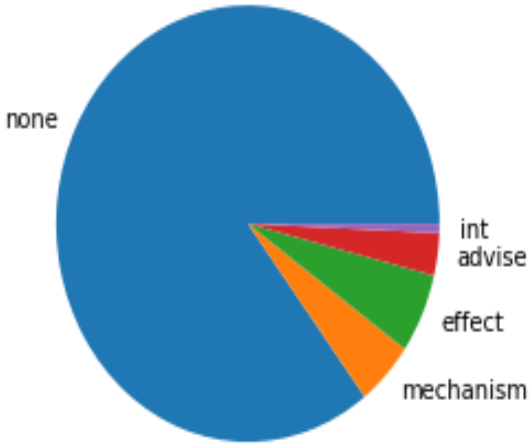
Performance metrics

```
get effect prf:
p - 0.8058823529411765
r - 0.8896103896103896
f - 0.845679012345679
get mechanism prf:
p - 0.7534246575342466
r - 0.88
f - 0.8118081180811809
```

```
get advise prf:
p - 0.8571428571428571
r - 0.8780487804878049
f - 0.8674698795180722
get int prf:
p - 0.9411764705882353
r - 0.8888888888888888
f - 0.9142857142857143
```

```
get none prf:
p - 0.9880404783808647
r - 0.9684400360685302
f - 0.9781420765027322
Overall model prf:
p - 0.7990543735224587
r - 0.8918205804749341
f - 0.8428927680798006
```

- Our dataset(DDI Corpus) has an unbalanced number of None DDI type



Performance measures

1. Precision

Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the

total number of positive predictions.

2. Recall

The ability of a model to find all the relevant cases within a data set. Mathematically, we define recall as the number of true positives divided by the number of true positives plus the number of false negatives

3. F - Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. We use a micro averaged F score to evaluate our model.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

Test cases and Validation:

Module 1

Case 1

Input:

```
<sentence id="DDI-DrugBank.d610.s0" text="Pharmacokinetic properties of abacavir were not altered by the addition of either lamivudine or zidovudine or the combination of lamivudine and zidovudine. ">
  <entity id="DDI-DrugBank.d610.s0.e0" charOffset="30-37" type="drug" text="abacavir"/>
  <entity id="DDI-DrugBank.d610.s0.e1" charOffset="82-91" type="drug" text="lamivudine"/>
  <entity id="DDI-DrugBank.d610.s0.e2" charOffset="96-105" type="drug" text="zidovudine"/>
  <entity id="DDI-DrugBank.d610.s0.e3" charOffset="129-138" type="drug" text="lamivudine"/>
  <entity id="DDI-DrugBank.d610.s0.e4" charOffset="144-153" type="drug" text="zidovudine"/>
  <pair id="DDI-DrugBank.d610.s0.p0" e1="DDI-DrugBank.d610.s0.e0" e2="DDI-DrugBank.d610.s0.e1" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p1" e1="DDI-DrugBank.d610.s0.e0" e2="DDI-DrugBank.d610.s0.e2" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p2" e1="DDI-DrugBank.d610.s0.e0" e2="DDI-DrugBank.d610.s0.e3" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p3" e1="DDI-DrugBank.d610.s0.e0" e2="DDI-DrugBank.d610.s0.e4" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p4" e1="DDI-DrugBank.d610.s0.e1" e2="DDI-DrugBank.d610.s0.e2" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p5" e1="DDI-DrugBank.d610.s0.e1" e2="DDI-DrugBank.d610.s0.e3" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p6" e1="DDI-DrugBank.d610.s0.e1" e2="DDI-DrugBank.d610.s0.e4" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p7" e1="DDI-DrugBank.d610.s0.e2" e2="DDI-DrugBank.d610.s0.e3" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p8" e1="DDI-DrugBank.d610.s0.e2" e2="DDI-DrugBank.d610.s0.e4" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s0.p9" e1="DDI-DrugBank.d610.s0.e3" e2="DDI-DrugBank.d610.s0.e4" ddi="false"/>
</sentence>
```


Intermediate Output:


```
DDI-DrugBank.d610.s0$none$abacavir$lamivudine$Pharmacokinetic properties of drug1 were not altered by the addition of either drug2 or drug0 or the combination of drug0 and drug0.
DDI-DrugBank.d610.s0$none$abacavir$zidovudine$Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug2 or the combination of drug0 and drug0.
DDI-DrugBank.d610.s0$none$abacavir$lamivudine$Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug0 or the combination of drug2 and drug0.
DDI-DrugBank.d610.s0$none$abacavir$zidovudine$Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug0 or the combination of drug0 and drug2.
DDI-DrugBank.d610.s0$none$lamivudine$zidovudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug2 or the combination of drug0 and drug0.
DDI-DrugBank.d610.s0$none$lamivudine$lamivudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug0 or the combination of drug2 and drug0.
DDI-DrugBank.d610.s0$none$lamivudine$zidovudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug0 or the combination of drug0 and drug2.
DDI-DrugBank.d610.s0$none$zidovudine$lamivudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug1 or the combination of drug2 and drug0.
DDI-DrugBank.d610.s0$none$zidovudine$zidovudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug1 or the combination of drug0 and drug2.
DDI-DrugBank.d610.s0$none$lamivudine$zidovudine$Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug0 or the combination of drug1 and drug2.
```


Final Output :


```
BERT Tokenized output
{[101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108, 108,
1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108,
108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914,
108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150,
17914, 108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234,
1150, 17914, 108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104,
```

Case 2

Input


```
<sentence id="DDI-DrugBank.d610.s1" text="No clinically significant changes to lamivudine or zidovudine pharmacokinetics were observed following concomitant administration of abacavir. ">
  <entity id="DDI-DrugBank.d610.s1.e0" charOffset="37-46" type="drug" text="lamivudine"/>
  <entity id="DDI-DrugBank.d610.s1.e1" charOffset="51-60" type="drug" text="zidovudine"/>
  <entity id="DDI-DrugBank.d610.s1.e2" charOffset="133-140" type="drug" text="abacavir"/>
  <pair id="DDI-DrugBank.d610.s1.p0" e1="DDI-DrugBank.d610.s1.e0" e2="DDI-DrugBank.d610.s1.e1" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s1.p1" e1="DDI-DrugBank.d610.s1.e0" e2="DDI-DrugBank.d610.s1.e2" ddi="false"/>
  <pair id="DDI-DrugBank.d610.s1.p2" e1="DDI-DrugBank.d610.s1.e1" e2="DDI-DrugBank.d610.s1.e2" ddi="false"/>
</sentence>
```


Intermediate Output

```
DDI-DrugBank.d610.s1$none$lamivudine$zidovudine$No clinically significant changes to drug1 or drug2 pharmacokinetics were observed following concomitant administration of drug0.
DDI-DrugBank.d610.s1$none$lamivudine$abacavir$No clinically significant changes to drug1 or drug0 pharmacokinetics were observed following concomitant administration of drug2.
DDI-DrugBank.d610.s1$none$zidovudine$abacavir$No clinically significant changes to drug0 or drug1 pharmacokinetics were observed following concomitant administration of drug2.
```

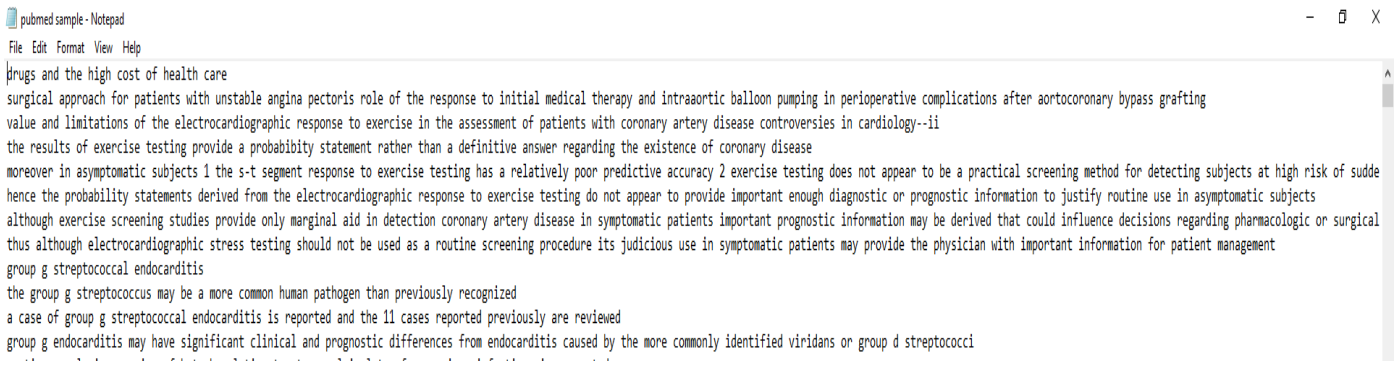
Final Output


```
BERT Tokenized output
{[101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108, 108,
1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914, 108,
108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150, 17914,
108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234, 1150,
17914, 108, 108, 1033, 1105, 108, 108, 1037, 1113, 170, 2366, 3142, 119, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0], [101, 1103, 2686, 4693, 1103, 5417, 1115, 1103, 27236, 6834, 1548, 3943, 1336, 4467, 1107, 1103, 3472, 1104, 1234,
```


Module 2:

Case 1

Input:



Intermediate Output

```
[TaggedDocument(words=['drugs', 'and', 'the', 'high', 'cost', 'of', 'health', 'care'], tags=[0]),
 TaggedDocument(words=['surgical', 'approach', 'for', 'patients', 'with', 'unstable', 'angina', 'pectoris', 'role', 'of', 'th
e', 'response', 'to', 'initial', 'medical', 'therapy', 'and', 'intraaortic', 'balloon', 'pumping', 'in', 'perioperative', 'co
mplications', 'after', 'aortocoronary', 'bypass', 'grafting'], tags=[1]),
 TaggedDocument(words=['value', 'and', 'limitations', 'of', 'the', 'response', 'to', 'exercise', 'in', 'the', 'assessment',
'of', 'patients', 'with', 'coronary', 'artery', 'disease', 'controversies', 'in', 'cardiology', 'ii'], tags=[2]),
 TaggedDocument(words=['the', 'results', 'of', 'exercise', 'testing', 'provide', 'probabibity', 'statement', 'rather', 'tha
n', 'definitive', 'answer', 'regarding', 'the', 'existence', 'of', 'coronary', 'disease'], tags=[3]),
```

Final Output

```
['Acetic acid']
['An antimicrobial agent used to treat susceptible infections of the external auditory canal.']
[ 0.09468366  0.09216903  0.11456724 -0.04980076]


['Acetohexamide']
['Used in the management of diabetes mellitus type 2 (adult-onset).']
[-0.0573249  -0.07823293 -0.12070145  0.024875  ]


['Acetohydroxamic acid']
['A synthetic urea derivative used to treat urea splitting bacterial infections of the urinary tract.']
[ 0.01365925  0.0306412  -0.12466621 -0.0331239 ]
```

Case 2

Final Output

```
['Acetic acid']
['An antimicrobial agent used to treat susceptible infections of the external auditory canal.']
[ 0.04670344  0.00381992 -0.01906684 -0.0496716  -0.01543501 -0.01917518
 -0.00033465 -0.0254167   0.01911375 -0.01553548]


['Acetohexamide']
['Used in the management of diabetes mellitus type 2 (adult-onset).']
[-0.02120379 -0.0088813   0.03869624 -0.04204399 -0.02698744  0.01787479
 -0.02220657 -0.0469635   0.00622886  0.04800335]


['Acetohydroxamic acid']
['A synthetic urea derivative used to treat urea splitting bacterial infections of the urinary tract.']
[ 0.03238273 -0.02587391  0.01618962 -0.00059579 -0.01458078  0.00330345
 -0.00371267  0.02099394  0.0022108   0.04190589]
```

Case 3

Final Output

```
['Bismuth subgallate']
['A medication used to deodorize flatulence and stools as well as hemostasis in soft tissue surgery.']
[-0.0228847   0.00965731  0.04986431 -0.02498963  0.00532944  0.04160086
 -0.02262641  0.03174618  0.02137806  0.01642643]


['Bismuth subnitrate']
['A medication used as an antacid.']
[ 0.0276839   0.0390879  -0.00369716  0.04307287  0.00066668  0.03519141
 -0.01955034  0.03870724  0.00455606  0.03497594]
```

Case 4

Final Output

```
['Chloroprocaine']
['A local anesthetic agent indicated for intrathecal injection in adults for the production of subarachnoid block, or spinal
anesthesia.']
[-0.03169756 -0.02366873 -0.00778214  0.00121389 -0.01593266  0.02688211
 -0.00636586  0.0253257  -0.02865078 -0.00623138  0.00741835 -0.03330389
 -0.02085052 -0.00730552 -0.00629711]


['Chloroquine']
['An antimalarial drug used to treat susceptible infections with P. vivax, P. malariae, P. ovale, and P. falciparum. It is al
so used for second line treatment for rheumatoid arthritis.']
[-0.0114064   0.03270555 -0.01751994  0.00331079  0.02723997  0.00093886
 -0.02498939  0.00414133  0.00249989 -0.01925991 -0.00259304  0.01170385
  0.0129979   0.026882   -0.03128897]
```

Case 5

Final Output

```
['Acetic acid']
['An antimicrobial agent used to treat susceptible infections of the external auditory canal.']
[-0.00340043  0.00054851 -0.00270276  0.00328364  0.00471617 -0.00277028
 -0.00106453 -0.00307904 -0.00132303 -0.00178873 -0.00273731 -0.00445481
 -0.00410731  0.0026263   0.00338543  0.00453486  0.00321395 -0.00495021
 -0.00354544 -0.00235927  0.00287571 -0.00232139 -0.00327961  0.00018743
 -0.00214969  0.00457792 -0.00027754  0.00367443 -0.00016999  0.00413335
 -0.00057316  0.00251156 -0.00378326 -0.00155643  0.00109739 -0.00091643
  0.00179497  0.00496106  0.00432376  0.00414048 -0.0006349  -0.00167338
 -0.00340131 -0.00369294  0.00220246 -0.00214092 -0.00081164 -0.00310177
  0.00172474 -0.00329738 -0.00368625 -0.00369382  0.00485082  0.0031589
 -0.00310202 -0.00460259  0.00498125 -0.00062634  0.00490172  0.00451507
  0.00245556  0.00128513 -0.00315972  0.00205939 -0.00048409  0.00249744
  0.00481931 -0.0028335  -0.00077747  0.00103288 -0.00096797 -0.00137803
  0.00264086  0.00397756 -0.00365166 -0.00264061  0.00157518  0.00496072
 -0.00371674 -0.00365441 -0.00023242 -0.0033212  -0.00461313 -0.00031549
  0.00108935  0.00448747 -0.00109565  0.00239843  0.00091771 -0.00284047
  0.0038945   0.0048294  -0.00328487 -0.00069263  0.00262663 -0.00310167
 -0.00112285 -0.00345419  0.00118939 -0.00238266]
```

Module 3:

Case 1

Input

```
Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug2 or the combination of drug0 and drug0.
Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug0 or the combination of drug2 and drug0.
Pharmacokinetic properties of drug1 were not altered by the addition of either drug0 or drug0 or the combination of drug0 and drug2.
Pharmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug2 or the combination of drug0 and drug0.
harmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug0 or the combination of drug2 and drug0.
Pharmacokinetic properties of drug0 were not altered by the addition of either drug1 or drug0 or the combination of drug0 and drug2.
Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug1 or the combination of drug2 and drug0.
Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug1 or the combination of drug0 and drug2.
Pharmacokinetic properties of drug0 were not altered by the addition of either drug0 or drug0 or the combination of drug1 and drug2.
No clinically significant changes to drug1 or drug2 pharmacokinetics were observed following concomitant administration of drug0.
No clinically significant changes to drug1 or drug0 pharmacokinetics were observed following concomitant administration of drug2.
No clinically significant changes to drug0 or drug1 pharmacokinetics were observed following concomitant administration of drug2.
drug1 has no effect on the pharmacokinetic properties of drug2.
```

Intermediate Output

```
traininput shape: (25966, 154)
Traininput[0]: [ 620  217 1127 1925   16    4    3    1    5 1109   88   22  247   36
  606    1  177    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0]
```

Final Output

```
Testing predictions
effect
effect
effect
effect
effect
effect
effect
effect
effect
none
none
none
advise
```

Case 2

Input

It is recommended that drug1 and drug2 not be administered simultaneously.
drug1: When studied in stable renal transplant patients, drug2, USP (MODIFIED) pharmacokinetics were unaffected by steady state dosing of drug0.
drug1: When studied in stable renal transplant patients, drug0, USP (MODIFIED) pharmacokinetics were unaffected by steady state dosing of drug2.
drug0: When studied in stable renal transplant patients, drug1, USP (MODIFIED) pharmacokinetics were unaffected by steady state dosing of drug2.
drug1/drug2: may be taken with Myfortic;
drug1/Ganciclovir: may be taken with drug2;

Intermediate Output

```
testinput shape: (5264, 154)
Testinput[0]: [217 945   2   4  86  36 274  29   6 461   2 253   1  10   3  10   6 264
   2   1   5   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0]
```

Final Output

```
advise
effect
effect
effect
effect
effect
```

References

1. Yu Zhu, Lishuang Li, Hongbin Lu, Anqioa Zhou and Xueyang Qin, "Extracting drug-drug interactions from texts with BioBERT and multiple entity-aware attentions", in Journal of Biomedical Informatics, Volume 106, June 2020
2. Xioa-Ying Yan, Peng-Wei Yin, Xiao-Meng Wu, Jia-Xin Han, "Prediction of drug-drug interaction types with the unified embedding features from drug similarity networks", in Frontiers in Pharmacology Journal, December 2021
3. Sunil Kumar Shahu, Ashish Anand, "Drug-drug interaction extraction from Biomedical texts using Long Short-Term Memory network", in Journal of Biomedical Informatics, 2018
4. Sangrak Lim, Kyubum Lee, Jaewoo Kang, "Drug-drug interaction extraction from the literature using a recursive neural network", in The Public Library of Science, 2018
5. Prashant Kumar Shukla, Piyush Kumar Shukla, Poonam Sharma, Paresh Rawat, Jashwant Samar, Rahul Moriwal, Manjit Kaur, "Efficient Prediction of drug-drug interaction using deep learning models", IET Syst Biol, August 2020
6. Suyu Mei, Kun Zhang, "A machine learning framework for predicting drug-drug interactions", Scientific reports, September 2021
7. I. Segura-Bedmar, P. Martinez, D. Sanchez-Cisneros," The 1st ddi extraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts", 2011
8. M. Herrero-Zazo, I. Segura-Bedmar, P. Martinez, T. Declerck, "The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions", J. Biomed. Inform. 46, 2013
9. S. Liu, B. Tang, Q. Chen, X. Wang, Drug-drug interaction extraction via convolutional neural network Comp. Math. Meth. Med., 2016.
10. D. Li, H. Ji, Syntax-aware multi-task graph convolutional networks for biomedical relation extraction, in: Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis, Association for Computational Linguistics, Hong Kong, 2019.
11. Prescription drug information, interactions and side effects. 2000; Available from: https://www.drugs.com/ Baxter K., Preston CL. Stockley's drug interactions Pharmaceutical Press; London; 2015.
12. S. Kim, H. Liu, L. Yeganova, W.J. Wilbur Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach J. Biomed. Inform., 55 (2015).

13. H. Gurulingappa, L. Toldo, A.M. Rajput, J.A. Kors, A. Taweel, Y. Tayrouz Automatic detection of adverse events to predict drug label changes using text and data mining techniques Pharmacoepidemiol. Drug Saf, (2013).

14. J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Biobert: pre-trained biomedical language representation model for biomedical text mining, 2019.

15. I. Segura-Bedmar, P. Martinez, D. Sanchez-Cisneros, The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts, vol, 2011.