Adverse Drug Reaction Extraction

Submitted in partial fulfillment of the requirements
for the degree of

Master of Science
in
Data Science

Charith Musku (cmusku)
Bhargav Kandlagunta (bkandla)
Bhagya Reddy (bhreddy)

Under the kind guidance of
Prof. Damir Cavar

School of Informatics and Computing
Indiana University Bloomington
December, 2019

CONTENTS

LIST OF FIGURES

ABSTRACT

The aim of the project is to identify and extract the adverse effects of the drugs from text data using sequencelabelling. The data for the problem is taken from CADEC[5] adverse drug reaction annotations. Data is processed and build BIO format tagged annotations. Different models were trained for this task. Conventional models CRF, BiLSTM were implemented along with several pre-trained models based on BERT.

1 INTRODUCTION

Due to limitations in clinical trials, not all the potential side effects of medications are discovered prior to the drug going to market [6]. Adverse drug reactions that remain unknown create major concerns in public health. They are responsible for thousands of incidents of death or serious injury, as well as millions of hospitalisations. Consequently, they cost billions of dollars to the healthcare systems around the world.

An Adverse Drug Reaction (ADR) is an illness or injury occurring after a drug(medication) is used at the recommended dosage, for recommended symptoms. Termed as "pharmacovigilance", is especially concerned with identifying previously unreported adverse reactions of the drugs after they are released into the market. Drug quality can be assessed and actions can be taken based on the results. Text mining over different sources of information, such as electronic health records and patient reports on health forums, can be one way of finding such potential adverse reactions. With enormous textual data from patient responses in survey collections, it is difficult to manually interpret the adverse reactions. Some of the current advancements in NLP techniques can be a potential solution to such problems.

2 DATASET AND APPROACH

2.1 About

CSIRO Adverse Drug Event Corpus (Cadec) is a new rich annotated corpus of medical forum posts on patient-reported Adverse Drug Events (ADEs). The corpus is sourced from posts on social media, and contains text that is largely written in colloquial language and often deviates from formal English grammar and punctuation rules. Annotations contain mentions of concepts such as drugs, adverse effects, symptoms, and diseases linked to their corresponding concepts in controlled vocabularies, i.e., SNOMED Clinical Terms and MedDRA.

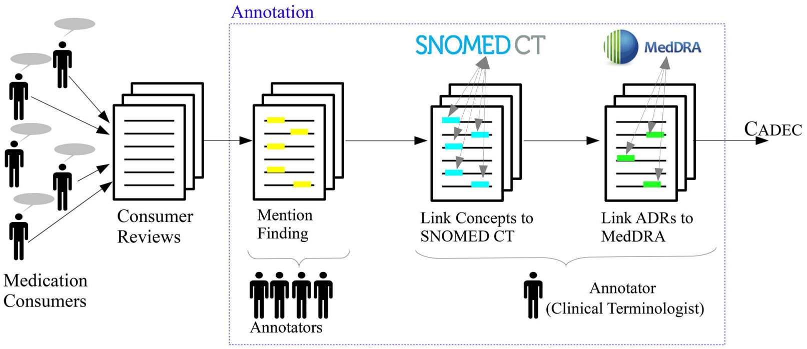
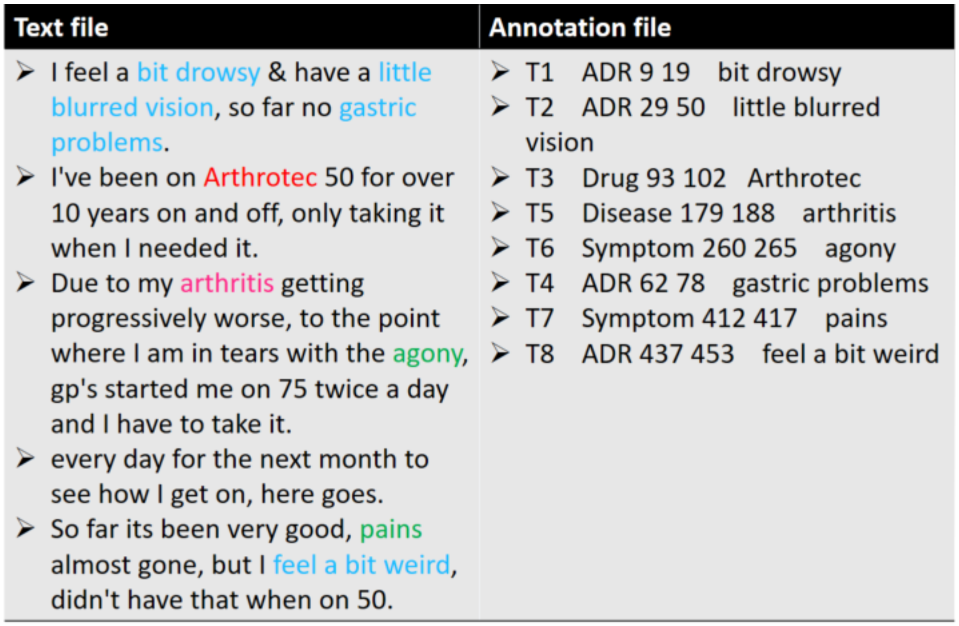


Figure 1: CADEC dataset preparation

The quality of the annotations is ensured by annotation guidelines, multi-stage annotations, measuring inter-annotator agreement, and final review of the annotations by a clinical terminologist. This corpus is useful for studies in the area of information extraction, or more generally text mining, from social media to detect possible adverse drug reactions from direct patient reports.

There are 1200 data samples in the dataset. Each text file contains a single patiet's response about the drug usage and any adverse reactions caused because of the drug. Typically each file has 5-6 lines of response. And every file has a corresponding annotation file that contains annotation tags for the text file content. A sample of test file and annotation file is shown in the figure 2.



Text file	Annotation file
➢ I feel a bit drowsy & have a little blurred vision, so far no gastric problems. ➢ I've been on Arthrotec 50 for over 10 years on and off, only taking it when I needed it. ➢ Due to my arthritis getting progressively worse, to the point where I am in tears with the agony, gp's started me on 75 twice a day and I have to take it. ➢ every day for the next month to see how I get on, here goes. ➢ So far its been very good, pains almost gone, but I feel a bit weird, didn't have that when on 50.	➢ T1 ADR 9 19 bit drowsy ➢ T2 ADR 29 50 little blurred vision ➢ T3 Drug 93 102 Arthrotec ➢ T5 Disease 179 188 arthritis ➢ T6 Symptom 260 265 agony ➢ T4 ADR 62 78 gastric problems ➢ T7 Symptom 412 417 pains ➢ T8 ADR 437 453 feel a bit weird

Figure 2: Sample data files

3 METHODOLOGIES

3.1 Data Pre-processing

- From the text/annotation files, need to extract all the annotation mappings for particular content.
- The input for our problem is taken as a sentence from the text file.
- The data needs to be processed before training the models.

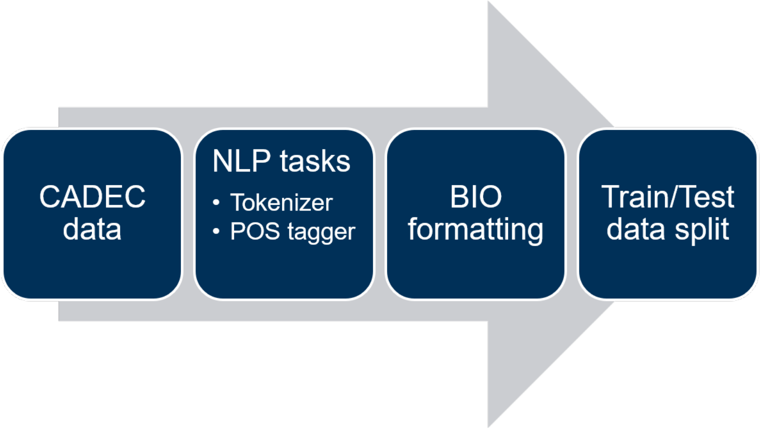


Figure 3: Data Pre-processing Pipeline

3.1.1 *Annotation mapping*

- As mentioned earlier all the annotation tags must be mapped to corresponding text content.
- Tags available in the dataset are:
  - ∗ ADR(Adverse Drug Reaction): bit drowsy,little blurred vision,feel a bit weird
  - ∗ Drug: Arthrotec,Tylenol
  - ∗ Disease: Arthritis,plantarfasciitis
  - ∗ Symptom: Discomfort,infection,inflammation
  - ∗ Finding: Stomach problems,joint soreness

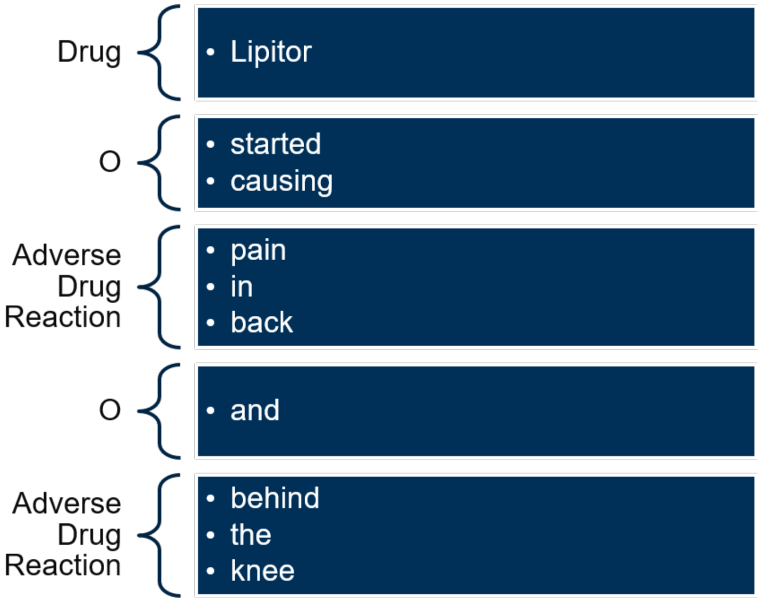


Figure 4: Example Annotation Mapping

3.1.2 *BIO Format*

- While mapping there is another thing that needs to be taken care of.
- Each tagged entity can span for more than one word.
- Thus, the annotations need to be in BIO format which is Beginning, Inside and Out.
  - ∗ O Outside concept. All tokens outside the concepts in which we are interested are labelled as O.
  - ∗ B- Begin of concept, for continuous and nonoverlapping spans.
  - ∗ I- Continuation of concept, for continuous and non-overlapping spans.

Example of BIO format data

- Input sentence: "Intermittent acute pain in both feet, for which I was prescribed Neurontin."
- Intermittent/O
- acute/B-ADR
- pain/I-ADR
- in/I-ADR

- both/I-ADR
- feet/I-ADR
- ,/O
- for/O
- which/O
- I/O
- was/O
- prescribed/O
- Neurontin/B-Drug
- ./O

3.1.3 *Tokenize*

- Each sentence has to be tokenized, which is separating all the tokens in the sentence.
- This is done using any nlp library available.

3.1.4 *POS tagging*

- After tokenizing the input sentences, parts-of-speech(POS) tag for each word has to extracted.
- This is done using nltk pos-tagger
- The POS tags will be useful for CRF model, that takes these tags as features.

After tokenizing and pos-tagging the data is stored in a file with format:
Token POS tag Label
For each

DATA SPLIT

- With 1200 files in the dataset, the split for train, dev and test are as 0.6,0.15 and 0.25 respectively of the entire data.

3.2 Visual Analysis

After the data split, the distributions of the tags are as follows. 'O' tag is excluded from the data.

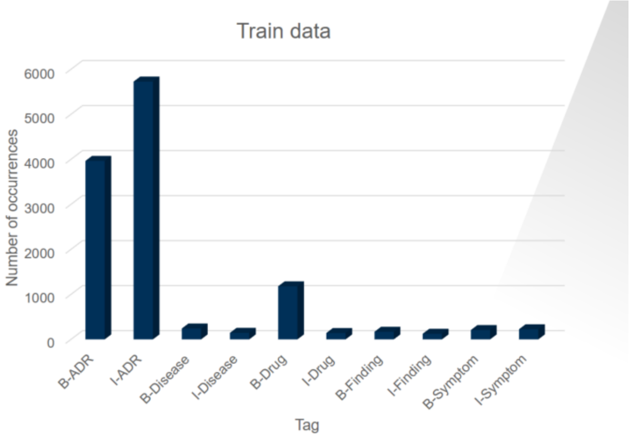


Figure 5: Tags distribution in train data

4 MODELS

Based on the problem statement and data available, several Machine Learning models were implemented and tuned to check the performance.

4.1 CRF

- Conditional Random Fields(CRF): They belong to sequence modeling family of Machine Learning Algorithms.
- Predicts or classifies each token of a sentence in a sequential manner by taking into consideration the words surrounding the current word and their corresponding features.
- CRF is one of the most efficient models used for sequence modeling tasks. Conditional Random Fields are a Discriminative model, and their underlying principle is that they apply Logistic Regression on sequential inputs.
- CRFs are similar to Logistic regression, except that CRF deals with a sequence of predictions unlike logistic regression.

- In logistic regression the inputs are considered to be independent whereas CRF makes use of the features of other variables in the sequence to predict the label for the current word.

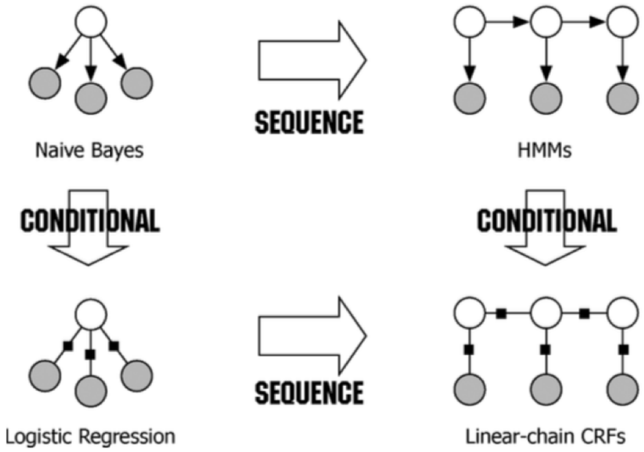


Figure 6: Relation between Sequence Models

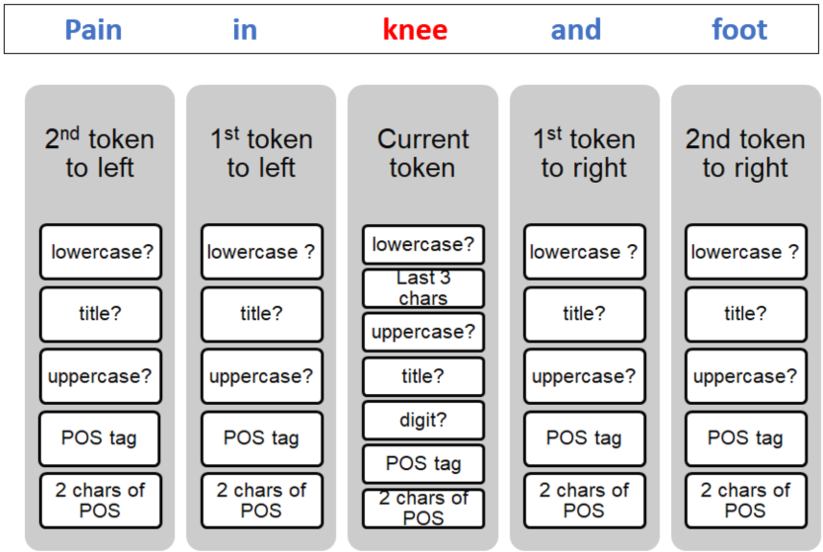


Figure 7: Features of CRF model

4.2 BiLSTM+CRF

- Bi Directional Long short-term memory + CRF
- A BiLSTM network with CRF layer on top of it proved to be state-of-the art techniques for sequence tagging problems before BERT.
- A BiLSTM is a RNN architecture that is extremely useful in studying sequential data(video frames, language).

- Using the context BiLSTM network predicts the probability distribution of each token in a sentence.
- Then CRF layer on top helps finding the best sequence given the distributions from the BiLSTM layer.

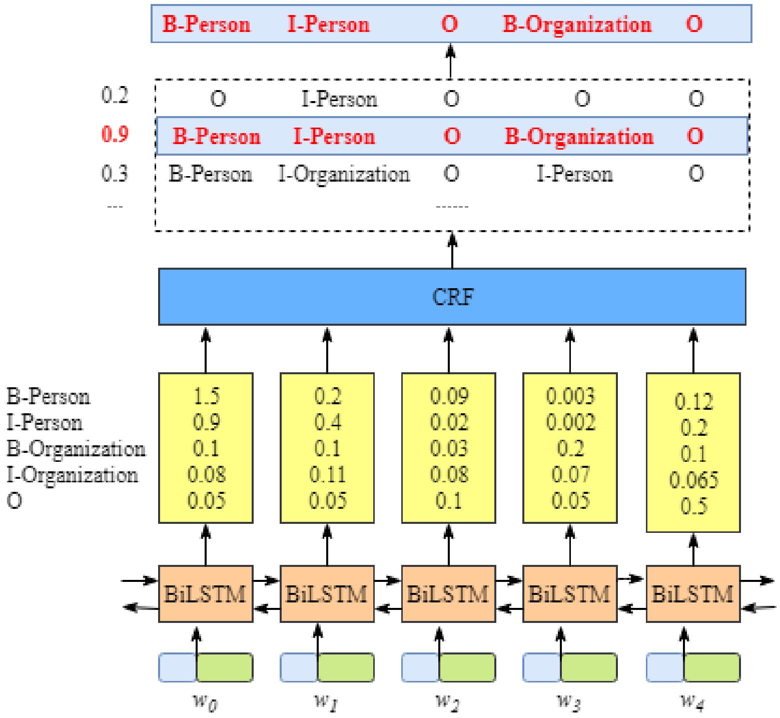


Figure 8: BiLSTM-CRF Architecture

4.3 BERT

4.3.1 *What is BERT?*

- Bidirectional Encoder Representations from Transformers. BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. The detailed workings of Transformer are described in a paper by Google.
- Attention based deep learning model that has given state-of-the-art results on a wide variety of natural language processing tasks.

- It has been pre-trained on Wikipedia and BooksCorpus and requires task-specific fine-tuning.
- Published by Google AI researchers in 2018.
- The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models.

4.3.2 *Why BERT?*

- It is a pre-trained model on a large corpus of natural language on two main tasks:
  - Predicting the masked work in a sentence.
  - Learning if a sentence is in context(that follows) with another sentence given a pair.
- Transfer learning has proved to be a great way to improve the deep learning model performances. BERT has implemented the same into NLP.
- This pre-trained model can be fine-tuned and used effectively for many downstream NLP tasks like NER tagging, Question Answering and Sentiment Analysis.
- We are currently using the CADEC data to fine tune the BERT model for NER task, to predict the tags for each word given a sentence.

4.3.3 *Architecture*

- BERT base – 12 layers (transformer blocks), 12 attention heads, and 110 million parameters.
- Every word is embedded(E) for input to the model.
- Each layer has multi-headed(12) attention computation of the word representation from the previous layer.

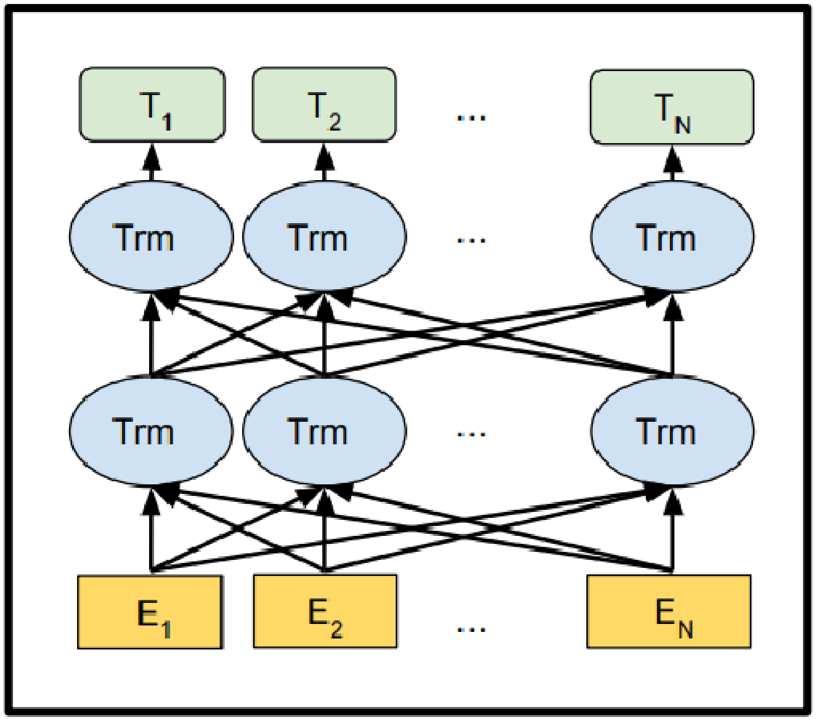


Figure 9: BERT architecture

4.3.4 *Attention Mechanism*

Attention is one of the most influential ideas in the Deep Learning community. Even though this mechanism is now used in various problems like image captioning and others,it was initially designed in the context of Neural Machine Translation using Seq2Seq Models. In terms of Computer vision, Human visual attention allows us to focus on a certain region with "high resolution" (i.e. look at the pointy ear in the yellow box) while perceiving the surrounding image in "low resolution" (i.e. now how about the snowy background and the outfit?), and then adjust the focal point or do the inference accordingly. Given a small patch of an image, pixels in the rest provide clues what should be displayed there. We expect to see a pointy ear in the yellow box because we have seen a dog's nose, another pointy ear on the right, and Shiba's mystery eyes (stuff in the red boxes). However, the sweater and blanket at the bottom would not be as helpful as those doggy features. Similarly, we can explain the relationship between words in one sentence or close context. When we see "eating", we expect to encounter a food word very soon.

The color term describes the food, but probably not so much with "eating" directly.

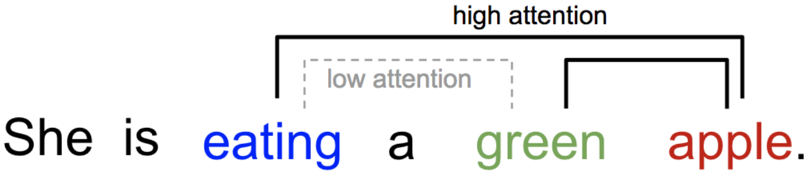
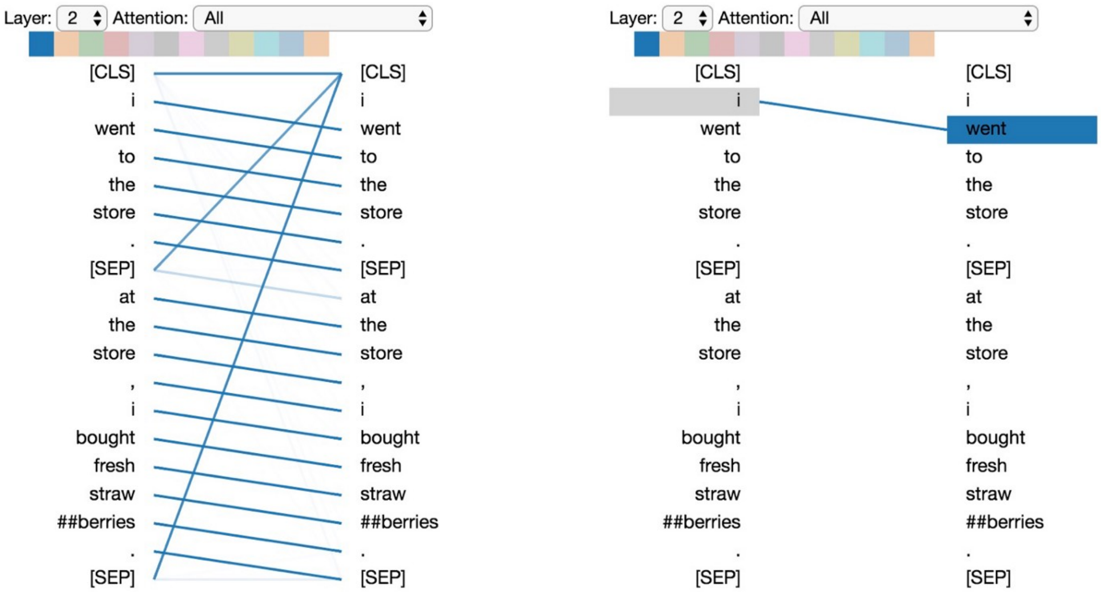


Figure 10: Sentence example of Attention

In a nutshell, attention in the deep learning can be broadly interpreted as a vector of importance weights: in order to predict or infer one element, such as a pixel in an image or a word in a sentence, we estimate using the attention vector how strongly it is correlated with (or "attends to" as you may have read in many papers) other elements and take the sum of their values weighted by the attention vector as the approximation of the target.


Layer: 2 Attention: All

Layer: 2 Attention: All

Figure 11: Attention in BERT

4.3.5 *Limitations*

- Though BERT has proved to be the state of the art model for many downstream tasks, it fails in domain specific tasks.
- This is because of the lack of domain related data corpus and the patters not present in the general language context.

- In our case we are dealing with biomedical data and thus the performance of BERT is not as expected.

4.4 BioBERT

4.4.1 *About*

BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining), which is a domain-specific language representation model pre-trained on large-scale biomedical corpora. With almost the same architecture across tasks, BioBERT largely outperforms BERT and previous state-of-the-art models in a variety of biomedical text mining tasks when pre-trained on biomedical corpora. While BERT obtains performance comparable to that of previous state-of-the-art models, BioBERT significantly outperforms them on the following three representative biomedical text mining tasks: biomedical named entity recognition (0.62% F1 score improvement), biomedical relation extraction (2.80% F1 score improvement) and biomedical question answering (12.24% MRR improvement).

- As the name suggests, this is a model developed to deal with biomedical language processing based on the BERT principles.
- In addition to the general data corpus as in BERT, this model pre trains on medical data from:
  - PubMed abstracts
  - PMC full text articles
- This helps the model to extract and learn the language patterns in biomedical context along with the vocabulary.

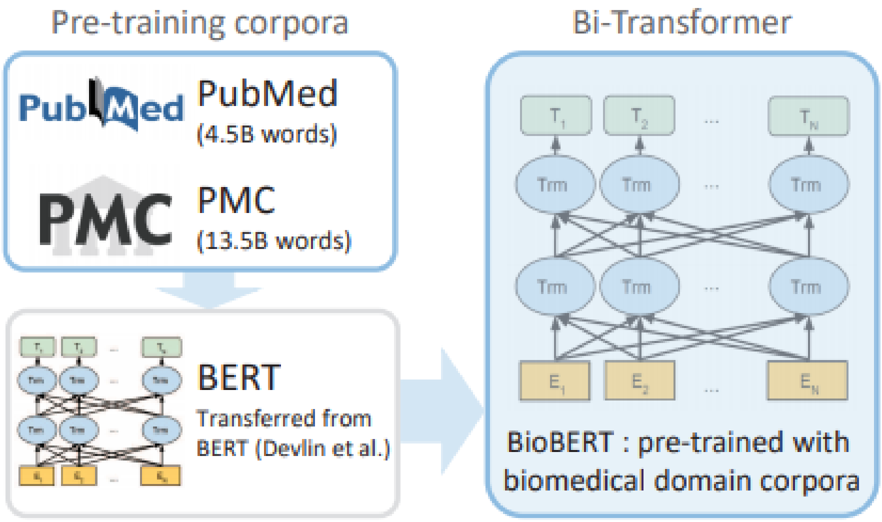
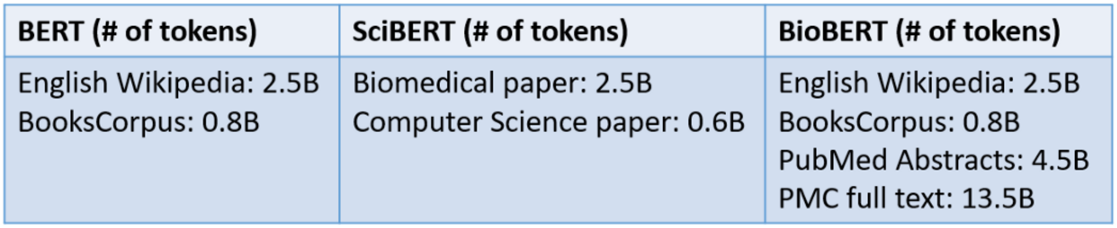


Figure 12: BioBERT Architecture

4.5 SciBERT

SciBERT, a pretrained language model based on BERT (Devlin et al., 2018) to address the lack of high-quality, large-scale labeled scientific data. SciBERT leverages unsupervised pretraining on a large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. It is useful in analyzing and identifying patters from scientific data. The pretrained data comparision of all the BERT models are as shown in the table below.



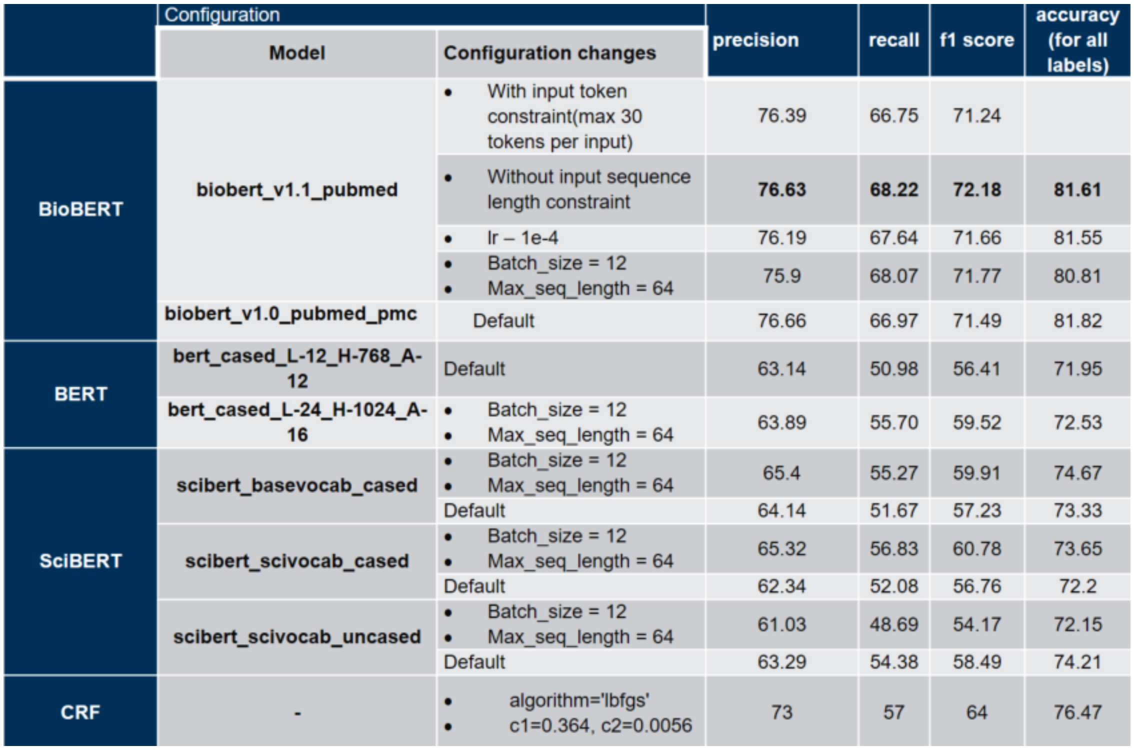
BERT (# of tokens)	SciBERT (# of tokens)	BioBERT (# of tokens)
English Wikipedia: 2.5B BooksCorpus: 0.8B	Biomedical paper: 2.5B Computer Science paper: 0.6B	English Wikipedia: 2.5B BooksCorpus: 0.8B PubMed Abstracts: 4.5B PMC full text: 13.5B

Figure 13: BERT models pre-trained data

5 EXPERIMENTS AND RESULTS

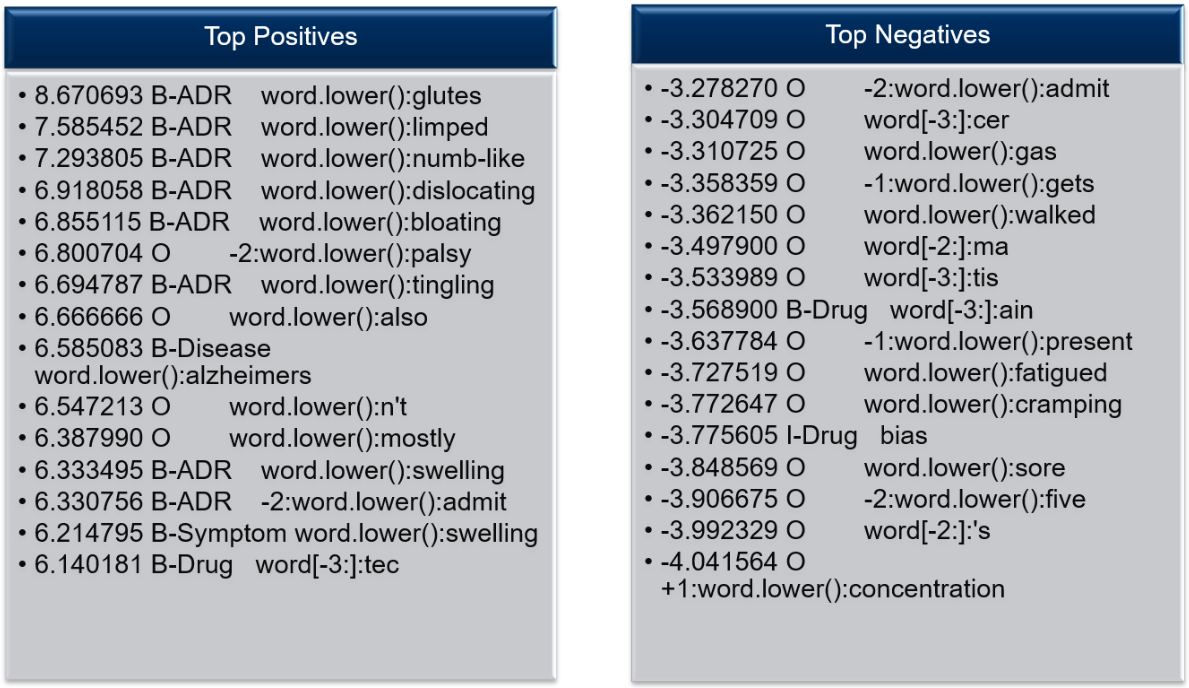
5.1 Model Cnfiguration

- Epochs: 8
- Max Sequence length: 128
- Batch Size train: 32
- Learning Rate: 5e-5
- All metric values are micro averaged



	Configuration		precision	recall	f1 score	accuracy (for all labels)
	Model	**Configuration changes**				
BioBERT	**biobert_v1.1_pubmed**	• With input token constraint(max 30 tokens per input)	76.39	66.75	71.24	
		• Without input sequence length constraint	**76.63**	**68.22**	**72.18**	**81.61**
		• lr – 1e-4	76.19	67.64	71.66	81.55
		• Batch_size = 12 • Max_seq_length = 64	75.9	68.07	71.77	80.81
	biobert_v1.0_pubmed_pmc	Default	76.66	66.97	71.49	81.82
BERT	**bert_cased_L-12_H-768_A-12**	Default	63.14	50.98	56.41	71.95
	bert_cased_L-24_H-1024_A-16	• Batch_size = 12 • Max_seq_length = 64	63.89	55.70	59.52	72.53
SciBERT	**scibert_basevocab_cased**	• Batch_size = 12 • Max_seq_length = 64	65.4	55.27	59.91	74.67
		Default	64.14	51.67	57.23	73.33
	scibert_scivocab_cased	• Batch_size = 12 • Max_seq_length = 64	65.32	56.83	60.78	73.65
		Default	62.34	52.08	56.76	72.2
	scibert_scivocab_uncased	• Batch_size = 12 • Max_seq_length = 64	61.03	48.69	54.17	72.15
		Default	63.29	54.38	58.49	74.21
CRF	-	• algorithm='lbfgs' • c1=0.364, c2=0.0056	73	57	64	76.47

Figure 14: Results - All Models



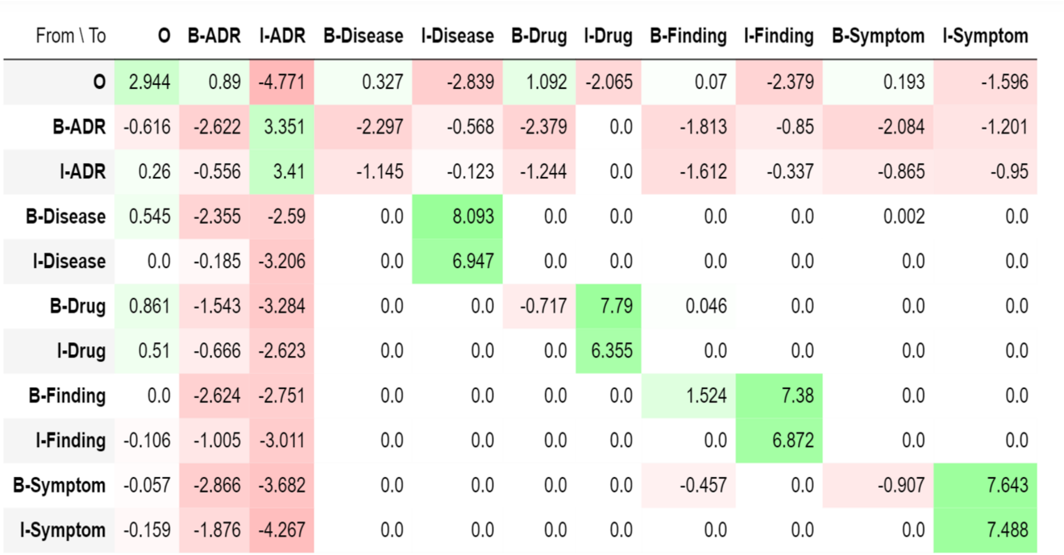
Top Positives

- 8.670693 B-ADR word.lower():glutes
- 7.585452 B-ADR word.lower():limped
- 7.293805 B-ADR word.lower():numb-like
- 6.918058 B-ADR word.lower():dislocating
- 6.855115 B-ADR word.lower():bloating
- 6.800704 O -2:word.lower():palsy
- 6.694787 B-ADR word.lower():tingling
- 6.666666 O word.lower():also
- 6.585083 B-Disease word.lower():alzheimers
- 6.547213 O word.lower():n't
- 6.387990 O word.lower():mostly
- 6.333495 B-ADR word.lower():swelling
- 6.330756 B-ADR -2:word.lower():admit
- 6.214795 B-Symptom word.lower():swelling
- 6.140181 B-Drug word[-3:]:tec

Top Negatives

- -3.278270 O -2:word.lower():admit
- -3.304709 O word[-3:]:cer
- -3.310725 O word.lower():gas
- -3.358359 O -1:word.lower():gets
- -3.362150 O word.lower():walked
- -3.497900 O word[-2:]:ma
- -3.533989 O word[-3:]:tis
- -3.568900 B-Drug word[-3:]:ain
- -3.637784 O -1:word.lower():present
- -3.727519 O word.lower():fatigued
- -3.772647 O word.lower():cramping
- -3.775605 I-Drug bias
- -3.848569 O word.lower():sore
- -3.906675 O -2:word.lower():five
- -3.992329 O word[-2:]:'s
- -4.041564 O +1:word.lower():concentration

Figure 15: Results - Best Model (CRF)



From \ To	O	B-ADR	I-ADR	B-Disease	I-Disease	B-Drug	I-Drug	B-Finding	I-Finding	B-Symptom	I-Symptom
O	2.944	0.89	-4.771	0.327	-2.839	1.092	-2.065	0.07	-2.379	0.193	-1.596
B-ADR	-0.616	-2.622	3.351	-2.297	-0.568	-2.379	0.0	-1.813	-0.85	-2.084	-1.201
I-ADR	0.26	-0.556	3.41	-1.145	-0.123	-1.244	0.0	-1.612	-0.337	-0.865	-0.95
B-Disease	0.545	-2.355	-2.59	0.0	8.093	0.0	0.0	0.0	0.0	0.002	0.0
I-Disease	0.0	-0.185	-3.206	0.0	6.947	0.0	0.0	0.0	0.0	0.0	0.0
B-Drug	0.861	-1.543	-3.284	0.0	0.0	-0.717	7.79	0.046	0.0	0.0	0.0
I-Drug	0.51	-0.666	-2.623	0.0	0.0	0.0	6.355	0.0	0.0	0.0	0.0
B-Finding	0.0	-2.624	-2.751	0.0	0.0	0.0	0.0	1.524	7.38	0.0	0.0
I-Finding	-0.106	-1.005	-3.011	0.0	0.0	0.0	0.0	0.0	6.872	0.0	0.0
B-Symptom	-0.057	-2.866	-3.682	0.0	0.0	0.0	0.0	-0.457	0.0	-0.907	7.643
I-Symptom	-0.159	-1.876	-4.267	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.488

Figure 16: Confusion Matrix - Best Model (CRF)

6 SUMMARY AND CONCLUSIONS

- Identifying the Adverse reactions of the drugs is very crucial in assessing the quality of the drug. This can only be done by learning from the patient responses.
- With advanced NLP techniques, there are a lot of choices in the current situation to deal such problems.
- With the experiments conducted on CADEC dataset, BioBERT stood out to be the most efficient model and would be a better choice for any biomedical dataset.
- Performance of the model can be further improved by training it on a large, balanced and diverse biomedical data corpus.
- Hyper parameters can be tuned to achieve better metric scores at a cost of increased training time and complexity.

REFERENCES

[1] Medication and Adverse Event Extraction from Noisy Text https://www.aclweb.org/anthology/U17-1009

[2] BioBERT https://arxiv.org/pdf/1901.08746.pdf

[3] BERT https://arxiv.org/abs/1810.04805

[4] SciBERT https://arxiv.org/abs/1903.10676

[5] CADEC dataset https://www.ncbi.nlm.nih.gov/pubmed/25817970

[6] Adverse Event Detection in Drug Development: Recommendations and Obligations Beyond Phase 3 https://ajph.aphapublications.org/doi/10.2105/AJPH.2007.124537

[7] BERT Explained – A list of Frequently Asked Questions https://yashuseth.blog/2019/06/12/bert-explained-faqs-understand-bert-working/

[8] Deconstructing BERT: Distilling 6 Patterns from 100 Million Parameters https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77

[9] Some examples of applying BERT in specific domain https://towardsdatascience.com/how-to-apply-bert-in-scientific-domain-2d9db0480bd9

[10] Review: BioBERT paper https://medium.com/@raghudeep/biobert-insights-b4c66fde8fa7

[11] Named Entity Recognition With Conditional Random Fields In Python https://www.depends-on-the-definition.com/named-entity-recognition-conditional-random-fields-python/

[12] Sequence Tagging With A LSTM-CRF https://www.depends-on-the-definition.com/sequence-tagging-lstm-crf/