Group12: Biomedical/Clinical Text Processing

Saksham Gupta Sathvik Bhagavan Harshit Gupta
Akshay Kumar Arya
170613, 170638, 170292, 17807074
{gupsak, sathvikb, hargupta, akshayka}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

In this report, we deep dive into Clinical Coding using NLP algorithms. We experiment with models of all types - traditional ML algorithms, bidirectional LSTMs and BERT transformers. The dataset used to test our models is the CodiEsp dataset. We provide extensive results for our proposed models. In the end, we give some suggestions for future work after a comprehensive error analysis of our models.

1 Introduction

Biomedical/Clinical text processing (BioNLP) refers to the study of how natural language processing(NLP) methods may be applied to texts and literature of the biomedical and clinical domain. The exponential increase in the number of electronic health records (EHRs) has created a tremendous opportunity to derive previously unknown healthcare insights. EHRs have much information about patients: family background, disease and treatment results, interpretation of test images, behaviour and much more. BioNLP has a wide range of real-life applications like gene-disease association, building EHR Question-Answer, and cause of death classification, to name a few. In this report, we are going to describe major problems in BioNLP briefly. We will mainly focus on Clinical Coding. Next, we discuss our proposed idea and its implementation details - dataset, models, experiment results, and error analysis. Lastly, we suggest some ideas to work on in the future. **Here** is the link to our code.

2 Related Work

2.1 Problem statements in NLP

Some problem statements in BioNLP are:

- **Negation Detection** - It is an important problem in BioNLP as many clinical statements are written as the absence of certain disease. Formally, it is a classification task.
- **Word Sense Disambiguation (WSD)** - Abbreviations are common in biomedical documents, and many are ambiguous because they have several potential expansions. It could be seen as a classification problem where choices are the multiple senses of the target word/phrase.
- **Information Retrieval (IE)** - It is the task of extracting and encoding information from clinical narratives, journals, EHRs, discharge summaries, or medical reports. The IR system extracts concepts, events, entities from free text and determines the relation between them [1]. The extracted and encoded information can be used for performing specific downstream tasks.
- **Named Entity Recognition (NER)** - Identification of entities like diseases, genes, chemicals, drugs and symptoms in the given text. It is particularly a complex problem as entities in the biomedical domain are often described using long phrases consisting of punctuation and characters.
- **Clinical Coding** - The task of translating clinical statements into a set of codes, as defined in international standards. We discuss Clinical Coding in the next section.

2.2 Clinical Coding using NLP techniques

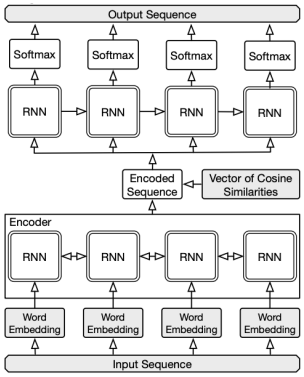
Using NLP in this domain can help in automatically extracting codes which will save a lot of time and if properly trained can outperform human coders as well. Formally this is a multilabel classification problem. This problem can also be viewed as a machine translation problem, where text is translated from the language in which medical records were written, into a generalised clinical coding language. There are many types of clinical codes, important amongst them are **ICD10** (International Classification of Diseases) and **CPT** (Current Procedural Terminology).

2.2.1 Using CNNs for ICD coding

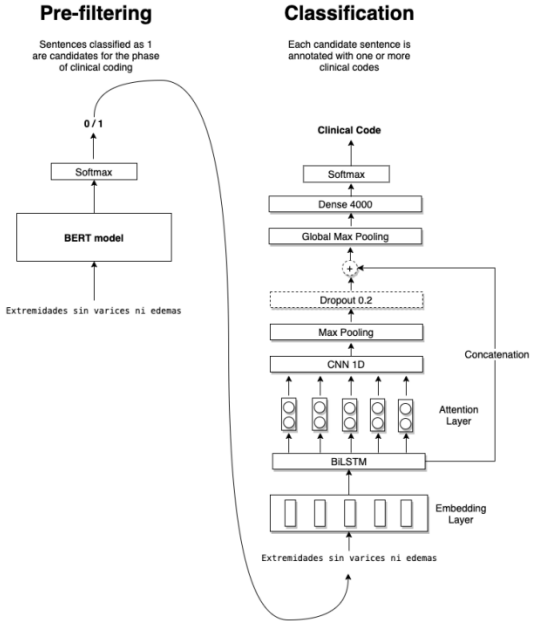
CNNs along with attention mechanism[2] have been used in the past for predicting ICD codes. This paper[3] proposes a **multi filter residual CNN** to perform ICD coding, which was built onto the previous state of the art[2]. The model was trained using the **MIMIC dataset**[4], and was compared to five state of the art models. It was concluded that the model obtained better performance than the baseline methods. The authors also highlighted that performance could have been improved further using recurrent transformers[5].

2.2.2 Sequence to sequence architecture

The paper [6] uses a seq2seq architecture to perform ICD-10 coding of death certificates written in english language. For the encoder, bidirectional LSTMs were used to convert the input sequence into an encoded vector. The decoder was a left to right LSTM which predicted the output sequence from the encoded vector, as well as a cosine similarity vector. The architecture is shown in figure 1(a). The dataset that was used was the **CepiDC (Cause of Death Corpus)**. The model performed better than the mean and median score of the other submitted models, even though no task specific feature engineering was done. This showed how deep learning models provide an edge over the traditional methods.



(a) Seq2seq model architecture used by [6]



(b) Fusion Model architecture used by [7]

Figure 1: Model architectures for clinical coding

2.2.3 Transformers for clinical coding

BertXML [8], which was introduced in 2020, was trained on a huge dataset of anonymous medical notes (around 7.5 million) of about 1 million patients. They doubled the input size of the model as compared to BioBert, which allowed intake of a complete EHR in one go. As the model was trained from scratch, it learnt vocabulary more specific to medical domain, which was more suited for EHR tasks. As a result, this model was a **State-of-the-Art model**, outperforming both BioBert and ClinicalBert in ICD code classification.

Some people also tried fusing Bert with LSTM and CNN. This was for a competition called as CodiEsp 2020 [7]. They used Bert as a prefiltering step, whose output indicated the probability of whether the sentence was likely to contain clinical code. The high probability sentences were passed into the classifier, which was experimented upon by having layers of LSTMs, CNNs with or without attention. The architecture has been shown in 1(b).

3 Proposed Idea

We experiment with various ML and NLP algorithms. We also propose a few models for Clinical Coding. We use a a freely available Clinical Coding dataset, **CodiEsp 2020 Competition [9]**. Other primary datasets like i2b2 and MIMIC-III had restricted access; some required course completion and/or other permissions; therefore we could access these datasets. We use only the CodiEsp dataset throughout our experiments. In the subsequent sections we give a detailed description and analysis of our experiments. Our proposal architectures are in Figure 3.

4 Dataset/Corpus

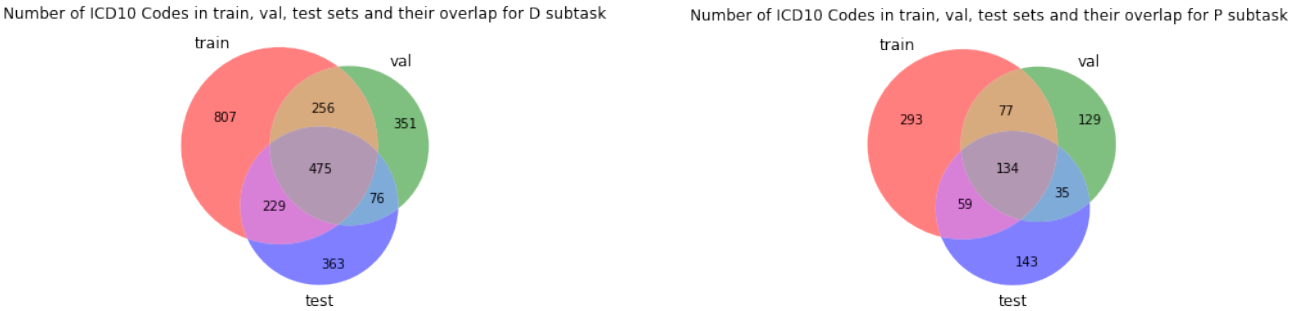
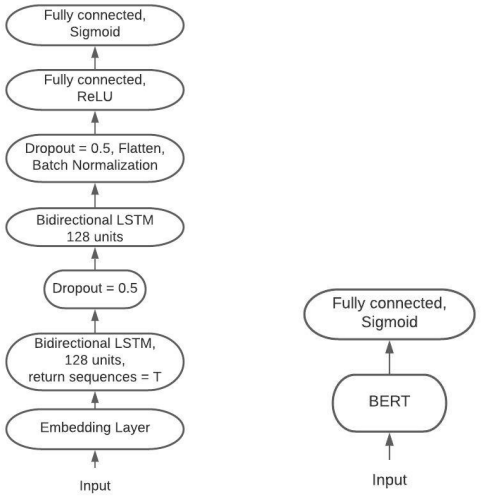


Figure 2: Label Distribution in the train, val, test sets

The CodiEsp data consists of annotated clinical documents. The dataset provides original Spanish texts along with machine translated English texts. The annotations are of two types: Procedural (P) and Diagnostic (D). Therefore, there are two subtasks for the datasets. In total, there are 500 train documents, 250 validation and 250 test documents. The dataset was manually annotated by practicing physicians and clinicians. Also the dataset covers diagnoses from different fields like oncology, radiology and cardiology. The following are some more critical insights from the data:

1. The documents with their tokenized sequence length has:
   (a) Mean: 342.63
   (b) Median: 318.5
   (c) Standard Deviation: 161.12
2. The corpus has 18,483 annotated codes, out of which 3427 are unique.

The label distribution for the train, validation and test set are summarized in Figure 2. As we can see, the amount of data is insufficient; the test set contains some labels which are present neither in the training nor validation set for both task D and P. This imbalance will affect predictions as the labels which are only present in the test set can never be predicted by any model.



(a) 2 layer BiLSTM model

(b) Bert Model

Figure 3: Model architectures proposed/tested

5 Experiments

We directly cannot feed text data into any model. We used a standard text preprocessing pipeline to compare the performance of different algorithms. We used two metrics to judge our models objectively: f1 score and hamming score. The preprocessing was as follows: We reduced the text size by removing stopwords (using the nltk library) and other special characters like punctuations, commas etc.

Firstly, We applied traditional Machine learning algorithms for classification [10] like Support Vector Classifier, Multinomial Naive Bayesian and Random Forest Classifier. We also experimented with relatively newer classifiers like **XGBoost Classifier** [11] and **AdaBoost Classifier**. We fed the text data using two techniques: **Bag of Words** and term frequency–inverse document frequency (**TF-IDF**). There was no significant change in the accuracy of the models for different hyperparameters for all the models mentioned above. As expected, AdaBoost and XGBoost gave the best results using Bag of Words method.

Secondly, we tried two deep neural networks (DNN). In addition to above mentioned preprocessing steps, we created a vocabulary of words for our data. We added four special tokens in the vocabulary corresponding to the start and end of a sentence, the unknown token (for rare words in test set for example) and a padding token (to make shorter documents equal to the required size). The first DNN model was a simple feed-forward neural network. The input to the model were indices of the tokens corresponding to the vocabulary. It had three dense layers with relu function, except for the last (it had sigmoid) and one batch normalization layer in between. The second was a 2 layer **bidirectional LSTM** model given in 3(a). The model fed the text input into a word embedding layer of dimension equal to 300, further fed into a double BiLSTM layer (hidden size was 512 and 256). Lastly, the output was fed into a dense layer to predict the final output.

Thirdly, we experimented with pretrained **BERT** transformer and fine-tuned other BERT based transformers - **Bio BERT** [12] and **Clinical BERT** [13] given in 3(b). We freezed the pretrained layers of all the three models and add randomly initialised dense layers on top. These dense layers were fine-tuned using training data. Unfortunately, the models did not give satisfactory results.

6 Results

6.1 Baseline Models

For baseline or traditional ML models, the results on validation and test dataset are given in Table 1 and 2.

Model Name	Subtask - D **Val Set**		Subtask - P **Val Set**	
	Hamming Score	F1 Score	Hamming Score	F1 Score
Multinomial Naive Bayesian	0.0192	0.0008	0.0239	0.0016
XGBoost Classifier	0.2091	0.0245	0.1936	0.0206
Support Vector Classifier	0.0050	0.0002	0.0009	0.0001
Random Forest Classifier	0.0047	0.0004	0.0061	0.0003
Logistic Regression	0.0421	0.0049	0.0694	0.0074
AdaBoost Classifier	**0.2715**	**0.0728**	**0.2049**	**0.0372**

Table 1: Results of traditional models on validation set

Model Name	Subtask - D **Test Set**		Subtask - P **Test Set**	
	Hamming Score	F1 Score	Hamming Score	F1 Score
Multinomial Naive Bayesian	0.0227	0.0015	0.0311	0.0016
XGBoost Classifier	0.2091	0.0245	0.1936	0.0206
Support Vector Classifier	0.0056	0.0002	0.0031	0.0003
Random Forest Classifier	0.0081	0.0005	0.0061	0.0003
Logistic Regression	0.0651	0.0078	0.0771	0.0068
AdaBoost Classifier	**0.2848**	**0.0722**	**0.2012**	0.**0344**

Table 2: Results of traditional models on test set

6.2 Neural Networks and Transformers

For the DNN based and transformer based models,the results on validation and test dataset are given in Table 3 and 4.

Model Name	Subtask - D **Val Set**		Subtask - P **Val Set**	
	Hamming Score	F1 Score	Hamming Score	F1 Score
Feed Forward NN	0.0216	0.0019	0.0281	0.0013
2 layer Bi-LSTM Model	0.0572	0.0007	-	-
BERT Transformer	0.0059	0.0072	0.0080	0.0069
Clinical-BERT Transformer	0.0069	0.0076	0.0091	0.0083
Bio-BERT Transformer	0.0061	0.0085	0.0095	0.0097

Table 3: Results of DNN and Transformer based models on validation set

Model Name	Subtask - D **Test Set**		Subtask - P **Test Set**	
	Hamming Score	F1 Score	Hamming Score	F1 Score
Feed Forward NN	0.0227	0.0015	0.0311	0.0016
2 layer Bi-LSTM Model	0.0590	0.0008	-	-
BERT Transformer	0.0054	0.0082	0.007	0.0085
Clinical-BERT Transformer	0.0061	0.0077	0.0098	0.0090
Bio-BERT Transformer	0.0058	0.0079	0.0091	0.0087

Table 4: Results of DNN and Transformer based models on test set

7 Error Analysis

We can see the hamming and F1 scores are not high. For the baseline models, **AdaBoost** gave the highest score and for the neural networks, **2 layer Bi-LSTM** gave the highest followed by BERT Transformer. Overall AdaBoost gave the highest score on the test set. This is because baseline models do work well or comparable to neural networks when the amount of data is small. By analysis, we can say the following might be the reasons for the failure of models based on Neural Networks:

1. Neural Networks are data-hungry. The amount of data that we trained is significantly less as the number of classes is more than 2000 and the number of documents in the training set is 500. So **Lack of training data** might be a reason for these low scores.
2. The dataset is **imbalanced** in the label distribution. As already explained above, we had anticipated that predictions might be affected heavily, and indeed it turned out to be the case.

So this might be mitigated by using a better and a bigger corpus like MIMIC-III. Only then we can comment on the performance of Neural Networks and their comparison with other baseline models.

8 Individual Contribution

Work	Saksham Gupta	Sathvik Bhagavan	Harshit Gupta
Code Implementation			
Data Preprocessing	✓	✓	
Baseline Models	✓		✓
Deep Learning based models		✓	
Transformer based models	✓	✓	
Report			
Introduction and Conclusion	✓	✓	✓
Related Work			✓
Proposed Idea	✓	✓	✓
Dataset and Corpus	✓	✓	
Experiments	✓	✓	✓
Error Analysis	✓	✓	✓
Conclusion		✓	

Table 5: Contribution of Team members

*Akshay Kumar Arya could not contribute due to health issues.

9 Conclusion

In this report, we introduced BioNLP and particularly delved deep into one of the problem statements, i.e. clinical Coding. We did a literature survey where we studied various datasets which are used for solving this task, studied various models and architectures including SOTA. We proposed to work on CodiEsp 2020 dataset as it is freely available without any restriction as most of the benchmark datasets required registration and other procedures to fulfill which we could not do in time. We proposed to test various existing algorithms and proposed a 2 layer Bi-LSTM model, BERT with a dense layer on top of it to check and evaluate the performance on the dataset selected. We described various experiments that we conducted and reported all the results. Then we discussed possible explanations for the performance obtained.

Our future work will include using a bigger and better dataset like MIMIC on the proposed algorithms and also experimenting with various pretrained embeddings like word2vec, glove, fasttext, bert etc. and its impact on performance of the models.

References

[1] J. Cowie and W. Lehnert, "Information extraction," *Communications of the ACM*, vol. 39, no. 1, pp. 80–91, 1996.

[2] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," *arXiv preprint arXiv:1802.05695*, 2018.

[3] F. Li and H. Yu, "Icd coding from clinical text using multi-filter residual convolutional neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8180–8187, 2020.

[4] H. Schäfer, "Multilingual icd-10 code assignment with transformer architectures using mimic-iii discharge summaries," 2020.

[5] Z. Dai, Z. Yang, Y. Yang, W. Cohen, J. Carbonell, Q. Le, and R. T.-X. Salakhutdinov, "Attentive language models beyond a fixed-length context. arxiv 2019," *arXiv preprint arXiv:1901.02860*.

[6] Z. Miftahutdinov and E. Tutubalina, "Kfu at clef ehealth 2017 task 1: Icd-10 coding of english death certificates with recurrent neural networks.," in *CLEF (Working Notes)*, 2017.

[7] M. Polignano, V. Suriano, P. Lops, M. de Gemmis, and G. Semeraro, "A study of machine learning models for clinical coding of medical reports at codiesp 2020," in *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum. CEUR Workshop Proceedings*, 2020.

[8] Z. Zhang, J. Liu, and N. Razavian, "Bert-xml: Large scale automated icd coding using bert pretraining," 2020.

[9] A. Miranda, A. Gonzalez-Agirre, and M. Krallinger, "CodiEsp corpus: Spanish clinical cases coded in ICD10 (CIE10) - eHealth CLEF2020," Apr. 2020. Funded by the Plan de Impulso de las Tecnologías del Lenguaje (Plan TL).

[10] `https://scikit-learn.org/stable/`.

[11] `https://xgboost.readthedocs.io/en/latest/#`.

[12] `https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT`.

[13] `https://huggingface.co/emilyalsentzer/Bio_Discharge_Summary_BERT`.