2. INTRODUCTION

The term Named Entity refers to “unique identifiers of entities”. It is a real-world object that can be denoted with a proper name.NER is widely used in downstream applications of NLP and artificial intelligence such as machine translation, information retrieval, and question answering. Named Entity Recognition(NER) involves the identification of proper names in texts and the classification of these names into a set of predefined categories. Here we have 5 sets of categories: Diseases, Genes, Protein, Chemical, and Biomedical.
A well-studied solution for a neural network to take into account an effectively infinite amount of context is the BI-LSTM. CNN’s have also been investigated for modeling character-level information, among other NLP tasks, and a combination of BI-LSTM, CNN, and CRF is very successful in the field of sequence labeling tasks in the past few years.
Most recent approaches to NER have focused on multi-task learning which jointly conducts other related NLP tasks like entity linking or chunking. Conditional random field (CRF) jointly models the label decision by capturing the dependencies across adjacent labels. [2]
Named entity recognition (BioNER) is one of the most fundamental tasks in biomedical text mining that aims to automatically recognize and classify biomedical entities. It is typically formulated as a sequence labeling problem whose goal is to assign a label to each word in a sentence. The BiLSTM (bidirectional long short-term memory) layer models the context information of each character. The hidden states of the BiLSTM layer are fed into the CRF layer to optimize sequence tagging with the help of adjacent tags. On POS, chunking, and NER data sets, the BI-LSTM-CRF model can deliver state-of-the-art (or close to) accuracy. [3][4]
CRF has shown to be very effective when combined with neural architectures for sequence labeling tasks. However, the models with unidirectional CRF (generally referred to as CRF) are capable of capturing the dependencies between labels in the forward direction only. [5]

3. PROBLEM STATEMENT

With the enormous volume of biological literature, an increasing growth phenomenon due to the high rate of new publications is one of the most common motivations for biomedical text mining. Biomedical text mining (BioNLP) refers to the methods and study of how text mining may be applied to texts and literature of the biomedical and molecular biology domains. Name Entity Recognition overcomes the primitive methods to manually identify and classify biological entities. Names and identifiers for biomolecules such as proteins and genes, chemical compounds and drugs, and disease names have all been used as entities. We studied different research publications that used the bi-LSTM model, powered by CRF. On scrutinizing further we finalized our project objective. [6]
In this project, our main focus is to depict the state-of-the-art performance of BI-LSTM CRF by testing its performance on both Single task framework and multi-task framework. [7] [8]
NOTE: Tensorflow 1.13.1 and Keras 2.2.4 ARE SPECIFICALLY REQUIRED TO RUN THE CRF MODEL WITHOUT PATCH THE KERAS LIBRARY

Literature Survey

S. No.	Author and Year (Reference)	Title (Study)	Concept/Theoretical model/ Framework	Methodology used	Dataset Details /Analysis	Relevant Finding	Limitations/ Future Research
1.	Jason P.C. Chiu, Eric Nichols Transactions of the Association for Computational Linguistics, vol. 4, pp. 357–370, 2016 Submission batch: 11/2015; Revision batch: 3/2016; Published 7/2016. https://arxiv.org/abs/1511.08308	Named Entity Recognition with Bidirectional LSTM-CNNs	The aim of this paper is to develop a neural network model, which incorporates a bidirectional LSTM and a character-level CNN and which benefits from robust training through dropout, achieves state-of-the-art results in named entity recognition with little feature engineering.The authors also propose a novel method of encoding partial lexicon matches in neural networks and compare it to existing approaches.	The neural network is inspired by the work of Collobert et al. (2011b), where lookup tables transform discrete features such as words and characters into continuous vector representations, which are then concatenated and fed into a neural network. Instead of a feed-forward network, we use the bi-directional long-short term memory (BLSTM) network. To induce character-level features, we use a convolutional neural network. The extracted features of each word are	The datasets used are the CoNLL-2003 NER shared task and the OntoNotes 5.0 datasets. The CoNLL-2003 dataset consists of newswire from the Reuters RCV1 corpus tagged with four types of named entities: location, organization, person, and miscellaneous. The OntoNotes 5.0 Dataset is much larger than CoNLL-2003 and consists of text from a wide variety of sources, such as broadcast conversation, broadcast news, newswire, magazine,	The models have surpassed the previous highest reported F1 scores for both CoNLL-2003 and OntoNotes. The GloVe 50d model provides an ultimate F1 score of 91.41 (± 0.21) for the CoNLL-2003 and gives an efficient 86.24 (± 0.35) output on the OntoNotes. The F1 Score for the Skip-gram 50d are 90.76 (± 0.23) and 85.70 (± 0.29) for the CoNLL-2003 and OntoNotes respectively.	More effective construction and application of lexicons and word embeddings are areas that require more research. In the future, we would also like to extend our model to perform similar tasks such as extended tagset NER and entity linking

				fed into a forward LSTM network and a backward LSTM network. The output of each network at each time step is decoded by a linear layer and a log-softmax layer into log-probabilities for each tag category. These two vectors are then simply added together to produce the final output.	telephone conversation, and Web text.		

2.	Buzhou Tang, Xiaolong Wang, Jun Yan2 and Qingcai Chen Entity recognition in Chinese clinical text using attention-based CNN-LSTM-CRF BMC Med Inform Decis Mak 19, 74 (2019). https://doi.org/10.1186/s12911-019-0787-y	Entity recognition in Chinese clinical text using attention-based CNN-LSTM-CRF	In this study, The authors propose a deep neural network for entity recognition in Chinese clinical text, which extends LSTM-CRF by introducing a CNN layer and an attention layer. The CNN layer is used to capture local context information of the Chinese character of interest, and the attention layer is used to determine the relativity strength of other Chinese characters to the Chinese character of interest. The effectiveness of their method is shown by a comparison with two benchmark datasets. The performances of all systems are measured by micro-averaged precision, recall and F1-score under two criteria: "strict" and "relaxed", where the "strict" criterion checks whether	In this paper, a DNN, called attention-based CNN-LSTM-CRF, is proposed to recognize entities in Chinese clinical text. Attention-based CNN-LSTM-CRF is an extension of LSTM-CRF by introducing a CNN (convolutional neural network) layer after the input layer to capture local context information of words of interest and an attention layer before the CRF layer to select relevant words in the same sentence.	In this paper, the authors have used namely two datasets: CCKS2017_CNER and ICRC_CNER. CCKS2017_CNER contains 400 Chinese clinical records with five categories of clinical entities, 300 records are treated as a training set and the remainder 100 records are treated as a test set. In this dataset, all clinical entities are contiguous, and the a total number of them is 39,359. ICRC_CNER contains 1176 Chinese clinical records with the other five categories of clinical entities, 600 records are treated as a training set, 176 records are treated as a	Our method achieves the highest "strict" F1-scores of 90.61% on CCKS2017_CNER and 83.32% on ICRC_CNER, outperforming CRF and LSTM-CRF by 0.44 and 0.32% respectively. When the CNN layer is removed from our method, the F-score slightly increases on CCKS2017, but slightly decreases on ICRC_CNER. When the attention layer is removed, the F-scores on both two datasets decrease slightly. When both CNN and attention layers are removed, the F-scores on both two datasets decrease greatly. The experimental results indicate that both CNN and	Although the method shows better overall performance than CRF and LSTM-CRF, it does not always achieve highest "strict" F1-score on all categories of clinical entities. The limitations of this study are: 1) the proposed method is also applicable to entity recognition in English text, but we do not compare it on English datasets. The experiments will be conducted in the future. 2) there are also some other extensions of LSTM-CRF on tasks in other domains, and the study is not compared to those tasks. Comparing our method with them

			predicted entities exactly match with gold ones in boundary and category, while the “relaxed” criterion relaxes the condition in boundary, and only checks whether predicted entities overlap with gold ones.		development set and the remainder 400 records are treated as a test set.	attention layers are individually beneficial to LSTM-CRF. The method performs well on some categories, such as “Test” and “Medication” on ICRC_CNER, “Symptom”, “Test” and “Body” on CCKS2017_CNER dataset. However, it also performs not very well on some categories, such as “Disease” and “Treatment” on both datasets.	and introducing their characteristics into our method to form new methods are other two cases of their future work.

3.	Renzo M. Rivera Zavala1 , Paloma Mart´ınez1 , Isabel Segura-Bedmar1 1Computer Science Department, University Carlos III of Madrid A Hybrid Bi-LSTM-CRF model for Knowledge Recognition from eHealth documents TASS 2018: Workshop on Semantic Analysis at SEPLN, septiembre 2018, págs. 65-70 http://ceur-ws.org/Vol-2172/p6_hybrid_bi_lstm_tass2018.pdf	A Hybrid Bi-LSTM-CRF model for Knowledge Recognition from eHealth documents	In this paper , the authors propose a hybrid Bi-LSTM and CRF model adding sense-disambiguation embedding and an extended tag encoding format to detect discontinuous entities, as well as overlapping or nested entities. To do this, they adapt the NeuroNER model have extended NeuroNER by adding context information, Partof-Speech (PoS) tags and information about overlapping or nested entities. use two pre-trained word embedding models: i) a word2vec model ii) a sense-disambiguation embedding model.	Pre-processing All texts were preprocessed in four steps. a)First, sentences were split by using Spacy b) sentences and their annotated entities were transformed to the BRAT format c) sentences were tokenized. d) each token in a sentence was annotated using the BMEWO-V extended tag encoding The Words Embeddings are implemented by Spanish Billion Words(Cardellino, 2016), which is a pretrained model of word embeddings Post-processing Once tokens have been annotated	The dataset used for evaluation is the the TASS-2018-Task 3 eHealth Knowledge Discovery. The training set is made up of 5 documents with 3276 entities annotations. The development set consists of 1 text document with 1958 entities annotations. The test set consists of 1 text document .There are two types of of entities: concepts and actions. For this reason, tokens can be annotated with different labels following the BMEWO-V encoding format.	Compared to NeuroNER , extended Neuro NER provides better Precision , Recall and Hence the F1 score . In the substask A (identification of key phrases), our system obtained the top micro F1 (0.872) .It significantly outperform the rest of participating systems. When contrasted with the established systems such as plubeda , upf-upc , VSP , Marcelo ; the extended NeuoNER model outperforms all of them with an incredible F1 score of 87%	The future goals aim to try the other embeddings models such as the FastText model, which contains morphological information. Moreover, the authors will extend the encoding format to capture distinct types of overlapping or nested entities.

				with their corresponding labels in the BMEWO-V encoding format, the entity mentions must be transformed to the BRAT format. V tags, which identify nested or overlapping entities, are generated as new annotations			
4.	Donghyeon Kim; Jinhyuk Lee; Chan Ho So; Hwisang Jeon; Minbyul Jeong; Yonghwa Choi; Wonjin Yoon; Mujeen Sung; Jaewoo Kang Date of Publication: 04 June 2019 https://doi.org/10.1109/ACCESS.2019.2920708	A Neural Named Entity Recognition and Multi-Type Normalization Tool for Biomedical Text Mining	They propose a neural biomedical named entity recognition and multi-type normalization tool called BERN. The BERN uses high-performance BioBERT named entity recognition models which recognize known entities and discover new entities. Also, probability-based decision rules are developed to identify the types of overlapping entities. Furthermore, various named	The RESTful Web service of BERN was implemented using Python and Node.js. BERN run BioBERT NER models which are pre-trained with TensorFlow3, on their server to recognize incoming biomedical text such as PubMed articles and raw text.Four BioBERT NER models for genes/proteins, diseases, drugs/chemicals, and	There are many use cases where BERN can be used. Discovery of new named entities: BioBERT NER models can be used to discover new entities from the latest biomedical literature. Information retrieval: BERN can serve as a fundamental NER+NEN model for various text mining tools. Relation	Their proposed tool BERN recognizes known entities and discovers new entities using BioBERT NER models. The BioBERT models outperform NER models of existing Web-based text mining tools in terms of F1-score on genes/proteins, diseases, drugs/chemicals, and species. After reviewing a vast number of cases of overlapping entities, they	For future work, they plan to use a multi-task NER model for higher NER performance. Also, they will develop a novel entity type decision model that uses transfer learning to consider not only the entity types and probabilities of overlapping entities but also the deeper contextual meaning of a text.

			entity normalization models are integrated into BERN for assigning a distinct identifier to each recognized entity. The BERN provides a Web service for tagging entities in PubMed articles or raw text. Researchers can use the BERN Web service for their text mining tasks, such as new named entity discovery, information retrieval, question answering, and relation extraction.	species, use 2.4 GB (4×0.6 GB) of GPU memory. They use 8 NVIDIA V100 GPUs for pre-training BioBERT, and they use a NVIDIA Titan X GPU for making predictions. And, they use the following training datasets to fine-tune each BioBERT NER model: BC2GM for genes, NCBI disease for diseases, BC4CHEMD for drugs/chemicals, and LINNAEUS for species. GNormPlus uses 8 to 16 GB, and tmVar 2.0 uses 4 to 8 GB of memory. And, the load time of the GNormPlus gene dictionary is about 5 seconds and the load time of the tmVar 2.0 part-of-speec	extraction: BERN can generate rich datasets for downstream biomedical text mining tasks such as relation extraction.• A useful text mining tool: Using APIs, researchers can obtain NER+NEN results for texts from highly accessible Web services	developed and used the decision rules on identifying the entity types of overlapping entities which occur frequently in multi-type NER results. For assigning a specific ID to each recognized entity, multiple normalization models are combined and integrated into BERN. The RESTful Web service of BERN is freely available and can be used for various types of input. Researchers can use BERN for text mining tasks such as new named entity discovery, information retrieval, question answering, and relation extraction.	

				h tagger is about 1 second. To reduce their load time, they run GNormPlus and tmVar 2.0 processes in the background on the server			
5.	R. Ramachandran, K. Arutchelvan Received: 22 November 2020 / Accepted: 2 March 2021 https://doi.org/10.1007/s12652-021-03078-z	Named entity recognition on bio-medical literature documents using hybrid based approach	In this article, a new hybrid based approach is proposed to identify named entity from the medical literature documents. New dictionary has been built for route of administration, dosage forms and symptoms to annotate the entities in the medical documents. The annotated entities are trained by the blank Spacy machine learning model. The trained model provide a decent accuracy when compared with the existing model. The hybrid model is validated with the dictionary and human (optional)to calculate the confusion matrix. It is able	Preprocessing is the heart of the proposed architecture which cleaned the raw data and provide the annotated sentence with entities. This step involves in the formation of unstructured data into meaningful format. JSON format has been used for this work. The raw data are tokenized by annotating the sentences with the entity. For training a model annotated sentences are essential. A custom annotated dataset was developed internally for the three entities: Symptoms,	The literature documents around 100 numbers are downloaded by using a python API which has been developed using the python beautiful soup library. The downloaded documents were in PDF format. The retrieved data are converted into raw text. The documents are split into sentences. Spacy phrase matcher is used to annotate the start and end position of the entities. The sentences are further filtered based on the presence of	This research work presented the detailed study of the NER on life science domain. It also highlighted the role of transfer learning to enhance the machine learning model. The proposed hybrid approach identified named entity and outperformed well than the existing baseline method. The transfer learning shows the increase of around 15% accuracy when compared to the baseline method. Baseline model and	In future, they plan to boost the quantity of entities. Enriching the dictionaries by adding more object will give more accuracy. The work will be extended to update the dictionary from the suggestions of domain expert and retrain the model instantly.

			to identify more entities than the prevailing model. The average F1 score for five entities of the proposed hybrid based approach 73.79%.	Route of Administration and Dosage Forms. After data preprocessing step, cleaned data is passed in to Bio-NER model. This phase train the model with the annotated sentences. The model is built based on the Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). The model is retrained by the dataset generated from the dictionary based approach.	the entities.	proposed hNER model was trained with 80% of annotated sentences and tested with 20% of annotated sentence. The F1 score of the experimented model has shown a progressive improvement. As an add-on, the validation tool is more useful to fnd the accuracy by domain expert.	

6.	Zhiheng Huang , Wei Xu and Kai Yu https://arxiv.org/pdf/1508.01991.pdf	Bidirectional LSTM-CRF Models for Sequence Tagging	In this paper, they propose a variety of Long Short-Term Memory (LSTM) based models for sequence tagging. These models include LSTM networks, bidirectional LSTM (BI-LSTM) networks, LSTM with a Conditional Random Field (CRF) layer (LSTM-CRF) and bidirectional LSTM with a CRF layer (BI-LSTM-CRF). Their work is the first to apply a bidirectional LSTM CRF (denoted as BI-LSTM-CRF) model to NLP benchmark sequence tagging data sets. They show that the BILSTM-CRF model can efficiently use both past and future input features thanks to a bidirectional LSTM component. It can also use sentence level tag information thanks to a CRF	All models used in this paper share a generic SGD forward and backward training procedure. They choose the most complicated model, BI-LSTMCRF, to illustrate the training algorithm. In each epoch, they divide the whole training data into batches and process one batch at a time. Each batch contains a list of sentences which is determined by the parameter of batch size. In their experiments, they use batch size of 100 which means to include sentences whose total length is no greater than 100. For each batch, they first run bidirectional LSTM-CRF model forward pass which includes the	They test LSTM, BI-LSTM, CRF, LSTM-CRF, and BI-LSTM-CRF models on three NLP tagging tasks: Penn TreeBank (PTB) POS tagging, CoNLL 2000 chunking, and CoNLL 2003 named entity tagging. They extract the same types of features for three data sets. The features can be grouped as spelling features and context features. As a result, they have 401K, 76K, and 341K features extracted for POS, chunking and NER data sets respectively.	For POS data set, they achieved state of the art tagging accuracy with or without the use of extra data resource.Their test accuracy is 97.55% which is significantly better than others in the confidence level of 95%. Their model can achieve the best F1 score of 90.10 with both Senna embedding and gazetteer features. With the same Senna embedding, BI-LSTM-CRF slightly outperforms Conv-CRF (90.10% vs. 89.59%). However, BI-LSTM-CRF significantly outperforms Conv-CRF (84.26% vs. 81.47%) if random embedding is used.	In this paper, thye systematically compared the performance of LSTM networks based models for se-quence tagging. They presented the first work of applying a BI-LSTM-CRF model to NLP benchmark sequence tagging data. Their model can produce state of the art (or close to) accuracy on POS, chunking and NER data sets. It can achieve accurate tagging accuracy without resorting to word embedding.

			layer. The BI-LSTMCRF model can produce state of the art (or close to) accuracy on POS, chunking and NER data sets. In addition, it is robust and has less dependence on word embedding as compared to previous observations.	forward pass for both forward state and backward state of LSTM.			
7.	G. Yang and H. Xu Date of Publication: 21 December 2020 doi: 10.1109/ACCESS.2020.3046253	A Residual BiLSTM Model for Named Entity Recognition	To produce word or character vectors, they have used both word2vec and BERT. Furthermore, we do tests to assess the performance of NER utilising different residual block architectures. The results of the experiments show that our model can effectively improve the performance of both Chinese and English NER without requiring any external knowledge.	A 3-layer residual BiLSTM model as an example to illustrate The residual structure is used.	CoNLL2003, MSRA, Weibo, OntoNotes 4.0, and OntoNotes 5.0 are the four most extensively-used datasets for evaluating our model on English and Chinese NER tasks, respectiv-ely.	For NER challenges, we developed a new residual BiLSTM model. Based on BiLSTMs, we present a new sort of residual block. We make attempts to innovate on the structure of residual networks based on BiLSTMs, in contrast to most other state-of-the-art models that include external knowledge or multi-task learning.	For the future they can combine the model with an attention mechanism. Also the model can be applied to other NLP tasks

8.	Víctor Suárez-Paniagua, Renzo M. Rivera Zavala, Isabel Segura-Bedmar, Paloma Martínez Journal of Biomedical Informatics, Volume 99, 2019, 103285 https://doi.org/10.1016/j.jbi.2019.103285	A two-stage deep learning approach for extracting entities and relationships from medical texts	A two-stage deep learning method for Named Entity Recognition (NER) and Relation Extraction (RE) from medical texts is presented in this paper. Many natural language understanding applications in the biomedical realm rely on these tasks.	Deep Learning approaches for NER may discover patterns automatically from corpora, collecting essential syntactic and semantic information. Also, the method of Biomedical Relation Extraction.	eHealth-KD dataset is used here. The dataset was divided into three sections: training (559 sentences), validation (285 sentences), and test (285 sentences) (300 sentences). The test set comprises three separate subsets for testing performance in each scenario at the same time.	A two-stage IE system based on medical literature is presented in this research. Our system is responsible for three tasks: entity detection, entity categorization, and relation extraction.	More syntactic information of the sentence, such as Part-of-Speech tags, Chunk labels, dependency types, through the embeddings can be added.
9.	Cho, H., Lee, H. Published 27 December 2019 https://doi.org/10.1186/s12859-019-3321-4	Biomedical named entity recognition using deep neural networks with contextual information	Traditional NER approaches use extra conditional random fields (CRF) to capture crucial correlations between surrounding labels; they don't always include all of the contextual information from text in the deep learning layers.	All methods in terms of precision, recall, and F-score are compared. They have performed strict matching at the IOB token level and strict and partial matching at the level of mention to compute these values.	All Datasets were extracted from - http://gcancer.org/clstmdata	An NER system for biological items has been developed here that incorporates n-grams with bi-directional long short-term memory (BiLSTM) and CRF; this system is referred to as contextual long short-term memory networks with CRF (CLSTM).	The Contextual Information that is received from the CRF model, can be further made more accurate, and we can use the Standard LSTMs which can only use prior contexts and cannot predict the future.

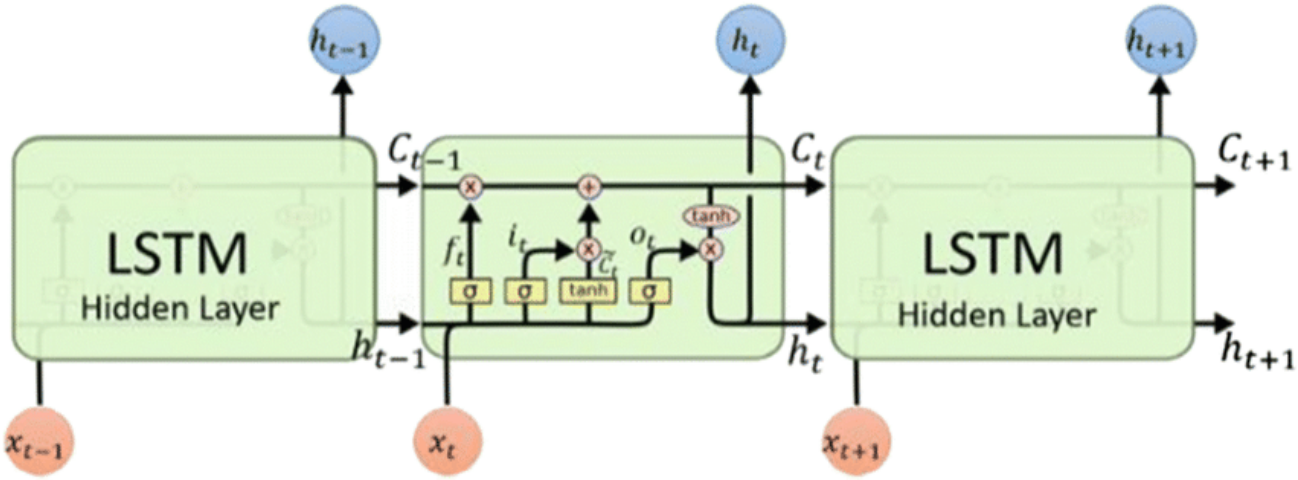
10.	Pir Dino Soomro, Sanotsh Kumar, Banbhrani, Arsalan Ali Shaikh, Hans Raj 2017 10.14569/IJACSA.2017.081220	Bio-NER: Biomedical Named Entity Recognition using Rule-Based and Statistical Learners	The purpose of extracting Bio-Medical Entities is to recognize the particular entities, whether word or phrases, from the unstructured data contained in the text. This research paper is aimed at Bio-Medical Named Entity by proposing the approach of Hybrid Machine Learning. The performances of different approaches viz., Machine learners like, Naïve Bayesian, Rule Based Learners i.e. PART, DTNB and NNGE, and Bayesian Network, are compared. Investigation and exploration of the data discovers that execution close to the best in class can be accomplished via a blend of Statistical Machine Learning and Rule Based Techniques utilizing straightforward characteristics.	This work proposes different approaches and methods, i.e. Machine Learning Hybrid Classification, Rule Based Non-tested Generalized Exemplars and Partial Decision Tree (PART) Learners for Bio-Medical Named Entity Recognition. For NER challenges, they developed a new residual BiLSTM model. Based on BiLSTMs, we present a new sort of residual block. They have made attempts to innovate on the structure of residual networks based on BiLSTMs, in contrast to most other state-of-the-art models that include external knowledge or multi-task learning.	The National Center for Biotechnology Information (NCBI) ailment corpus which is unreservedly accessible by NCBI on which this test or experiment is based. The corpus incorporates 793 synopses compositions which comprise of 2783 sentences and an aggregate of 6900 malady names	Single sets of a classifier which examined, it states that 87.4% of F-score accomplished by Naive Bayesian Decision Table on characteristics, for example, affixes, contextual, orthographic and N-grams. fusion of Naive Bayesian+Bayesian, Network+Non-Nested, Generalized Exemplars accomplished the 88.5% of F-scoreOverall accuracy was 89.0 percent on the Training dataset, 84.0 percent on the Development dataset, and 86.0 percent on the Testing dataset, respectively. The execution of sets of classifiers using vote WEKA Data Mining Tool was investigated using this Classifiers	To check the effectiveness of our proposed method for Drug Name Recognition. The authors plan to apply and check the effectiveness of our proposed method for Drug Name Recognition.

						blending of two, three, four, and five.	
11.	Rrubaa Panchendrarajan Aravindh Amaresan 2018 https://www.researchgate.net/publication/333384813_Bidirectional_LSTM-CRF_for_Named_Entity_Recognition	Bidirectional LSTM-CRF for Named Entity Recognition	Named Entity Recognition (NER) is a challenging sequence labeling task which requires a deep understanding of the orthographic and distributional representation of words. This model includes bidirectional LSTM (BI-LSTM) with a bidirectional Conditional Random Field (BI-CRF) layer. It is truly an end-to-end model not relying on any other additional labeled data.	BI-LSTM was successfully used to a voice recognition problem. CNNs have also been studied for modelling character-level information, as well as a mix of BI-LSTM, CNN, and CRF for various NLP tasks.	Our system is competitive on the CoNLL-2003 dataset for English. The dataset contains four different types of named entities: Person (PER), Organization (ORG), Location (LOC) and Miscellaneous (MISC). Sentences in the dataset are represented in the IOB format.	The obtained model is competitive and outperforms the majority of existing techniques that do not use externally labelled data. Furthermore, with a short quantity of training material, backward CRF is more capable of detecting complicated labels such as words existing inside a named entity and names of Miscellaneous things, according to the assessments.	There are several potential directions for future work. First, the performance of the model can be further enhanced by converting the dataset from IOB to IOBES tagging scheme. Moreover, it can be explored in multi-task learning approaches to combine more useful and correlated information among different NLP tasks.
12.	Muhammad Raza Khan Morteza Ziyadi Mohamed AbdelHady 2020 https://arxiv.org/abs/2001.08904	MT-BioNER : Multi-task Learning for Biomedical Named Entity Recognition using Deep Bidirectional Transformers	Healthcare industry is going through a technological transformation especially through the increasing adoption of conversational agents including voice assistants (such as Cortana, Alexa, Google Assistant, and	Slot tagging and Named Entity Recognition (NER) extract semantic elements by filling in specified slots in a semantic frame with the words of an input sentence/utterance.	We evaluate the performance of the proposed approach on four benchmark datasets.The datasets are BC2GM,BC5CDR, NCBI-Disease, JNLPBA.	They defined the training of a slot tagger as a multi-task learning problem including many data sets encompassing various slot kinds. They also reported training and scoring times and compared them to	Exploring the impact of overlap between the datasets on the overall model performance. We will like to explore ways to tackle overlap between entities that can degrade

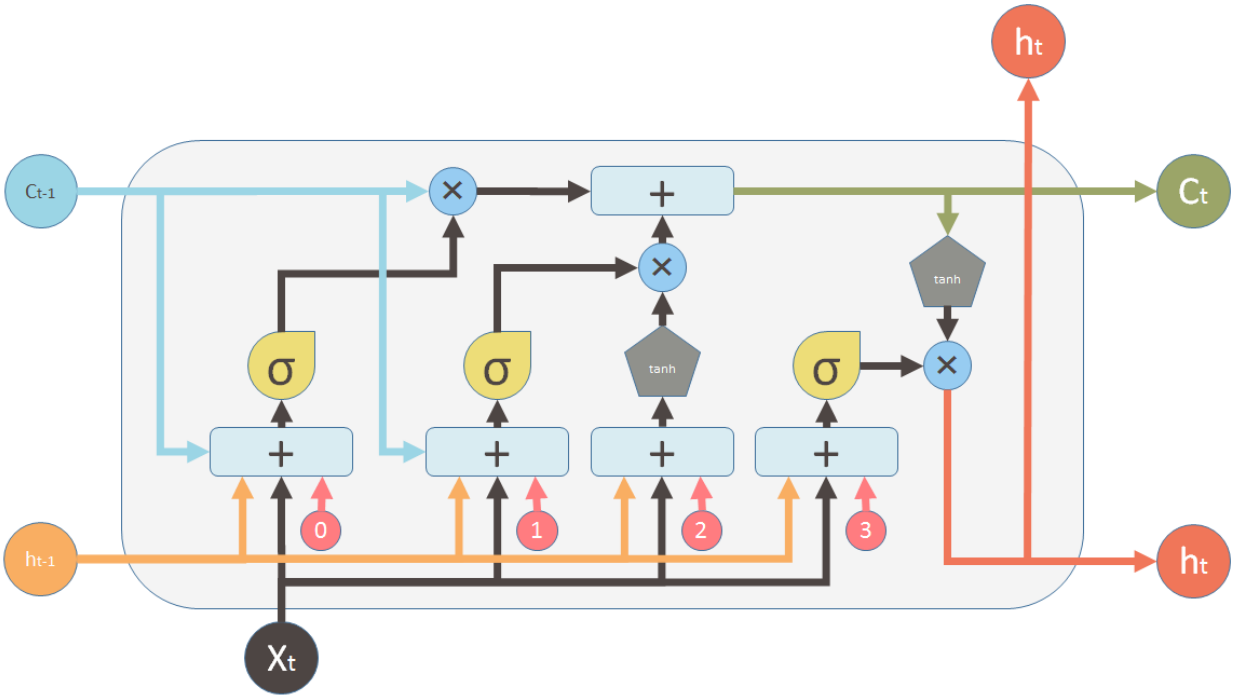
			Siri) and the medical chatbots. In this paper, we presented a multi-task transformer based neural architecture for slot tagging that overcomes the problems like limited-memory devices which require some model compression, training a large amount of labelled data. We formulated the training of a slot tagger using multiple data sets covering different slot types as a multi-task learning problem.			previous improvements .	the model performance. We will also like to perform comparative analysis of different models on same input sentences to highlight the plus points of our model over other models. We also want to analyze the performance of our NER model on general domain conversational systems in future work as well.

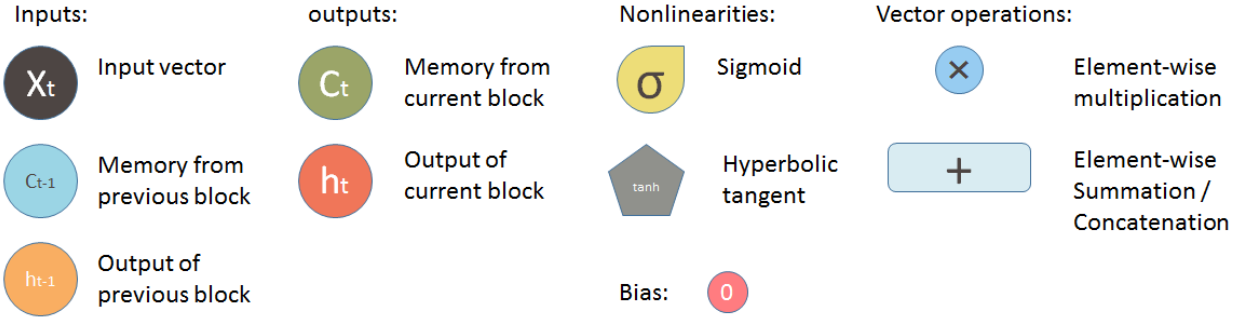
3.1 ARCHITECTURE DIAGRAM -

Long short-term memory neural network is a specific type of recurrent neural network that models dependencies between elements in a sequence through recurrent connections.



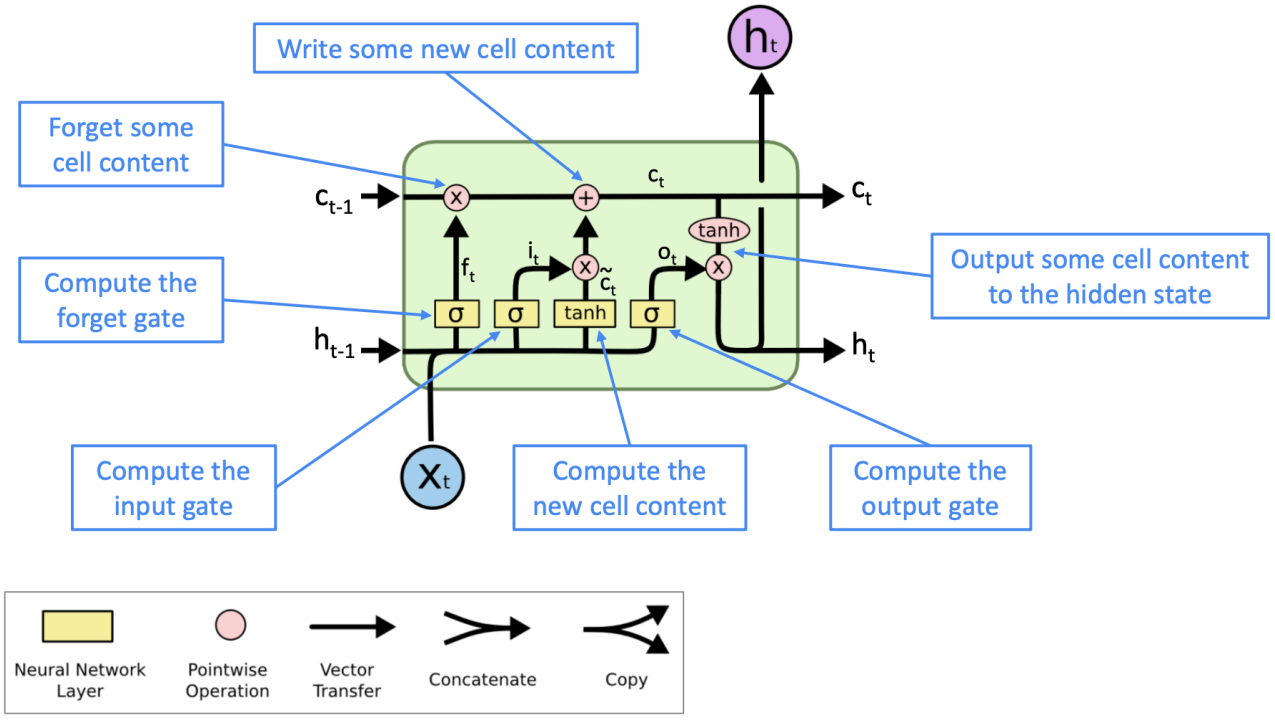
The repeating module in an LSTM containing four interacting layers





Building blocks of LSTM and its internal operations

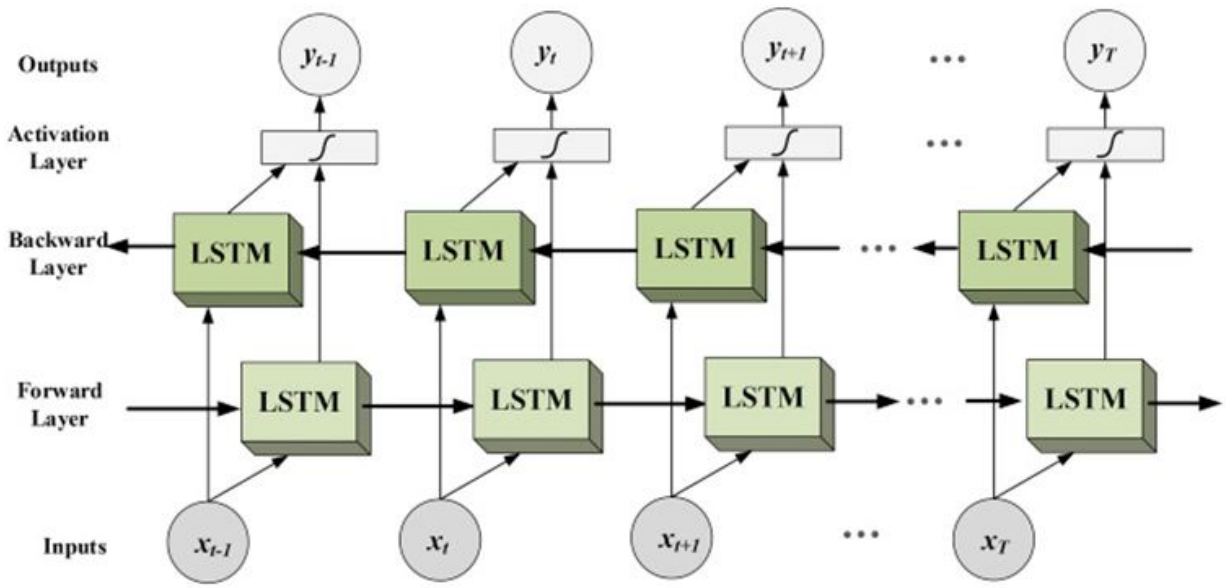
The network takes three inputs. X_t is the input of the current time step. h_{t-1} is the output from the previous LSTM unit and C_{t-1} is the "memory" of the previous unit. As for outputs, h_t is the output of the current network. C_t is the memory of the current unit. Therefore, this single unit makes a decision by considering the current input, previous output, and previous memory. And it generates a new output and alters its memory.



A single LSTM Cell showing its details

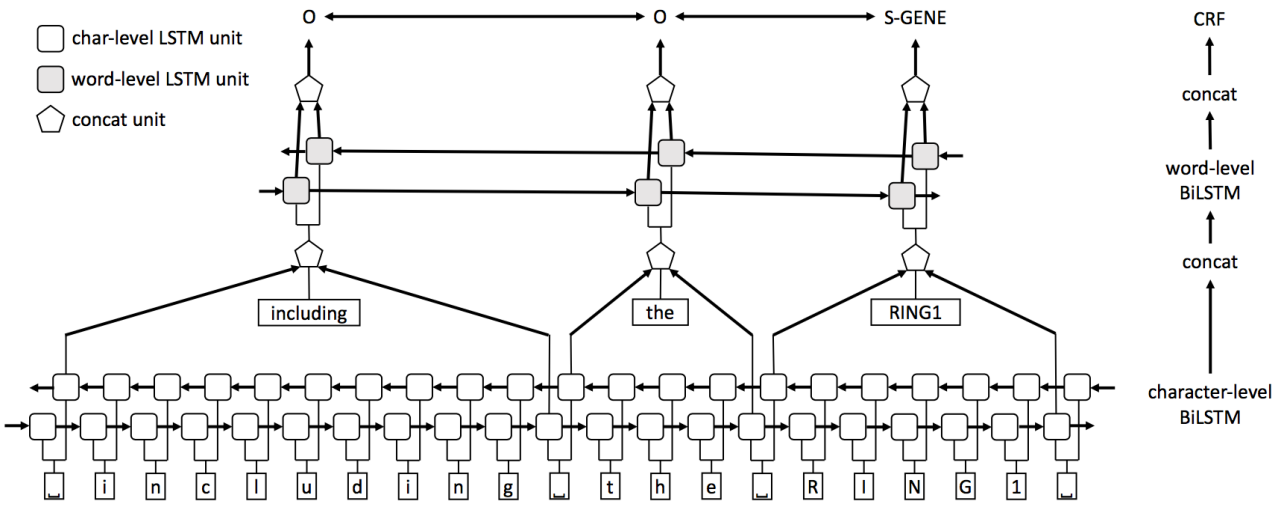
Bidirectional long-short term memory (bi-LSTM) is the process of making any neural network have the sequence information in both directions backward (future to past) or forward(past to future).

In bidirectional, our input flows in two directions, making a bi - LSTM different from the regular LSTM. With the regular LSTM, we can make input flow in one direction, either backward or forward. However, in bi-directional, we can make the input flow in both directions to preserve the future and the past information.



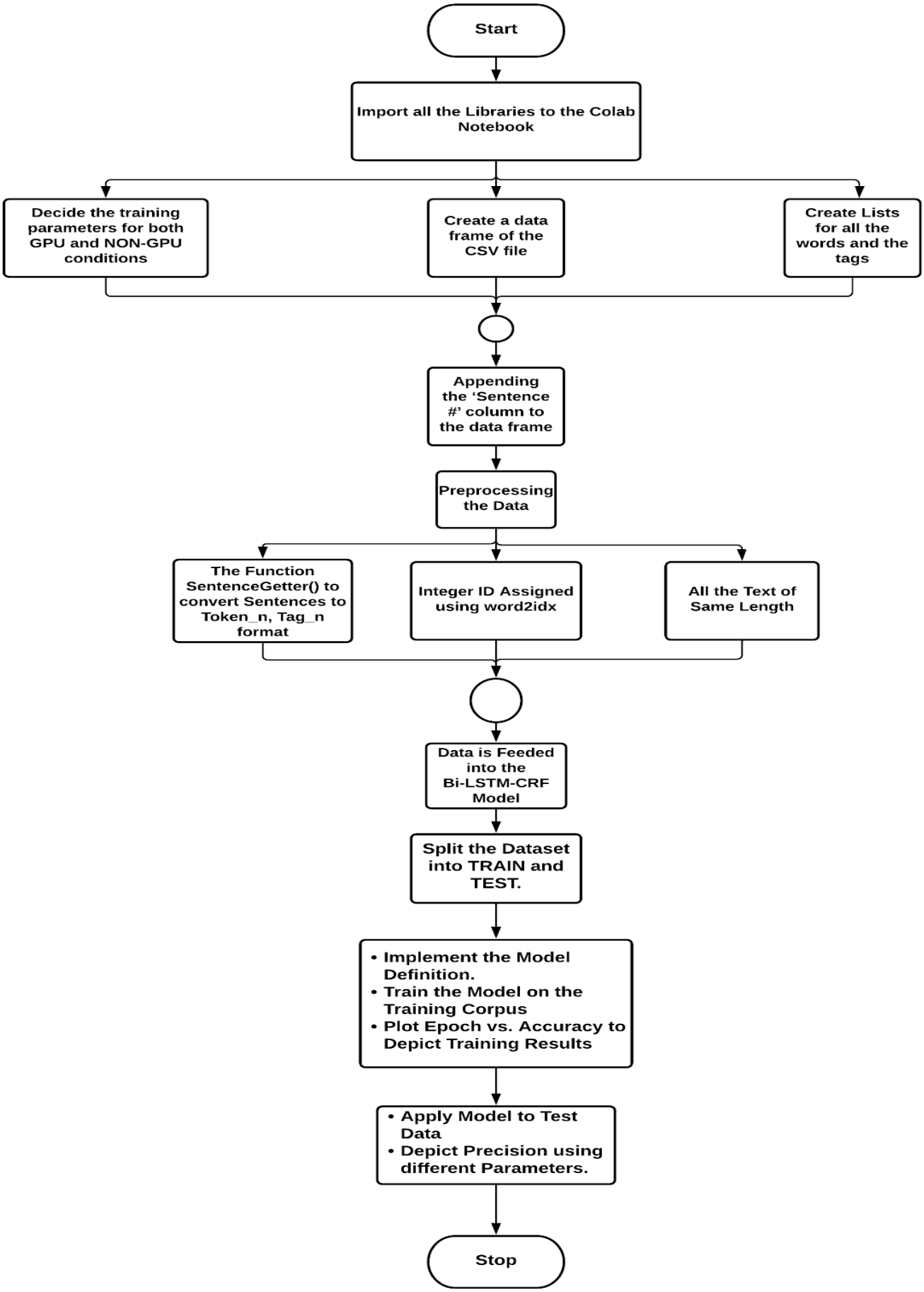
A Bi-LSTM Network Model

The architecture of a single-task neural network. The input is a sentence from biomedical literature. Rectangles denote character and word embeddings; empty round rectangles denote the first character-level BiLSTM; shaded round rectangles denote the second word-level BiLSTM; pentagons denote the concatenation units. The tags on the top, e.g. 'O', 'S-GENE', are the output of the final CRF layer, which are the entity labels we get for each word in the sentence. [9] [10]



Architecture of a single-task neural network

3.2 FLOW DIAGRAM -



3.3 PSEUDOCODE -

Input the data from the .tsv file into the data frame

```
      data = pd.read_csv("bc2gm_train.tsv",sep='\t',quoting=3,
error_bad_lines=False)
```

Append the Sentence column to the dataframe

```
for index,row in data.iterrows():
  row['Sentence #']="Sentence: "+str(count)
  if(row['Word']=="."):
    count=count+1
```

Define SentenceGetter class for Preprocessing

```
class SentenceGetter(object):
```

Define init and get_self function inside the SentenceGetter

```
def __init__(self, data):
        self.n_sent = 1
        self.data = data
        self.empty = False
        agg_func = lambda s: [(w,t) for w,t in zip(s["Word"].values.tolist(),

s["Tag"].values.tolist())]
        self.grouped = self.data.groupby("Sentence #").apply(agg_func)
        self.sentences = [s for s in self.grouped]

    def get_next(self):
        try:
            s = self.grouped["Sentence: {}".format(self.n_sent)]
            self.n_sent += 1
            return s
        except:
            return None
```

Tokenize the sentences by mapping the sentences to a sequence of numbers and then pad the sequence to feed the data to the Bi-LSTM-CRF model

```
y = pad_sequences(maxlen=max_len, sequences=y, padding="post",
value=tag2idx["O"])
```

Split the training and testing dataset

```
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.1)
```

Implement the model definition and print the model summary after the compilation of model

```
input = Input(shape=(max_len,))
model = Embedding(input_dim=n_words + 1, output_dim=20,
                  input_length=max_len, mask_zero=True)(input)
model = Bidirectional(LSTM(units=50,
return_sequences=True,recurrent_dropout=0.1))(model)
model = TimeDistributed(Dense(50, activation="relu"))(model)
model.compile(optimizer="rmsprop", loss=crf.loss_function,
metrics=[crf.accuracy])
model.summary()
```

Train the model on the entire training corpus

```
history = model.fit(X_tr, np.array(y_tr), batch_size=32, epochs=5,
                    validation_split=0.1, verbose=1)
```

Evaluate the model by applying it to testing data

```
test_pred = model.predict(X_te, verbose=1)
```

Depict the performance of the model using metrics like precision, recall, and f1- score

```
print("F1-score: {:.1%}".format(f1_score(test_labels, pred_labels)))
print(classification_report(test_labels, pred_labels))
```

[11] [12] [13]

4. EXPERIMENT AND RESULTS -

4.1 DATASET

	Word	Tag	Sentence #
0	DPP6	O	Sentence: 1
1	as	O	Sentence: 1
2	a	O	Sentence: 1
3	candidate	O	Sentence: 1
4	gene	O	Sentence: 1
5	for	O	Sentence: 1
6	neuroleptic	O	Sentence: 1
7	-	O	Sentence: 1
8	induced	O	Sentence: 1
9	tardive	O	Sentence: 1
10	dyskinesia	O	Sentence: 1
11	.	O	Sentence: 1
12	We	O	Sentence: 2
13	implemented	O	Sentence: 2
14	a	O	Sentence: 2
15	two	O	Sentence: 2
16	-	O	Sentence: 2
17	step	O	Sentence: 2
18	approach	O	Sentence: 2
19	to	O	Sentence: 2
20	detect	O	Sentence: 2
21	potential	O	Sentence: 2
22	predictor	O	Sentence: 2
23	gene	O	Sentence: 2
24	variants	O	Sentence: 2
25	for	O	Sentence: 2
26	neuroleptic	O	Sentence: 2
27	-	O	Sentence: 2
28	induced	O	Sentence: 2
29	tardive	O	Sentence: 2
30	dyskinesia	O	Sentence: 2
31	(	O	Sentence: 2
32	TD	O	Sentence: 2
33	)	O	Sentence: 2
34	in	O	Sentence: 2
35	schizophrenic	O	Sentence: 2
36	subjects	O	Sentence: 2
37	.	O	Sentence: 2
38	First	O	Sentence: 3
39	,	O	Sentence: 3
40	we	O	Sentence: 3
41	screened	O	Sentence: 3
42	associations	O	Sentence: 3
43	by	O	Sentence: 3
44	using	O	Sentence: 3
45	a	O	Sentence: 3
46	genome	O	Sentence: 3
47	-	O	Sentence: 3
48	wide	O	Sentence: 3
49	(	O	Sentence: 3

DATASET DESCRIPTION

The complete BioNER database can be found here
https://www.kaggle.com/adityaanup/biobert-named-entity-recognition-datasets

The dataset consists of 4 columns:

Index
Word
Tag
Sentence Number

Tags of entities are encoded in an IOBES-annotation scheme. Each entity is labeled with a B or an I to detect multi-word entities, where B denotes the beginning of an entity and I denote the inside of an entity, while E denotes the End of the entity. BIE is a scheme used to represent a complete entity. S is used to represent a singly tokenized entity
O denotes all other words which are not named entities.

The complete list of datasets that will be used in the project are

Dataset	Size	Entity types and counts
BC2GM	20,000 sentences	Gene/Protein (24,583)
BC4CHEMD	10,000 abstracts	Chemical (84,310)
BC5CDR	1,500 articles	Chemical (15,935), Disease (12,852)
NCBI-Disease	793 abstracts	Disease (6,881)
JNLPBA	2,404 abstracts	Gene/Protein (35,336), Cell Line (4,330), DNA (10,589), Cell Type (8,649), RNA (1,069)

For **Review-2**, we have trained the model only on the **BC4CHEMD**, which is an annotated database only for chemicals. [14] [15]

Methodology - How to run the Code :

1. Open the colab notebook provided in the drive link.
2. Install TensorFlow version =1.13.1 and Keras veras 2.2.0 using the following commands:

```
pip install tensorflow==1.13.1
pip install keras==2.2.0
```

3. Restart the Runtime using the option provided under the Runtime sub-menu or ctrl+M
4. After the Runtime is allotted, switch to GPU using the Change Runtime Type under the Runtime sub-menu.
5. Now upload the dataset of your choice to the colab notebook.
6. Change the name of the training file to that of the uploaded dataset in cell 3.
7. Now to execute the entire notebook, use Ctrl+F9 and the code would start running.

4.2 OUTPUT (5 Different set of outputs for 5 Datasets)

OUTPUT FOR BC4CHEMD DATASET

After Pre-processing, we define the model definition and after the establishment of the 3 layers (Lstm, BI-Lstm, CRF), the model summary is printed using the input and output from CRF

```
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 75)                0
_________________________________________________________________
embedding_1 (Embedding)      (None, 75, 20)            810560
_________________________________________________________________
bidirectional_1 (Bidirection (None, 75, 100)           28400
_________________________________________________________________
time_distributed_1 (TimeDist (None, 75, 50)            5050
_________________________________________________________________
crf_1 (CRF)                  (None, 75, 5)             290
=================================================================
Total params: 844,300
Trainable params: 844,300
Non-trainable params: 0
_________________________________________________________________
```

After creating the model, the model is trained for 5 epochs on the entire training corpus on about 30000 samples. The model achieved a significant validation accuracy of 96.71%

```
Train on 30267 samples, validate on 3363 samples
Epoch 1/5
30267/30267 [==============================] - 172s 6ms/step - loss: 7.8300 - crf_viterbi_accuracy: 0.9428 - val_loss: 7.5081 - val_crf_viterbi_accuracy: 0.9636
Epoch 2/5
30267/30267 [==============================] - 169s 6ms/step - loss: 7.7170 - crf_viterbi_accuracy: 0.9726 - val_loss: 7.4853 - val_crf_viterbi_accuracy: 0.9678
Epoch 3/5
30267/30267 [==============================] - 170s 6ms/step - loss: 7.6980 - crf_viterbi_accuracy: 0.9815 - val_loss: 7.4721 - val_crf_viterbi_accuracy: 0.9728
Epoch 4/5
30267/30267 [==============================] - 170s 6ms/step - loss: 7.6904 - crf_viterbi_accuracy: 0.9857 - val_loss: 7.4666 - val_crf_viterbi_accuracy: 0.9798
Epoch 5/5
30267/30267 [==============================] - 169s 6ms/step - loss: 7.6863 - crf_viterbi_accuracy: 0.9880 - val_loss: 7.4636 - val_crf_viterbi_accuracy: 0.9806
```

While evaluating, we consider performance metrics such as precision, recall, and F1 score as accuracy cannot be considered as the best metric for measuring the performance of the model. The average F1 score we obtain is 82%.

```
[ ]  print("F1-score: {:.1%}".format(f1_score(test_labels, pred_labels)))
```

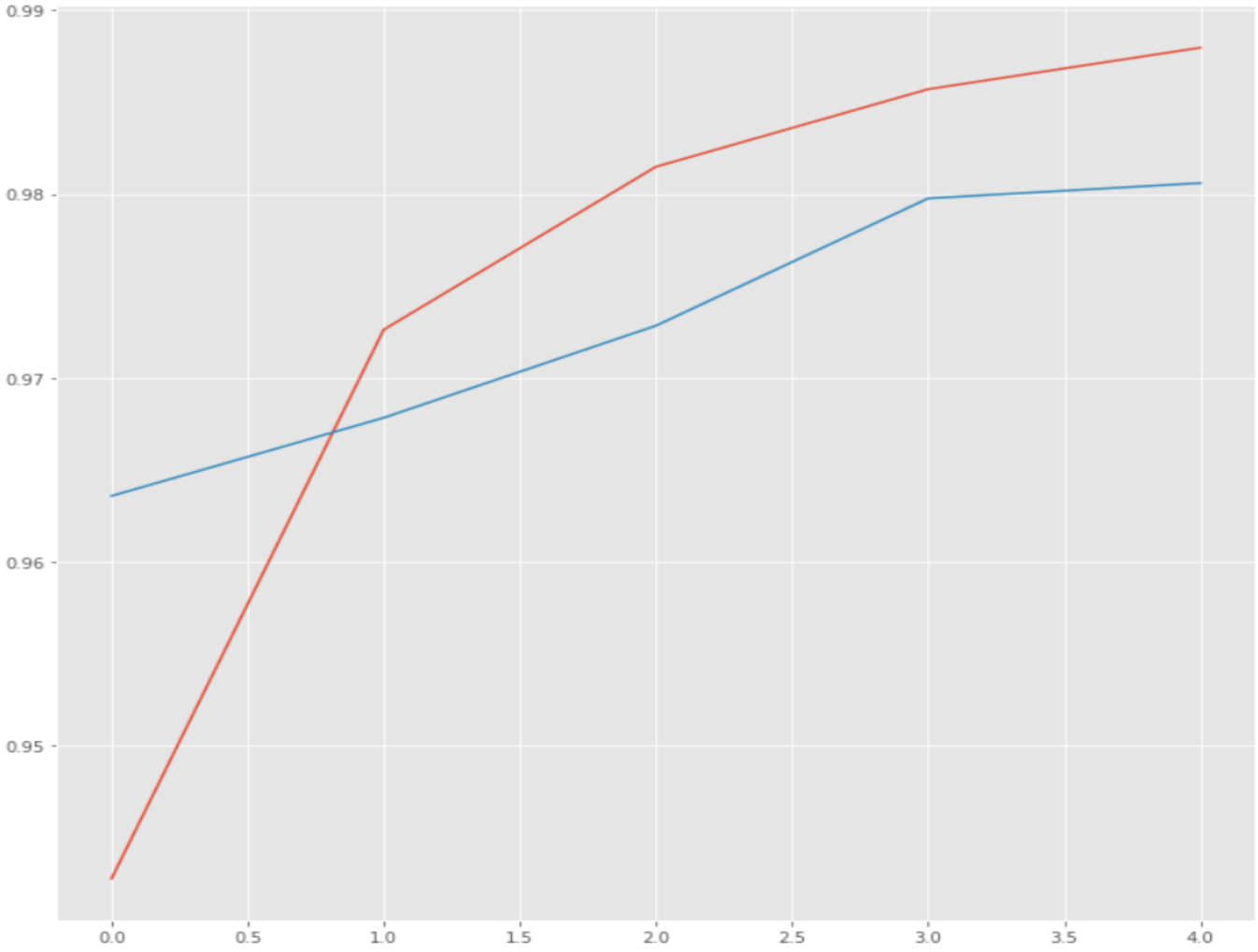
```
F1-score: 82.0%
```

```
[ ]  print(classification_report(test_labels, pred_labels))
```

```
                precision    recall  f1-score   support

      Chemical       0.83      0.81      0.82      2998

     micro avg       0.83      0.81      0.82      2998
     macro avg       0.83      0.81      0.82      2998
  weighted avg       0.83      0.81      0.82      2998
```

In the observations above, we see the performance metrics. Here the average performance comes out to be the same because **Chemical** is the only entity used while training and testing the model.

PERFORMANCE PLOT (x-axis: Epoch & y-axis: Accuracy)

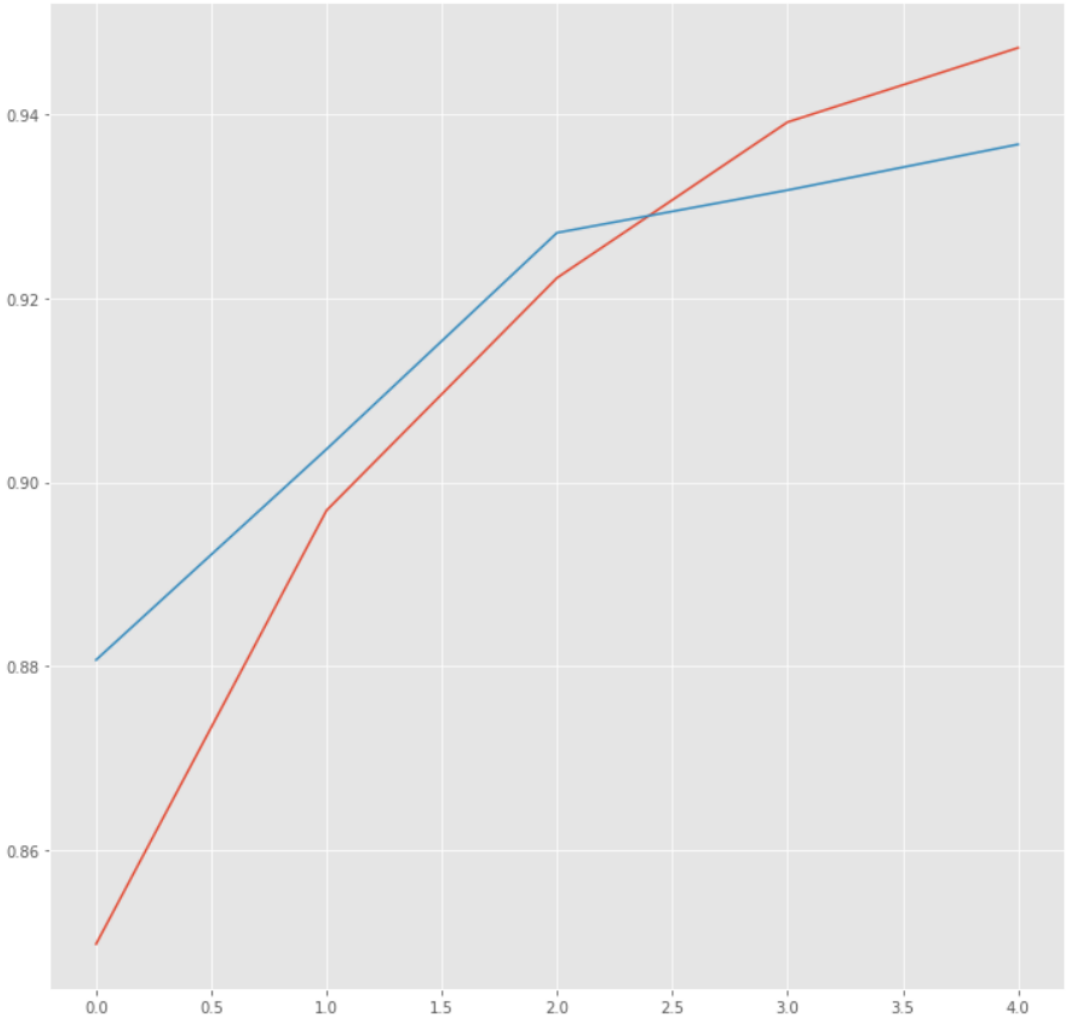


The **Red line refers to Training** and the **blue line refers to testing**.

OUTPUT FOR BC5CDR DATASET

```
Train on 4713 samples, validate on 524 samples
Epoch 1/5
4713/4713 [==============================] - 29s 6ms/step - loss: 12.9208 - crf_viterbi_accuracy: 0.8436 - val_loss: 12.8825 - val_crf_viterbi_accuracy: 0.8792
Epoch 2/5
4713/4713 [==============================] - 27s 6ms/step - loss: 12.5712 - crf_viterbi_accuracy: 0.8965 - val_loss: 12.7664 - val_crf_viterbi_accuracy: 0.9115
Epoch 3/5
4713/4713 [==============================] - 27s 6ms/step - loss: 12.4891 - crf_viterbi_accuracy: 0.9232 - val_loss: 12.7097 - val_crf_viterbi_accuracy: 0.9334
Epoch 4/5
4713/4713 [==============================] - 27s 6ms/step - loss: 12.4352 - crf_viterbi_accuracy: 0.9408 - val_loss: 12.6735 - val_crf_viterbi_accuracy: 0.9386
Epoch 5/5
4713/4713 [==============================] - 27s 6ms/step - loss: 12.3977 - crf_viterbi_accuracy: 0.9522 - val_loss: 12.6620 - val_crf_viterbi_accuracy: 0.9372
```

PERFORMANCE PLOT (x-axis: Epoch & y-axis: Accuracy)



The **Red line refers to Training** and the **blue line refers to testing**.

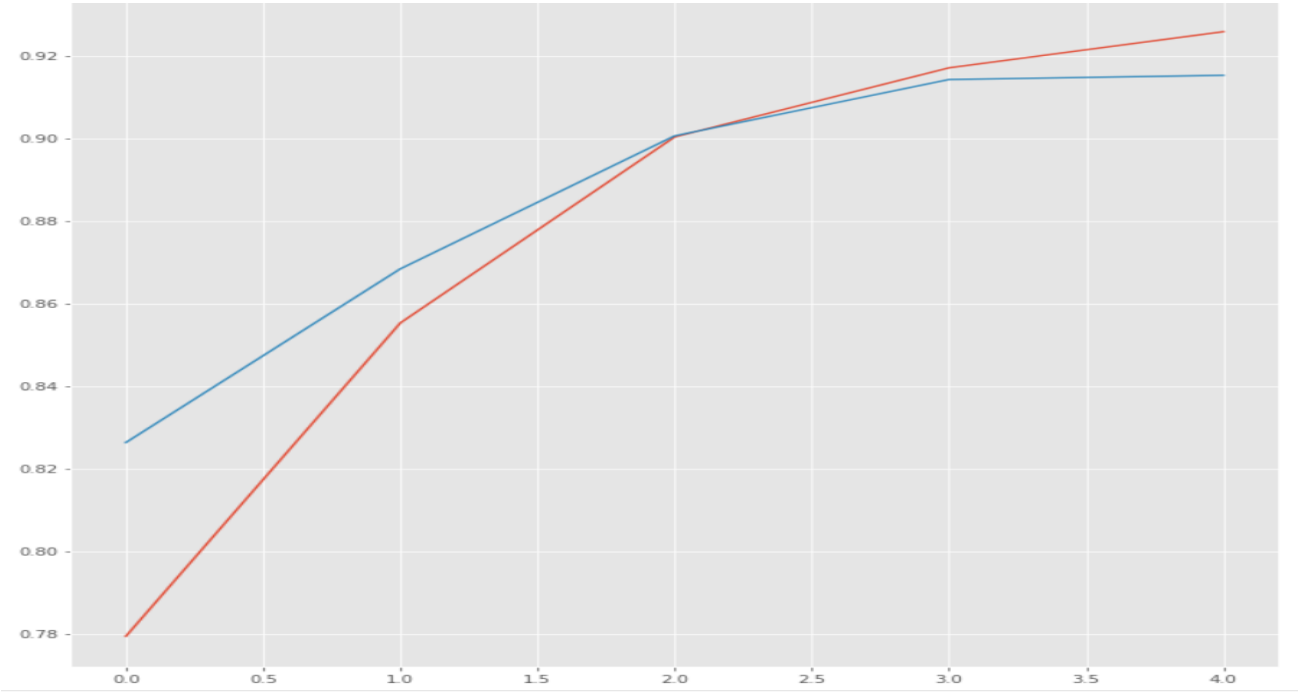
```
print(classification_report(test_labels, pred_labels))
```

	precision	recall	f1-score	support
Chemical	0.65	0.83	0.73	479
Disease	0.27	0.65	0.38	406
micro avg	0.42	0.75	0.53	885
macro avg	0.46	0.74	0.55	885
weighted avg	0.47	0.75	0.57	885

OUTPUT FOR JNLPBA DATASET

```
Train on 13523 samples, validate on 1503 samples
Epoch 1/5
13523/13523 [==============================] - 89s 7ms/step - loss: 8.3209 - crf_viterbi_accuracy: 0.7794 - val_loss: 7.7838 - val_crf_viterbi_accuracy: 0.8263
Epoch 2/5
13523/13523 [==============================] - 85s 6ms/step - loss: 7.8659 - crf_viterbi_accuracy: 0.8552 - val_loss: 7.6184 - val_crf_viterbi_accuracy: 0.8683
Epoch 3/5
13523/13523 [==============================] - 86s 6ms/step - loss: 7.7325 - crf_viterbi_accuracy: 0.9003 - val_loss: 7.5132 - val_crf_viterbi_accuracy: 0.9006
Epoch 4/5
13523/13523 [==============================] - 87s 6ms/step - loss: 7.6693 - crf_viterbi_accuracy: 0.9170 - val_loss: 7.4714 - val_crf_viterbi_accuracy: 0.9142
Epoch 5/5
13523/13523 [==============================] - 86s 6ms/step - loss: 7.6354 - crf_viterbi_accuracy: 0.9258 - val_loss: 7.4498 - val_crf_viterbi_accuracy: 0.9152
```

PERFORMANCE PLOT (x-axis: Epoch & y-axis: Accuracy)



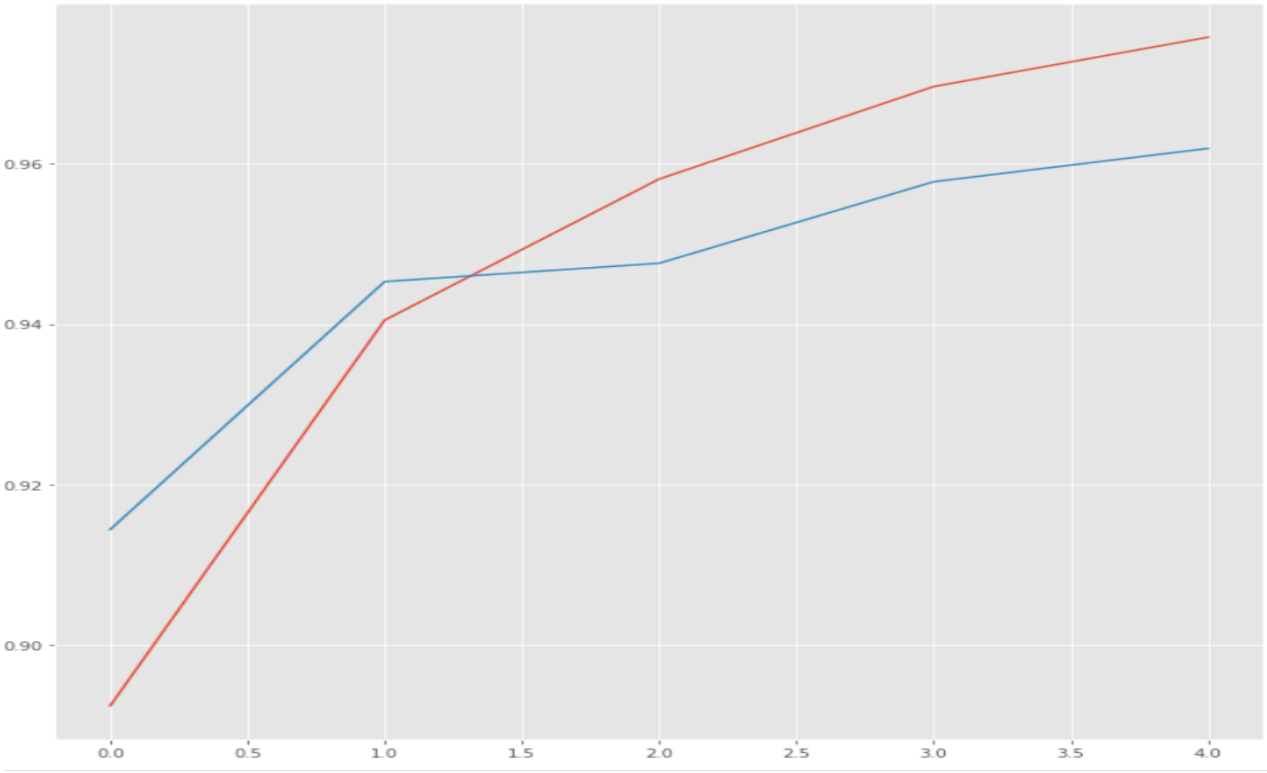
The **Red line refers to Training** and the **blue line refers to testing**.

	precision	recall	f1-score	support
DNA	0.66	0.65	0.66	848
RNA	0.76	0.59	0.67	76
cell_line	0.00	0.54	0.00	317
cell_type	0.76	0.70	0.73	675
protein	0.76	0.82	0.79	2677
micro avg	0.04	0.75	0.08	4593
macro avg	0.59	0.66	0.57	4593
weighted avg	0.69	0.75	0.70	4593

OUTPUT FOR NCBI_DISEASE DATASET

```
Train on 5086 samples, validate on 566 samples
Epoch 1/5
5086/5086 [==============================] - 31s 6ms/step - loss: 14.9266 - crf_viterbi_accuracy: 0.8924 - val_loss: 13.1810 - val_crf_viterbi_accuracy: 0.9144
Epoch 2/5
5086/5086 [==============================] - 27s 5ms/step - loss: 14.6511 - crf_viterbi_accuracy: 0.9405 - val_loss: 13.1026 - val_crf_viterbi_accuracy: 0.9453
Epoch 3/5
5086/5086 [==============================] - 27s 5ms/step - loss: 14.5904 - crf_viterbi_accuracy: 0.9581 - val_loss: 13.0738 - val_crf_viterbi_accuracy: 0.9476
Epoch 4/5
5086/5086 [==============================] - 27s 5ms/step - loss: 14.5606 - crf_viterbi_accuracy: 0.9696 - val_loss: 13.0495 - val_crf_viterbi_accuracy: 0.9577
Epoch 5/5
5086/5086 [==============================] - 27s 5ms/step - loss: 14.5413 - crf_viterbi_accuracy: 0.9757 - val_loss: 13.0334 - val_crf_viterbi_accuracy: 0.9619
```

PERFORMANCE PLOT (x-axis: Epoch & y-axis: Accuracy)



The **Red line refers to Training** and the **blue line refers to testing**.

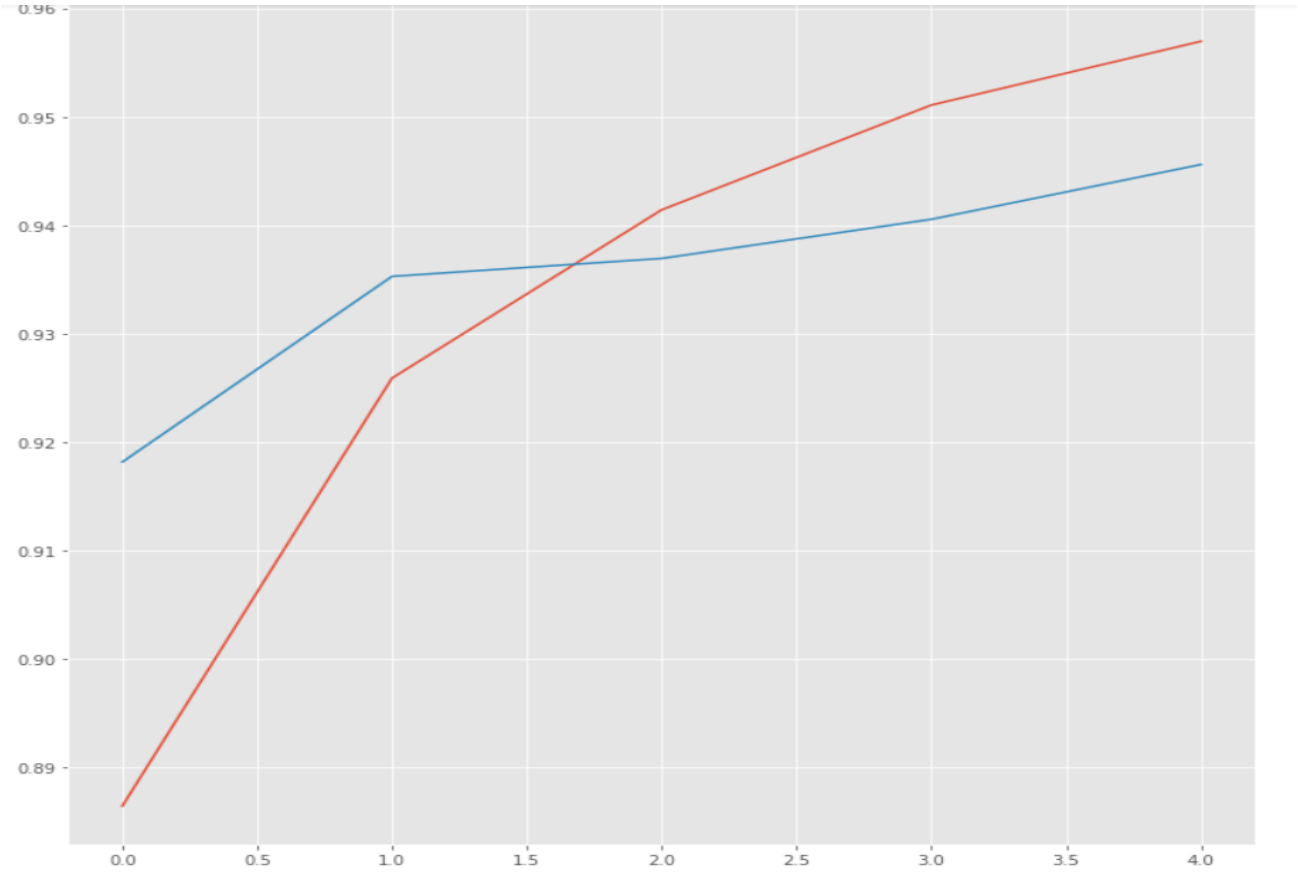
```
print(classification_report(test_labels, pred_labels))
```

```
                precision    recall  f1-score   support

       Disease       0.32      0.74      0.45       487

     micro avg       0.32      0.74      0.45       487
     macro avg       0.32      0.74      0.45       487
  weighted avg       0.32      0.74      0.45       487
```

OUTPUT FOR BC2GM DATASET

```
Train on 12204 samples, validate on 1357 samples
Epoch 1/5
12204/12204 [==============================] - 72s 6ms/step - loss: 8.5102 - crf_viterbi_accuracy: 0.8865 - val_loss: 8.0445 - val_crf_viterbi_accuracy: 0.9182
Epoch 2/5
12204/12204 [==============================] - 67s 5ms/step - loss: 8.3066 - crf_viterbi_accuracy: 0.9259 - val_loss: 7.9693 - val_crf_viterbi_accuracy: 0.9353
Epoch 3/5
12204/12204 [==============================] - 66s 5ms/step - loss: 8.2493 - crf_viterbi_accuracy: 0.9414 - val_loss: 7.9418 - val_crf_viterbi_accuracy: 0.9370
Epoch 4/5
12204/12204 [==============================] - 67s 5ms/step - loss: 8.2187 - crf_viterbi_accuracy: 0.9511 - val_loss: 7.9209 - val_crf_viterbi_accuracy: 0.9406
Epoch 5/5
12204/12204 [==============================] - 67s 5ms/step - loss: 8.1999 - crf_viterbi_accuracy: 0.9570 - val_loss: 7.9101 - val_crf_viterbi_accuracy: 0.9456
```

PERFORMANCE PLOT (x-axis: Epoch & y-axis: Accuracy)



The **Red line refers to Training** and the **blue line refers to testing**.

```
print(classification_report(test_labels, pred_labels))
```

```
              precision    recall  f1-score   support

        GENE       0.01      0.51      0.02      1569

   micro avg       0.01      0.51      0.02      1569
   macro avg       0.01      0.51      0.02      1569
weighted avg       0.01      0.51      0.02      1569
```

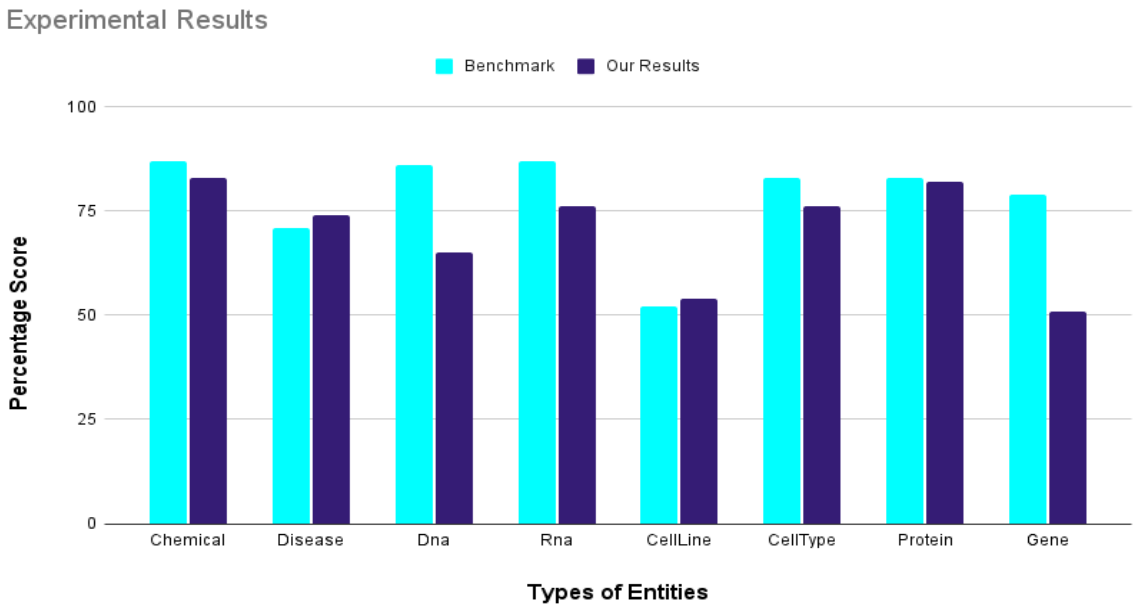
PREDICTION AND TESTING RESULTS (For BC4CHEMD Dataset)

The table below shows 2 sentences from the testing data. The true output is the one annotated by the creators of the database and the predicted output is the one predicted by our model. We have chosen these examples specifically to depict that our model can not only successfully predict single token entities but also the multi-word entity accurately. But as the model isn't 100% accurate, it predicts co as '**S-Chemical'** instead of '**O'.**

```
Word            ||True ||Pred
==============================
Co              : O     S-Chemical
-               : O     O
immunoprecipitation: O     O
and             : O     O
pull            : O     O
-               : O     O
down            : O     O
studies         : O     O
showed          : O     O
that            : O     O
only            : O     O
Ca              : B-Chemical B-Chemical
(               : I-Chemical I-Chemical
2               : I-Chemical I-Chemical
+               : I-Chemical I-Chemical
)               : E-Chemical E-Chemical
-               : O     O
bound           : O     O
calmodulin      : O     O
was             : O     O
able            : O     O
to              : O     O
bind            : O     O
DdCAD           : O     O
-               : O     O
1               : O     O
.               : O     O
```

```
Word            ||True ||Pred
==============================
Fourier         : O     O
transform       : O     O
infrared        : O     O
spectroscopy    : O     O
is              : O     O
employed        : O     O
to              : O     O
analyze         : O     O
the             : O     O
conformational  : O     O
changes         : O     O
of              : O     O
TGF             : O     O
-               : O     O
beta            : O     O
1               : O     O
on              : O     O
the             : O     O
surface         : O     O
of              : O     O
AuNPs           : O     O
.               : O     O
```

Performance Analysis



Here we showed the analysis of the performance of our model vs the benchmark score for the 8 entities. The graph shows that our model is not even close to perfect. For entities like DNA, RNA, and GENE the obtained score is much lower than the benchmark score. That shows the model needs to be more flexible and fitted using better training parameters with the use of better-annotated datasets. On the other hand, entities such as Chemical , Disease, and Protein show up to the mark of Benchmark datasets.

5. CONCLUSION

The performance of the model is excellent as per the metrics and the testing data proves it too. Having a CRF layer on the bidirectional layer provides an additional boost to the accuracy of the result showing the power of the Keras library. [16] It is shown that our embedding layer can improve the performance slightly, even though it still can not beat the performance of using pre-trained word representation, such as Benchmark Level and. It can also be observed that the performance of the LSTM-CRF model will not improve, with larger annotated datasets. As shown in the results, our output can be significantly improved with the use of better and bigger datasets while using higher GPUs to reduce the training time.

GLOSSARY

Precision: Precision talks about how precise/accurate your model is out of those predicted positive, how many of them are positive. It's calculated as :

= True Positive / Total Predicted Positive

Recall: Recall calculates how many of the Actual Positives our model capture through labeling it as Positive (True Positive). It is calculated as :

= True Positive / Total Actual Positive

F1 score: F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall and there is an uneven class distribution.It is calculated as :

= 2* (Precision * Recall) / (Precision + Recall)

6. REFERENCES

[1.] Bolucu, Necva, et al. “Bidirectional LSTM-CNNs with Extended Features for Named Entity Recognition.” *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, 2019, doi:10.1109/ebbt.2019.8741631.

[2.] Duan, Jianyong, et al. “Chinese Spelling Check via Bidirectional LSTM-CRF.” *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, doi:10.1109/itaic.2019.8785520.

[3.] Huang, Zhiheng, et al. “Bidirectional LSTM-CRF Models for Sequence Tagging.” *ArXiv.org*, 9 Aug. 2015, arxiv.org/abs/1508.01991.

[4.] Kim, Seon-Wu, and Sung-Pil Choi. “Research on Joint Models for Korean Word Spacing and POS (Part-Of-Speech) Tagging Based on Bidirectional LSTM-CRF.” *Journal of KIISE*, vol. 45, no. 8, 2018, pp. 792–800., doi:10.5626/jok.2018.45.8.792.

[5.] Jeong, Yewon, and Jong-Hyeok Lee. “Extending Word Representations with Predicted Affix Features for Bidirectional LSTM-CRF-Based Korean Named Entity Recognition.” *KIISE Transactions on Computing Practices*, vol. 26, no. 9, 2020, pp. 408–413., doi:10.5626/ktcp.2020.26.9.408.

[6.] Liu, Yupeng, et al. “Semi-Markov CRF Model Based on Stacked Neural Bi-LSTM for Sequence Labeling.” *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, 2020, doi:10.1109/iicspi51290.2020.9332321.

[7.] Misawa, Shotaro, et al. “Character-Based Bidirectional LSTM-CRF with Words and Characters For Japanese Named Entity Recognition.” *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, 2017, doi:10.18653/v1/w17-4114.

[8.] Qin, Ying, and Yingfei Zeng. “Research of Clinical Named Entity Recognition Based on Bi-LSTM-CRF.” *Journal of Shanghai Jiaotong University (Science)*, vol. 23, no. 3, 2018, pp.

392–397., doi:10.1007/s12204-018-1954-5.

[9.] Sterbak, Tobias. “Sequence Tagging with LSTM-CRFs.” *Depends on the Definition*, 20 Apr. 2020, www.depends-on-the-definition.com/sequence-tagging-lstm-crf/.

[10.] Utkarsh, Kumar. “Named Entity Recognition Using Bidirectional LSTM-CRF.” *Medium*, Medium, 15 Mar. 2020, utkarsh-kumar2407.medium.com/named-entity-recognition-using-bidirectional-lstm-crf-9f4942746b3c.

[11.] Wang, Xuan, et al. “Cross-Type Biomedical Named Entity Recognition with Deep Multi-Task Learning.” *Bioinformatics*, vol. 35, no. 10, 2018, pp. 1745–1752., doi:10.1093/bioinformatics/bty869.

[12.] Wang, Yan, et al. “Biomedical Event Trigger Detection Based on Bidirectional LSTM and CRF.” *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017, doi:10.1109/bibm.2017.8217689.

[13.] Yang, Li, et al. “Post Text Processing of Chinese Speech Recognition Based on Bidirectional LSTM Networks and CRF.” *Electronics*, vol. 8, no. 11, 2019, p. 1248., doi:10.3390/electronics8111248.

[14.] Yang, Xuemin, et al. “Bidirectional LSTM-CRF for Biomedical Named Entity Recognition.” *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2018, doi:10.1109/fskd.2018.8687117.

[15.] Yuzhimanhua. “Yuzhimanhua/Multi-BioNER: Cross-Type Biomedical Named Entity Recognition with Deep Multi-Task Learning (Bioinformatics'19).” *GitHub*, github.com/yuzhimanhua/Multi-BioNER.

[16.] Zeyer, Albert, et al. “Towards Online-Recognition with Deep Bidirectional LSTM Acoustic Models.” *Interspeech 2016*, 2016, doi:10.21437/interspeech.2016-759.