# Multiple Approaches of Named Entity Recognition

**Anand Shrivastava[1] G. D. Makwana[2]**
[1,2]GTU - Graduate School of Engineering and Technology, Ahmedabad, India

*Abstract*— The Named Entity Recognition (NER) is a unique structure where annotated sequences can contain inside each other. Named Entity Recognition is a challenging task in nature language processing (NLP). The document is annotated in two different fashion , from specific to general which is known as inside to outside and general to specific which is known as outside to inside. These approaches are validated on various datasets. Word2Vec to Bert is used For generating word or character vector, Another well known approach - Long Short Term memory (LSTM) is widely used in Natural Language Processing (NLP) and here it further enhanced in by turning it to Bidirectional LSTM - BiLSTM and cloubed with Conditional Random Field (CRF) to generate more powerful model for NER with highest accuracy possible. Major impact of Named Entity Recognition (NER) could be on the medical sector. Currently the model achieves state-of-the-art performance. Research in biomedical BioBERT model has three major types: memorization, synonym generalization, and concept generalization. By applying statistical debiasing technique to overcome the model bias over a dataset. By leveraging the current deep bidirectional transformer model like BERT and GPT-3 the requirement for manually annotated dataset can be reduced , the BioNER model requires the manually annotated multiple entity type dataset, The dataset can be available with single type of entity which make difficult to train model for multiple entity, hence it requires to use two different kind of dataset and this issue is targeted by TaughtNER model a knowledge distillation based model which allows to finetune a single multi task student model by leveraging the both the ground truth. Multiple text mining tools help researchers to extract biomedical documents like tmTool, ezTag. Another type of model is proposed which aims to resolve the overlapping entity recognition issue which is called BERN, a neural biomedical named entity recognition and multi type normalization tool. BERN used high performance BioBERT which recognised known entities and discovered new entities.

***Key words:*** The Named Entity Recognition (NER), Nature Language Processing (NLP)

## I. INTRODUCTION

Named entity recognition (NER) is a well-established tech- nique in natural language processing (NLP) which involves finding and classifying named entities in text. These tasks have been widely applied in various domains for identifica- tion of names of people, organizations, temporal expressions, geographic locations, or specialized entities in scientific doc- uments [1]. From a practical point of view, finding named entities or, more generally, entity mentions is useful in solving more complex problems such as information retrieval, knowl- edge base population, or natural language understanding (NLU). From a supervised learning perspective, named entity recognition is an example of structured prediction, that is, the prediction of structured objects from the data rather than simple categories (classification) or numeric values (regres- sion). Typically, the task is simplified by the assumption that named entity mentions can- not overlap. As a fundamental task of natural language processing tasks, named entity recognition (NER) aims to identify the named entities from unlabeled sentences or texts. Named entities are a series of special semantic types such as person (PER), organization (ORG) and location (LOC), etc. Thus, NER is a typical classification task that trains a model with texts in which named entities have been labeled rightly, and then predicts the named entities in other unlabeled texts. NER has received much attention for it will impact the performance of other downstream NLP tasks, such as relation extraction, entity linking [1], etc. In recent years, deep learning technologies have been widely used in a variety of NLP and computer vision (CV) tasks. The most popular RNN model is Long Short-Term Memory (LSTM) that has achieved success in many NLP tasks, . For NER, the state-of-the-art models are usually based on BiLSTM-CRF which uses BiLSTM to extract the features of input sentences and connect them to a conditional random field (CRF) layer to jointly predict target labels[2]. Biomedical named entity recognition (BioNER), which involves identifying biomedical named entities in unstructured text, is a core task to do so since entities extracted by BioNER systems are utilized as important features in many downstream tasks such as drug-drug interaction extraction. One important desideratum of BioNER models is to be able to generalize to unseen entity mentions. (Corresponding author: Marco Postiglione. postiglione@unina co/marcopost-it/TaughtNet-disease-chem-gene[4] There are over 29 million articles in PubMed as of May 2019, and the amount of biomedical literature has been growing rapidly in recent years. Fast and precise text mining tools can reduce the amount of effort and time it takes researchers to find and extract useful information from the vast amount of biomedical literature. Researchers have used named entity recognition (NER) and named entity normalization (NEN) models to develop effective biomedical text mining tools for information retrieval , question answering , relation extraction , and so on[5].

## II. RELATED WORK AND METHODS

They proposed a number of techniques based on CRF, namely layering, cascading, and joined label tagging. In layering, several CRF models are trained to detect subsequent levels of named entities. Cascading works by training separate CRF models to identify separate types of named entity. A major drawback of cascading is that it cannot detect nested enti- ties of the same type for the reason that type-specific CRF models generate flat predictions. The final method, joined label tagging, trains a single CRF model[1], but expands the set of all possible labels by joining nested tags from all levels into a single label. Two years later, Finkel and Manning formulated the task of nested NER as a parsing problem, and proposed a CRF-based algorithm with O(n 3 ) time complexity for solving it. Hypothesis based model [2] Another group of methods models the representation of

entity mentions in a sentence as a hypergraph. Such a hypergraph consists of various types of node that represent specific mention properties, such as its left and right boundaries, labels, or words that are included in the mention. The construction of an optimal mention hypergraph was solved using log linear modeling based on a number of handcrafted features. This work was later improved by Muis and Lu , who introduced a novel encoding scheme for nested mentions that assigns labels to the gaps between words. Wang and Lu proposed a modified hypergraph representation that did not suffer from structural ambiguity, and included representation generated by bidirectional LSTM [2] for span, word, and character-level features. Katiyar and Cardie proposed a method for building entity mention hypergraphs which varied from the previ- ous approaches inspired by Lu and Roth. Their decoder layer constructs a simple hypergraph based on the BILOU (beginning-inside- last-outside-unary) tagging scheme. Hypergraph-based approaches are flexible and capable of modeling many types of nested structure. NEURAL METHODS In recent years, several neural architectures for nested named entity recognition have been developed. A similarly exhaustive approach is presented in the research of Sohrab and Miwa, in which an LSTM encoder is used to build word and sequence represen- tations, and the output layer predicts a label for each possible sequence. The idea of combining several models into one nested named entity recognition system is often exploited. train two neural models - one for identifying the anchor words of a mention, and the other for detecting word regions around those anchors. The first model, detector is responsible for identifying which word segments are named entity mentions: the second, classifier is responsible for assigning categories to those mentions. The second model assigns a label to each entity candidate, based on its right and left context as input features. Another popular approach involves the use of transition- based parsing. combine this method with a neural model to predict a set of nested entity tags represented as a forest where each outermost entity is a root of a tree, and nested entities are its children. Methods not falling into any of the above categories are usually based on complex, multilayer neural architectures.detects nested entities in the inside-out order with a stack of LSTM layers. Each layer has a CRF [1] output generating predictions for the current level of nesting, and its hidden states are passed to the next layer until no new entities are detected. have trained an LSTM-based multitask model to jointly detect the boundaries of named entities and classify them. The main contribution of their work is the modification of a Viterbi decoding algorithm in the CRF layer to recursively identify nested entities in the outside-in manner. 3) Model intersection - In this policy, an entity mention is included in the resulting set, S if and only if it is present both in Sin and Sout.

From the results of we can see that applying the same residual structure of ResNets to BiLSTM is not as effective as CNN. A token represents an English word or a Chinese character in our model. In the look-up layer, we map each token to a vector as inputs of the BiLSTM. Word2vec, GloVe and BERT are all available and adopted to generate input vectors. Specifically, BERT can generate dynamic vectors of English word and Chinese characters. Word2vec can generate fixed vectors of Chinese characters. GloVe can generate fixed vectors for English words. On account of that many English named entities are out of vocabulary words, we use the same method proposed by to help generating fixed vectors, which uses BiLSTM to encode each character in a word. RESIDUAL BiLSTM BLOCKS [2] will illustrate the structure of the residual BiLSTM blocks we proposed. We take the l-th blocks as an example to illustrate the structure of residual BiLSTM blocks.

We use BERT to compare general and domain-specific PLMs in terms of generalization in BioNER. DICTIONARY MODELS Traditional approaches in the field of BioNER are based on pre-defined dictionaries . To compare the generalization abilities of traditional and recent approaches, we set two types of simple dictionary-based extractors as baseline models. DICTtrain uses all the entity mentions in a training set (i. DICTsyn expands the dictionary to use entity mentions in the training set as well as their synonyms, which are pre-defined in biomedical databases. BioBERT outperforms other baseline models on NCBI- disease based on overall performance. BERT performs less than domain-specific PLMs, but far superior to dictionary models. DICTsyn outperforms DICTtrain in recall due to its larger biomedical dictionary, but the precision scores decrease in general. Note that the performance of DICTsyn on Mem is lower than that of DICTtrain as there exists annotation inconsistency between benchmarks and biomedical databases.

In the first scenario, data can be easily retrieved (at least for the English language), allowing for the collection of enormous amounts of raw text to train the model; in the second scenario, data is more difficult to gather and share due to privacy concerns, but is closer to the real world of medical practise than the idealized information found in textbooks and journals . paper[3] mainly focused on pretrain a novel language model, but rather to design a fine-tuning framework which, based on knowledge distillation, allows us to accomplish the NER task for multiple entities by exploiting pretrained language models and heterogeneous publicly available healthcare datasets, each of them referring to a different entity type. To the best of our knowledge, is the first work adopting the multi-task learning framework with a pre-trained language model. To solve such false-positive problem, the authors propose CollaboNet, a network composed of multiple models, each one built on a different dataset for a different task, which collaborates during training and inferences to output the final prediction. De- spite the promising results, this framework requires “collabora- tor” models to be stored in memory at inference time and to pro- vide their outputs when a prediction is required, resulting in low efficiency in computational and memory consumption terms. To overcome the low-precision and the computational and memory consumption challenges, inspired by CollaboNet, we developed TaughtNet[4], a training framework which allows us to fine-tune a single transformer language model for multi-task BioNER based on Knowledge Distillation. In simple terms, we train single-task models on different datasets, but they do not collaborate to provide the outputs of predictions, but rather to “teach” to a single multi-task “student” how to predict the entity types in which they are experts. D. Knowledge Distillation (KD) as a teacher-student framework which allows the knowl- edge embedded in a large “teacher” model to be shared with its small “student”. Modeling the behavior of teacher and student with functions $f_T(\cdot)$ and $f_S(\cdot)$, respectively, the objective of KD is to minimize the following objective function: $L = x \in X\ L(f_S(x), f_T(x))$, (1) where X is the training dataset and$L(\cdot)$ denotes the loss function computing the difference between the two behavior

function outputs for the input x ∈ X . With the primary aim to "compress" the knowledge embedded in a large model — which shows good performance but is too large to be used in real scenarios — into a smaller one, the application of KG in NLP and pre-trained models has been extensively studied try to transfer the knowledge embedded in an English BERT model to the German language. In, a fine- tuned BERT teacher is used as extra supervision to improve the text generation performance of conventional Seq2Seq student models. To the best of our knowledge, TaughtNet is the first approach exploiting KD in a NER scenario to transfer the knowledge encoded in a variety of teachers, specialized in single entity types, into a single student, which learns to recognize all the entity types. The multi-teacher scenario in the application of the KD approach has been thoroughly investigated The use of an en- semble knowledge distillation framework in results in better student accuracy thanks to the encouragement of heterogeneity in feature learning.

In previous studies, NER models were used to recognize entities in biomedical text even when the entities overlapped. In recent years, Wang and Lu , and Katiyar and Cardie proposed models for learning time-efficient hypergraph representations of overlapping entity mentions. proposed a model which consists of Bi-LSTM [2] and an expectation-maximization (EM) marginal CRF, and recognizes disjoint or partially overlapping sets of entity types.Named Entity normalization models for biomedical text mining, there are various types of entities which are referred to as different names in biomedical text. tmTool, PubTerm, ezTag, and BERN [5] commonly use GNorm- Plus for gene normalization, SR4GN for species normalization (only dictionary lookup for BERN), and tmVar for mutation normalization. GNormPlus uses exact match and bag-of-words match to pair recognized names with concepts in Entrez Gene . On the other hand, text mining tools use different models for disease and chemical normalization. ezTag uses Tag- gerOne to normalize disease and chemical entities, and TaggerOne jointly performs NER and normalization using semi-Markov models. tmTool and PubTerm use DNorm, which is based on pairwise learning to rank (pLTR), to normalize disease entities. BERN uses the sieve-based entity linking approach of D`Souza and Ng to normalize disease entities. tmChem , which is used by tmTool, PubTerm and BERN[5], converts recognized chemical entity names and chemical entity names in the lexicon of tmChem to lowercase letters, and removes whitespace and punctuation. BIOBERT FOR NAMED ENTITY RECOGNITION BioBERT NER models used by BERN [5] recognize known enti- ties and discover new entities using WordPiece embed- dings. As a result, the WordPiece embeddings can be used to extract features of rare or unknown words, which is very helpful in discovering new entities. The BioBERT NER[3] models are fine-tuned as follows: p(yi = k|Ti) = softmax(TiW>+b)k , k = 0, 1, . BioBERT NER models compute the probabilities of the following seven tags: IOB2 tags (``I``nside, ``O``utside, ``B``egin) , ``X`` (a sub-token of WordPiece), ``[CLS]`` (the first token of every sequence for classification), ``[SEP]`` (a delimiter between sentences), and ``PAD`` (padding) of each word in a sentence. Note that the BioBERT NER models make predictions for ``I,`` ``O,`` and ``B`` tags but not for the ``X,`` ``[CLS],`` ``[SEP],`` or ``PAD`` tags. As a result, BERN can discover new entities using BioBERT NER models. As shown in Table 1, the BioBERT NER models used by BERN obtain the highest F1-scores on the test sets for all types except species. The BioBERT NER models of BERN outperform the NER models of tmTool, PubTerm, and ezTag on test sets of genes/proteins (BC2GM 5. Note that BioBERT NER models can recognize all types of entities if there is training data.

## III. DECISION RULES

After conducting the case studies above, we developed deci- sion rules for overlapping entities in multi-type NER results. , BERN uses decision rules for determining which entities to choose if overlapping entities are found. 3 shows the decision rules that BERN uses for the multi-type NER results. 9%)), BERN tags the mutation and entities that the mentions are most likely to be entities predicted by BioBERT NER models. 1%)), only the entities with the highest probability of being an actual entity are tagged; the probability is calculated by BioBERT NER models. 0 used by BERN [5] generally achieve much higher precision (over 97%) on mutations than on other entity types.

## IV. EXPERIMENT OVER DIFFERENT DATASET

Experiments were conducted on four nested named entity recognition datasets: GENIA (biomedical domain), NNE (news domain), PolEval (mixed texts, a Polish corpus), and GermEval (news and Wikipedia, a German corpus). Each dataset was split into three parts: training, validation, and test sets. The validation set is utilized for hyperparameter selection, early stopping, and finding the optimal threshold parameters for probability-based and linear classifier selection policies. The test set is used only for the final evaluation, and for reporting the results of the model. Since random initialization of the network`s weights and shuffling of batches during training affects the final results of the model, reporting a single result might misrepresent the actual performance of our approach. We compared the results of our algorithm with other neural and non-neural methods, using scores provided by their authors or, in the case of NNE dataset, our own evaluation based on the released source code of the models. We employed a training scheduler with a decreasing learning rate and early stopping, based on the validation set performance. We consider the choice of the number of BiLSTM [2] layers, the hidden size of a BiLSTM layer and a contextual word representation among the most commonly used pre-trained language models: Flair , ELMo, and BERT. proved to be particularly effective, setting new state-of- the-art results on several popular named entity recognition datasets, such as CoNLL-2003 , OntoNotes, and WNUT 2017, as well as the CoNLL datasets for the German and Dutch languages. The initial configuration of our inside-out and outside-in models followed, where applicable, the hyperparameters of the best performing architecture from Akbik et al. Therefore, the optimal hyperparameters for an iterative model should not be different from the those of a traditional NER model. In those experiments, we trained separate inside-out and outside-in iterative models with different hyperparameter values. As with the other experiments, we repeated the procedure three times and reported the average F1-scores. We tested three different word representations, four LSTM hidden sizes (from 128 to 512 neurons), and architectures

from one to four stacked BiLSTM layers. In each experiment, the initial values have been used for the remaining hyperparameters.

## V. RESULTS ON ENGLISH NER DATASETS

In this section, we perform our model on English NER Datasets ConLL-2003 and OntoNotes 5.0. We take the same approach proposed by to generate English input vectors for English NER, where the inputs of English NER are composed of pre-trained word vectors from GloVe1 and character vectors learned by a BiLSTM network. The results are shown in Table 4 and Table 5. Our model achieves a F1-score of 92.22% and 89.65% on CoNLL-2003 and OntoNotes 5.0 respectively, which outperform the baselines on the both datasets. Our model also outperforms the residual LSTM [2] model in significantly. Meanwhile, we can observe that stacked BiLSTM model performs worse that . It demonstrates that shortcut connection can improve the performance of stacked BiLSTM, and the residual structure in our model is more effective and reasonable than which uses the same structure of ResNets. Since most NLP task can benefit from BERT, we also adopt BERT 2 to generate dynamic input vector for our model on the ConLL-2003 dataset. We use the official BERT tools3 offered by Google to program which adopts AdamW algorithm for optimization. On account of that our model is more complex to fine tune with BERT, we use the method proposed by which contains two steps to fine tune a complex model with BERT. Table 6 shows the F1-scores on ConLL-2003.

The baselines also adopt BERT or ELMo as the input. We can see that our model work with BERT more effectively than baselines, which again shows the effectiveness and robustness of our model.

In the first place, we train three students for three different biomedical entity types: diseases, chemical compounds and genetic information. Thereafter, we train several student architectures with different size and parameters, and report our results in the Results subsection. Specifically, we report: (1) a comparison of our best student with several state-of-the-art base- lines; (2) results of different students with different architectures and size; (3) a comparison of all the students in terms of their level of agreement on predictions; (4) an error analysis w. r. t different error types; and (5) an explainability experiment which investigates how the inner workings change from the teachers to the student. A. Datasets and Teachers We evaluate the performance of our approach with three benchmark datasets, each of which has been constructed from PubMed abstract: NCBI-Disease , BC5CDR , BC2GM . For each one of the datasets, we trained our teachers by fine- tuning for 30 epochs a RoBERTa-large architecture which had been pre-trained on PubMed and PMC and MIMIC-III with a BPE Vocab learnt from PubMed . A summary of the datasets, in terms of size and entity-type, and of the teachers, in terms of their precision, recall and F1 scores, is provided in Table III. In particular, training/development/test splits of NCBI-disease and BC5CDR corpora are the same as their original version, while the training set of BC2GM has been modified because the original corpus does not provide a development set. Thus, 2,500 sentences are split off from the training data to generate the development set. C. Metrics Quality: For the evaluation of the quality of the named entity recognition approaches, we used the Precision, Recall and F1 metrics computed with the seqeval Python framework. In simple terms, Precision is the percentage of entities which are correctly found by the system, while Recall is the percentage of entities of the test set which are found by the system. Memory occupation and inference time: The efficiency of models has been evaluated based on their size (in terms of MB of memory occupied) and the samples-per-second (SPS) required during the training and inference phases. We experimented with several model architectures and weights with varying size.

## VI. CONCLUSION

An iterative deep learning model for nested named entity recognition in which the representation of words is constructed from the character and word-level features, and a vector of encoded entity types identified in the previous iterations. We introduce a new type of residual block based on BiLSTMs. Being different from most other state-of-the-art models that introduce external knowledge or multi-task learning, we make efforts to innovate on the structure of residual network based on BiLSTMs. TaughtNet has the objective to integrate various publicly available single-task healthcare datasets in a single BERT architecture which can be used as a fast and highly performing BioNER engine in real applications, such as conversational agents or knowledge graph development. Experimental results demonstrate that not only does Taugh- tNet surpass strong state-of-the-art baselines, but it also is a valuable option when constrained by strict computational and memory requirements thanks to its ability to train lightweight models that distill the knowledge from high-performing single- task teachers. 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. Researchers can use BERN for text mining tasks such as new named entity discovery, information retrieval, question answering, and relation extraction.

## REFERENCES

[1] S. Dadas and J. Protasiewicz, "A Bidirectional Iterative Algorithm for Nested Named Entity Recognition," in IEEE Access, Vol. 8, pp. 135091-135102, 2020, doi: 10.1109/ACCESS.2020.3011598.
[2] G. Yang and H. Xu, "A Residual BiLSTM Model for Named Entity Recognition," in IEEE Access, Vol. 8, pp. 227710-227718, 2020, doi: 10.1109/ACCESS.2020.3046253.
[3] H. Kim and J. Kang, "How Do Your Biomedical Named Entity Recognition Models Generalize to Novel Entities?," in IEEE Access, Vol. 10, pp. 31513-31523, 2022, doi: 10.1109/ACCESS.2022.3157854.

[4] V. Moscato, M. Postiglione, C. Sansone and G. Sperlí, "TaughtNet: Learning Multi-Task Biomedical Named Entity Recognition From Single-Task Teachers," in IEEE Journal of Biomedical and Health Informatics, Vol. 27, No. 5, pp. 2512-2523, May 2023, doi: 10.1109/JBHI.2023.3244044.
[5] D. Kim et al., "A Neural Named Entity Recognition and Multi-Type Normalization Tool for Biomedical Text Mining," in IEEE Access, Vol. 7, pp. 73729-73740, 2019, doi: 10.1109/ACCESS.2019.2920708.