TeluguNER: Leveraging Multi-Domain Named Entity Recognition with Deep Transformers

Suma Reddy Duggenpudi[1], **Subba Reddy Oota**[1,2], **Mounika Marreddy**[1] **Radhika Mamidi**[1]
[1]IIIT Hyderabad, India; [2]INRIA, Bordeaux, France
sumareddy.duggenpudi@research.iiit.ac.in, subba-reddy.oota@inria.fr
mounika.marreddy@research.iiit.ac.in, radhika.mamidi@iiit.ac.in

Abstract

Named Entity Recognition (NER) is a successful and well-researched problem in English due to the availability of resources. The transformer models, specifically the masked-language models (MLM), have shown remarkable performance in NER in recent times. With growing data in different online platforms, there is a need for NER in other languages too. NER remains underexplored in Indian languages due to the lack of resources and tools. Our contributions in this paper include (i) Two annotated NER datasets for the Telugu language in multiple domains: Newswire Dataset (ND) and Medical Dataset (MD), and we combined ND and MD to form a Combined Dataset (CD) (ii) Comparison of the finetuned Telugu pretrained transformer models (*BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te*) with other baseline models (CRF, LSTM-CRF, and BiLSTM-CRF) (iii) Further investigation of the performance of Telugu pretrained transformer models against the multilingual models mBERT (Devlin et al., 2018), XLM-R (Conneau et al., 2020), and IndicBERT (Kakwani et al., 2020). We find that pretrained Telugu language models (*BERT-Te* and *RoBERTa*) outperform the existing pretrained multilingual and baseline models in NER. On a large dataset (CD) of 38,363 sentences, the *BERT-Te* achieves a high F1-score of 0.80 (entity-level) and 0.75 (token-level). Further, these pretrained Telugu models have shown state-of-the-art performance on various Telugu NER datasets. We open-source our dataset, pretrained models, and code[1].

1 Introduction

Named Entity Recognition (NER) aims to identify various named entities from the raw text. Typically these named entities are broadly categorized into person names, locations, organizations, and other categories depending on the domain. Identifying these named entities is necessary and is proven to be very helpful in Natural Language Processing (NLP), Information Retrieval (IR), and Information Extraction (IE). Moreover, when so much data is generated daily today, NER becomes very important in processing and extracting meaningful information from the text. However, most NER work is limited to the resource-rich English language due to the availability of annotated datasets, efficient feature representations, and tools to process the data.

English has many huge annotated datasets like CoNLL-2003 (Sang and De Meulder, 2003), OntoNotes (Weischedel et al., 2013) and WNUT (Derczynski et al., 2017). Traditional models like Conditional Random Fields (CRF) (Lafferty et al., 2001) have been used for NER modeling by training them on these datasets. With the development in deep learning, solutions like Lample et al. (2016) and Ma and Hovy (2016) used Long Short-Term Memory (LSTMs) for sequence-labelling tasks like NER. Further, the combination of the LSTM-CRF model proposed by Huang et al. (2015) has achieved even better performance. Recently, transformer models (Devlin et al., 2019) have proven to be achieving similar results to the state-of-the-art models (Akbik et al., 2018; Peters et al., 2018). Hence, we can infer that there has been extensive and rapid research in NER for English with significant advancements. However, NER developed in English cannot be generalized and extended due to the rich morphological nature of Indian languages.

Unlike English, most of the resources created for Indian languages are for machine translation. However, in the NER task, the meaning of context, the roles of named entities, differentiations amongst categories, and syntactic and semantic structures will be lost if we translate English sentences to Telugu. Examples of Telugu language NER sentences, their WX notation (a standard notation used

[1]https://github.com/mors-ner/anonymous_telner

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Student Research Workshop, pages 262 - 272
May 22-27, 2022

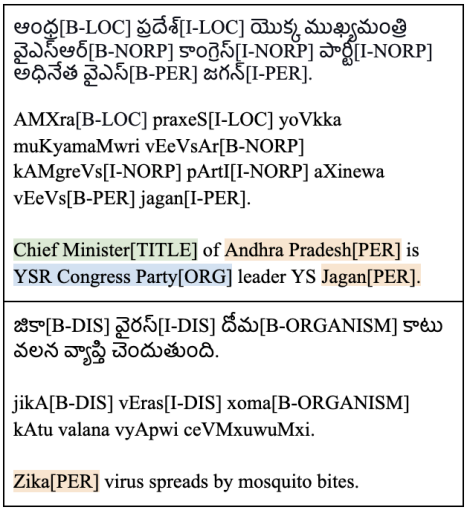


Figure 1: Example sentences of NER tags in Telugu (top), WX notation (middle) and their English translations (bottom) with NER tagging using CoreNLP (Manning et al., 2014) respectively.

for Indian languages)[2], and their English translations are reported in Figure 1. From the examples, we can notice that Telugu's context and the actual NER tags are not captured by English-translated sentences when given to the Stanford CoreNLP NER tool [3]. Therefore, we understand the need for NER to address these challenges even in morphologically rich languages like Telugu. Hence, we created an annotated dataset for NER in Telugu, which will be a good resource for those working in Telugu NLP areas such as Dialog Systems, Text Summarization, Machine Translation, and Question Answering. Furthermore, we used pretrained Telugu transformer models (Marreddy et al., 2021) and finetuned on the Telugu NER dataset to achieve NER in multiple domains.

In this paper, we aim at creating resources for NER in Telugu. Overall, we make the following contributions to this paper: (1) We publicly release two diverse annotated NER datasets, which will be pioneering resources for building automated NER systems in Telugu, (2) We build NER models using Telugu pretrained transformer models to analyze the entity-level and token-level class performance across the multi-domain datasets and (3) We achieve the state-of-the-art results on existing NER datasets.

Our extensive experiments also lead us to these crucial insights: (i) Telugu pretrained transformer models fine-tuned for the NER task outperform the existing baseline methods. (ii) It is widely known that language-specific models (*BERT-Te* and *RoBERTa-Te*) outperform the existing pretrained multilingual models (mBERT, XLM-R, and IndicBERT), this holds to be true for Telugu as well. (iii) *ELECTRA-Te* performs on par with the existing pretrained multilingual models.

[2] https://en.wikipedia.org/wiki/WX_notation

[3] https://corenlp.run/

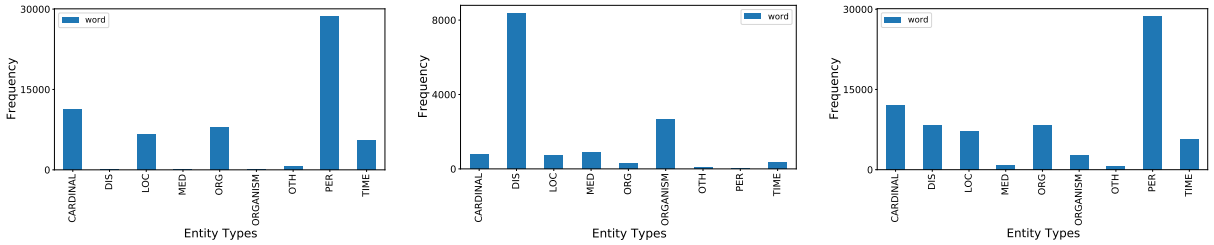
2 Related Work

Traditional Methods: The early NER experiments were studied to identify specific categories of named entities like Proper Names (Wakao et al., 1996), Organizations, and Locations (Grishman, 1995). They were based on rules, heuristics, and gazetteers. However, they could not handle out-of-gazetteer and ambiguous cases. Unlike earlier work, Lafferty et al. (2001) and Rabiner (1989) proposed CRF and HMM models to handle numerous sequence to sequence tasks such as NER and POS tagging. Nevertheless, the main limitation of these models is the computational complexity and that they cannot handle unknown words.

Later, it was found that deep learning (DL) based models like LSTM-CRF (Lample et al., 2016) and BiLSTM-CRF (Huang et al., 2015) focused on long-term dependencies and handled the feedback mechanism on sequence labeling tasks with high accuracy. However, these models compute token representation one by one (sequentially), which hinders the full exploitation of parallel computation and bidirectional context.

Transformers Based NER: In recent years, Transformers (Vaswani et al., 2017) have successfully performed various NLP tasks like Machine Translation, Language Modelling, and Semantic Role Labeling. Recently introduced Bidirectional Encoder Representations from Transformers (BERT), developed by Devlin et al. (2019), is a powerful language modeling technique to handle Masked-Language Modelling (MLM) and next-sentence prediction tasks. Furthermore, by fine-tuning the BERT model on the CoNLL dataset, a high F1 score of 92.8% was reported in Devlin et al. (2019) for NER. The success of BERT led to other variations like RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2019).

NER for Telugu: Though NER is a well-researched problem in English, very few works describe NER for Telugu. Existing NER sys-



(a) Newswire (b) Medical (c) Combined

Figure 2: Frequency of named entities across three datasets: *(a)* Newswire *(b)* Medical, and *(c)* Combined Dataset

tems mainly use small datasets and limited categories like Person, Location, and Organisation. In addition, these systems are developed based on heuristics (Sasidhar et al., 2011), traditional ML (Shishtla et al., 2008; Srikanth and Murthy, 2008) or DL (Reddy et al., 2018) methods.

To the best of our knowledge, we are the first to create such a large and diverse annotated dataset of 38,363 sentences for the NER task in Telugu. Further, we create a multi-domain dataset that incorporates both Newswire and Medical domains. Finally, we take inspiration from the transformer models and use *BERT-Te* to model NER in Telugu.

3 Annotated Dataset for NER task

Existing NER datasets are small and mainly focus on limited categories like Person (PER), Location (LOC), and Organisation (ORG). There are two significant existing datasets for NER in Telugu: (i) WikiAnn (Pan et al., 2017) (ii) LREC-NER (Reddy et al., 2018). The WikiAnn dataset has PER, LOC, and ORG entity types, with a total of $6,495$ annotated sentences. On the other hand, even though the LREC-NER dataset has $32,610$ sentences, it consists only of PER, ORG, LOC, and Miscellaneous Named Entity category (MISC).

Hence, we came up with three datasets consisting of diverse named entity categories for NER in Telugu: (i) Newswire Dataset (ND), (ii) Medical Dataset (MD), and (iii) Combined Dataset [Newswire+Medical] (CD).

The ND focuses on the general named entity categories in the news domain, while the MD focuses on data related to the biomedical domain. Ultimately, by combining ND and MD, we form the CD. Detailed statistics of the three datasets are shown in Figures 2a, 2b, and 2c. Further, details regarding the dataset have been discussed below.

Data Collection and Preprocessing: For the ND, we crawled around 50,000 sentences from Telugu360[4], GreatAndhra[5], and Eenadu[6] websites that generally publish articles related to current affairs, sports, movies, gossips, and the latest news. However, while doing so, we noticed that in the prevailing COVID-19 situation, much information on the Telugu websites focuses on health and diseases. So then, we created a separate dataset by crawling 20,000 sentences for MD. We collected this data from Boldsky[7] and Telugu-Wikipedia[8] websites. After crawling, we cleaned and preprocessed the data by removing the unwanted URLs, hashtags, hyperlinks, English text, and duplicate sentences.

Entity Types in Datasets: After analyzing the preprocessed data, we identified the following named entity categories that would best suit to describe the data:

1. **Diseases and Symptoms** (DIS): Names of diseases and symptoms comprise this category (Patil, 2020). It is a part of MD and CD. Ex: *Tuberculosis* is an airborne disease.

2. **Cardinal** (CARDINAL): The number based entities that represent quantities fall into this category (Weischedel et al., 2013). It is a part of ND, MD and CD. Ex: Mahua tree reaches *20 meters* height.

3. **Medical and Pharmacological Terms** (MED): Names of medical procedures, treatments and medicines fall under MED (Patil, 2020). It is a part of MD and CD. Ex: *Laparoscopy* is a safe procedure.

4. **Organisms** (ORGANISM): Names of all living organisms, along with their biological equivalent terms constitute ORGAN-

[4] https://www.telugu360.com
[5] https://telugu.greatandhra.com
[6] https://www.eenadu.net
[7] https://telugu.boldsky.com/health/
[8] https://te.wikipedia.org/wiki/

ISM (Patil, 2020). It is a part of MD and CD. Ex: *Coronavirus* causes COVID-19.

5. **Location** (LOC): The names of places can be classified as LOC (Sang and De Meulder, 2003). It is a part of ND, MD and CD. Ex: *India* is a beautiful country.

6. **Organization** (ORG): The names of organizations belong to this category (Sang and De Meulder, 2003). It is a part of ND, MD and CD. Ex: *Vodafone* is a telecom company.

7. **Person** (PER): The names of people fall under PER (Sang and De Meulder, 2003). It is a part of ND and CD. Ex: *Priyanka* is an actress.

8. **Date and Time** (TIME): The words used to specify particular time and other precise temporal objects can be classified into this category (Loper and Bird, 2002). It is a part ND, MD and CD. Ex: I have a party on *June 20*.

9. **Other Miscellaneous Named Entities** (OTH): Other named entities that do not fit into the above categories form OTH (Sang and De Meulder, 2003). Ex:- *Names of currencies*. It is a part of ND and CD.

Data Annotation and Statistics: Usually, named entities can be of a single word or multiple words (chunks). Hence, we used the IOB2 tagging format for annotation to capture these types of named entities. IOB2 is similar to the BIO (Ramshaw and Marcus, 1999) format. The only difference is that in IOB2, the B- tag is used at the start of all chunks.

Dataset	Sentences	Words	Named Entities	Entity Types
Newswire Data	34,109	345,202	60,491	12
Medical Data	4,254	40,352	14,260	14
Combined Data	38,363	385,554	74,751	18

Table 1: Dataset Statistics for the NER task

We provided the data to an *Elancer IT Solutions Private Limited*[9] company for NER annotation. In order to perform the annotation process, *Elancer IT Solutions Private Limited* chose five native speakers of Telugu with excellent fluency, the company itself properly remunerates all the annotators. We provided the annotators with detailed annotation guidelines and example sentences. As a first step, we gave 100 sentences to all the annotators to verify their proficiency in the annotation. The Fleiss Kappa Score (Fleiss and Cohen, 1973) for this step was 0.92, and any minor issues found were conveyed as feedback to the annotator. After this step, five qualified native Telugu speakers provided annotations for $58,712$ sentences using provided annotation guidelines. As part of the annotation, we requested annotators to provide the named entities for every sentence. However, $20,349$ sentences are removed from the final dataset due to the following reasons: (i) redundant sentences, (ii) sentences that do not have one or no named entity, and (iii) sentences with bad quality tags. Finally, there were $38,363$ annotated sentences for the dataset, out of which $4,254$ sentences belong to the MD, and $34,109$ sentences belong to the ND. Table 1 includes the detailed statistics of all datasets. The Inter-Annotator agreement for this annotation was 0.91. Finally, we performed our experiments on the ND, MD, and CD datasets.

4 Methodology

4.1 Approaches

This section presents the eight models we investigated for the NER study in more detail and their configuration.

CRF: The CRF (Lafferty et al., 2001) concept has been successfully adopted as a popular solution for sequence tagging tasks and is also a primary solution in NER. We use One-Hot Vector representations as input for the CRF model, and the output is a sequence of tags associated with each input word. The following hyperparameters were used for training the CRF model viz obtained from sklearn_crfsuite [10] library:- (i) Training Algorithm: *Gradient Descent with L-BFGS method (Liu and Nocedal, 1989)*, (ii) Coefficients of L1 and L2 regularization: *c1 = 0.1* and *c2 = 0.1*, and (iii) Maximum iterations: *1000*.

LSTM-CRF: In this model, we combined the LSTM with CRF to form an LSTM-CRF model (Huang et al., 2015). We used LSTM and other required layers from the Keras library [11], while the CRF layer from keras_contrib [12] library. For input, we compare the performance of both One-Hot vectors, which are trained from scratch, and Telugu FastText embeddings (Marreddy et al.,

[9] http://elancerits.com/

[10] https://sklearn-crfsuite.readthedocs.io/en/latest/

[11] https://keras.io

[12] https://github.com/keras-team/keras-contrib

2021) (each word dimension is 200), while the output is a sequence of tags associated with each input word.

The following hyperparameters were used to train the model:- (i) Activation function: *Sigmoid*, (ii) Recurrent Dropout: *0.5*, (iii) Loss: *Negative log-likelihood*, (iv) Number of epochs: *50*, (v) Optimizer: *RMSProp*, (vi) Batch size: *64*, (vii) Hidden units in LSTM layer: *128*, and (viii) Hidden units in Dense Layer: *128*.

BiLSTM-CRF: We combine BiLSTM with CRF to form a BiLSTM-CRF model (Huang et al., 2015). Due to the additional context that BiLSTM receives, it generally performs better than the LSTM-CRF model. We used the same setup and hyperparameters as the LSTM-CRF model.

BERT-Telugu (*BERT-Te*): Like Pretrained BERT (Devlin et al., 2019) (a pretrained model trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia), we chose a model based on the Transformer structure of BERT-base-cased for Telugu (large corpora of 8 million sentences) (Marreddy et al., 2021). The BERT-base-cased model consists of 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million parameters in total. For this study, we finetune a *BERT-Te* model on each dataset separately. In order to finetune a *BERT-Te* model, we observe that the following hyper-parameters yields best performances: (i) Batch size: *32*, (ii) Learning rate: $3e^{-5}$, (iii) Number of training epochs: *10*, (iv) ϵ constant set to $1e^{-8}$ to avoid division by zero in the AdamW calculation when the gradient approaches zero, and (iv) *AdamW* as optimizer. We stopped training to overcome the over-fitting problem if the validation loss did not decrease for five consecutive epochs.

RoBERTa-Telugu (*RoBERTa-Te*): Similar to *BERT-Te*, we chose *RoBERTa-Te*, a pretrained RoBERTa-base model for Telugu (Marreddy et al., 2021). We then finetuned this Telugu RoBERTa model on NER datasets as well. Testing on the ND, MD, and CD, we found that parameters similar to *BERT-Te* reported the best macro-F1 score.

ELECTRA-Telugu (*ELECTRA-Te*): Here, we used a pretrained model created on Telugu Corpus (Marreddy et al., 2021) called *ELECTRA-Te*, and then we made it more relevant by finetuning it on NER datasets. We use the same hyper-parameters as *BERT-Te* when finetuning the *ELECTRA-Te* model.

It is to be noted that casing has no impact in Telugu script.

4.2 Dataset Splitting

To make sure our model is time sensitive, we used the data from the most recent articles of the dataset for testing (7,672 sentences), and the older data for training (30,691 sentences). We achieve this by dividing our data into 20% and 80% ratio based on the recency. We then use the latest data (20%) for testing and the remaining data (80%) for training and validation. We calculated the average of 5-folds on the 80% of train data and reported the results on the 20% of the latest data for each model.

4.3 Evaluation Metrics

Seqeval (Entity-Level): To assess the performance of the chunking task i.e. NER, we use the *seqeval* (Nakayama, 2018) tool to measure classification metrics for sequence labeling evaluation. For measuring these classification metrics, the first step is to predict all the sequences of NER tags on the test dataset using each trained model. To understand how each class performs, we choose macro averaging that gives each class equal weight for evaluating the system's performance across the 9-classes. Here, we report the macro-average precision, recall, and F1-score to measure the per entity classification performance.

Token-Level: We measure the NER system using the most typical evaluation method to calculate precision, recall, and F1-score at a token level. The final macro-average precision, recall, and F1-score values are reported at token level between empirical and predicted tokens on the test dataset.

5 Results

This section presents the entity and token-level macro-averaged classification metrics for models trained on ND, MD, and CD in Tables 2 and 3. To further examine each class's performance, we show the performance of eight models on each dataset in section 5.1 and answer several research questions.

Entity-Level Results: We make the following observations from Table 2: (i) The CRF model, LSTM-CRF and BiLSTM-CRF models are on par in performance, where the input representations of LSTM models are One-hot and FastText (FT). (ii)

Model Type→	CRF			LSTM-CRF			LSTM-CRF-FT			BiLSTM-CRF			BiLSTM-CRF-FT			**BERT-Te**			RoBERTa-Te			ELECTRA-Te		
Dataset↓	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Newswire Dataset	0.72	0.54	0.61	0.57	0.69	0.62	0.61	0.62	0.61	0.59	0.69	0.63	0.63	0.61	0.62	**0.83**	**0.83**	**0.83**	0.80	0.79	0.79	0.78	0.78	0.78
Medical Dataset	0.71	0.46	0.54	0.64	0.52	0.56	0.51	0.52	0.51	0.60	0.54	0.56	0.55	0.47	0.51	0.71	0.74	0.72	**0.74**	**0.73**	**0.73**	0.72	0.73	0.72
Combined Dataset	0.83	0.60	0.68	0.72	0.67	0.69	0.69	0.58	0.63	0.69	0.68	0.68	0.69	0.64	0.66	**0.79**	**0.81**	**0.80**	0.78	0.78	0.77	0.76	0.77	0.76

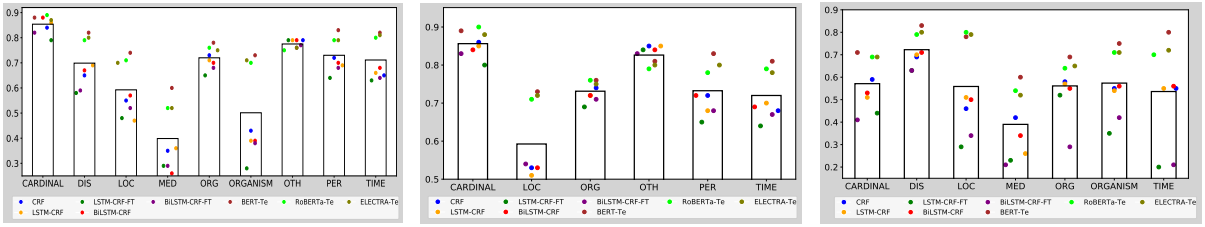
P = Precision, R = Recall, F1 = F1-score

Table 2: Telugu NER Results Entity-Level classification.

Model Type→	CRF			LSTM-CRF			LSTM-CRF-FT			BiLSTM-CRF			BiLSTM-CRF-FT			**BERT-Te**			RoBERTa-Te			ELECTRA-Te		
Dataset↓	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Newswire Dataset	0.69	0.52	0.58	0.53	0.55	0.54	0.59	0.51	0.53	0.60	0.58	0.58	0.60	0.52	0.56	**0.72**	**0.72**	**0.72**	0.69	0.72	0.70	0.71	0.70	0.70
Medical Dataset	0.67	0.53	0.52	0.49	0.45	0.44	0.59	0.40	0.48	0.44	0.40	0.42	0.63	0.35	0.49	**0.71**	**0.79**	**0.75**	0.68	0.75	0.71	0.69	0.73	0.71
Combined Dataset	0.78	0.53	0.60	0.62	0.57	0.59	0.62	0.54	0.57	0.59	0.62	0.60	0.60	0.56	0.58	**0.74**	**0.76**	**0.75**	0.72	0.72	0.72	0.73	0.72	0.72

P = Precision, R = Recall, F1 = F1-score

Table 3: Telugu NER Results: Token-Level classification.



(a) Combined Dataset (b) Newswire Dataset (c) Medical Dataset

Figure 3: Distribution of F1 scores across three datasets: (a) Combined Dataset, (b) Newswire Dataset, and (c) Medical Dataset.

Wrt to precision, recall & f1-score, finetuned Telugu pretrained transformer models such as *BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te* show an improved performance than CRF, LSTM-CRF, and BiLSTM-CRF models. (iii) Specifically, the *BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te* models yield the highest, second-highest, and third-highest recall and F1 scores for all the classes except for OTH and CARDINAL categories, as shown in Figures 3(a) and 3(b). (iv) We observe that the *BERT-Te* model is better than all the models for ND (0.83) and CD (0.80) in terms of F1-score, whereas *RoBERTa-Te* model performs the best on MD (0.73). This demonstrates that the pre-training models capture the word context better. (v) The performance of all models on MD is comparatively low compared to ND and CD. This can be explained by analyzing entity class differences across the eight training models as discussed in 5.1.

Token-Level Results: Table 3 illustrates the token-level classification performance for three NER datasets using eight trained models. We observe from Table 3 that: (i) For all three datasets, the F1-scores (0.65, 0.73, 0.75) show that the *BERT-Te* model predicts the NER tags with high accuracy at token level. (ii) Similar to entity-level results, Telugu pretrained transformer models outperform the baseline CRF and LSTM-CRF based models. (iii) Since the number of classes in token-level is 2X than entity-level classes, we observe a comparatively low F1-score at token-level than entity-level.

5.1 Do Telugu pretrained transformer models outperform the baseline models for the NER task?

Class Distribution Performance: To understand the performance of models on each class, we show the individual class performance wrt entity-level macro-average classification metrics, including precision, recall, and F1-score.

Entity-Level Class Distribution: Figures 3(a), 3(b), and 3(c) display each class performance at entity-level wrt F1-score on three datasets. We also report the F1-score of three best performing models such as *BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te* for each class at entity-level on three datasets in Figures 4, 5, and 6. Further, we showcase the recall of each class at entity-level on three datasets (refer to Figures 10, 11, and 12 in Appendix). Overall, the results indicate that the transformer-based models outperform CRF and LSTM-CRF based models in terms of recall and F1 score across the three datasets. *BERT-Te* achieves the highest recall and F1-score in 7 out of the 9 classes. However, the CRF and LSTM-CRF based models have similar performance but display relatively lower class performance in terms of recall and F1-score when compared to the finetuned Telugu pretrained models. Specifically, LSTM-CRF and BiLSTM-CRF models with FT as

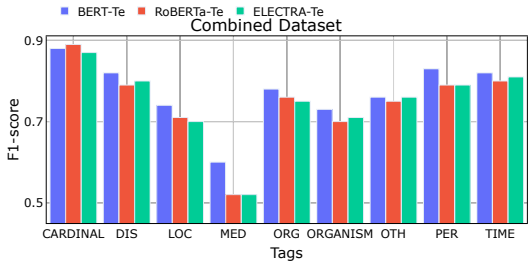


Figure 4: Distribution of F1 scores across three best-performing systems on Combined Dataset.

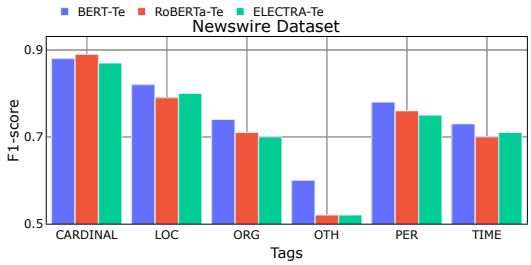


Figure 5: Distribution of F1 scores across three best-performing systems on Newswire Dataset.

input have shown a trend of lower performance in most classes.

Token-Level Class Distribution: Figure 7 shows the token-level class performance wrt F1-score across eight models on CD. Similar to entity-level, the transformer-based models *BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te* outperform the other models wrt F1-score in Figure 7. *BERT-Te* and *RoBERTa-Te* show an increasing F1-score performance for every class, while LSTM-CRF-FT and BiLSTM-CRF-FT report an overall lower F1-score across all the classes.

Model	#Sentences	#Parameters
mBERT	2.5TB	110M
XLM-R	2.5TB	125M
IndicBERT	452.8M	11M
BERT-Te	8.2M	108M
RoBERTa-Te	8.2M	125M
ELECTRA-Te	8.2M	14M

Table 4: Models and their Training Corpus size for the NER task

5.2 Do Telugu pretrained transformer models outperform the existing multilingual transformer models for the NER task?

Here, we compare the performance of three finetuned Telugu pretrained models (*BERT-Te*, *RoBERTa-Te*, and *ELECTRA-Te*) with existing multilingual transformer models (mBERT (Devlin et al., 2018), XLM-R (Conneau et al., 2020), and IndicBERT (Kakwani et al., 2020)) for the NER

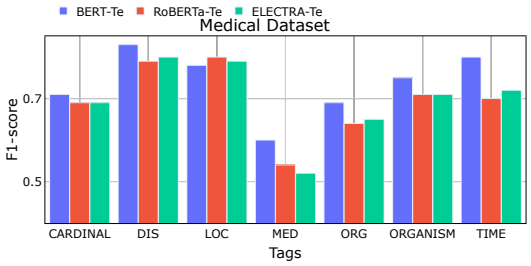


Figure 6: Distribution of F1 scores across three best-performing systems on Medical Dataset.

task. Figure 8 showcases the entity-level class performance across the three datasets. From Figure 8, we observe that *BERT-Te*, and *RoBERTa-Te* outperform mBERT, XLM-R, and IndicBERT across the three datasets. On the other hand, the *ELECTRA-Te* model has a similar performance as mBERT and XLM-R. Further, we report the pretrained model parameters of each model, as depicted in Table 4. Here, we noticed that *ELECTRA-Te* and IndicBERT models have comparatively fewer parameters than other models.

5.3 Do Telugu pretrained transformer models outperform the state-of-the-art Telugu NER systems?

In this section, we evaluate the performance of the Telugu Transformer models on the existing NER datasets: (i) WikiAnn (Pan et al., 2017) and (ii) LREC-NER (Reddy et al., 2018) and compare it with the previous state-of-the-art results. We report the various models and their performance against the datasets mentioned above in Table 5. From Table 5, we observe that *BERT-Te* and *RoBERTa-Te* deliver state-of-the-art performance on the WikiAnn dataset. Due to the simplicity of the LREC-NER dataset, all the Transformer models display 100% accurate predictions.

5.4 Quantitative Analysis

Figure 9 shows the macro F1-score of the BERT-Te model with varying training data set sizes across three datasets: CD, ND, and MD. We ran the model with three different settings - 25%, 50%, and 75% of the data for training and subsequently tested with the remaining data. As expected, the macro F1-score of the proposed model increases with the size of the training set. At 25% of the data, it is 0.74, at 50% of the data, it stands at 0.77, and finally, at 75% of the data, it stands at 0.80 for the CD. Similarly, we can observe an increasing level of performance for the ND and MD by varying the

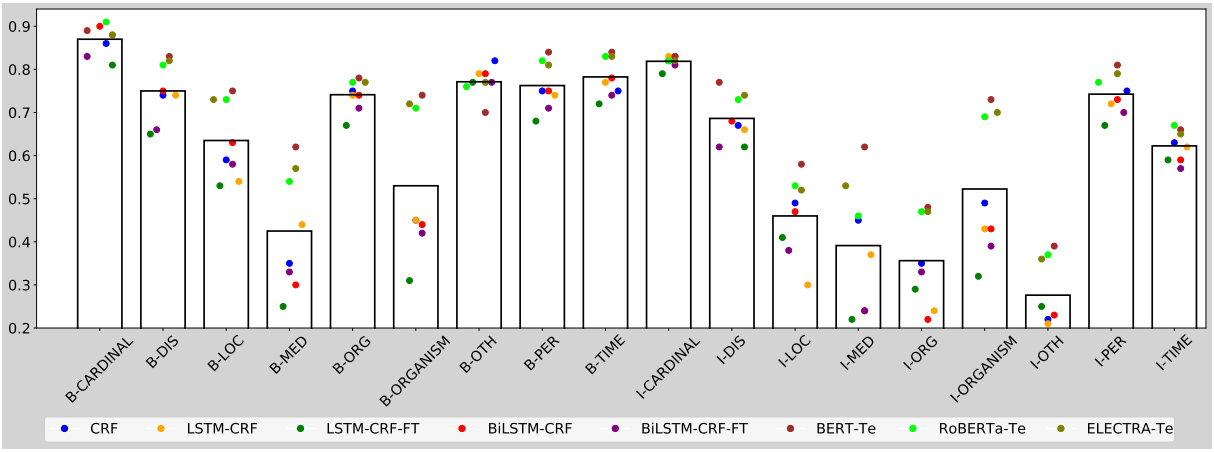


Figure 7: Combined-Dataset: Distribution of F1 scores at Token-Level.

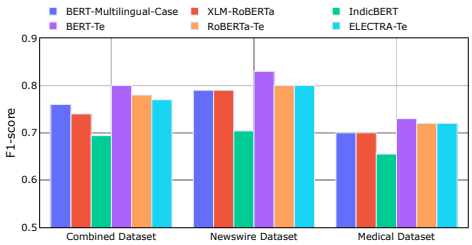


Figure 8: Entity-Level: Comparison of F1-score performance of (i) mBERT, (ii) XLM-R, (iii) IndicBERT, (iv) *BERT-Te*, (v) *RoBERTa-Te*, and (vi) *ELECTRA-Te* embeddings across three datasets: CD, ND, and MD. The *BERT-Te* fine-tuned on NER shows a higher F1-score compared to all the models.

Dataset	WikiAnn	LREC-NER
Model	F1-score	F1-score
LSTM-CRF (Reddy et al., 2018)	57.03	85.13
mBERT (Kakwani et al., 2020)	84.31	100
XLM-R (Kakwani et al., 2020)	81.71	100
IndicBERT base (Kakwani et al., 2020)	84.38	100
IndicBERT large (Kakwani et al., 2020)	80.12	100
BERT-Te	**87.03**	100
RoBERTa-Te	**87.16**	100

Table 5: Models comparison on existing Telugu NER datasets

size of the training set. However, the increase in performance is marginal as the BERT-Te model yields a similar level of performance with a smaller training dataset, possibly because the pretrained transformer captures the named entities mentioned in unstructured text into predefined categories.

5.5 Error Analysis

We analyzed the error cases in detail for three datasets using our best-performing model - *BERT-Te*. Tables 6, 7, and 8 reports the entity-level confusion matrices for the CD, ND, and MD. Table 6 shows that 2.8% of the LOC class were predicted as ORG and 1.45% as PER. Similarly, 4.5% were predicted as LOC for the ORG class, and 1.09% were predicted as PER. We can even observe a similar analysis from Table 7, where the model confused LOC, PER, and ORG tags. It is mainly because many last names derive from places in Telugu, and many Organisations are named after Person Names.

In the medical dataset, we observe from Table 8 that, for the DIS class, 1.1% were predicted as MED, and 1.7% were predicted as ORGANISM which indicates that the *BERT-Te* model gets confused with DIS, MED, and ORGANISM classes.

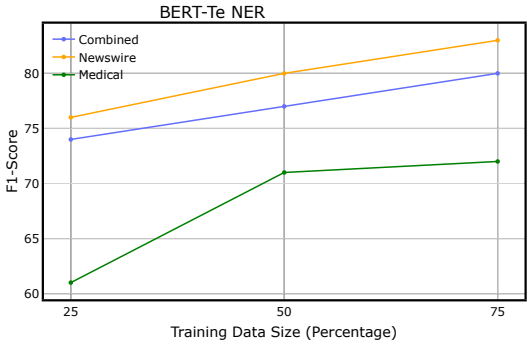


Figure 9: Entity-Level: Effect of changing the training set size on the *BERT-Te* model performance across three datasets: CD, ND, and MD.

		Predicted								
		CARDINAL	DIS	LOC	MED	ORG	ORGANISM	OTH	PER	TIME
Actual	CARDINAL	6386	14	1	0	0	0	14	20	108
	DISL	10	10153	1	110	23	121	17	5	2
	LOC	12	0	8809	8	268	44	4	138	20
	MED	0	24	6	692	14	7	19	2	6
	ORG	22	0	442	10	9077	0	0	106	3
	ORGANSIM	0	134	23	54	0	2763	12	14	0
	OTH	3	23	0	0	0	4	326	4	0
	PER	30	52	124	0	85	9	0	29895	5
	TIME	87	5	7	0	0	0	4	14	5162

Table 6: Combined: Confusion matrix for BERT-Te

6 Conclusion

This paper presented annotated datasets and an empirical study of the performance of various fine-

		Predicted					
		CARDINAL	LOC	ORG	OTH	PER	TIME
Actual	CARDINAL	5729	1	25	2	22	104
	LOC	5	7308	194	5	106	12
	ORG	34	552	8417	0	79	3
	OTH	4	11	0	302	0	0
	PER	30	133	112	11	29293	21
	TIME	72	18	0	0	17	4608

Table 7: Newswire: Confusion matrix for BERT-Te

		Predicted						
		CARDINAL	DIS	LOC	MED	ORG	ORGANISM	TIME
Actual	CARDINAL	474	4	0	0	0	9	8
	DIS	3	10333	2	127	10	185	0
	LOC	5	7	1044	11	17	39	0
	MED	8	103	16	735	7	66	0
	ORG	0	16	3	0	250	0	0
	ORGANISM	0	179	32	43	0	2995	0
	TIME	6	0	8	0	0	0	404

Table 8: Medical: Confusion matrix for BERT-Te

tuned Telugu pretrained transformer models for the NER task. We compare these results with the commonly used architectures like CRF, LSTM-CRF, and BiLSTM-CRF models in all three datasets. We even compare these pretrained Telugu models to existing multilingual models like mBERT, XLM-R, and IndicBERT. We conclude that finetuned Telugu pretrained transformer models outperform all the other models across multiple domains and they give state-of-the-art performance on existing datasets. We also notice that *ELECTRA-Te* yields significantly equal performance when compared with multilingual models even after being trained on a much smaller corpus. In the future, we would like to perform Fine-Grained NER and also expand NER to more domains for the Telugu language.

7 Ethical Statement

We created two Telugu NER datasets corresponding to two different domains (Newswire and Medical), and we open source the two datasets. The code and datasets can be downloaded from https://github.com/mors-ner/anonymous_telner.

We reused publicly available datasets (WikiAnn and LREC-NER) to compare state-of-the-art methods.

WikiAnn dataset can be downloaded from https://drive.google.com/drive/folders/1Q-xdT99SeaCghihGa7nRkcXGwRGUIsKN?usp=sharing. WikiAnn dataset is licensed under https://opendatacommons.org/licenses/by/. Please read their terms of use[13] for more details.

[13]https://elisa-ie.github.io/wikiann/

LREC-NER dataset can be downloaded from http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5. LREC-NER dataset is licensed under a Creative Commons License. Please read their terms of use[14] for more details.

Fair Compensation: We provided the data to an *Elancer IT Solutions Private Limited*[15] company for NER annotation. In order to perform the annotation process, *Elancer IT Solutions Private Limited* chose five native speakers of Telugu with excellent fluency, the company itself properly remunerates all the annotators.

Privacy Concerns: We have gone through the privacy policy of various websites mentioned in the paper. For example, the website privacy policy of www.greatandhra.com is provided here [16]. We do not foresee any harmful uses of using the data from these websites.

8 Limitations & Social Impact

Multilingual pretrained models are usually evaluated by their capacity for knowledge transfer across languages. This can be done either by training the NER model on English data only or English+Telugu NER data using (for example) mBERT representations. It allows the model to benefit from high resource languages. During the testing phase, the NER model is evaluated in Telugu only. However, this paper evaluated the NER model where training and testing on Telugu data only. In the future, it would be interesting to evaluate how the knowledge transfer from the high resource languages model performs in Telugu to assess the usefulness of the proposed datasets better.

This paper studies NER with two large, strongly annotated datasets corresponding to two different domains. Further, we compared our model to existing small labeled Telugu NER datasets. Our investigation neither introduces any social/ethical bias to the model nor amplifies any bias in the data. We do not foresee any direct social consequences or ethical issues.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling.

[14]http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5

[15]http://elancerits.com/

[16]https://www.greatandhra.com/privacy.php

In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Multilingual bert -r. *https://github.com/google-research/bert/blob/master/multilingual.md*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Ralph Grishman. 1995. The nyu system for muc-6 or where's the syntax? In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics-Volume 1*, pages 63–70.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Mounika Marreddy, Subba Reddy Oota, Lakshmi Sireesha Vakada, Venkata Charan Chinni, and Radhika Mamidi. 2021. Clickbait detection in telugu: Overcoming nlp challenges in resource-poor languages using benchmarked techniques. In *2021 International Joint Conference on Neural Networks (IJCNN)*.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Ashish Patil. 2020. Medical dataset for named entity recognition in cord 19 research challenge. `https://www.kaggle.com/finalepoch/medical-ner`.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural Language Processing using Very Large Corpora*, pages 157–176. Springer.

Aniketh Janardhan Reddy, Monica Adusumilli, Sai Kiranmai Gorla, Lalita Bhanu Murthy Neti, and Aruna Malapati. 2018. Named entity recognition for telugu using lstm-crf. In *WILDRE4–4th Workshop on Indian Language Data: Resources and Evaluation*, page 6.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

B Sasidhar, PM Yohan, A Vinaya Babu, and A Govardhan. 2011. Named entity recognition in telugu language using language dependent features and rule based approach. *International Journal of Computer Applications*, 22(8):30–34.

Praneeth M Shishtla, Karthik Gali, Prasad Pingali, and Vasudeva Varma. 2008. Experiments in telugu ner: A conditional random field approach. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

P Srikanth and Kavi Narayana Murthy. 2008. Named entity recognition for telugu. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Takahiro Wakao, Robert Gaizauskas, and Yorick Wilks. 1996. Evaluation of an algorithm for the recognition and classification of proper names. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

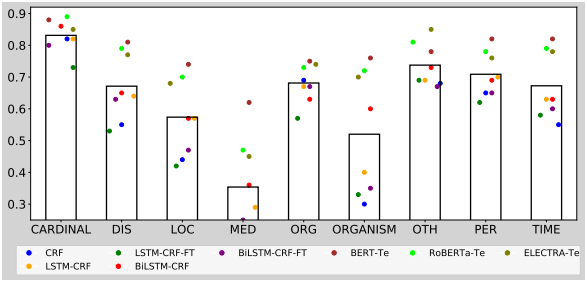


Figure 10: Combined Dataset: Distribution of Recall

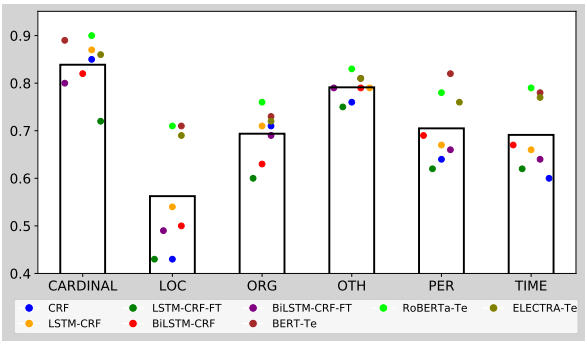


Figure 11: Newswire Dataset: Distribution of Recall

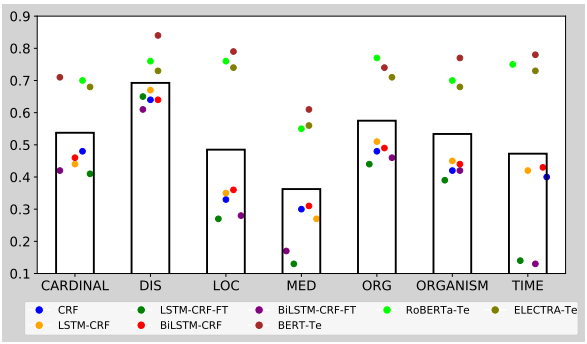


Figure 12: Medical Dataset: Distribution of Recall

A Entity-Level Class Distribution Performance