IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 January 2024, accepted 29 January 2024, date of publication 5 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362645

RESEARCH ARTICLE

ACRF: Aggregated Conditional Random Field for Out of Vocab (OOV) Token Representation for Hindi NER

SUMIT SINGH, (Graduate Student Member, IEEE), AND UMA SHANKER TIWARY, (Senior Member, IEEE)
Indian Institute of Information Technology Allahabad, Allahabad 211012, India
Corresponding author: Sumit Singh (pse2017004@iiita.ac.in)

ABSTRACT Named entities are random, like emerging entities and complex entities. Most of the large language model's tokenizers have fixed vocab; hence, they tokenize out-of-vocab (OOV) words into multiple sub-words during tokenization. During fine-tuning for any downstream task, these sub-words (tokens) make the named entity classification more complex since, for each sub-word, an extra entity type is assigned for utilizing the word embedding of the sub-word. This work attempts to reduce this complexity by aggregating token embeddings of each word. In this work, we have applied Aggregated-CRF (ACRF), where a conditional random field (CRF) is applied at the top of aggregated token embeddings for named entity prediction. Aggregation is done at embeddings of all tokens generated by a tokenizer corresponding to a word. The experiment was done with two Hindi datasets (HiNER and Hindi Multiconer2). This work showed that the ACRF is better than vanilla CRF (where token embeddings are not aggregated). Also, our result outperformed the existing best result at HiNER data, which was done by applying a cross-entropy classification layer. Further, An analysis of the impact of tokenization has been conducted, both generally and according to entity types for each word present in test data, and the results show that ACRF performed better for the words which tokenized in more than one sub-words (OOV) compared to vanilla CRF. In addition, this work conducts a comparative analysis between two transformer-based models, MuRIL-large and XLM-roberta-large and investigates how these models adopt aggregation strategy based on OOV.

INDEX TERMS CRF, LLM, NER, NLP, transformer.

I. INTRODUCTION

Named Entity Recognition (NER) is an essential lower-level task [1] in Natural Language Processing (NLP), used to extract and categorize named entities from structured and unstructured text into a predefined set of classes such as person, location, organization, numeral and temporal entities. An example of the named entity recognition task is illustrated in Fig.1, and the corresponding Hindi example is illustrated in Fig.2.

Text summarization [2], [3], Web Scraping [4] question-and-answer applications [5], and machine translation [6], [7] are just a few of the uses for named entity recognition.

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

Recent NER models are based on deep learning like LSTM-CRF [8] and other transformer-based language models [9]. These models used Conditional Random Field (CRF) [10] and Cross-Entropy (CE) [11] for decoding tags [12]. These models required data for training for the NER task. NER data consists of annotated entity types for each word present in an example. According to our requirement, these entity types can be from any predefined set, and non-entities are annotated as other (''O''). A broader range of entity-type sets is considered a good data distribution for training; also, a balanced data distribution with a large number of examples in training data is considered a good dataset. However, the Hindi language is taken as a low-resource language since data availability of the Hindi language is much less than the available data of rich resource languages like English.

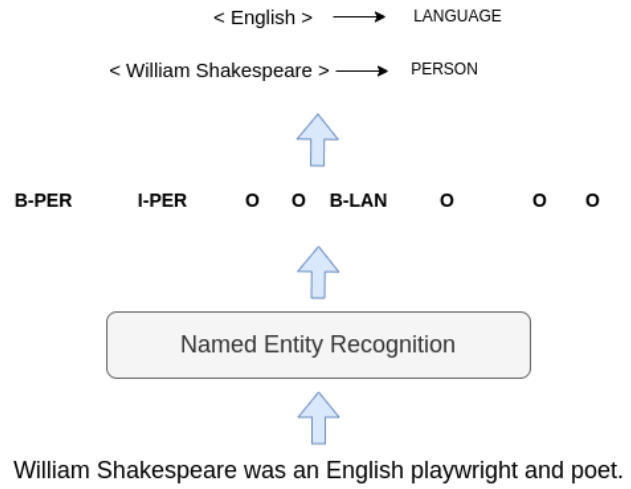

FIGURE 1. An illustration of the named entity recognition task.

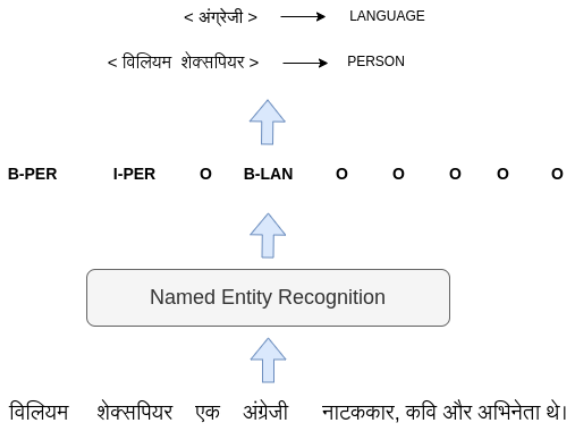

FIGURE 2. An illustration of the named entity recognition task (Hindi text).

Recently publically available NER datasets for the Hindi language are HiNER [13] (with 76025 training examples and 11 entity-type), MultiCoNER1 [14] (with 15300 training examples and six entity-type) and Multiconer2 [15] (with 9,632 training examples, six course-grained entity-type and 36 fine-grained entity-type). This work experiments with feeding word embeddings to the CRF layer (tag-decoder) with and without aggregated word embeddings for the NER task at HiNER [13] and Multiconer2 [15] datasets.

HiNER [13] established a benchmark of weighted F1-score of 88.78, and their best result was 89.2 weighted F1-score, which utilized xlm-roberta-large (XLM-R) for feature extraction. More detail about the data is given in the section III. However, the best score achieved by the Hindi Multiconer2 dataset with cross-entropy classification layer (without the use of any external knowledge-base) is 77.62 weighted f1-score with the MuRIL language model and 43.55 weighted f1-score with the XLM-R language model [16].

The motivation behind our approach emerged due to two reasons. The first one is that all transformer models use

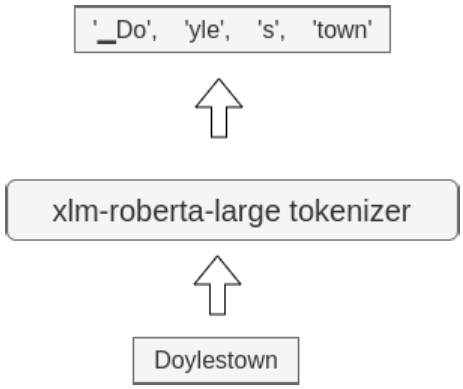

FIGURE 3. An example of the XLM-R tokenizer for OOV word.

sub-word tokenizer.[1] An example of a subword tokenizer for XLM-R model has been shown in Fig. 3. Doylestown is a village (Location); however, it is not part of the vocab of the XLM-R tokenizer; hence, the tokenizer splits it into subword tokens.

Second, there are numerous emerging named entities [17] or rarely occurring entities, and they are not part of the vocab of the corresponding model tokenizer. Both reasons affect the NER task complexity by assigning an extra label to a subword token (if a word is tokenized into multiple tokens). These extra tokens can be bigger based on the tokenizer and entity form.

This work studied at HiNER and Multiconer2 dataset, and found statistics of these extra Entity types (tags). These statistics are shown in Table 2 for HiNER dataset and in Table 3 for Multiconer2 dataset. It is apparently shown in Table 2 and 3 that XLM-R tokenizer has more OOV than MuRIL tokenizer.

Our methodology is based on the architecture of NER models as it does not require a Knowledge-Base (KB) or gazetteers [17], [18]. Knowledge-based methods require domain-specific prior knowledge base [19].

In this work, to handle out-of-vocabulary (OOV) words in the appropriate way, Aggregated CRF (ACRF) is applied where a CRF layer (Tag decoder) at the top of aggregated token embeddings for each word is applied for named entity recognition.

We provided entity-wise and tokenization-length[2]-wise F1-score, for example, score generated by entities whose tokenization-length is one, two, three. It is clear from the definition of tokenization-length that in-vocab entities have tokenization-length of one, and OOV entities have tokenization-length of more than one.

Following are some key points that outline our contribution.

Our first contribution is to produce state-of-the-art results on the HiNER dataset by applying Aggregated-CRF. When using our architecture (Aggregated-CRF), XLM-R performs

[1]Subword tokenizer split a word into multiple tokens (subwords) if the word is not present on tokenizer vocab (OOV).

[2]This is the number of tokens generated by tokenizer for a word.

better than XLM-R (benchmarks) by 0.77% F1-score while MuRIL performs better than MuRIL (benchmarks) by 1.71% F1-score. The other contribution of this work is an analysis of the effect of aggregation with CRF tag decoder based on transformer-based pretrained model's tokenizers at two recent Hindi datasets for the NER task.

In the next section, related work is defined. Section III describes the HiNER and Multiconer2 datasets. Our methodology is defined in Section IV. Experimental setup and evaluation metrics are described in Section V and VI. Thereafter, results and analysis are explained in Section VII, and Section VIII concludes this paper.

II. RELATED WORK

Recent advancements in Deep learning-based approaches for NLP showed the strong capability of solving downstream tasks like NER [9]. These models automatically extract the hidden features, due to which the accuracies of these models are high compared to the traditional NER approaches.

While most state-of-the-art is based on transformer-based self-supervised pretrained models [20] and pretraining requires a large corpus, rich resource languages have seen significant research due to the availability of large corpora. However, for the Hindi language, there is a lack of structured resources in this domain [21], such as large corpora, web content, or data for the downstream task, necessitating the use of more robust methods. If we uncover the NER task independent of data availability, it can be divided into two subtasks. The First is to encode the sequences into knowledge space, also known as a feature extractor. The second one is tag decoders, which decode sequence knowledge into Entity types (labels). For encoding, there are two types of models. The first one is those encoders initialized with random weights for training like LSTM [22], GRU [23], CNN [24] and its variants, and the second one is pretrained large language models (LLMs). For NLP tasks most LLMs are transformer-based, like Bert [8], mbert [8], xlm-roberta (XLM-R) [25], MuRIL [26], indicbert [27]. LLMs are pre-trained on large data in self-supervised ways with millions of parameters, and these models are domain and language-specific. Transformer-based pretrained models generate contextual word-embeddings since the representation of a word-embedding of a word in a context (sentence) is dependent on the context. Some sequence tagging-specific LLM architectures like LUKE [19] are based on entity-aware pretraining and using Wikipedia entities to decode the extracted features with encoders. Indic-bert, MuRIL, mbert and XLM-R are pretrained for Hindi languages, and among these models, the performance of XLM-R and MuRIL in [26] and [28] motivated us to choose these language models for this work. Also, for the Hindi Multiconer2 [15] dataset, MuRIL performed best when any external KB is not used [16].

Earlier tag decoders are linear statistical models, which include Hidden Markov Models (HMM), Maximum entropy Markov models (MEMMs) [29], and Conditional Random Fields (CRF) [10]. CRF achieved popularity for the NER task and achieved state-of-the-art results for the NER task [8], [9], [30]. Convolutional network-based model [31] consists of a convolutional network and a CRF layer on the top of the convolutional network for sequence tagging. References [8], [9], and [30] utilize CRF at the top of bidirectional and provide promising results with LSTM. The zero-shot generalisation of large language models (LLMs) [32] has also revolutionised natural language processing (NLP). An analysis of zero-shot NER using ChatGPT was conducted in [33] and [34], and so far, the findings were not as good as the Benchmark results using the transformer encoder-based models.

For Hindi data like HiNER, the best score was achieved by finetuning XLM-R, and the second-best score was achieved by finetuning MuRIL-base. These models have used cross entropy as a classification of tags at the top of base models. For Multiconer1 and Multiconer2 datasets, MuRIL, without a knowledge base, scored best [14], [16], [28]. However, XLM-R with augmented retrieval got the best score [14], [35].

In this work, an experiment with aggregation over the word embeddings corresponding to each token of a word is done with a CRF tag encoder.

III. DATASET

Our work done with two state-of-the-art datasets HiNER [13] and Multiconer2 [15].

HiNER [13] data contain 108,608 annotated examples, divided into training, validation and testing data by a ratio of 70, 10, and 20. This data contains 11 tags, and their statistics are defined in Table 1. HiNER follows conll format with I-O-B encoding. OOV-based statistics for each tag of the Hiner test data are tabulated in Table 2.

TABLE 1. Entity distribution for HiNER dataset.

Tag	Training	Validation	Testing	Total
PERSON	26310	3771	7524	37605
LOCATION	137995	20100	40187	198282
NUMEX	17194	2555	4662	24411
ORGANIZATION	18508	2645	5356	26509
MISC	4070	553	1080	5703
LANGUAGE	4187	571	1190	5948
GAME	1214	180	369	1763
TIMEX	13047	1762	3653	18462
RELIGION	823	133	234	1190
LITERATURE	597	74	181	852
FESTIVAL	203	30	40	273
Total	224148	32374	64476	320998

Multiconer2 [15] dataset has 9632 training, 514 validation and 18399 testing examples. A total of 33 tags are annotated for the sentences in I-O-B format. OOV-based Entity distribution for the Multiconer2 test data is tabulated in Table 3.

IV. METHODOLOGY

Overall Methodology shown in Fig. 4. The details of our Methodology can be divided into the subsequent subsections.

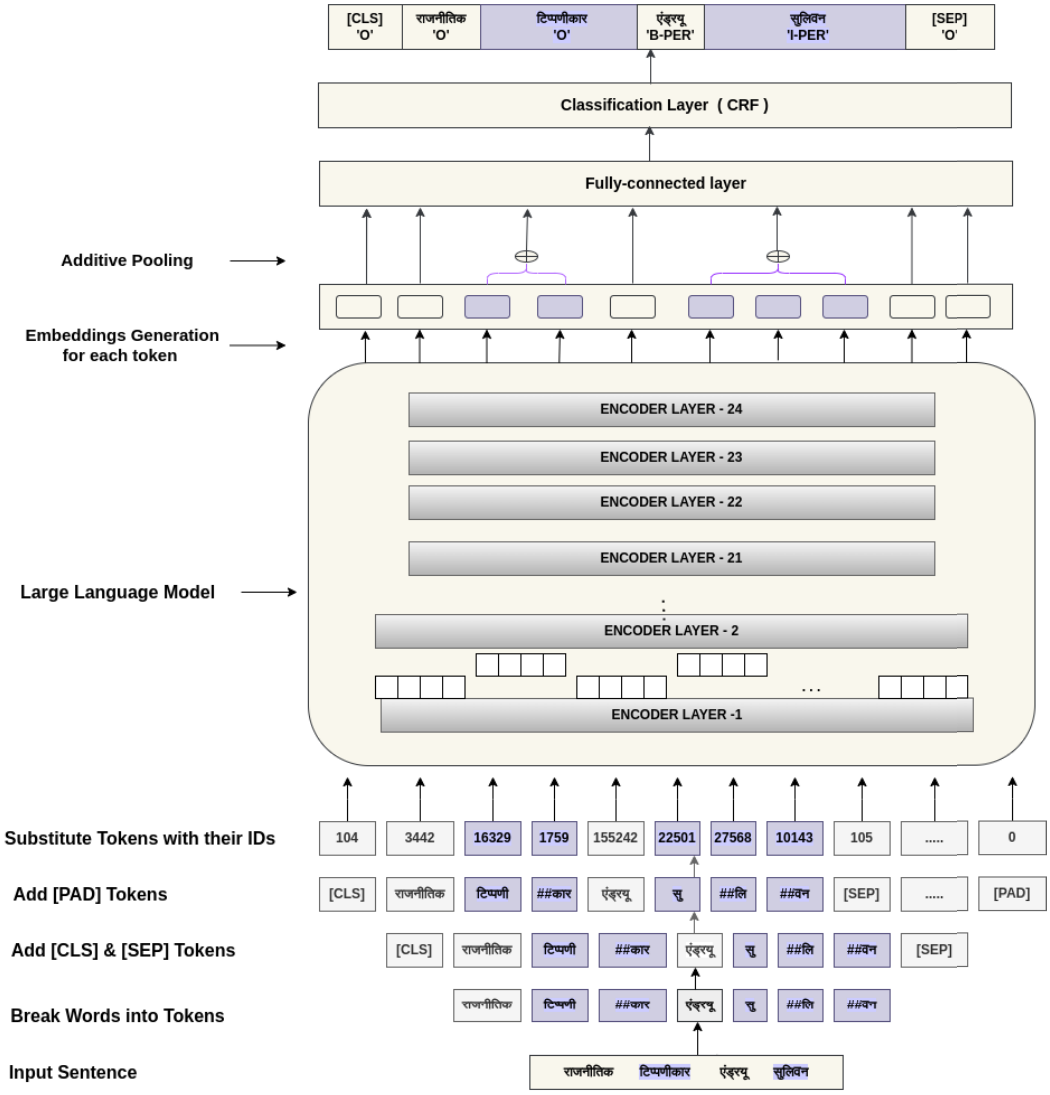

FIGURE 4. Generalized transformer-based NER model architecture.

A. TRANSFORMER-BASED ENCODER SELECTION

Our work has taken advantage of pretrained models based on transformer encoders for encoding the input text into deep feature vectors. The architecture of the transformer is based on a multi-head self-attention mechanism, and selected encoders are pretrained in a self-supervised way with a large amount of data. The chosen models of this work were pretrained with a large amount of text data, including Hindi, and performed state-of-the-art results for various Hindi NLP tasks. Our work experiments with MuRIL and XLM-R models for finetuning the task, motivated by [13], [16], and [28]. The selection of encoders is independent of the overall architecture.

B. TOKENIZATION

After selecting encoders, we processed the input sentence with padding to LLM max length and generated an attention mask for each sentence. Labels are assigned an integer value starting from zero. Thereafter, we tokenize each sentence of a dataset with the selected encoder tokenizer. Tokenizer also assigns an index to each token. Along with tokenization, Word IDs list has also been created. Indices of tokens corresponding to each word can be found using Word IDs, which can be created during tokenization; it is the list in which indices correspond to each token stored indices of parent words in a sentence. Word IDs are beneficial during the pooling of word features after feature extraction with the encoder. For a given sentence S containing m words, the sub-word tokenizer splits it into tokens T.

$$T < t_1, t_2, \ldots t_n > = \text{tokenizer}(S < w_1, w_2, \ldots w_m >) \quad (1)$$

C. FEATURE EXTRACTION

Tokenized sentence T is fed into transformer-based encoders (XLM-R Large and MuRIL in our work). The encoder generates contextual features in the form of word embeddings

TABLE 2. Entity-wise OOV information with XLM-R and MuRIL tokenizers for HiNER test Data.

Tag / Model Tokenizer	Total words	XLM-R Tokenizer		MuRIL Tokenizer	
		In vocab	OOV	In vocab	OOV
B-FESTIVAL	39	7	32	23	16
B-GAME	369	227	142	301	68
B-LANGUAGE	1190	839	351	1091	99
B-LITERATURE	179	58	121	113	66
B-LOCATION	40053	12749	27304	29527	10526
B-MISC	1065	486	579	863	202
B-NUMEX	4637	3831	806	4286	351
B-ORGANIZATION	5348	2688	2660	3826	1522
B-PERSON	7490	2170	5320	4700	2790
B-RELIGION	230	118	112	224	6
B-TIMEX	3644	3025	619	3408	236
I-FESTIVAL	26	12	14	22	4
I-GAME	231	148	83	217	14
I-LANGUAGE	603	592	11	598	5
I-LITERATURE	171	100	71	135	36
I-LOCATION	4728	2533	2195	3520	1208
I-MISC	72	34	38	66	6
I-NUMEX	1816	1163	653	1614	202
I-ORGANIZATION	3847	2435	1412	3232	615
I-PERSON	5486	1992	3494	3675	1811
I-RELIGION	20	16	4	17	3
I-TIMEX	1912	1482	430	1767	145

for each token t_i.

$$\text{embed} < e_1, e_2, e_3 \ldots e_n > = \text{encoder}(T) \tag{2}$$

D. FEATURES AGGREGATION

The tokens corresponding to i_{th} word of sentences are added together to form the word representation WR_i. Given a sentence sub-word (token) representation[3] T, the final contextual word representation[4] WR_i express as

$$\text{WR}_\text{i} = \sum_{j=\text{START}_i}^{\text{END}_i} e_j \tag{3}$$

where START_i and END_i are the start and end indices of the sub-words constituting word w_i. We have used additive pooling for aggregation.

E. TAG DECODER

A conditional random field is globally conditioned on the mentioned sequence. The correlation between neighbouring tags can be taken into account using CRF as a probability model for sequence prediction to provide the overall tag sequence. Also, in feature-based supervised learning techniques, CRFs have been employed extensively. A CRF layer is frequently used as the tag decoder in NER models based on deep learning; for example, CRF used at the top of the bidirectional LSTM layer [18], CNN layer [17] and XLM-R [35]. Fig. 5 shows the structures of CRF.

Aggregated features from the previous stage were fed into a linear layer, which transforms each word representation into logits with size (1, total number of labels in our data).

$$\text{logits} = \text{linear\_layer}(\text{WR}) \tag{4}$$

Thereafter, a CRF layer was applied at logits to compute the likelihood of the tag sequence and loss for the input sentence.

$$\text{prob} = \text{CRF}(\text{logits}) \tag{5}$$

Here, the prob vector is the best likelihood of a sequence of tags for the given sequence of words. In a sequence tagging task like NER, the neighbour tags can help the model learn current tag information in a given sequence of tags. For example, given in Fig. 1, the second tag I-PER confirms that the previous tag is either B-PER or I-PER according to BIO format. To learn this global information at a sequence level, we need to learn the tag result according to the score of the whole tag sequence.

[3]Tokenized sentence.

[4]Word embedding generated by the encoder for each word.

TABLE 3. Entity-wise OOV information with XLM-R and MuRIL tokenizers for Multiconer2 test Data.

Tag /Model Tokenizer	Total Words	XLM-R tokenizer		MuRIL tokenizer	
		In vocab	OOV	In vocab	OOV
B-AerospaceManufacturer	85	2	83	10	75
B-AnatomicalStructure	490	62	428	193	297
B-ArtWork	426	148	278	242	184
B-Artist	1852	253	1599	988	864
B-Athlete	1174	135	1039	584	590
B-CarManufacturer	146	17	129	76	70
B-Cleric	189	28	161	128	61
B-Clothing	77	3	74	24	53
B-Disease	634	26	608	120	514
B-Drink	136	24	112	81	55
B-Facility	844	147	697	400	444
B-Food	428	52	376	220	208
B-HumanSettlement	5826	909	4917	3044	2782
B-MedicalProcedure	337	90	247	176	161
B-Medication/Vaccine	377	37	340	91	286
B-MusicalGRP	174	37	137	90	84
B-MusicalWork	44	1	43	22	22
B-ORG	1847	838	1009	1279	568
B-OtherLOC	231	44	187	127	104
B-OtherPER	741	109	632	428	313
B-OtherPROD	780	138	642	333	447
B-Politician	1155	215	940	672	483
B-PrivateCorp	84	59	25	69	15
B-PublicCorp	418	75	343	167	251
B-Scientist	132	19	113	70	62
B-Software	703	188	515	470	233
B-SportsGRP	1141	93	1048	605	536
B-SportsManager	493	45	448	265	228
B-Station	256	82	174	122	134
B-Symptom	140	19	121	52	88
B-Vehicle	190	19	171	40	150
B-VisualWork	753	277	476	457	296
B-WrittenWork	878	316	562	566	312
I-AerospaceManufacturer	94	21	73	41	53
I-AnatomicalStructure	109	11	98	86	23
I-ArtWork	1020	449	571	641	379
I-Artist	1931	323	1608	802	1129
I-Athlete	1264	86	1178	347	917
I-CarManufacturer	69	14	55	54	15
I-Cleric	200	54	146	130	70
I-Clothing	7	4	3	6	1
I-Disease	304	110	194	159	145
I-Drink	63	39	24	61	2
I-Facility	735	287	448	569	166
I-Food	151	70	81	109	42
I-HumanSettlement	1712	762	950	1332	380
I-MedicalProcedure	224	99	125	149	75
I-Medication/Vaccine	139	25	114	78	61
I-MusicalGRP	144	19	125	56	88
I-MusicalWork	29	11	18	23	6
I-ORG	3048	1573	1475	2559	489
I-OtherLOC	259	165	94	213	46
I-OtherPER	887	205	682	493	394
I-OtherPROD	436	193	243	343	93
I-Politician	1420	319	1101	719	701
I-PrivateCorp	81	28	53	55	26
I-PublicCorp	252	89	163	159	93
I-Scientist	175	22	153	62	113
I-Software	360	116	244	208	152
I-SportsGRP	1882	622	1260	961	921
I-SportsManager	659	36	623	191	468
I-Station	533	236	297	476	57
I-Symptom	29	4	25	6	23
I-Vehicle	100	16	84	71	29
I-VisualWork	1378	511	867	778	600
I-WrittenWork	1122	540	582	823	299

V. EXPERIMENTAL SETUP

In the HiNER dataset, the average tokenized length[5] of all words is 1.5 for XLM-R and 1.2 for MuRIL. In Multiconer2

[5]Tokenized length of a word is the number of tokens generated after tokenization of that word.

TABLE 4. Hyper-parameters for MuRIL and XLM-R setups for Hiner dataset.

Hyper parameters	XLM-R (CRF)	MuRIL (CRF)	XLM-R (Aggregated-CRF)	MuRIL (Aggregated-CRF)
Baseline language model	XLM-Roberta-large	google/MuRIL-large-cased	XLM-Roberta-large	google/MuRIL-large-cased
Classification	cross-entropy	cross-entropy	cross-entropy	cross-entropy
Learning rate for language models	2e-05	2e-05	4e-05	2e-05
Batch size	64	64	64	64
Learning rate for the classification layer	2e-05	2e-05	4e-05	2e-05
training epochs	10	10	10	10
Model_Max_Len for tokenizer	92	92	92	92
Optimizer	AdamW	AdamW	AdamW	AdamW
Dropout rate	0.1	0.1	0.1	0.1

TABLE 5. Hyper-parameters for MuRIL and XLM-R setups for Multiconer2 dataset.

Hyper parameters	XLM-R (CRF)	MuRIL (CRF)	XLM-R (Aggregated-CRF)	MuRIL (Aggregated-CRF)
Baseline language model	XLM-Roberta-large	google/MuRIL-large-cased	XLM-Roberta-large	google/MuRIL-large-cased
Classification	cross-entropy	cross-entropy	cross-entropy	cross-entropy
Learning rate for language models	1e-05	5e-05	1e-05	5e-06
Batch size	64	64	64	64
Learning rate for the classification layer	5e-03	8e-03	5e-3	2e-03
training epochs	20	20	20	20
Model_Max_Len for tokenizer	92	92	92	92
Optimizer	AdamW	AdamW	AdamW	AdamW
Dropout rate	0.1	0.1	0.1	0.1
Seed	7	7	7	3

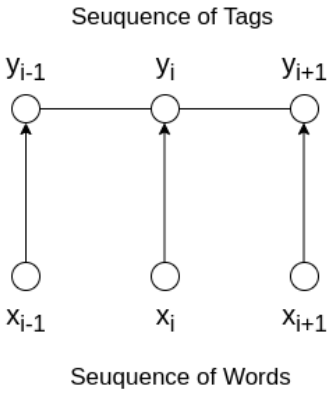

FIGURE 5. Structure of CRF.

dataset, the average tokenized length of all words is 2.54 for XLM-R and 1.78 for MuRIL.

Therefore, word embeddings of three subwords (tokens) corresponding to each word are utilized for aggregation. If a word is tokenized into one token, then it is utilised in its original form since, here, aggregation is not required. For aggregation, the summation of each subword embeddings is applied to correspond to each word of the examples. For training, experiments were done with batch sizes of 16, 32, and 64, with the learning rate of 5e-6, 5e-5, 1e-5, 2e-5, 4e-5, 8e-5 and with different random seeds. AdamW [36] optimizer optimizes the weights during backpropagation. Tokenizer max length set to 92 for better utilization of GPU as it does not affect results since only a few examples had their length over 92. CRF is implemented using Pytorch-CRF.[6] All instances were fine-tuned for 20 epochs with different combinations of hyperparameters. In each case, the models that yield the lowest validation loss throughout an epoch are the ones that are chosen as the best. Details of hyperparameters and experimental setup for the best model of each architecture are tabulated in Table 4 for HiNER dataset and in Table 5 for Multiconer2 dataset.

VI. EVALUATION METRICS

For evaluation, strict F1[7] score applied. **Strict match** is used to evaluate sequence tagging when examples contain both entity type and boundary to an entity, and we have to predict both entity type and boundary correctly. For illustration, Let a sentence is given, and we need to find the entity present in the sentence in I-O-B format.

Sentence: ''Taj Mahal situated in Uttar Pradesh.''

Here, ''Uttar Pradesh'' is a Location entity. Therefore, according to the I-O-B format, the NER model should predict: ''Uttar'' as B-LOC and ''Pradesh'' as I-LOC.

Here, Prefix B (Begain) and I (Inside) depict the entity's boundaries, and LOC depicts the Type of the entity. So, during the matching of the ground truth and prediction, if we consider both boundary and type, it is called as strict match, and the corresponding F1-score will be a strict F1-score. However, for matching other combinations of ground truth and prediction, some other metrics are defined, like partial match. Strict match metric is used in the HiNER paper; therefore, we have mentioned it explicitly in this work. It is also the default metric for NER evaluation.

It can be computed using the following formula:

Strict F1-Score $= 2 \times$ (Strict Precision $\times$ Strict Recall) / (Strict Precision + Strict Recall)

After computing the strict F1-score of each class, we have also computed the average of the strict F1-score of all classes.

VII. RESULT AND ANALYSIS

Next subsections explain the results for both the datasets HiNER and Multiconer2. All the F1-scores are strict F1-scores.

[6]https://pytorch-crf.readthedocs.io/en/stable/

[7]https://github.com/MantisAI/nervaluate

TABLE 6. Strict F1-score (%) for XLM-R (benchmark), MuRIL (benchmark), XLM-R_{CRF}, MuRIL$_{CRF}$, XLM-$R_{Aggregated\ CRF}$, and MuRIL$_{Aggregated\ CRF}$ for each tag along with overall weighted, macro and micro average of all tags (For HiNER Dataset).

Labels/Models	XLM-R (CE) (benchmark)	MuRIL (CE) (benchmark)	XLM-R (CRF)	MuRIL (CRF)	XLM-R (Aggregated CRF)	MuRIL (Aggregated CRF)
FESTIVAL	46.73	0	36.11	39.64	61.15	26.37
GAME	59.63	40.88	30.2	58.94	65.63	68.17
LANGUAGE	91.42	90.08	56.51	91.61	84.45	93.37
LITERATURE	56.69	44.23	16.81	53.76	70.71	67.03
LOCATION	94.86	94.81	66.39	94.52	95.05	95.5
MISC	67.86	62.84	44.49	66.98	71.93	69.17
NUMEX	69.1	68.31	51.12	69.22	70.84	70.43
ORGANIZATION	78.76	78.26	52.11	77	81.52	80.86
PERSON	85.14	84.6	54.43	85.48	85.97	86.49
RELIGION	72.27	53.43	40.58	80.61	73.75	83.98
TIMEX	80.63	81.17	53.36	81.37	83.56	82.53
Weighted	88.78	88.27	60.96	88.52	89.55	89.88
Macro	73.01	63.51	45.65	72.65	76.78	74.9
Micro	88.73	88.27	55.97	88.43	89.44	89.73

TABLE 7. Tokenized length-wise F1-score (%) for XLM-R_{CRF}, MuRIL$_{CRF}$, XLM-$R_{Aggregated\ CRF}$, and MuRIL$_{Aggregated\ CRF}$ (HiNER dataset).

Architecture	MuRIL (CRF)			XLM-R (CRF)			MuRIL (Aggregation-CRF)			XLM-R (Aggregation-CRF)		
Word Tokenized Length	1	2	3	1	2	3	1	2	3	1	2	3
B-FESTIVAL	0.5652	0.6	0.5714	0.25	0	0.5	0.2708	0.6316	0.5	0.375	0.4444	0.7407
B-GAME	0.7393	0.6857	0.8421	0.325	0.2128	0.1455	0.7635	0.6	0.8462	0.7403	0.806	0.8222
B-LANGUAGE	0.9549	0.8919	0.56	0.7965	0.5818	0.8184	0.9605	0.9367	0.7647	0.9751	0.9091	0.9409
B-LITERATURE	0.7426	0.7273	0.75	0.5977	0.7156	0.7547	0.8145	0.7324	0.72	0.8	0.8657	0.8136
B-LOCATION	0.9667	0.9433	0.9549	0.6369	0.7579	0.8465	0.9777	0.9551	0.962	0.9719	0.9575	0.979
B-MISC	0.7252	0.6723	0.6918	0.4382	0.6122	0.5217	0.7287	0.6916	0.7347	0.7923	0.7396	0.664
B-NUMEX	0.8004	0.8758	0.9096	0.4776	0.6431	0.416	0.8005	0.8746	0.9091	0.8104	0.8642	0.7088
B-ORGANIZATION	0.8494	0.8122	0.8877	0.5517	0.5726	0.6473	0.8632	0.8383	0.8882	0.8625	0.8089	0.8676
B-PERSON	0.9221	0.8886	0.8643	0.6005	0.6988	0.7149	0.9256	0.902	0.8693	0.92	0.9085	0.9011
B-RELIGION	0.8174	0.6	0.8	0.6243	0.3871	0.3636	0.808	0.4444	0.8	0.8538	0.7474	0.5641
B-TIMEX	0.8686	0.7629	0.8148	0.6263	0.6112	0.4651	0.8654	0.7059	0.7903	0.8799	0.8589	0.7815
I-FESTIVAL	0.4651	0.8	0	0	0	0	0.5926	0.5714	0	0.4286	0.2	0.4444
I-GAME	0.6923	0.2759	0	0	0	0	0.749	0.2963	0.2857	0.6991	0.7544	0.3529
I-LANGUAGE	0.9601	0.5714	0	0	0	0	0.9617	0.6667	0	0.9679	0.5455	0.6667
I-LITERATURE	0.7336	0.7391	0.5	0	0	0	0.7273	0.7692	0.3077	0.8122	0.7907	0.8485
I-LOCATION	0.8562	0.8141	0.8792	0	0	0	0.8827	0.8824	0.8792	0.8988	0.8686	0.8231
I-MISC	0.4779	0.2222	0	0	0	0	0.4037	0.5	0	0.6462	0.6415	0.3333
I-NUMEX	0.7751	0.8119	0.6939	0	0	0	0.7733	0.8379	0.6939	0.7872	0.7194	0.8639
I-ORGANIZATION	0.8268	0.7911	0.815	0	0	0	0.8456	0.7698	0.8393	0.8484	0.7903	0.8251
I-PERSON	0.9384	0.9087	0.9232	0	0	0	0.909	0.8973	0.9092	0.9424	0.9204	0.9253
I-RELIGION	0.6818	0	0	0	0	0	0.6977	0.8	0	0.0206	0	0
I-TIMEX	0.8519	0.8039	0.7742	0	0	0	0.855	0.8683	0.7692	0.864	0.8654	0.8119
OTHER	0.9876	0.9781	0.9414	0.9499	0.9333	0.8179	0.9881	0.9799	0.9443	0.9883	0.9804	0.9553
Weighted Average	0.9747	0.9566	0.9272	0.8875	0.8544	0.7425	0.9764	0.9613	0.9307	0.9783	0.9655	0.9456

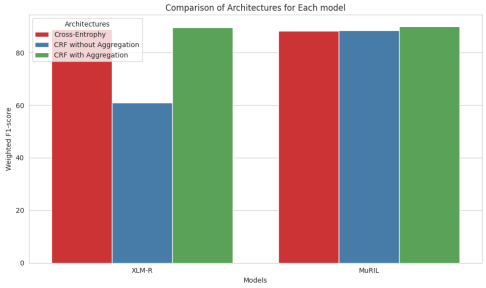

FIGURE 6. Architecture-wise comparison of weighted average F1-score for XLM-R and MuRIL for HiNER Dataset.

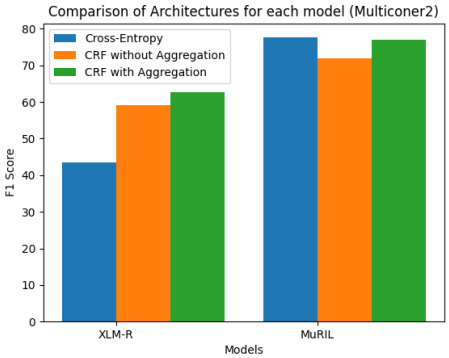

FIGURE 7. Architecture-wise comparison of weighted average F1-score for XLM-R and MuRIL for multiconer2 Dataset.

A. RESULTS FOR THE HINER DATASET

1) COMPARISON WITH BENCHMARK RESULTS

Table 6 shows the average results of the top three models of both architectures CRF and Aggregated-CRF, along with benchmarks (cross-entropy without aggregation) architecture for MuRIL and XLM-R. Table 6 contains tag-wise results along with their weighted average, macro average and micro F1-scores. XLM-R and MuRIL with Aggregated-CRF

TABLE 8. Strict F1-score (%) for XLM-R (benchmark), MuRIL (benchmark), XLM-R_{CRF}, $MuRIL_{CRF}$, XLM-$R_{Aggregated\ CRF}$, and $MuRIL_{Aggregated\ CRF}$ for each tag along with overall weighted average of all tags (For Multiconer2 Dataset).

Labels/Models	XLM-R (CE)	MuRIL (CE)	XLM-R (CRF)	MuRIL (CRF)	XLM-R(Aggregated CRF)	MuRIL(Aggregated CRF)
AerospaceManufacturer	4.76	23.17	9.52	5.62	5.31	13.33
AnatomicalStructure	31.36	81.85	54.26	76.04	56	80.22
ArtWork	0	3.27	5.48	0.59	1.52	1.42
Artist	53.43	74.28	55.03	65.66	49.11	73.21
Athlete	68.35	86.24	71.18	77.23	17.74	76.7
CarManufacturer	43.68	92.88	62.3	88.52	70.77	80.71
Cleric	0	79.9	47.69	71	69.45	83.02
Clothing	0	76.76	65.14	74.56	81.05	79.75
Disease	35.87	80.62	56.01	75.38	68.24	77.74
Drink	0	78.15	58.79	70.97	77.18	65.52
Facility	15.65	67.49	40.23	54.11	52.91	60.41
Food	32.56	71.12	51.51	66.75	63.97	66.51
HumanSettlement	65.91	92.3	78.34	88.34	86.56	92.72
MedicalProcedure	18.08	78.89	55.94	70.61	72.67	83.2
Medication/Vaccine	14.37	80.68	52.41	75.68	65.05	77.4
MusicalGRP	25.63	90.96	50.23	85.64	75.84	88.77
MusicalWork	3.08	48.72	26.76	52.31	54.87	52.63
ORG	53.47	82.46	69.01	75.5	77.38	83.13
OtherLOC	0	55.56	47.38	62.93	58.69	77.58
OtherPER	23.07	39.41	30.33	40.5	22.98	46.44
OtherPROD	3.25	69.33	34.3	58.68	52.97	60.02
Politician	28.99	66.89	41.58	59.28	41.92	60.93
PrivateCorp	0	17.24	44.3	66.29	66.67	91.36
PublicCorp	29.48	83.94	56.86	75.17	45.71	70.65
Scientist	0	61.64	43.1	41.4	32.61	66.67
Software	40.97	87.29	64.85	78.61	64.9	86.58
SportsGRP	62.65	96.05	81.07	91.91	78.86	95.24
SportsManager	0	31.62	8.12	9.86	49.31	40.45
Station	62.39	87.71	68.51	76.01	87.04	90.2
Symptom	1.05	65.48	42.55	70.89	68.75	88.89
Vehicle	13.47	82.64	42.79	76	16.41	77.42
VisualWork	51.61	83.68	51.9	70.57	47.49	71.43
WrittenWork	41.63	83.59	55.21	73.87	72.98	81.4
Average weight	43.55	77.62	59.21	71.85	62.62	77.09

architecture outperform both XLM-R and MuRIL benchmarks over all three evaluation metrics: weighted average, macro average and micro F1-score. Also, MuRIL with CRF architecture outperforms the MuRIL benchmark over all the evaluation metrics weighted average, macro average and micro F1-score. Architecture-wise comparison of weighted average F1-score for both models is also shown in Fig. 6. Scores of both architectures, CRF and Aggregated-CRF, are averages of the top three performances of XLM-R and MuRIL. When using Aggregated-CRF, XLM-R performs better than XLM-R (benchmarks) by 0.77 F1-score, while MuRIL performs better than MuRIL (benchmarks) by 1.71 F1-score.

2) COMPARISON BETWEEN CRF AND AGGREGATED-CRF

Results for comparison between the CRF and Aggregated-CRF have been shown in Fig. 6 and details tabulated in Table 6 as CRF architecture performed well for MuRIL; however, results of XLM-R with CRF architecture were not good comparatively, and it might be happening because of the large number (135130) OOV words found by XLM-R tokenizer during the tokenization of test data while for MuRIL it was 55224. However, with aggregated architecture, both XLM-R and MuRIL performed better than CRF architecture, and the improvement of XLM-R with aggregated architecture is notable (Fig. 6).

3) TOKENIZATION LENGTH-WISE COMPARISON

Tokenization Length-wise comparison between the CRF and Aggregated-CRF has been tabulated in Table 7. Model tokenizers tokenize each word of test data into one, two or more subwords. For analysis of the effect of these lengths, all words of test data were divided into classes based on their tokenized length (number of subwords, a word tokenized into.) and a strict F1-score was calculated for each tag in Table 7. For Aggregated-CRF architecture, all scores are almost uniform if the tokenized length increases from one to three. However, for CRF architecture, the weighted F1-score decreases if the tokenized length increases. Words with tokenized lengths greater than three are not considered for analysis since their support is lesser.

B. RESULTS FOR THE MULTICONER2 DATASET

1) COMPARISON WITH BENCHMARK (CROSS-ENTROPY) RESULTS

Table 8 shows the tag-wise results of the best models of both architectures, CRF and Aggregated-CRF, along with benchmarks (cross-entropy without aggregation) architecture for

Sr. No	Example from test Data	Ground Truth	Prediction with Aggregated CRF Architecture	Prediction with CRF Architecture
	Hiner data with XLM-R Language Model			
A1	'सबलपुर', 'पुनपुन', ',', 'पटना', ',', 'बिहार', 'स्थित', 'एक', 'गाँव', 'है।'	B-LOCATION', 'B-LOCATION', 'O', 'B-LOCATION', 'O', 'B-LOCATION', 'O', 'O', 'O', 'O'	B-LOCATION', 'B-LOCATION', 'O', 'B-LOCATION', 'O', 'B-LOCATION', 'O', 'O', 'O', 'O'	B-LOCATION', 'O', 'O', 'O', 'O', 'B-LOCATION', 'B-LOCATION', 'O', 'O', 'O'
A2	'यह', 'अली', 'सरदार', 'जाफ़री', 'द्वारा', 'लिखी', 'गयी', 'पुस्तक', 'है।'	O', 'B-PERSON', 'I-PERSON', 'I-PERSON', 'O', 'O', 'O', 'O', 'O'	O', 'B-PERSON', 'I-PERSON', 'I-PERSON', 'O', 'O', 'O', 'O', 'O'	O', 'B-PERSON', 'O', 'O', 'O', 'O', 'O', 'O', 'O'
A3	'सीबोर्गियम', 'एक', 'रासायनिक', 'तत्व', 'है।'	O', 'B-NUMEX', 'O', 'O', 'O'	O', 'B-NUMEX', 'O', 'O', 'O'	O', 'O', 'O', 'O', 'O'
A4	'नाना', 'पालसिकर', 'हिन्दी', 'फ़िल्मों', 'के', 'एक', 'अभिनेता', 'हैं।'	B-PERSON', 'I-PERSON', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O'	B-PERSON', 'I-PERSON', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O'	B-PERSON', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O', 'O'
A5	'इंडियन', 'एयरलाइंस', 'कालोनी', 'दिल्ली', 'का', 'एक', 'आवासीय', 'क्षेत्र', 'है।'	B-LOCATION', 'I-LOCATION', 'I-LOCATION', 'B-LOCATION', 'O', 'O', 'O', 'O'	B-LOCATION', 'I-LOCATION', 'I-LOCATION', 'B-LOCATION', 'O', 'O', 'O', 'O', 'O'	B-ORGANIZATION', 'B-LOCATION', 'O', 'O', 'O', 'O', 'O', 'O'
	Hiner data with MuRIL Language Model			
B1	'पालनी', 'पर्वत', 'भारत', 'के', 'पश्चिमी', 'घाट', 'की', 'पर्वतमाला', 'का', 'एक', 'पर्वत', 'है।'	B-LOCATION', 'I-LOCATION', 'B-LOCATION', 'O', 'B-LOCATION', 'I-LOCATION', 'O', 'O', 'O', 'O', 'O'	B-LOCATION', 'I-LOCATION', 'B-LOCATION', 'O', 'B-LOCATION', 'I-LOCATION', 'O', 'O', 'O', 'O', 'O'	O', 'I-LOCATION', 'B-LOCATION', 'O', 'B-LOCATION', 'I-LOCATION', 'O', 'O', 'O', 'O', 'O'
B2	क्रिकेट', 'प्रशंसक', 'और', 'विशेषज्ञ', 'इसके', 'गूढ़', 'को', 'समझने', 'में', 'अब', 'भी', 'लगे', 'हुए', 'हैं.'	B-GAME', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'	B-GAME', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'	O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'
B3	इसराइली', 'आम', 'चुनाव', 'में', 'मतदान', 'जारी', '.'	B-MISC', 'O', 'O', 'O', 'O', 'O', 'O'	B-MISC', 'O', 'O', 'O', 'O', 'O', 'O'	B-LOCATION', 'O', 'O', 'O', 'O', 'O', 'O'
B4	प्रैसीडेंसी', 'कॉलेज', ',', 'कोलकाता', '.'	B-ORGANIZATION', 'I-ORGANIZATION', 'O', 'B-LOCATION', 'O'	B-ORGANIZATION', 'I-ORGANIZATION', 'O', 'B-LOCATION', 'O'	B-LOCATION', 'I-LOCATION', 'O', 'B-LOCATION', 'O'
B5	नवनीत', 'हिन्दी', 'की', 'मासिक', 'पत्रिका', 'है।', 'इसके', 'सम्पादक', 'वरिष्ठ', 'लेखक-पत्रकार', 'विश्वनाथ', 'सचदेव', 'हैं।'	B-LITERATURE', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-PERSON', 'I-PERSON', 'O'	B-LITERATURE', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-PERSON', 'I-PERSON', 'O'	O', 'B-LANGUAGE', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B-PERSON', 'I-PERSON', 'O'
	multiconer2 data with XLM-R Language Model			
C1	उर्मिला', '(', 'अभिनेत्री', 'पर', 'आधारित', 'माला', 'सिन्हा', ')'	O', 'O', 'O', 'O', 'O', 'B-Artist', 'I-Artist', 'O'	O', 'O', 'O', 'O', 'O', 'B-Artist', 'I-Artist', 'O'	B-Artist', 'O', 'O', 'O', 'O', 'O', 'I-Artist', 'O'
C2	टूर्नामेंट', 'मैडीसन', 'स्क्वायर', 'गार्डन', 'न्यूयॉर्क', 'शहर', 'में', 'हुआ।'	O', 'B-Facility', 'I-Facility', 'I-Facility', 'B-HumanSettlement', 'I-HumanSettlement', 'O', 'O'	O', 'B-Facility', 'I-Facility', 'I-Facility', 'B-HumanSettlement', 'I-HumanSettlement', 'O', 'O'	I-Facility', 'B-Facility', 'I-Facility', 'I-Facility', 'B-HumanSettlement', 'I-HumanSettlement', 'O', 'O'
C3	दोनों', 'ब्लैक', 'बॉक्स', 'मलबे', 'से', 'बरामद', 'किए', 'गए', 'थे।'	O', 'B-OtherPROD', 'I-OtherPROD', 'O', 'O', 'O', 'O', 'O', 'O'	O', 'B-OtherPROD', 'I-OtherPROD', 'O', 'O', 'O', 'O', 'O', 'O'	O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'
C4	इसे', 'साण्टा', 'फे', 'में', 'फिलमाया', 'गया', 'था।'	O', 'B-HumanSettlement', 'I-HumanSettlement', 'O', 'O', 'O', 'O'	O', 'B-HumanSettlement', 'I-HumanSettlement', 'O', 'O', 'O', 'O'	O', 'B-HumanSettlement', 'B-HumanSettlement', 'O', 'O', 'O', 'O'
C5	परितारिका', 'काले', 'रेटिकुलेशन', 'के', 'साथ', 'कांस्य', 'है।'	B-AnatomicalStructure', 'O', 'O', 'O', 'O', 'O', 'O'	B-AnatomicalStructure', 'O', 'O', 'O', 'O', 'O', 'O'	O', 'O', 'O', 'O', 'O', 'O', 'O'
	multiconer2 data with MuRIL Language Model			
D1	अर्नस्ट', 'नेचिडा', 'फुटबॉल', 'खिलाड़ी', 'और', 'कोच'	B-SportsManager', 'I-SportsManager', 'O', 'O', 'O', 'O'	B-SportsManager', 'I-SportsManager', 'O', 'O', 'O', 'O'	B-Scientist', 'I-Athlete', 'O', 'O', 'O', 'O'
D2	ऊबे', 'यामागुची', 'प्रीफेक्चर', 'जापान', 'में', 'रेलवे', 'स्टेशन', 'है।'	B-HumanSettlement', 'B-ORG', 'I-ORG', 'B-HumanSettlement', 'O', 'B-Station', 'I-Station', 'O'	B-HumanSettlement', 'B-ORG', 'I-ORG', 'B-HumanSettlement', 'O', 'B-Station', 'I-Station', 'O'	O', 'B-ORG', 'I-ORG', 'B-HumanSettlement', 'O', 'B-Station', 'I-Station', 'O'
D3	तीव्र', 'फ्रैक्चर', 'गंभीर', 'पीठ', 'दर्द', 'का', 'कारण', 'होगा।'	O', 'O', 'O', 'B-Symptom', 'I-Symptom', 'O', 'O', 'O'	O', 'O', 'O', 'B-Symptom', 'I-Symptom', 'O', 'O', 'O'	O', 'O', 'O', 'O', 'I-Symptom', 'O', 'O', 'O'
D4	वह', 'स्वामी', 'दयानंद', 'भी', 'मिले', 'थे।'	O', 'B-Cleric', 'I-Cleric', 'O', 'O', 'O'	O', 'B-Cleric', 'I-Cleric', 'O', 'O', 'O'	O', 'B-Cleric', 'B-Cleric', 'O', 'O', 'O'
D5	फिल्म', 'ने', 'शम्मी', 'कपूर', 'का', 'एक', 'स्टार', 'बनाया।'	O', 'O', 'B-Artist', 'I-Artist', 'O', 'O', 'O', 'O'	O', 'O', 'B-Artist', 'I-Artist', 'O', 'O', 'O', 'O'	O', 'O', 'B-Artist', 'I-Politician', 'O', 'O', 'O', 'O'

FIGURE 8. **Comparative evaluation of a few examples from testing data.**

MuRIL and XLM-R. XLM-R with Aggregated-CRF architecture outperforms XLM-R benchmarks (cross-entropy) on comparison of weighted F1-score. However, MuRIL with Aggregated-CRF architecture is comparable with the cross-entropy architecture. When using Aggregated-CRF, XLM-R performs better than XLM-R (CE) by 19 F1-scores, while MuRIL performed comparably with MuRIL (CE). Architecture-wise comparison of weighted average F1-score for both models is shown in Fig. 7.

2) COMPARISON BETWEEN CRF AND AGGREGATED-CRF

Results for comparison between the CRF and Aggregated-CRF for the Multiconer2 dataset have been shown in Fig. 7 and details tabulated in Table 8. With the Aggregated-CRF architecture, both XLM-R and MuRIL (with weighted average F1-score of 66.62 and 77.02) performed better than CRF architecture (with weighted average F1-score of 59.21 and 71.85). After tokenization, the average length of each entity type in the test data is 2.58 for the XLM-R tokenizer and 1.78 for the MuRIL tokenizer. MuRIL tokenizer has less OOV compared to the XLM-R tokenizer. This might be the reason for MuRIL's better performance.

3) TOKENIZATION LENGTH-WISE COMPARISON

The overall performance of Aggregated-CRF architecture is better than CRF architecture for both types of tokens (OOV and in vocab). However, if we compare the tokenization length-wise results, the result of entities in Vocab is almost comparable with the result of entities that are OOV. It might be possible that in the Multiconer2 dataset, the number of entities that are OOV, are in a bigger ratio (Table 3).

C. ERROR ANALYSIS

The overall results show the advancement of the Aggregated-CRF architecture. A comparative examination of a few test data examples is displayed in Fig. 8 together with the ground truth and prediction for both the datasets and the language models for both architectures. Serial numbers A1 to A5 have the predictions of examples of the HiNER

dataset with XLM-R LM. These rows compare the prediction of CRF and Aggregated-CRF architectures. In row A2, the Aggregated-CRF architecture predicts the whole entity PERSON correctly; however, CRF architecture didn't predict the whole entity PERSON. Similarly, rows B1 to B5 have the predictions of examples of the HiNER dataset with MuRIL LM. Row B5 depict that with MuRIL CRF architecture, it is not able to predict entity LITERATURE correctly for the corresponding example. However, Aggregated-CRF architecture predicts it correctly. Similarly, Rows C1 to C5 and D1 to D5 of Fig. 8 depict some examples of the Multiconer2 dataset with XLM-R and MuRIL LMs. Rows C4 and D4 also depict that Aggregated-CRF predicts the whole entity; however, CRF architecture separates one entity into two entities.

The results above suggest that the aggregation method has been shown to be beneficial for the NER task if tokenized sentences contain more OOV words.

VIII. CONCLUSION

This work addresses the challenges posed by named entity recognition in the context of large language models and tokenization. These models tokenize out-of-vocabulary words into multiple sub-words, complicating the task of named entity recognition. To mitigate this complexity, the study proposes a technique that aggregates token embeddings for each word and introduces a conditional random field (CRF) layer on top of these aggregated word embeddings for named entity prediction.

The study focuses on HiNER and Multiconer2 datasets, and contrasts the suggested strategy with existing work for both datasets. In the case of more OOV, utilising a CRF layer with aggregated word embeddings performs better than conventional CRF models without aggregation. In addition, we have compared our method to the existing methods and found that it performed better for the HiNER dataset and yielded state-of-the-art results for the Multiconer2 dataset.

The paper thoroughly examines the effects of tokenization on a word-by-word basis and finds that words which are tokenized into multiple sub-words (OOV) perform better with the Aggregated-CRF architecture. A comparison between the two transformer-based models MuRIL and XLM-R is also included in the work.

Overall, this work emphasises the benefits of word embedding aggregation and the use of CRF layers while providing useful insights into improving named entity recognition in the setting of large language models.

REFERENCES

[1] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1064–1074. [Online]. Available: https://aclanthology.org/P16-1101

[2] C. Chen, W. E. Zhang, A. S. Shakeri, and M. Fiza, "The exploration of knowledge-preserving prompts for document summarisation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[3] C. Aone, "A trainable summarizer with knowledge acquired from robust NLP techniques," *Adv. In Autom. Text Summarization*, pp. 71–80, 1999. [Online]. Available: https://cir.nii.ac.jp/crid/1571135650129558656

[4] B. Bhardwaj, S. I. Ahmed, J. Jaiharie, R. S. Dadhich, and M. Ganesan, "web scraping using summarization and named entity recognition (NER)," in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, vol. 1, Mar. 2021, pp. 261–265.

[5] M. Al-Smadi, I. Al-Dalabih, Y. Jararweh, and P. Juola, "Leveraging linked open data to automatically answer Arabic questions," *IEEE Access*, vol. 7, pp. 177122–177136, 2019.

[6] A. Reddy and R. C. Rose, "Integration of statistical models for dictation of document translations in a machine-aided human translation task," *IEEE Trans. Audio, Speech, Language Process*, vol. 18, no. 8, pp. 2015–2027, Nov. 2010.

[7] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in *Proc. 7th Int. Workshop MT Other Lang. Technol. Tools, Improving MT Through Other Lang. Technol. Tools Resour. Tools Building MT*, 2003, pp. 1–8.

[8] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.

[9] J. Li, A. Sun, J. Han, and C. Li, "A Survey on deep learning for named entity recognition," *IEEE Trans. On Knowl. and Data Eng.*, vol. 34, no. 1, pp. 50–70, Mar. 2020.

[10] C. Sutton, K. Rohanimanesh, and A. McCallum, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," in *Proc. 21st Int. Conf. Mach. Learn.*, Jul. 2004, p. 99. [Online]. Available: https://api.semanticscholar.org/CorpusID:219683473

[11] T. Qian, M. Zhang, Y. Lou, and D. Hua, "A joint model for named entity recognition with sentence-level entity type attentions," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1438–1448, Mar. 2021.

[12] R. Sharma, S. Morwal, and B. Agarwal, "Named entity recognition using neural language model and CRF for Hindi language," *Comput. Speech Lang.*, vol. 74, Jul. 2022, Art. no. 101356. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230822000055

[13] R. Murthy, P. Bhattacharjee, R. Sharnagat, J. Khatri, D. Kanojia, and P. Bhattacharyya, "HiNER: A large Hindi named entity recognition dataset," in *Proc. Int. Conf. Lang. Resour. Eval.*, Jun. 2022, pp. 1–10.

[14] S. Malmasi, A. Fang, B. Fetahu, S. Kar, and O. Rokhlenko, "SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER)," in *Proc. 16th Int. Workshop Semantic Eval. (SemEval)*, 2022, pp. 1412–1437.

[15] B. Fetahu, S. Kar, Z. Chen, O. Rokhlenko, and S. Malmasi, "SemEval-2023 Task 2: Fine-grained multilingual named entity recognition (MultiCoNER 2)," 2023, *arXiv:2305.06586*.

[16] S. Singh and U. Tiwary, "Silp_nlp at SemEval-2023 task 2: Cross-lingual knowledge transfer for mono-lingual learning," in *Proc. The 17th Int. Workshop Semantic Eval. (SemEval)*, 2023, pp. 1183–1189.

[17] T. Meng, A. Fang, O. Rokhlenko, and S. Malmasi, "GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 1499–1512.

[18] B. Chen, J.-Y. Ma, J. Qi, W. Guo, Z.-H. Ling, and Q. Liu, "USTC-NELSLIP at SemEval-2022 task 11: Gazetteer-adapted integration network for multilingual complex named entity recognition," in *Proc. 16th Int. Workshop Semantic Eval. (SemEval-)*, 2022, pp. 1613–1622.

[19] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "LUKE: Deep contextualized entity representations with entity-aware self-attention," 2020, *arXiv:2010.01057*.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[21] M. Sabane, A. Ranade, O. Litake, P. Patil, R. Joshi, and D. Kadam, "Enhancing low resource NER using assisting language and transfer learning," in *Proc. 2nd Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)*, May 2023, pp. 1666–1671.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[24] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.

[25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 8440–8451.

[26] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, and P. Talukdar, "MuRIL: Multilingual representations for Indian languages," 2021, *arXiv:2103.10730*.

[27] D. Kakwani, A. Kunchukuttan, S. Golla, N. C. Gokul, A. Bhattacharyya, M. M. Khapra, and P. Kumar, "IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages," in *Proc. Findings Assoc. Comput. Linguistics, (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 4948–4961. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.445/

[28] S. Singh, P. Jawale, and U. Tiwary, "Silpa_nlp at SemEval-2022 tasks 11: Transformer based NER models for Hindi and Bangla languages," in *Proc. 16th Int. Workshop Semantic Eval. (SemEval)*, 2022, pp. 1536–1542.

[29] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proc. ICML*, Nov. 2001, vol. 17, no. 2000, pp. 591–589.

[30] R. Panchendrarajan, "Bidirectional LSTM-CRF for named entity recognition," in *Proc. 32nd Pacific Asia Conf. Lang. Inf. Comput.*, 2018, pp. 1–10. [Online]. Available: https://aclanthology.org/Y18-1061

[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[32] OpenAI. *Introducing Chatgpt*. Accessed: Dec. 1, 2023. [Online]. Available: https://openai.com/blog/chatgpt

[33] T. Xie, Q. Li, J. Zhang, Y. Zhang, Z. Liu, and H. Wang, "Empirical study of zero-shot NER with ChatGPT," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 7935–7956.

[34] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is ChatGPT a general-purpose natural language processing task solver?" in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 1339–1384.

[35] X. Wang, Y. Shen, J. Cai, T. Wang, X. Wang, P. Xie, F. Huang, W. Lu, Y. Zhuang, K. Tu, W. Lu, and Y. Jiang, "DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition," in *Proc. 16th Int. Workshop Semantic Eval. (SemEval-)*, 2022, pp. 1457–1468.

[36] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019, *arXiv:1711.05101*.

SUMIT SINGH (Graduate Student Member, IEEE) received the bachelor's degree in science and the master's degree in computer application and in information technology (specializing in software engineering). He is currently a Research Scholar with the Indian Institute of Information Technology Allahabad, Allahabad. His work specializes in Indic languages. He also works on deep learning-based pretrained models, such as transformers and other sequence learning models. His research interests include named entity recognition, natural language generation, and question-answering.

UMA SHANKER TIWARY (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronics Engineering, Institute of Technology, Banaras Hindu University, Varanasi, India, in 1991. He was a Lecturer with the Department of Electronics and Communication, J. K. Institute of Applied Physics and Technology, University of Allahabad, from September 1988 to March 1992, where he was a Reader in computer science with the J. K. Institute of Applied Physics and Technology, from March 1992 to June 2002. He was also a Visiting Scientist with the Department of Computer Science and Engineering, IIT Kanpur, from December 1995 to July 1996. He was an Associate Professor with the Indian Institute of Information Technology Allahabad, Allahabad, India, from July 2002 to December 2006, where he has been a Professor with the Department of Information Technology, since December 2006. He is holding research and teaching experience for more than 30 years, in which he is very much involved in image processing, computer vision, medical image processing, pattern recognition and script analysis, digital signal processing, speech and language processing, wavelet transforms, soft computing and fuzzy logic, neurocomputing and softcomputers, speech-driven computers, natural language processing, brain simulation, cognitive science, and affective computing.