SemEval 2022 Task 11: MultiCoNER - Multilingual Complex Named Entity Recognition

1 Abstract

We present the system description for our submission towards the Multi-CoNER Shared Task 11 at SemEval 2022. The task focuses on detecting semantically ambiguous and complex entities in short and low-context settings. We leveraged existing state of the art pre-trained language models along with incorporating additional data and features extracted from the inputs to improve performance. All the codes to generate reproducible results on our models are provided in the zip file submission. Additionally, our code is also uploaded to this Github repository.

2 Group Details

Subtask Id: SemEval Task 11

Quintet	
Name	Roll No
Kaizer Rahaman	19IE10044
Sreeya	19AG10008

3 Individual Contributions of Students

- **Kaizer Rahaman (19IE10044):** Implemented XLM RoBERTa with Bi-LSTM on CRF fusion for Farsi,German,Dutch,Spanish,Hindi.Experimented with fine tuning for Hindi Bangla.
- **Sreeya (19AG10008) :**Worked on hyper parameter tuning for XLM RoBERTa with Bi-LSTM on CRF fusion on Korean,Turkish, Bangla,English,Chinese and Russian and demonstrated fine tuning for English language .

4 Introduction:

The Multilingual Complex Named Entity Recognition (MultiCoNER) which is the 11th shared task of SemEval 2022 encourages participants to build complex Named Entity Recognition systems for any number of languages out of the 11 languages' data given by them. The languages are: English, Spanish, Dutch, Russian, Turkish, Korean, Farsi, German, Chinese, Hindi, and Bangla. We have mainly focused on three different languages, namely, English, Hindi, and Bengali. Formally, given an input sentence, we need to predict a set of NER tags for each token of the input sentence. The task focuses on detecting semantically ambiguous and complex entities in short and low-context settings.

5 Dataset Descrpition

Multilingual Complex Named Entity Recognition dataset follows the CoNLL formatIn a data file, samples are separated by blank lines. Each data instance is tokenized and each line contains a single token with the associated label in the last (4th) column. Second and third columns (-) are ignored. Entities are labeled using the BIO scheme. That means, a token tagged as O is not part of an entity, B-X means the token is the first token of an X entity, I-X means the token is in the boundary (but not the first token) of an X type entity having multiple tokens.

6 Task Description

6.1 Baseline Architecture

Transformer (Vaswani et al., 2017) based pre-trained language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2019), and DeBERTa (He et al., 2021), have proven to be very powerful in learning robust context-based representations of lexicons and applying these to achieve state of the art performance on a variety of downstream tasks. We leverage these models for learning contextual

representations of the sequences for which the named entities are to be identified. One of the experiments included concatenation of these contextual representations to an encoded feature vector of additional features (one of Dependency Parse based features and Parts-of-Speech based features). This concatenated vector was then passed through a couple of dense layers and a CRF layer to get the final named entity for a particular word token.

We model the tagging decisions jointly using a conditional random field (Lafferty et al., 2001). For an input sequence i.e., the contextual embedding that we obtain from the transformer,**X = (x1, x2, ..., xn)**
we consider P to be the matrix of scores output by the set of linear layers after the transformer model. P is of size n x k, where k is the number of distinct tags, and Pi,j corresponds to the score of the jth tag for the ith word in a sentence. For a sequence of predictions **y = (y1, y2, ..., y3)**
the score is defined as S(X, y) = Xn i=1 where A is a matrix of transition scores such that Ai,j represents the score of a transition from the tag i to tag j.

$$S(X, y) = \sum_{i=1}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

A softmax over all possible tag sequences yields a probability for the sequence y:

$$p(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in Y_x} e^{s(X,y)}}$$

During training, we maximize the log-probability of the correct tag sequence: where Yx

$$log(p(y|X)) = s(X, y) - log\Big(\sum_{\tilde{y} \in Y_x} e^{s(X,y)}\Big)$$

represents all possible tag sequences for a sentence X. From the formulation above, it is evident that we encourage the network to produce a valid sequence of output labels. While decoding, we use the standard Viterbi algorithm for obtaining the optimal tag sequence for the input sequence of words

6.2 Parts of Speech Features

With a similar motive as before, i.e., to better capture the syntactic structure of the sentences, we

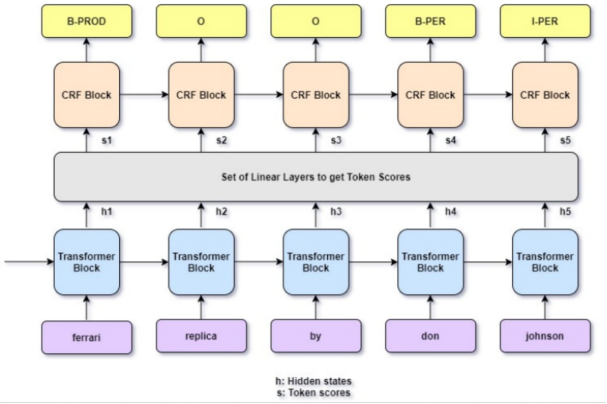

Figure 1: Model Architecture.

experimented with Part-Of-Speech (POS) Features. We used the open-source tool Spacy to obtain POS labels for each lexicon, which were then labeled encoded according to descending order of occurrences. The encoded feature vector is then concatenated to the output of the transformer model and fed to the subsequent layers.

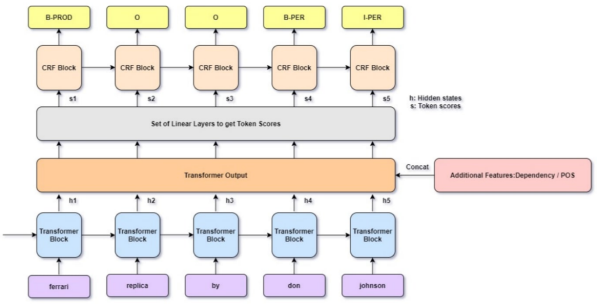

Figure 2: Model Architecture for Additional Features

6.3 Bi-LSTM CRF Fusion

Instead of directly using the contextual representation of the word tokens obtained from the transformer to feed into fully connected layers followed by CRF, we embed a bidirectional long short-term memory layer just after the transformer layer to make use of the robust representations and train a sequential model based on the LSTM architecture. The bi-LSTM effectively captures the sequential relationships amongst the contextual transformer embeddings whereas CRF provides the optimal joint prediction of all the labels in the sentence.

6.4 Hyper-Parameter Tuning

An integral part of learning the best model for certain data is to get the optimal setting of the

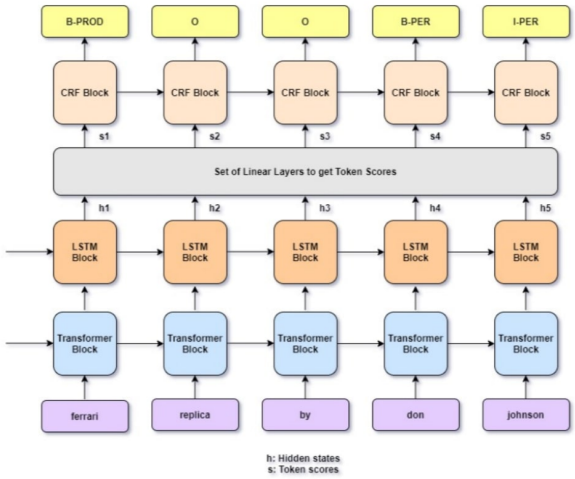

Figure 3: LSTM with CRF model architecture.

hyper-parameters. We carried out thorough hyper-parameter tuning with the following experiments in our implementation.

1. Batch size and Learning rate: To obtain the optimal batch size and learning rate setting, we carry out a thorough grid search on three settings each of batch size and learning rate. The batch sizes used were 16, 32, and 64, whereas the learning rates used were 1e-5, 2e-5, and 3e-5.

2. Feedforward layers: We experiment with 5 different settings of feedforward layers to find the optimal architecture. We embedded 2 feedforward layers before the transformer and CRF layer. The detailed experiments are mentioned in the results section.

7 Results and Discussions

This section is organized to enumerate the results of different transformer-based architectures and some additional features and hyper-parameter tuning for Multilingual Complex Named Entity Recognition. We use one Tesla k80 GPU with 16 GB RAM to perform all the experiments. All the experiments are done on the google colab platform. We present the values of the following metrics to evaluate our system:

1. Micro-precision measures the precision of the aggregated contributions of all classes. It's short for micro-averaged precision. A precision score of 1 means the model's predictions are perfect; all samples classified as the positive class are truly positive.

2. Micro-recall measures the recall of the aggregated contributions of all classes. It's short for micro-averaged recall. Micro-recall score of 1 means the model's predictions are perfect; all truly positive samples were predicted as the positive class.

3. Micro F1-score transformer-based (short for mianded F1 score) assesses the quality of multi-label binary problems. It measures the F1-score of the aggregated contributions of all classes.

4. MD Precision: Mention Detection (Kummerfeld et al., 2011) the Precision measures the precision of the task where the model has only to identify the entity boundaries regardless of the entity type.

5. MD Recall: Mention Detection Recall measures the recall of the task where the model has only to identify the entity boundaries regardless of the entity type.

6. MD F1-score: Mention Detection F1-score measures the f1-score of the task where the model has only to identify the entity boundaries regardless of the entity type.

First, we tried to experiment only with the transformer-based architectures on BERT base to get an intuition on the models performed for the task of complex named entity recognition. As shown in Table 1, for all the 11 languages. etoolbox

Language	Micro@P	Micro@R	Micro@F1	MD@R	MD@P	MD@F1
English	0.745	0.766	0.755	0.843	0.843	0.831
Hindi	0.597	0.0.599	0.598	0.0.736	0.734	0.0.735
Bangla	0.570	0.629	0.599	0.769	0.700	0.733
German	0.772	0.558	0.648	0.691	0.787	0.736
Russian	0.538	0.560	0.549	0.565	0.633	0.597
Spanish	0.686	0.710	0.698	0.703	0.779	0.739
Chinease	0.734	0.697	0.715	0.741	0.774	0.757
Dutch	0.775	0.732	0.753	0.790	0.854	0.821
Korean	0.52	0.495	0.507	0.564	0.55	0.557
Turkish	0.604	0.590	0.597	0.591	0637.	0.613
Farsi	0.672	0.507	0.578	0.631	0.623	0.627

Table 1: Results of Transformer Model

Language	Micro@P	Micro@R	Micro@F1	MD@R	MD@P	MD@F1
English	0.748	0.745	0.747	0.837	0.842	0.839
Hindi	0.619	0.0.647	0.632	0.0.771	0.738	0.0.754
Bangla	0.598	0.645	0.62	0.793	0.736	0.764

Table 2: Results with POS Features

Table 2 enumerates the results for transformer models along with the parts-of-speech features. Implemented XLM RoBERTa on English,Hindi and Bangla.

Language	Micro@P	Micro@R	Micro@F1	MD@R	MD@P	MD@F1
English	.762	0.741	0.751	0.827	0.850	0.838
Hindi	564	0.600	0.581	0.759	0.714	0.736
Bangla	0.615	0.611	0.612	0.750	0.755	0.753
German	0.7309	0.712	0.721	0.78353	0.8567	0.845
Farsi	0.588	0.567	0.577	0.624	0.647	0.635
Russian	0.688	0.652	0.669	0.728	0.768	0.747
Spanish	0.727	0.664	0.694	0.7600	0.8316	0.794
Chinese	0.662	0.628	0.644	0.724	0.763	0.743
Korean	0.582	0.629	0.604	0.76	0.703	0.731
Turkish	0.756	0.720	0.738	0.762	0.8015	0.781
Dutch	0.733	0.767	0.77	0.839	0.845	0.842

Table 3: Results with Bi-LSTM + CRF fusion

Table 3 provides the results for the architecture where we used a bi-directional LSTM over transformer along with CRF to get the most probable sequence of named entity tags for the individual language.

Language	Micro@P	Micro@R	Micro@F1	MD@R	MD@P	MD@F1
English	.73954	0.70407	0.72137	0.76748	0.80615	0.78634
Hindi	0.62454	0.61091	0.61765	0.72727	0.74349	0.73529
Bangla	0.59521	0.59596	0.59558	0.70707	0.70618	0.70662
German	0.71227	0.65133	0.68044	0.73043	0.79876	0.76307
Farsi	0.58381	0.59587	0.58978	0.64628	0.63320	0.63967
Russian	0.58381	0.59587	0.58978	0.64628	0.63320	0.63967
Spanish	0.72468	0.66922	0.69584	0.73980	0.80110	0.76923
Chinese	0.77152	0.72756	0.74890	0.78845	0.83609	0.81157
Korean	0.55287	0.51916	0.5354	0.55747	0.59367	0.57500
Turkish	0.64482	0.58521	0.61357	0.62621	0.68999	0.65655
Dutch	0.81375	0.73639	0.77314	0.79257	0.87584	0.83212

Table 4: Results with Multilingual language model

Table 4 provides the results for Multilingual language model performed on 11 languages all together.

Model Name	Language	Batch Size	Learning Rate	First layer neurons	Second layer neurons	micro @P	micro @R	micro @F1	MD @R	MD @P	MD @F1
XLM-RoBERTa-base	English	16	3.00E-05	512	512	0.753	**0.762**	0.757	**0.850**	0.840	**0.844**
XLM-RoBERTa-base	English	16	3.00E-05	512	256	**0.785**	0.753	0.768	0.826	**0.861**	0.843
XLM-RoBERTa-base	English	16	3.00E-05	512	128	0.783	**0.762**	**0.772**	0.824	0.847	0.836
XLM-RoBERTa-base	English	16	3.00E-05	256	256	0.759	0.754	0.756	0.832	0.837	0.834
XLM-RoBERTa-base	English	16	3.00E-05	256	128	0.753	0.742	0.747	0.828	0.839	0.833
XLM-RoBERTa-base	Bangla	16	2.00E-05	512	512	0.532	0.606	0.566	**0.774**	0.679	0.723
XLM-RoBERTa-base	Bangla	16	2.00E-05	512	256	0.555	0.596	0.575	0.755	0.704	0.728
XLM-RoBERTa-base	Bangla	16	2.00E-05	512	128	0.565	**0.607**	**0.585**	0.768	0.714	0.740
XLM-RoBERTa-base	Bangla	16	2.00E-05	256	256	0.566	0.604	0.584	0.771	0.724	**0.747**
XLM-RoBERTa-base	Bangla	16	2.00E-05	256	128	**0.569**	0.581	0.575	0.742	**0.727**	0.735
XLM-RoBERTa-base	Hindi	32	2.00E-05	512	512	0.584	0.599	0.591	0.731	0.713	0.722
XLM-RoBERTa-base	Hindi	32	2.00E-05	512	256	0.545	0.591	0.567	**0.742**	0.684	0.712
XLM-RoBERTa-base	Hindi	32	2.00E-05	512	128	**0.612**	**0.625**	**0.619**	0.739	0.724	0.731
XLM-RoBERTa-base	Hindi	32	2.00E-05	256	256	0.608	0.599	0.603	0.732	**0.743**	**0.737**
XLM-RoBERTa-base	Hindi	32	2.00E-05	256	128	0.558	0.578	0.568	0.728	0.703	0.715

Figure 4: Result of fine tuning

Table 5 shows the results of fine-tuning the baseline architecture with different settings on feedforward layers. These experiments were done for XLM RoBERTa base ,model outperformed baseline models in most of the settings for all the three languages. Moreover, due to time constraints, we leave the analysis of fine-tuning and finding the optimal setting of feedforward layers with only XLM RoBERTa base model.

8 Conclusion

This paper presents various analysis for constructing Named Entity Recognition for complex entities. We use different transformer-based models for NER and observe that XLM RoBERTa base model outperforms the other model for almost other languages for most number of metrics.

We also use the contextual encodings obtained from bidirectional LSTM to build our model. With an intuition of capturing the syntactic structure of the sentences, we try to obtain features from dependency graph as well as parts of speech tag.

Finally, we carry thorough hyper-parameter tuning to find the optimal setting of the learning rate, batch size and the best set of feed forward layer to get the token score that are subsequently fed into the CRF layer. Our proposed model performs fair enough and we achieve good score of micro average F1 and Mention Detection F1 for 11 languages.

9 References:

1. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

2. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

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

4. Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.