Infrrd.ai at SemEval-2022 Task 11: A system for named entity recognition using data augmentation, transformer-based sequence labeling model, and EnsembleCRF

JiangLong He, Akshay Uppal, Mamatha N,
Shiv Vignesh, Deepak Kumar, Aditya Kumar Sarda
Infrrd.ai
{jianglong,akshayuppal,mamathan}@infrrd.ai
{shivvignesh,deepakumar,adityasarda}@infrrd.ai

Abstract

In low-resource languages, the amount of training data is limited. Hence, the model has to perform well in unseen sentences and syntax on which the model has not trained. We propose a method that addresses the problem through an encoder and an ensemble of language models. A language-specific language model performed poorly when compared to a multilingual language model. So, the multilingual language model checkpoint is fine-tuned to a specific language. A novel approach of one hot encoder is introduced between the model outputs and the CRF to combine the results in an ensemble format. Our team, **Infrrd.ai**, competed in the MultiCoNER competition. The results are encouraging where the team is positioned within the top 10 positions. There is less than a 4% percent difference from the third position in most of the tracks that we participated in. The proposed method shows that the ensemble of models with a multilingual language model as the base with the help of an encoder performs better than a single language-specific model.

1 Introduction

In conll-2003 (Sang and De Meulder, 2003), a shared task was conducted to identify the named entities such as person (PER), location (LOC), organization (ORG), and miscellaneous (MISC). Over a period of time, there was an improvement in the developed systems (Marrero et al., 2009) which resulted in an improved performance that resulted in an increased number of entities. The named entities which needed to be identified and extracted were now six. They are person (PER), location (LOC), organization (ORG), group (GRP), product (PROD), and creative work (CW). Some of these named entities are created by coining a new word that may be non-existent or a combination of existing words in the entire corpus of words in a language. People are most likely interested in coining a new term from the list of words to have an identity tag of a location, an event, or similar concepts. The newly coined words become novel or emerging entities in the list of entities (Derczynski et al., 2017). These words form a most part of ambiguous words, where the word belongs to a particular entity or not, and is difficult to judge. For example, Microsoft is an organization, Windows is a product, and Microsoft Windows is a software product. Apart from this, creating a non-existent word as an entity is an expression of the creativity of the creator which belongs to the creative word entities. A competition was conducted in WNUT 2017 to identify the novel and emerging entities that look for unseen entities as described earlier (Derczynski et al., 2017). These types of entities are complex enough that even a person may miss the context of the entity and represent them differently.

The MultiCoNER competition brings in the additional dimension of complexity of low resources (Malmasi et al., 2022a,b). In a low resource language, the amount of training data available is limited within which the model has to learn to discriminate between the entities and identify them correctly. The model is exposed in this scenario to understand the unseen word that was not part of the training data and doesn't have annotated information to predict. Hence, the development of a model to examine the competition test data is more challenging than ever.

Our contributions and observations are summarized as follows:

- We explore various word-level data augmentation strategies such as LwTR, SR, MR, SiS, and bert-based token augmentation to improve the dataset size when training the transformer-based sequence labeling models. It is shown that the data augmentation increases the model generalization on the test set.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1501 - 1510

- Performing transfer learning using a multilingual sequence labeling model as an initializer improves the performance on language-specific tracks.
- Introduced the ensemble learning model, 'EnsembleCRF,' that solves the IOB scheme constraint when using majority voting. By learning to optimally combine the model predictions, EnsembleCRF also learns to avoid mistakes made by single sequence labeling models.

2 Related Work

The transformer-based language models fared better in the identification and extraction of entities from a given text. Since there is a necessity for a large amount of training data which provides a much-needed boost in accuracy. Sometimes, a model trained on a huge data in one or more languages is used in another or different language. These types of models are commonly known as cross-lingual language model (Conneau and Lample, 2019) or multilingual language model (Conneau et al., 2019). A fine-tuned language model has a better performance compared to a multilingual language model. But, the multilingual language model is more adaptable across different languages, which is not available for a fine-tuned language model. The researchers started exploring the amount of data required to train a language model. In some cases, the amount of data available in a language with annotation is very limited. These languages are termed low-resource languages. Due to this limitation, the model may not be aware of the complete set of words in the low-resource languages. Here, we are exploring to understand the performance of a transformer-based language model in low-resource constraints. The challenges involved in recognizing complex entities in low-resource environments (Meng et al., 2021; Fetahu et al., 2021) have led to the creation of the competition data (Malmasi et al., 2022a,b).

In low-resource scenarios, different approaches have been adapted to overcome the constraints. The available fine-tuned transformer-based model such as BERT is bootstrapped to improve the accuracy of NER (Yu et al., 2020). A prompt-guided attention layer is used as part of a transformer model by creating a semantic-aware answer space for tuning the model for further betterment (Chen et al., 2021). Sentence reconstruction approach to enhance low resource sequence tagging by utilizing the knowledge of high resource data (Perl et al., 2020). A common approach used on low resource languages is to use cross-lingual transfer learning, where a model trained on high resource language is used as the reference. An active learning mechanism was used to improve the performance of NER (Chaudhary et al., 2019). A teacher-student knowledge transfer model technique has shown to give effective results on low resource NER tasks (Izsak et al., 2019). An unsupervised cross-language transfer learning method where the encoders trained on the source and target language together using adversarial learning followed by augmented fine-tuning technique (Bari et al., 2020).

3 Methodology

Individual pre-trained models were used to evaluate the training and the dev set. Based on the empirical results, we decided to train a baseline language with multiple languages for 20 epochs and then perform transfer learning to train the monolingual models.

3.1 Main architecture

The training set with the following split, 151470 train + 8800 dev + 16830 test sentences, is tokenized and fed to the model xlm-roberta-large to generate the baseline multilingual checkpoint. The embeddings of the transformer model are then passed through the dropout layers. We have three types of dropouts in the mix as shown in Figure 1. The standard dropout with the probability of 0.3, the word dropout with the probability of 0.05, and finally the locked dropout with the probability of 0.5 were used. These embeddings are then linearly reprojected into a vector of size 1024. The reprojected vector is passed through a BiLSTM layer with 256 nodes, which generates a vector of size 512. This output vector is then passed through a CRF layer to generate the class label with IOB sequence tags.

The model is trained for 20 epochs to obtain the starting checkpoint for all the monolingual models. The performance of the monolingual models got a significant boost with cross-language training. We tried to train the last, the last two, the last three, and the last four layers of the transformer but did not get a significant boost while training more layers, so for the final step of training, we resorted to training the last layer of the transformer.

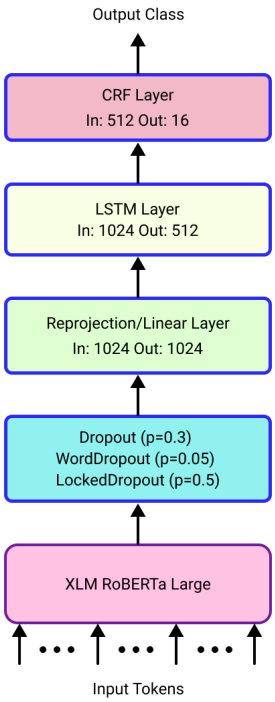

Figure 1: The architecture of the multilingual model was developed to train the training set of the competition data.

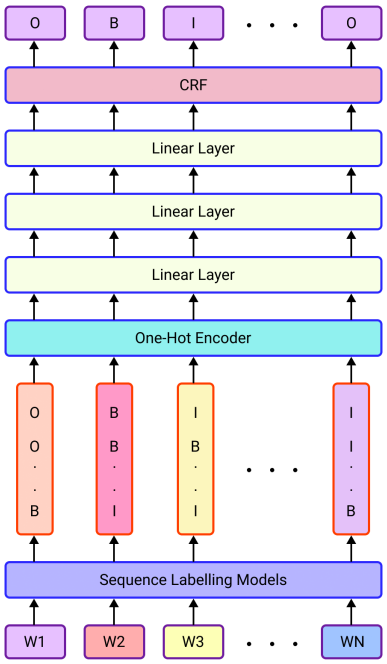

Figure 2: The architecture of EnsembleCRF model. Given a sentence as input, each of the sequence labeling models will output the name entity prediction in IOB format. One hot encoder combines them to generate ensemble output from a Conditional Random Field model.

The generated baseline multilingual checkpoint is loaded and fed with the augmented dataset for each monolingual task to obtain the respective language models. We have used different augmentation techniques described in the Training and Evaluation section. All the sentences provided by the competition organizers were used to generate the augmented data.

3.2 Ensemble architecture

A simple ensemble strategy of *Majority Voting* is developed. Given a set of M sequence labeling models denoted as $C = \{c_1, c_2, ..., c_M\}$ and an input sentence denoted as $S = \{w_1, w_2, ..., w_n\}$, where each w is a word from S. Each model from C will output a sequence of predictions for each word w in sentence S. Let $O^s_{c_i} = \{O^{c_i}_{w_1}, O^{c_i}_{w_2}, ..., O^{c_i}_{w_n}\}$ denote the prediction output of model C_i on sentence S. $O^{c_i}_{w_j}$ denotes the prediction of model C_i on word w_j in IOB format. The set of outputs for all models in C on sentence S will be denoted as

$$O_S = \{O^s_{c_1}, O^s_{c_2}, ..., O^s_{c_M}\} \quad (1)$$

The *Majority Voting* strategy takes all model's predictions of word w_j and outputs the most frequent prediction as to the final prediction for w_j. An obvious issue with *Majority Voting* is the IOB scheme constraint. The final ensemble result is not guaranteed to be passing all the constraints where (I) tag must follow and (B) tag and the entity of neighbor (B) and (I) tag must be the same. We introduce an ensemble learning approach via sequence labeling called 'EnsembleCRF' as shown in Figure 2.

The model outputs are stacked together and passed through a one-hot encoder, three linear layers, and CRF. The CRF layer is trained to optimally combine the model predictions to form a new set of predictions. The addition of the three linear layers helped in the performance improvement. The EnsembleCRF model is of the form

$$C_{en} = EnsembleCRF(C = \{c_1, c_2, ..., c_M\}, D_{en} = \{X, Y\}) \quad (2)$$

D_{en} is the ensemble learning dataset composed of X and Y. $X = \{O_{s_1}, O_{s_2}, ..., O_{s_k}\}$ is created by using model set C and set of input sentences $\{S\} = \{S_1, S_2, ..., S_k\}$ with size K. Each element of X is defined as equation (1). Y is the ground

Table 1: The language models used for different languages with the strategies adapted and the hyperparameters used for training the model. The last column shows the macro-averaged F1 score on the competition test set.

Model No	Language Model	BiLSTM	CRF	Transfer Learning	External Dataset	Aug. Data	Train with Dev	Learning Rate	F1-score
English									
1	roberta-large	Y	Y	N	N	Y	N	3e-3	0.7072
2	roberta-large	Y	Y	N	N	Y	Y	3e-3	0.715
3	xlm-roberta-large	Y	Y	Y	N	Y	N	1e-3	0.739
4	xlm-roberta-large	Y	Y	Y	Y	Y	Y	3e-3	0.7273
Spanish									
1	xlm-roberta-large + flair-es	Y	Y	N	N	N	N	1e-3	0.6031
2	xlm-roberta-large	Y	Y	N	N	N	N	3e-3	0.6505
3	mbert-uncased	Y	N	N	N	N	N	3e-3	0.6724
4	xlm-roberta-large	Y	Y	Y	N	Y[a]	Y	3e-3	0.738
Dutch									
1	dutchembedding + xlm-roberta-base	Y	Y	Y	N	N	N	3e-3	0.7603
2	xlm-roberta-large	Y	Y	Y	N	N	Y	3e-3	0.7603
3	xlm-roberta-base	Y	Y	Y	N	Y	Y	3e-3	0.7246
Korean									
1	xlm-roberta-base	N	N	N	N	N	N	3e-3	0.6315
2	xlm-roberta-base	Y	Y	N	N	N	N	3e-3	0.6481
3	xlm-roberta-base	Y	Y	Y	N	Y	N	3e-3	0.6407
4	xlm-roberta-large	Y	Y	Y	N	N	Y	3e-3	0.6729
5	xlm-roberta-large	Y	Y	Y	N	Y	Y	3e-3	0.6688
German									
1	germanembedding + xlm-roberta-base	Y	Y	Y	N	N	N	3e-3	0.7683
2	xlm-roberta-large	Y	Y	Y	N	N	Y	3e-3	0.7446
3	xlm-roberta-base	Y	Y	Y	N	Y	Y	3e-3	0.7402

[a] additionally translated sentences are used

truth entity label in IOB format. During the training phase for some models in set C, we included both provided training and dev datasets. Thus, we choose to perform the ensemble learning using an augmented dev set created using the data augmentation strategies explained in Section 4. Since the second layer classifier is a CRF layer, we solved the problem of breaking the IOB constraints. By learning to optimally combine the model predictions, EnsembleCRF also learns to avoid mistakes made by single sequence labeling models.

We experimented with creating D_{en} with not only the augmented dev dataset but also the augmented training dataset. However, we found that there is no positive correlation between the number of models in C and the macro-averaged F1 score on the test dataset. Treating every possible combination of set C as a hyperparameter to optimize will yield the optimal result.

4 Training and Evaluation

We have used the flair framework (Akbik et al., 2019) which uses pytorch and huggingface transformers to build and experiment with our approaches. The roberta transformer model was used as the base model. However, for some monolingual language training, we stacked a language-specific embedding layer provided by flair. For Dutch, German, and Spanish, flair language-specific embeddings were prepended before transformer embedding for experimentation.

We experimented with two different initial checkpoints loaded to train the transformer model. One checkpoint was from the huggingface library (hug), for both roberta-base and roberta-large. The other was to load the xlm-roberta model trained on the competition dataset as the initial checkpoint for the training of monolingual tasks. However, the models trained with the initial checkpoint from xlm-roberta performed better due to the transfer of knowledge from the multilingual checkpoint to the monolingual checkpoint. The language models used in the experiment for different languages with hyperparameters and macro averaged F1-score on the competition test set are tabulated in Tables 1 and 2.

4.1 Competition Data

The dataset provided by the competition organizers had 15300 train sentences and 800 dev sentences for each language. However, the entity distribution per language varied (Malmasi et al., 2022a,b). The number of sentences used for training is very less when compared to the number of sentences for testing, which provides the low-resource constraint designed by the competition organizers.

Table 2: The language models used for different languages with the strategies adapted and the hyperparameters used for training the model. The last column shows the macro-averaged F1 score on the competition test set (cont.).

Model No	Language Model	BiLSTM	CRF	Transfer Learning	External Dataset	Aug. Data	Train with Dev	Learning Rate	F1 score
Chinese									
1	bert-base-chinese	Y	Y	N	N	N	Y	3e-5	0.6468
2	bert-base-chinese	N	N	N	N	N	N	3e-3	0.608
3	yechen/ bert-large-chinese	Y	Y	N	N	N	Y	1e-3	0.6237
4	hfl/ chinese-roberta-wwm-ext	Y	Y	N	N	N	Y	1e-3	0.617
5	xlm-roberta-large	Y	Y	Y	N	N	Y	3e-3	0.645
Hindi									
1	monsoon-nlp/hindi-bert	Y	Y	N	N	N	N	3e-3	0.501
2	mbert-cased	Y	Y	N	N	N	N	3e-3	0.493
3	neuralspace-reverie/ indic-transformers-hi-bert	Y	Y	N	N	N	N	3e-2	0.4846
4	indic-distilbert	Y	Y	N	N	N	N	3e-2	0.5087
5	flax-community/roberta-hindi roberta-hindi	Y	Y	N	N	N	N	3e-3	0.2448
Bangla									
1	indic-distilbert	Y	Y	N	N	N	N	1e-3	0.4121
2	xlm-roberta-large	Y	Y	N	N	N	N	3e-3	0.5915
3	xlm-roberta-large	Y	Y	Y	N	N	N	3e-3	0.6019
Multilingual									
1	xlm-roberta-large	N	N	N	N	N	N	3e-3	0.6648
2	xlm-roberta-large	Y	Y	N	N	N	N	3e-3	0.6829
3	xllm-roberta-large	Y	Y	N	N	N	Y	3e-3	0.6924
4	xlm-roberta-large	Y	Y	N	N	Y	Y	3e-3	0.6704

4.2 Data Augmentation

Transformer-based language models require huge amounts of data to produce a good performance, but this requires a lot of labeled data. In the real world, such large labeled datasets are not available easily, especially in some specific domains. We need expert knowledge to annotate the data, which is time-consuming. However, we made use of simple data augmentation techniques for token-level (Dai and Adel, 2020). Here, the method concentrates on expanding the training data using smaller training sets and applying transformations to the training instances without changing their labels. We made use of all the techniques (Dai and Adel, 2020) namely Label-wise token replacement (LwTR), Synonym replacement (SR), Mention replacement (MR), Shuffle within segments (SiS), as well as the mixture of all the techniques to augment the training and development datasets. This produced improvement for a few of the languages even over strong baselines, where no augmentation was used. Although there is no clear single winner, applying all augmentation techniques outperformed single augmentation techniques on an average. We have tabulated the results and their explanation in Section 5.

We also made use of other open-source datasets (Samal, 2021) which were related to different domains like history, political parties, and particularly different from the competition training datasets in context. In addition to this, we made use of `nlpaug` (Ma, 2019) to generate the synthetic data without manual effort. A bert-based model was used to augment the original sentences which were later processed to match the number of token labels. These external datasets did not provide improvements to the performance of the baseline model.

5 Discussion and Results

The training strategies for all the tracks fall into these categories namely external dataset, data augmentations, model architecture searching, transfer learning, and ensemble learning as described in Sections 3 and 4. We used a gazetteer as the last option, which didn't improve the performance.

5.1 English

We trained 12 models using a combination of data augmentation, transfer learning, ensemble learning, and model architecture search. Out of the 12 trained models, we observed that using a multilingual model checkpoint for transfer learning on English data gives better performance on the dev set. We also observed that adding BiLSTM and CRF layer gives slightly better performance than using the linear layer as the classifier. Data augmentation didn't show any difference in the performance on the dev set but the model trained using data augmentation performs better in the test set evaluation.

Table 3: The macro-averaged F1 score for English language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.9122
2	USTC-NELSLIP	0.8547
3	PAI	0.7837
4	ML-HUB	0.7814
5	RACAI	0.7578
6	**Infrrd.ai**	**0.7471**
7	EURECOM	0.7457
8	Sliced	0.7454
9	MaChAmp	0.7448
10	Raccoons	0.7418
11	YNUNLP	0.7317
12	LMN	0.725
13	brotherhood	0.7235
14	L3i	0.7196
15	Multilinguals	0.7174
16	KDDIE	0.7173
17	MarSan_AI	0.7145
18	Cardiff NLP	0.7094
19	Lone Wolf	0.6977
20	MIDAS	0.6962
21	UC3M-PUCPR	0.6924
22	CSECU-DSG	0.6924
23	Sartipi-Sedighin	0.6751
24	Enigma	0.6719
25	DANGNT-SGU	0.6689
26	AaltoNLP	0.6685
27	SPDB Innovation Lab	0.6511
28	silpa_nlp	0.6342
29	BaselineExtending-Pokemons	0.6324
30	MultiCoNER Baseline	0.612
31	AutoNER	0.5572

The final model is an EnsembleCRF model trained with 4 Sequence Labeling models. There was a drop in the F1 score when all 12 models were used. So, we eventually kept the 2 models trained with the dev set and for the rest 10 models, we picked the best 2 models on the dev set. For all models in Table 1, we set the maximum epoch to be 100 with a mini-batch size of 50. We used stochastic gradient descent as the optimizer. We skipped the warmup learning rate as it did not show any improvement on the macro-averaged F1 score of the dev dataset. We observed the training to terminate in around the 15th to 20th epoch. For the EnsembleCRF model, we used Adam optimizer with a learning rate of 1e-3 and a weight decay of 0.01. We set the model to train for a maximum of 100 epochs with a mini-batch size of 126. The results of the proposed method for the English language are tabulated in Table 3.

Table 4: The macro-averaged F1 score for Spanish language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.8994
2	USTC-NELSLIP	0.8544
3	RACAI	0.7562
4	**Infrrd.ai**	**0.7526**
5	MaChAmp	0.752
6	Sliced	0.7511
7	YNUNLP	0.7317
8	brotherhood	0.7069
9	L3i	0.6893
10	PA Ph&Tech	0.6893
11	MarSan_AI	0.683
12	SPDB Innovation Lab	0.6731
13	CSECU-DSG	0.6562
14	EURECOM	0.6277
15	Multilinguals	0.612
16	Sartipi-Sedighin	0.607
17	BaselineExtending-Pokemons	0.6008
18	MultiCoNER Baseline	0.574
19	UC3M-PUCPR	0.5679

Table 5: The macro-averaged F1 score for Dutch language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.905
2	USTC-NELSLIP	0.8767
3	RACAI	0.7841
4	Sliced	0.7766
5	MaChAmp	0.7699
6	**Infrrd.ai**	**0.764**
7	YNUNLP	0.7582
8	brotherhood	0.7304
9	PA Ph&Tech	0.7205
10	MarSan_AI	0.7113
11	L3i	0.7096
12	CSECU-DSG	0.6794
13	EURECOM	0.667
14	BaselineExtending-Pokemons	0.6325
15	MultiCoNER Baseline	0.616
16	Sartipi-Sedighin	0.5837

5.2 Spanish

Since Spanish and English languages have a lexical similarity of about 30–50%, we tried translating the English dataset to Spanish and included it in the model training. Unfortunately, the translation experiment did not help in improving the performance of the model. The final model for the Spanish language included all the 4 techniques of token-level augmented data (Dai and Adel, 2020) along with train and dev datasets. We added up augmented datasets to our training pipeline to provide more exposure and to increase the diversity of available data. We used stochastic gradient descent as the op-

timizer and trained for about 20 epochs after which the performance turned out to be constant. The results on the test set are tabulated in Table 4.

5.3 Dutch

All the strategies with a language-specific embedding layer were used for experimentation namely roberta-base, roberta-large, and xlm-roberta-large models. The final model is an ensemble CRF model trained with 2 sequence labeling models, an xlm-roberta trained on the Dutch dataset and an xlm-roberta trained on the multilingual dataset. We trained the Dutch model on both the train and dev datasets provided by the competition organizers. A test set is created by splitting the (train and dev) dataset internally while training. The model was trained for 20 epochs. The generated predictions on the test set are propagated through an Ensemble-CRF model to ensure consistency in labeling and to improve the labeling accuracy. The results are tabulated in Table 5.

5.4 Korean

We trained 6 models using a combination of strategies as mentioned in Sections 3 and 4. The final model is an xlm-roberta-large followed by a BiLSTM and CRF while using the multilingual model as an initial checkpoint. The Korean model was trained with the training and dev set provided for 30 epochs with a learning rate of 1e-3 and a weight decay of 0.10. Monte Carlo Dropout (MCD) ensemble (Gal and Ghahramani, 2016) was also used for Korean as an ensemble strategy, 15 different inferences were done with varying architecture with a dropout of 0.3 and a majority voting strategy was used for the final submission. Upon analysis, the best performing model and the entire 15 model ensemble had similar performances. We could potentially cherry-pick models from all the 15 possible candidates to improve the scores but time being a limiting factor, it was dropped. The results are tabulated in Table 6.

Table 6: The macro-averaged F1 score for Korean language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.8859
2	USTC-NELSLIP	0.8636
3	RACAI	0.7174
4	CMB AI Lab	0.707
5	Sliced	0.7066
6	YNUNLP	0.7033
7	C-3PO	0.6749
8	UA-KO	0.6749
9	brotherhood	0.6741
10	**Infrrd.ai**	**0.6729**
11	MaChAmp	0.6545
12	EURECOM	0.6496
13	L3i	0.6268
14	MarSan_AI	0.6226
15	CSECU-DSG	0.6205
16	AaltoNLP	0.6182
17	BaselineExtending-Pokemons	0.5895
18	MultiCoNER Baseline	0.546

Table 7: The macro-averaged F1 score for German language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.9065
2	USTC-NELSLIP	0.8905
3	RACAI	0.7939
4	Sliced	0.789
5	MaChAmp	0.7838
6	YNUNLP	0.7732
7	L3i	0.7723
8	ML-HUB	0.7614
9	brotherhood	0.7594
10	**Infrrd.ai**	**0.759**
11	EURECOM	0.7443
12	MarSan_AI	0.7312
13	CSECU-DSG	0.7249
14	AaltoNLP	0.7137
15	PA Ph&Tech	0.6675
16	BaselineExtending-Pokemons	0.6659
17	MultiCoNER Baseline	0.634

5.5 German

The German language embedding layer was used with roberta-base, roberta-large, and xlm-roberta models, and all the strategies were evaluated. The submitted model is an ensemble CRF model trained with 2 sequence labeling models, an xlm-roberta trained on the German dataset and an xlm-roberta trained on the multilingual dataset. We trained the German model on both the train and dev datasets provided by the competition organizers. A test set is created by splitting the (train and dev) dataset internally while training. The model was trained for 20 epochs. The generated predictions on the test set are propagated through an EnsembleCRF model to ensure consistency in labeling and to improve the labeling accuracy. The results are tabulated in Table 7.

5.6 Chinese

The amount of work carried out on the Chinese dataset is limited due to time limitations. Most of the experiments were limited to model architecture

Table 8: The macro-averaged F1 score for Chinese language sentences.

Position	Team Name	Macro-averaged F1 score
1	USTC-NELSLIP	0.8169
2	CASIA	0.797
3	OPDAI	0.7954
4	DAMO-NLP	0.7806
5	NetEase.AI	0.7777
6	CMB AI Lab	0.7636
7	NCUEE-NLP	0.7418
8	QTrade AI	0.74
9	CSECU-DSG	0.6722
10	Multilinguals	0.6695
11	L3i	0.6691
12	Sliced	0.6521
13	**Infrrd.ai**	**0.6468**
14	MaChAmp	0.6381
15	EURECOM	0.634
16	RACAI	0.627
17	YNUNLP	0.6138
18	brotherhood	0.6086
19	MarSan_AI	0.5664
20	SPDB Innovation Lab	0.5574
21	BaselineExtending-Pokemons	0.528
22	MultiCoNER Baseline	0.511

searching and transfer learning. We tried various pre-trained Chinese language models that include pre-trained Chinese language models trained on other NER datasets. The best architecture observed is a Chinese BERT model followed by BiLSTM and CRF. Our final model was set to train for a maximum of 50 epochs with a mini-batch size of 24. We use AdamW optimizer (Loshchilov and Hutter, 2017) with a weight decay rate of 0.01. We also used Warmup Learning Rate Scheduler with 10% total training steps as a linear warmup period and rest steps with linear decay. The training terminates at the 24th epoch due to an early stopping mechanism. Our final model with the proposed method was trained for 24 epochs, and the results are tabulated in 8.

5.7 Hindi

We tried various Hindi language-based transformer-word-embeddings to include in the flair framework. Word-embeddings like hindi-bert from monsoon-nlp, multilingual-bert-cased, hindi-bert, and distil-bert of indic transformers were evaluated on the dev set. But none of these outperformed our proposed architecture model. We also tried adding language-specific embeddings which would usually help the model better understand the data. But this did not improve our baseline model performance. Hence, we did not include any additional language-specific embeddings. Our final model is xlm-roberta, which was trained using multilingual train and dev datasets. The model was trained for 30 epochs, and the results are tabulated in Table 9.

Table 9: The macro-averaged F1 score for Hindi language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.8623
2	USTC-NELSLIP	0.8464
3	RACAI	0.6808
4	Sliced	0.67
5	NetEase.AI	0.6663
6	**Infrrd.ai**	**0.6572**
7	brotherhood	0.6423
8	YNUNLP	0.6339
9	OPDAI	0.6294
10	MaChAmp	0.6173
11	CSECU-DSG	0.5768
12	MarSan_AI	0.5631
13	EURECOM	0.5278
14	silpa_nlp	0.5149
15	BaselineExtending-Pokemons	0.499
16	L3i	0.4973
17	Enigma	0.4862
18	MultiCoNER Baseline	0.469

Table 10: The macro-averaged F1 score for Bangla language sentences.

Position	Team Name	Macro-averaged F1 score
1	USTC-NELSLIP	0.8424
2	DAMO-NLP	0.8351
3	NetEase.AI	0.7088
4	RACAI	0.6628
5	**Infrrd.ai**	**0.6399**
6	YNUNLP	0.638
7	Sliced	0.6305
8	Team Atreides	0.5975
9	brotherhood	0.5863
10	MaChAmp	0.5646
11	MarSan_AI	0.5422
12	EURECOM	0.5257
13	AaltoNLP	0.5179
14	silpa_nlp	0.5139
15	CSECU-DSG	0.5055
16	BaselineExtending-Pokemons	0.4507
17	L3i	0.4481
18	Enigma	0.4268
19	MultiCoNER Baseline	0.391

5.8 Bangla

It was a challenging task to find good embeddings to represent the Bangla language. We performed a few experiments by including Bangla Indic transformer word embeddings in the flair framework. Similar to the Hindi language, even this embedding did not perform better than our proposed method. We also tried adding language-specific embeddings

Table 11: The macro-averaged F1 score for multiple language sentences.

Position	Team Name	Macro-averaged F1 score
1	DAMO-NLP	0.8531
2	USTC-NELSLIP	0.853
3	QTrade AI	0.7766
4	SeqL	0.7549
5	CMB AI Lab	0.7369
6	UM6P-CS	0.7249
7	RACAI	0.721
8	Cardiff NLP	0.7165
9	Sliced	0.7107
10	IIE_KDSEC	0.7089
11	BaselineExtending-Pokemons	0.7069
12	OPDAI	0.6948
13	brotherhood	0.6942
14	MarSan_AI	0.6928
15	**Infrrd.ai**	**0.6924**
16	HaveNoIdea	0.6879
17	EURECOM	0.6808
18	MaChAmp	0.6768
19	YNUNLP	0.6685
20	DSUG	0.6522
21	UPB	0.6473
22	CSECU-DSG	0.644
23	NSU-AI	0.6423
24	SPDB Innovation Lab	0.6322
25	L3i	0.6123
26	MultiCoNER Baseline	0.541
27	HaveNoIdea	0.5403

which would usually help the model better understand the data. But this did not improve our baseline model performance. Hence, we did not include any additional language-specific embeddings. Our final model is xlm-roberta, which was trained using multilingual train and dev datasets. The model was trained for 40 epochs, and the results are tabulated in Table 10.

5.9 Multilingual

The multilingual task was challenging in itself and our choice of framework made it even harder since we did not have a multi gpu support to conduct all the experiments. The experiments conducted are bucketed mainly into three parts, the architecture search, the data strategy, and the ensemble strategy. After experimenting with various architectures and embeddings, we resorted to xlm-roberta-large for the task. The data strategy was tricky considering all the languages. Open source datasets and translation APIs didn't provide improvements.

We decided to train a stable model and use it as an initial checkpoint for all the other languages. The model was trained for 30 epochs with a learning rate of 1e-3. The final model is xlm-roberta-large followed by a BiLSTM and CRF trained with the entire corpus of train and dev set. Various attempts were made to include the above-mentioned data augmentation techniques but due to the huge model and data along with limited time and resources, we could only do very limited experiments for this model. We tried with the MCD ensemble and took 15 inferences through the varying architectures and used the majority voting strategy to obtain the final submissions. The single best-performing model was at par with the MCD ensemble with majority voting. The results for multiple language sentences are tabulated in Table 11.

6 Conclusion

The recognition of entities from multiple languages with low resources is more complex. The problem lies with the ambiguous entities formed by newly coined words. The syntax of grammar in the sentences was not followed to capture the context between the words. We tried a single multilingual transformer approach, which didn't provide much-expected results. We had used gazetteers for all the languages sourced from the training and wiki data. Both of them didn't produce improvements over the model results.

We trained a multilingual transformer model and performed transfer learning to the individual languages. Since there were multiple languages in the task. We performed unique experiments on one language and then adapted it to the others based on the performance. In the discussion section, the approaches used for experiments are covered and vary for the individual languages. Overall, we created an ensemble of different models which resulted in improvements over the single model. The ensemble architecture covered different types of transformer-based language models. The results reached closer to the top positions with this approach. We also observed that the data augmentation used to improve the performance for a few languages and drop in the performance for the other languages. Our results are above 15% on an average in the participated sub-tasks over the MultiCoNER Baseline results.

We would like to explore the multilingual T5 transformer model, which couldn't be covered during the competition. We would like to explore different augmentation techniques with external data, which couldn't be completed due to time constraints.

References

Hugging face. https://huggingface.co/docs/transformers/index. Accessed: 2022-02-23.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

M Saiful Bari, Shafiq Joty, and Prathyusha Jwalapuram. 2020. Zero-resource cross-lingual named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7415–7423.

Aditi Chaudhary, Jiateng Xie, Zaid Sheikh, Graham Neubig, and Jaime G Carbonell. 2019. A little annotation does a lot of good: A study in bootstrapping low-resource named entity recognizers. *arXiv preprint arXiv:1908.08983*.

Xiang Chen, Ningyu Zhang, Lei Li, Xin Xie, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Lightner: A lightweight generative framework with prompt-guided attention for low-resource ner. *arXiv preprint arXiv:2109.00720*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

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.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Peter Izsak, Shira Guskin, and Moshe Wasserblat. 2019. Training compact models for low resource entity tagging using pre-trained language models. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 44–47. IEEE.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: a Large-scale Multilingual dataset for Complex Named Entity Recognition.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Mónica Marrero, Sonia Sánchez-Cuadrado, Jorge Morato Lara, and George Andreadakis. 2009. Evaluation of named entity extraction systems. *Advances in Computational Linguistics, Research in Computing Science*, 41:47–58.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512.

Tal Perl, Sriram Chaudhury, and Raja Giryes. 2020. Low resource sequence tagging using sentence reconstruction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2692–2698.

Debasis Samal. 2021. Named entity recognition dataset (ner). https://www.kaggle.com/debasisdotcom/name-entity-recognition-ner-dataset. Accessed: 2022-02-23.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003, Edmonton, Canada*, pages 142–147.

Houjin Yu, Xian-Ling Mao, Zewen Chi, Wei Wei, and Heyan Huang. 2020. A robust and domain-adaptive approach for low-resource named entity recognition. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 297–304. IEEE.