UM6P-CS at SemEval-2022 Task 11: Enhancing Multilingual and Code-Mixed Complex Named Entity Recognition via Pseudo Labels using Multilingual Transformer

Abdellah El Mekki[1] **Abdelkader El Mahdaouy**[1] **Mohammed Akallouch**[1],[2]
Ismail Berrada[1] **Ahmed Khoumsi** [3]
[1]School of Computer Sciences, Mohammed VI Polytechnic University, Morocco
[2]Faculty of Sciences Dhar EL Mahraz, Sidi Mohamed Ben Abdellah University, Morocco
[3]Dept. Electrical & Computer Engineering, University of Sherbrooke, Canada
{firstname.lastname}@um6p.ma

Abstract

Building real-world complex Named Entity Recognition (NER) systems is a challenging task. This is due to the complexity and ambiguity of named entities that appear in various contexts such as short input sentences, emerging entities, and complex entities. Besides, real-world queries are mostly malformed, as they can be code-mixed or multilingual, among other scenarios. In this paper, we introduce our submitted system to the Multilingual Complex Named Entity Recognition (MultiCoNER) shared task. We approach the complex NER for multilingual and code-mixed queries, by relying on the contextualized representation provided by the multilingual Transformer XLM-RoBERTa. In addition to the CRF-based token classification layer, we incorporate a span classification loss to recognize named entities spans. Furthermore, we use a self-training mechanism to generate weakly-annotated data from a large unlabeled dataset. Our proposed system is ranked 6th and 8th in the multilingual and code-mixed MultiCoNER's tracks respectively.

1 Introduction

Recent named entity recognition (NER) models have achieved great performance for many languages and using various benchmark datasets such as CoNLL2003 and OntoNotes 5.0 (Devlin et al., 2019). However, it is unclear whether or not these systems can handle ambiguous and complex entities, especially in the case of short and low-context settings (Augenstein et al., 2017). It is also unclear weather these systems can be deployed in real-world scenarios where the input data can be in different languages or code-mixed (Luken et al., 2018; Hanselowski et al., 2018). In fact, to illustrate these issues, if consider the example of complex named entities such as the titles of creative works (movies, songs, books ...), they are hard to be recognized by simple NER systems. This is due to their syntactic ambiguity and the form they can take from one context to another. For instance, they can be as an imperative clause ("Dial M for Murder") or a proposition ("On the beach") which refers to the name of a movie. Thus, it is important to check the performance of NER systems in these scenarios.

The complexity of named entities can be due to three main reasons:

1. **Complex entities:** These entities can be represented as complex infinitives (To Kill a Mockingbird) or full clauses (Mr.Smith Goes to Washington). Additionally, they can be represented as noun phrases or gerunds. State-of-the-art systems (Aguilar et al., 2017) have shown that it is hard to recognize such entities.

2. **Ambiguous entities and contexts:** These types of entities are context-dependent as they can refer to named entities in some contexts, but not in others. “Among Us” which refers to the name of a video game is an example of this challenge. This situation is even more challenging (Mayhew et al., 2019) in the case of short sentences with minimal context such as questions or search queries, which most of the time lack some features such as capitalization or punctuation.

3. **Emerging entities:** This challenge mimics the real-world scenario with many unseen entities, as new named entities are always appearing due to the release of new books, songs, or movies within a short period of time.

It is well known that the state-of-the-art performance reached by current NER systems is mainly due to the presence of easy entities and well-formed input texts (Augenstein et al., 2017). Nevertheless, they yield weak performance when applied in multilingual/code-switched input texts having

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1511 - 1517
July 14-15, 2022 ©2022 Association for Computational Linguistics

complex or unseen entities. It is also worth mentioning that the reached state-of-the-art results by current NER systems have been achieved using Transformers-based pre-trained language models (Devlin et al., 2019) that are known to encode the context of input texts and tokens.

In this paper, we introduce our participating system to the MultiCoNER shared task (Malmasi et al., 2022b), in particular, the complex MultiCoNER tracks of multilingual and code-switched queries. Our system relies on a deep learning model based on the multilingual Transformer-based pre-trained language model XLM-RoBERTa (Conneau et al., 2020). To handle the complexity of the two tracks, our model is trained using two optimization objectives, as well as for self-training. The main components of our system can be summarized as follow:

- **Optimization of entities span loss:** We train our model to recognize entities spans as an auxiliary task. The aim is to help the model detect the full entities expressions in an input sentence.

- **Incorporation of Conditional Random Field (CRF) (Lafferty et al., 2001) layer:** We incorporate a CRF layer on top of the Transformer representations to fully exploit the mutual information between tokens in an input sentence.

- **Self-training on unlabeled data:** we create a weakly-supervised data based on the model predictions on unlabeled data.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the dataset and the sub-tasks of SemEval-2022 Task 11 (Malmasi et al., 2022b). In section 4, we present our system overview. Section 5 summarizes and discusses the obtained results for both multilingual and code-mixed tracks. Finally, Section 5 concludes the paper.

2 Related Work

During the last years, neural network-based approaches have contributed to improving the performance of NER systems (Panchendrarajan and Amaresan, 2018; Devlin et al., 2019). This was mainly achieved thanks to word embeddings. Static word embeddings are fed to BiLSTM-CRF models and have helped eliminate manual feature engineering while achieving better performance. On the other hand, Transformer-based Language Models (Devlin et al., 2019; Conneau et al., 2020) have greatly improved the NER results thanks to their contextualized word representations.

However, these models may fail when recognizing new or complex entities (Luken et al., 2018; Hanselowski et al., 2018). These challenges are reflecting the real-world setting. Recently, Meng et al. (2021) have proposed an approach to tackle these challenges by using a contextual Gazetteer Representation encoder which can be fused with word-level models. The results have shown that this method enhances the F1-score by +49% in the uncased setting. This work has been mainly applied to the English language. Finally, in another work, Fetahu et al. (2021) have explored the code-mixed NER scenario using multilingual Transformers. They have combined the Mixture-of-Experts model with existing multilingual Transformers models to incorporate the multi-lingual gazetteers. The experiments have demonstrated that their proposed approach enhances the F1-score by +31% in the Code-Mixed NER over the baseline model.

3 Data

The dataset of the MultiCoNER (Multilingual Complex Named Entity Recognition) shared task (Malmasi et al., 2022a) is provided to tackle thirteen different tracks, eleven tracks cover the monolingual cases, while the remaining two tracks cover code-mixed and multilingual data. In this paper, we focus on the last two tracks.

The multilingual track covers eleven languages (English, Spanish, Dutch, Russian, Turkish, Korean, Farsi, German, Chinese, Hindi, and Bangla), while the code-mixed track covers a subset of these languages. In both tracks, the entities are annotated into six types: PER, LOC, GRP, CORP, PROD, and CW. Table 1 presents the size of the train and test datasets provided for these tracks. We notice that the multilingual track has more data than the code-mixed track, and the test datasets are larger than the train datasets for both tracks.

The provided datasets are labeled using the IOB format which is used for sequence labeling tasks. Table 2 presents the distribution of the entities and the spans in the train datasets for both tracks.

	Multilingual	Code-mixed
Train	168300	1500
Test	471911	100000

Table 1: The size of the train and the test datasets for the multilingual and code-mixed tracks.

4 Methodology

Before discussing the methodology of our proposed system, we assume that we have n labeled sentences with named-entities $D = \{(x_i^t, y_i^t)\}_1^n$ and m unlabeled sentences $U = \{(x_i^u)\}_1^m$, where $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,l}\}$ represents a sentence containing l tokens, while $y_i = \{y_{i,1}, y_{i,2}, ..., y_{i,l}\}$ are the l labels corresponding to these l tokens.

The proposed system incorporates 4 components used on top of the pre-trained Transformer encoder. In the following, we describe each component of our system.

4.1 Multilingual Transformer encoder

To encode each word in the input sentence, we use the XLM-RoBERTa (XLM-R) (Conneau et al., 2020). It is a multilingual pre-trained transformer encoder network. We choose to use this encoder for the following reasons: 1) XLM-R is the state-of-the-art encoder in the multilingual and code-mixed settings, and 2) It has been trained on 100 languages including the languages covered in the multilingual and code-mixed tracks. This ensures a good contextualized representation for the input sentences despite their language.

This model was mainly trained using the Mask Language Modeling (Devlin et al., 2019). For an input sentence, 15% of the words are randomly masked, then the model tries to predict the masked words. As a result of this training process, the model learns the representations of dimension d for the input words of 100 languages that can be fine-tunned on a downstream tasks such as sequence classification or sequence labeling.

4.2 Span classification module

Span classification is a span-wise classifier, where the aim is to classify whether or not, a sequence of tokens are representing a named entity span based on their semantics. It is a binary classification task as the model predicts **1** if the span is a named entity while it predicts **0** if not.

Given $H = [h_1, h_2, ..., h_k]$, the vector representations of the k sub-tokens contained in a span S, to learn a representation that encodes all the span tokens, we follow the same approach used in (Essefar et al., 2021; El Mekki et al., 2021b; El Mahdaouy et al., 2021), where an attention layer (Bahdanau et al., 2015) learns the span representations SH based on its tokens, as follows:

$$C = tanh(HW^a)$$

$$\alpha = softmax(C^T W^\alpha)$$

$$SH = \alpha \cdot H^T$$

where $W^a \in \mathbb{R}^{d\times 1}$, $W^\alpha \in \mathbb{R}^{k\times k}$ are the trainable parameters of the attention layer, $C \in \mathbb{R}^{k\times 1}$ and $\alpha \in [0,1]^k$ weights the word representations according to their importance for the task at hand.

The vector representations SH of all the spans in an input sentence are then fed to a feed-forward neural network which classifies whether or not, the input representation refers to a named-entity span.

As illustrated in table 2, we notice that the majority of spans do not represent a named entity. To tackle this imbalanced data issue, we follow (Li et al., 2020) in using the Focal Loss (FL) (Lin et al., 2017). The FL is given by:

$$FL(y, \hat{p}) = -\alpha_y \left(1 - \hat{p}_y\right)^\gamma \log(\hat{p}_y) \qquad (1)$$

where, $y \in \{0, 1\}$ denotes the span's label, $\hat{p} = (\hat{p}_0, \hat{p}_1)$ is a vector representing the predicted probability distribution over the labels, α_y is the weight of label y, and γ controls the contribution of high-confidence predictions in the loss. The performed experiments showed that $\gamma = 0.5$ gives the best result.

As the provided dataset in this shared task has been annotated based on named entities, we adjust it for the span classification task. Therefore, we label all the named entities spans as **1** while the rest of spans has been labeled as **0**.

4.3 NER classification using CRF-Layer

Most NER systems using Transformers rely on using the first sub-token of each word as input to the classification layer (Devlin et al., 2019). In our system, we follow the work of (Ács et al., 2021) in using a pooling of the sub-tokens of each word in the input sentence.

Given $H = [h_1, h_2, ..., h_p]$, the vector representations of the all the sub-tokens contained in a word

		Classification tasks							
		Named entity						**Entity Span**	
	Class	**LOC**	**PER**	**PROD**	**GRP**	**CORP**	**CW**	**Entity Span**	**Not Entity Span**
Mutilingual	%	22.02	18.76	15.0	14.0	14.11	16.12	38.82	61.17
Code-Mixed	%	18.27	16.63	17.88	13.9	16.63	16.69	41.71	58.28

Table 2: The distribution of named entities and entities spans in the train dataset for the multilingual and code-mixed tracks

w, an attention layer learns the word pooled representations based on its sub-tokens following the same method explained in section 4.2.

The pooled representations are then fed to the classification layer which is a Conditional Random Field (CRF) layer in our system. We opt for CRF mainly because the Softmax layer does not take into consideration the dependencies between tokens. The self-attention mechanism, performed by the Transformer encoder, encodes these dependencies in the output vectors of the input sentence. While the CRF which is common in sequence labeling tasks ensures output consistency, it transforms the sequence of input word representations to a sequence of probability distributions, therefore, each label prediction depends on the other predictions in the same input sentence.

4.4 Self-training

To take profit from the provided unlabeled dataset in this shared task, we generate a weakly-annotated dataset and re-train the developed model on it. This method has been applied differently in several works (Khalifa et al., 2021; El Mekki et al., 2021a; Huang et al., 2021). In our work, we apply the following pipeline:

1. We train a model M (based on span classification and NER-CRF explained in the previous subsections) using the provided labeled data D.

2. We use the trained model M to predict the labels of the provided unlabeled data. Then we build a weakly-annotated data U

3. We concatenate the datasets T and U and re-train the sequence labeling model.

It is worth mentioning that during the self-training phase, we remove the span classification module.

5 Experiments and results

5.1 Experimental setup

We use the PyTorch framework and the Transformers libraries for the implementation of our proposed system. The training of the model is performed on a server with a single Nvidia Tesla P100 with 16GB of RAM. XLM-RoBERTa Large is used as our multilingual Transformer encoder. Adam optimizer with a learning rate of $1e-5$ is used for all experiments. The system is trained with a batch size of 16 and for 20 epochs.

For the multilingual track, we train our model on the provided labeled multilingual data, then we use the best epoch's model to leverage pseudo-labels from the unlabeled test data, and we re-train the model again from scratch. Besides, for the code-mixed track, we combine the provided data with the training data of the multilingual track and we follow the same training pipeline of the multilingual track.

5.2 Results

Table 3 presents the submitted results for the multilingual and code-mixed tracks using our system. The first row in the table presents the baseline results on the test set published by the shared task organizers. The performance achieved by our best submission largely outperforms the baseline results in both tracks. In fact, the performance using our model boosts the baseline score by 18.39 and 21.11 F1-score points in the multilingual track and the code-mixed track, respectively. The table also reports the performance of our system during the three stages (ablation study):

- The BERT-CRF model that incorporates the named-entity recognition classification layer with CRF,
- The span classification objective, and
- The self-training.

	Multilingual track			Code-mixed track		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Baseline	-	-	54.10	-	-	58.10
BERT-CRF	68.59	69.30	68.00	78.47	78.92	78.78
+ Span Classification	70.71	70.56	70.25	78.30	78.77	78.52
+ Self-training	**73.21**	**72.51**	**72.49**	**79.38**	**79.08**	**79.21**

Table 3: Official Complex NER F1-scores on the multilingual and code-mixed tracks using the proposed system.

	Multilingual track			Code-mixed track		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
LOC	72.24	81.41	76.01	80.37	83.04	81.68
PER	85.20	81.40	83.13	88.02	88.67	88.35
PROD	70.54	70.24	70.00	82.68	80.51	81.58
GRP	69.78	63.6	66.26	70.85	73.02	71.92
CW	67.58	68.74	67.95	75.00	74.27	74.63
CORP	73.91	69.68	71.60	79.36	74.99	77.11

Table 4: Official Complex NER F1-scores per entity on the multilingual and code-mixed tracks using the proposed system.

The results show that the span classification stage enhances the performance of both tracks: the F1-scores achieved using the span classification are 70.25% for multilingual track and 78.52 % for the code-mixed track. However, we notice that the span classification has significantly boosted the F1-score compared to the BERT-CRF model for the multilingual track, while there is a small performance loss in the case of the code-mixed track. When performing self-training on the predictions extracted from the model using the span classification stage, a large gain has been achieved in both tracks. For the multilingual track, the F1-score obtained using self-training is 72.49% with a gain of 3.18% compared to the system without self-training. For the code-mixed track, our system has achieved the F1-score of 79.21% with a gain of 0.87% compared to the system without self-training.

Finally, Table 4 presents the performance of our best submission for the multilingual and code-mixed tracks. The proposed system fails the most in predicting the **GRP** entities for both tracks, meanwhile, it gives its best performance when predicting the **PER** entities.

6 Conclusion

In this paper, we present our Named Entity Recognition (NER) system for complex scenarios on multilingual and code-mixed queries. Our system relies on 4 components: a multilingual transformer encoder, an entity span classification module, a CRF-layer, and a self-training mechanism that leverages information from unlabeled data. We use our system to submit our predictions in the SemEval-2022 Task 11 within the multilingual and code-mixed tracks. The results show that the use of multilingual Transformer and self-training enhances the results in both multilingual and code-mixed cases. Moreover, the incorporation of the span classification module and the CRF layer allow better recognition of complex named entities.

References

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153, Copenhagen, Denmark. Association for Computational Linguistics.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech Language*, 44:61–83.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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

Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 334–339, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Abdellah El Mekki, Abdelkader El Mahdaouy, Ismail Berrada, and Ahmed Khoumsi. 2021a. Domain adaptation for Arabic cross-domain and cross-dialect sentiment analysis from contextualized word embedding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2824–2837, Online. Association for Computational Linguistics.

Abdellah El Mekki, Abdelkader El Mahdaouy, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021b. BERT-based multi-task model for country and province level MSA and dialectal Arabic identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 271–275, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Kabil Essefar, Abdellah El Mekki, Abdelkader El Mahdaouy, Nabil El Mamoun, and Ismail Berrada. 2021. CS-UM6P at SemEval-2021 task 7: Deep multi-task learning model for detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1135–1140, Online. Association for Computational Linguistics.

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.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

Kaiyu Huang, Junpeng Liu, Degen Huang, Deyi Xiong, Zhuang Liu, and Jinsong Su. 2021. Enhancing Chinese word segmentation via pseudo labels for practicability. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4369–4381, Online. Association for Computational Linguistics.

Muhammad Khalifa, Muhammad Abdul-Mageed, and Khaled Shaalan. 2021. Self-training pre-trained language models for zero- and few-shot multi-dialectal Arabic sequence labeling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 769–782, Online. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Los Alamitos, CA, USA. IEEE Computer Society.

Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. QED: A fact verification system for the FEVER shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 156–160, Brussels, Belgium. Association for Computational Linguistics.

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.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6256–6261, Hong Kong, China. Association for Computational Linguistics.

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.

Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.