Multilingual Fine-Grained Named Entity Recognition

Viorica-Camelia Lupancu, Adrian Iftene

Abstract

The "MultiCoNER II Multilingual Complex Named Entity Recognition" task[1] within SemEval 2023 competition focuses on identifying complex named entities (NEs), such as the titles of creative works (e.g., songs, books, movies), people with different titles (e.g., politicians, scientists, artists, athletes), different categories of products (e.g., food, drinks, clothing), and so on, in several languages. In the context of SemEval, our team, *FII_Better*, presented an exploration of a base transformer model's capabilities regarding the task, focused more specifically on five languages (English, Spanish, Swedish, German, and Italian). We took DistilBERT (a distilled version of BERT) and BERT (Bidirectional Encoder Representations from Transformers) as two examples of basic transformer models, using DistilBERT as a baseline and BERT as the platform to create an improved model. In this process, we managed to get fair results in the chosen languages. We have managed to get moderate results in the English track (we ranked 17th out of 34), while our results in the other tracks could be further improved in the future (overall third to last).

MSC 2020: 68T50.

1 Introduction

Named entity recognition (NER) involves identifying and classifying significant tokens (words) within a given text [1]–[3]. For instance, in news articles, identifying the names of individuals, organizations, and places is often essential. The highlighted named entities in the

doi:10.56415/csjm.v31.16

[1] https://multiconer.github.io

following example contain valuable information and can be utilized in natural language processing (NLP) applications:

Last month ***Sky West*** *moved to her husband's hometown in* ***West Virginia****.*

such as information extraction [4], [5], question answering, text summarization, machine translation, and semantic web search, which heavily rely on NER. Named entity recognition allows for the identification of named entities such as Sky West, which is particularly useful in machine translation as it prevents erroneous word-by-word translations. It is impressive that state-of-the-art NER systems rely heavily on hand-crafted features and domain-specific knowledge [6], [7]. Over the past few decades, the scope of named entity recognition has undergone significant evolution. Initially, NER was limited to the extraction of proper nouns from news-related content, such as names of people, organizations, and locations. However, with the expansion of NLP into other domains, these traditional named entity classes proved to be insufficient. For instance, articles about science or technology require additional named entity classes beyond the original three. Additionally, it's worth noting that named entities are not limited to proper nouns. In certain fields of study, like medicine, terms such as pneumonia, common cold, or cholesterol could also be considered named entities.

The MultiCoNER II shared task [8] aims at building NER systems for 12 languages, namely English, Spanish, Hindi, Bangla, Chinese, Swedish, Farsi, French, Italian, Portuguese, Ukrainian, and German. The task has 12 monolingual tracks and a multilingual one. The dataset contains sentences from the wiki domain, which are usually short and low-context sentences [9]. Moreover, these short sentences usually contain semantically ambiguous and complex entities, which makes the problem more difficult. Usually, retrieving knowledge related to such ambiguous concepts in any form is a definite method of understanding and disambiguating them. Thus, the ideal NER model would be capable of taking on hard samples if the option of additional context information was available. The rest of the paper is organized as follows: Section 2 briefly presents studies related to NER, either in a multi-

lingual context or not; Section 3 presents the dataset, the required pre-processing, and plausible methods for it; Section 4 resumes the results of the conducted experiments, with their interpretations, followed by Section 5 with the conclusions.

2 Related Work

There is a limited amount of research focused on identifying entity types beyond the conventional ones (persons, locations, organizations). Complex NEs, like chemicals, ingredients, diseases, or active substances are not straightforward nouns and pose greater challenges in terms of identification [10]. They have the ability to manifest as various linguistic constituents and have a very different surface from the traditional NEs. Their ambiguity makes it challenging to recognize them. Additionally, nowadays an increasing number of individuals are sharing information online on diverse topics, highlighting the growing significance of NER for these unconventional entities, given the data collected from social media, where people openly express their interests [11], [12]. Efforts have been made to explore the capacity of contemporary NER systems to demonstrate effective generalization across diverse genres. This attempt also found out, as expected, that a notable correlation exists between the size of the training corpus and the performance of NER systems, so by having a bigger corpus, the results may be more accurate [13]. The job of handling NEs by extracting them from the text has been done by transformers. In the last few years, new technologies have appeared, including a Google research releasing mT5, their own version of a transformer, which outperforms the previously released multilingual transformers [14]. Among those, BERT is one of the most powerful unsupervised models. A multilingual variant of it, trained in over 100 languages and enhanced with context awareness thanks to a CRF layer on top, has been leveraged before for such a task with promising results [15].

The "Multilingual Complex Named Entity Recognition (MultiCoNER)" task[2] was first introduced in the context of the SemEval

[2] `https://multiconer.github.io/multiconer_1/`

2022 competition [16]. This task was divided into 13 tracks and aimed to explore techniques for recognizing complex NEs, such as titles of creative works (movies, books, songs, etc.), products, and groups. It was conducted across 11 different languages (Bangla, German, English, Spanish, Farsi, Hindi, Korean, Dutch, Russian, Turkish, and Chinese), considering both monolingual and multi-lingual scenarios. The dataset used for this task is the MultiCoNER dataset. It contains 2.3 million samples and includes data from three domains: Wikipedia sentences, questions, and search queries, along with those 11 monolingual subsets, the multilingual and code-mixed subsets. The multilingual subset consists of randomly chosen instances from all 11 languages blended together to form a unified subset. On the other hand, the code-mixed subset holds code-mixed samples, where the tagged entities originate from one language while the remaining text inside the instance is written in a different one. The dataset defines a six-class NER tagset, as follows: **LOC** – location or physical facilities; **CW** – titles of creative works such as movies, songs, and book titles; **CORP** – corporations and businesses; **GRP** – all other groups; **PER** – people names; **PROD** – consumer products.

In last year's MultiCoNER shared task, the two winning systems employed different strategies. [17] used a large-scale retrieval approach to gather relevant paragraphs related to the target sentence, which were concatenated and used as input to a transformer-CRF system. The aim was to build a multilingual knowledge base relying on Wikipedia. That knowledge base served the purpose of offering relevant contextual information to enhance the performance of the NER model. On the other hand, [18] employed a gazetteer-augmented BiLSTM model in conjunction with a transformer model to classify target sentences. The BiLSTM was pre-trained to generate token embeddings similar to the accompanying transformer, using sequence labels based on gazetteer matches.

In the context of the SemEval 2023 competition, we decided to focus on implementing a transfer learning approach for the BERT transformer. The concept of transfer learning involves using a pre-trained large neural network in an unsupervised manner, which next is fine-tuned for a specific task. In our case, BERT is the neural net-

work pre-trained on two tasks: masked language modeling and next-sentence prediction. Therefore, we fine-tuned this network on the NER dataset provided by the organizers. The proposed implementation uses Python programming language and is based on `Transformers` package, which is backed by the three most popular deep learning libraries – `Jax`, `PyTorch`, and `TensorFlow` – with a seamless integration between them. From `Transformers` library we made use of `BertForTokenClassification`, which is a model that has BERT as its base architecture, with a token classification head on top (a linear layer on top of the hidden-states output), allowing it to make predictions at the token level, rather than the sequence level. Named entity recognition is typically treated as a token classification problem, that's why we chose to use it.

3 Dataset and Methods

Although we explored a few options, we opted for the BERT transformer model for our approach. In this section, we present statistics from the dataset, as well as the steps we went through before choosing the BERT model and using the data for training.

3.1 Dataset

The dataset that we are using, MultiCoNER v2, is a large multilingual dataset (2.2 million unique instances and 26 million tokens) used for NER, that includes filtered data from public resources, Wikipedia, specifically focusing on difficult low-context sentences across 12 languages and multilingual subset. Additionally, the data underwent further post-processing to enhance its quality. A snippet with annotated entities from the dataset can be seen in Figure 1 below.

The 12 languages are part of a variety of languages with diverse typologies and writing systems, including both well-resourced languages like English and low-resourced languages like Farsi. There is a separate subset for each of the 12 languages and a multilingual subset (see Table 1), which consists of randomly collected instances from all the languages combined. From each language's test subset, a maximum

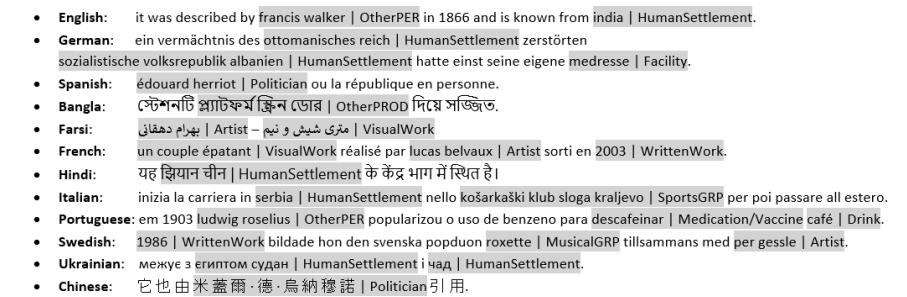

Figure 1. Examples from all the languages existing in MultiCoNER II

of 35,000 instances were randomly selected, resulting in a total of 358, 668 instances in the multilingual test subset. MultiCoNER II expanded on the challenges of MultiCoNER by adding fine-grained NER classes and the inclusion of noisy input. The dataset defines the following NER tagset with the 33 fine-grained classes which are listed into the 6 coarse types: **Location (LOC)** – Facility, OtherLOC, HumanSettlement, Station; **Creative Work (CW)** – VisualWork, MusicalWork, WrittenWork, ArtWork, Software; **Group (GRP)** – MusicalGRP, PublicCORP, PrivateCORP, AerospaceManufacturer, SportsGRP, CarManufacturer, ORG; **Person (PER)** – Scientist, Artist, Athlete, Politician, Cleric, SportsManager, OtherPER; **Product (PROD)** – Clothing, Vehicle, Food, Drink, OtherPROD; **Medical (MED)** – Medication/-Vaccine, MedicalProcedure, AnatomicalStructure, Symptom, Disease.

The fine-grained tagset facilitates the incorporation of various types of entities, including complex entity structures like Creative Work, as well as entities that require contextual information for disambiguation, such as Scientists and Athletes within the PER coarse-grained class.

3.1.1 Pre-processing

We have concatenated the training data from all of the languages into a single CONLL file. Then, to make reading and processing easier, we have converted the data into CSV format. At this step, we took note of the number of 2, 671, 439 total entries (tokens), spread between 67 fine-grained labels that are in the BIO scheme, which stands for Beginning-Inside-Outside. Each tag indicates whether the corresponding word is

Table 1. MultiCoNER II dataset statistics

Language	Train	Dev	Test
BN-Bangla	9,708	507	19,859
DE-German	9,785	512	20,145
EN-English	16,778	871	249,980
ES-Spanish	16,453	854	246,900
FA-Farsi	16,321	855	219,168
FR-French	16,548	857	249,786
HI-Hindi	9,632	514	18,399
IT-Italian	16,579	858	247,881
PT-Portuguese	16,469	854	229,490
SV-Swedish	16,363	856	231,190
UK-Ukrainian	16,429	851	238,296
ZH-Chinese	9,759	506	20,265
MULTI-Multilingual	170,824	8,895	358,668
Total	341,648	17,790	2,350,027

inside, outside, or at the beginning of a specific named entity. This scheme is used because named entities usually comprise more than one word. Finally, we have grouped the entries by sentence number and have used this format of the data going forward with the training. This dataset had a final size equal to $166,413$ in unique instances or better-said sentences.

3.1.2 Preparation

Having processed our dataset, it was now time to prepare it for training. We started by having two maps ready: `labels_to_ids` which would associate each unique NE tag a unique number (having 67 total tags, we simply numbered them from 0 to 66) and `ids_to_labels` being the reverse map of the first one. Then, for each pair *(sentence, labels)* in the dataset, we encoded the sentence's words using a tokenizer with a padding of 128 and converted the labels to their numeric form using our first mapping. The encoded words are then converted into tensors and each of them will be associated with the numeric labels which, similarly,

are also converted into tensors. The padding values, as well as word pieces that are not in the first part of the word after tokenization, are attributed a custom value of −100. Considering the final transformed model, we ended up using the *bert-base-uncased* tokenizer. The training set was turned into a `DataLoader` instance (from `PyTorch`), and at this point, it was ready to be used.

3.2 Methods

With a dataset of this size, we have run into difficulties trying to emulate the recommended baseline results with our resources, as such we opted to try out different pre-trained transformer models of small size to test which one would have the potential to be scalable within our limitations. Among the most popular and lightweight ones, we have decided to develop a model of our own based on the DistilBERT transformer. Using it as a base, we have created a baseline model for English that has been fine-tuned on the EN training data and obtained decent enough results to begin building upon it. The results of this baseline model are shown in Table 2. For the training parameters, we have used a learning rate of $1e-2$, a batch size of 32, several epochs of 8, and a `SGD` (Stochastic Gradient Descent) optimizer.

Table 2. Initial fine-tuned DistilBERT weighted results

Precision	**Recall**	**F1**	**Accuracy**
0.61	0.59	0.57	0.89

With this experience, we went ahead and looked into what the BERT transformer would be capable of, by comparison. We have used the `bert-base-uncased` transformer model as a start and began transfer learning, this time, using the entire collection of training data for all of the languages. We were very pleased with the initial results of the model (see Table 3). This initial run used a learning rate of $1e-05$, a training batch size of 4, and a validation batch size of 2, just 1 epoch and the `Adam` optimizer.

Further testing used the same hyperparameters, with the only difference being the number of epochs we trained the model for. Thus,

Table 3. Initial fine-tuned BERT weighted results

Precision	Recall	F1	Accuracy
0.91	0.90	0.91	0.90

the best model we managed to train in the competition is a multilingual one, trained on all 12 training subsets, using a learning rate of $1e-05$, a training batch size of 4, and a validation batch size of 2, 3 epochs and the `Adam` optimizer. For this model, we gained a training accuracy of 0.9125 and a validation accuracy of 0.9015. The training process finished in over 2 hours. Additional research made after the competition, along with further experiments regarding the dataset and hyperparameters, as well as the improved results, can be found in the "Analysis" subsection of Section 4.

4 Results

4.1 Analysis

For the practice phase of the competition, we have submitted for each track a file that contains only the predicted tag for every token. Two scores are noteworthy, one regarding each token to its predicted tag (see Table 4) and another one regarding the predicted tag being in the correct tagset (see Table 5). We can observe that, compared to the prediction of fine-grained tags for each individual in all of the languages, the coarse-grained tagset has increased scores. This indicates that while the exact tag may not be predicted, another tag within the same tagset is successfully predicted. Analyzing the macro-averaged F1 score from Table 4, we can notice that it is below 45 for languages such as Bangla, Farsi, Hindi, Ukrainian, and Chinese, which have diverse typology and writing systems, along with a smaller number of training instances and, therefore, the lower results.

As we can notice from Table 1, for some of the languages, the number of training instances is less than 16k (as most of them have). After the evaluation phase of the competition ended, the labeled test dataset was available, so to balance the dataset, instances from the test

Table 4. Macro-averaged results of practice phase for predicted **fine-grained tagset**, using the model trained on initial dataset, with 3 epochs

Lang.	Prec.	Recall	F1
EN	67.70	62.97	64.42
BN	61.78	34.49	40.17
DE	63.26	57.41	58.78
ES	66.26	60.26	62.67
FA	37.39	25.21	28.16
FR	66.92	61.05	62.54
HI	44.75	23.62	29.28
IT	68.18	63.22	64.94
PT	68.07	58.01	61.87
SV	62.94	54.69	57.30
UK	56.85	39.63	44.75
ZH	25.72	9.59	12.95
Multi	57.49	45.85	48.99

Table 5. Macro-averaged results of practice phase for predicted **coarse-grained tagset**, using the model trained on initial dataset, with 3 epochs

Lang.	Prec.	Recall	F1
EN	79.76	77.84	78.69
BN	66.96	42.57	49.84
DE	73.39	69.45	71.03
ES	74.83	70.34	72.36
FA	55.92	36.76	42.18
FR	75.47	73.02	74.19
HI	56.58	29.34	36.94
IT	78.18	74.13	76.05
PT	74.95	68.27	71.35
SV	76.16	64.65	69.28
UK	75.00	50.83	58.28
ZH	40.54	14.64	19.58
Multi	68.98	55.99	59.98

dataset were added to the training one, and therefore, new changes appeared in training and testing files for Bangla, German, Hindi, and Chinese languages (see Table 6). A new model was trained using the balanced dataset and the same values for all hyperparameters as before. Indeed, the training accuracy increased from 0.9125 to 0.9137, but not with a noticeable impact. Next, we trained another model using the 16k dataset, but this time increasing the number of epochs, from 3 to 5. After 3 hours of training, we found the new accuracy value: 0.9342. This showed a visible impact and a new question was raised:

What happens if we further increase the number of instances, for all the languages this time?

Analyzing the new form of the dataset, we came to the conclusion that we could increase the number of training instances for each language to approximately 25k. This led to another form of Multi-CoNER II dataset, which can be seen in Table 7. Having the second change inside the dataset, a new training process was started. For the number of epochs, we kept the same value, i.e., 5, because we clearly

Table 6. Dataset statistics after the first change (**the 16k dataset**)

Language	**Train**	**Dev**	**Test**
BN-Bangla	9,708+6,5=16,208	507	19,859-6,5=13,359
DE-German	9,785+6,6=16,385	512	20,145-6,6=13,545
EN-English	16,778	871	249,980
ES-Spanish	16,453	854	246,900
FA-Farsi	16,321	855	219,168
FR-French	16,548	857	249,786
HI-Hindi	9,632+6,5=16,132	514	18,399-6,5=11,899
IT-Italian	16,579	858	247,881
PT-Portuguese	16,469	854	229,490
SV-Swedish	16,363	856	231,190
UK-Ukrainian	16,429	851	238,296
ZH-Chinese	9,759+6,6=16,359	506	20,265-6,6=13,665
Total	197,024	8,895	1,965,159

noticed an improvement in the previous model. To train this model, approximately 5 hours were needed, and the value of training accuracy increased to 0.9433.

Further, we started to increase the number of epochs from 5 to 7 and train the 25k dataset again. This time, the training process lasted almost 18 hours, finishing with an accuracy of 0.9562, the best so far. Increasing the number of epochs from 7 to 8 and the batch size from 4 to 8, the training time decreased by almost 2 hours (due to the increase in batch size), but the accuracy obtained during training did not increase significantly, having a value of 0.9578. Keeping the number of epochs the same (i.e., 8), but increasing the number of batches from 8 to 64, a new model was trained in 13 hours, but unfortunately, the accuracy during training dropped to 0.9334.

Having a better-trained model, the next step was to generate prediction files for the development (dev) dataset. After obtaining predictions for all languages, they were submitted to the right track on CodaLab. The new results for the fine-grained tagset can be seen in Table 8, and the ones for the coarse-grained tagset are displayed in

Table 7. Dataset statistics after the second change (**the 25k dataset**)

Language	**Train**	**Dev**	**Test**
BN-Bangla	16,208+9=25,208	507	13,359-9=4,359
DE-German	16,385+9=25,385	512	13,545-9=4,545
EN-English	16,778+9=25,778	871	249,980-9=240,980
ES-Spanish	16,453+9=25,453	854	246,900-9=237,900
FA-Farsi	16,321+9=25,321	855	219,168-9=210,168
FR-French	16,548+9=25,548	857	249,786-9=240,786
HI-Hindi	16,132+9=25,132	514	11,899-9=2,899
IT-Italian	16,579+9=25,579	858	247,881-9=238,881
PT-Portuguese	16,469+9=25,469	854	229,490-9=220,490
SV-Swedish	16,363+9=25,363	856	231,190-9=222,190
UK-Ukrainian	16,429+9=25,429	851	238,296-9=229,296
ZH-Chinese	16,359+9=25,359	506	13,665-9=4,665
Total	305,024	8,895	1,857,159

Table 8. Macro-averaged results of dev files for predicted **fine-grained tagset**, using the model trained on 25k dataset, with 7 epochs

Lang.	**Prec.**	**Recall**	**F1**
EN	69.55	68.20	68.26
BN	77.14	69.88	72.56
DE	73.96	71.02	71.72
ES	71.12	67.56	68.48
FA	51.64	40.62	43.55
FR	71.83	66.66	68.29
HI	72.58	57.97	63.47
IT	69.31	67.89	68.22
PT	70.09	66.39	67.57
SV	69.33	66.38	66.92
UK	67.46	52.55	57.04
ZH	38.11	16.75	22.16
Multi	66.82	59.29	61.49

Table 9. Macro-averaged results of dev files for predicted **coarse-grained tagset**, using the model trained on 25k dataset, with 7 epochs

Lang.	**Prec.**	**Recall**	**F1**
EN	79.19	80.35	79.68
BN	79.60	75.89	77.48
DE	81.66	80.17	80.78
ES	77.90	75.73	76.79
FA	61.24	52.35	55.79
FR	79.31	76.79	77.98
HI	74.02	60.27	65.93
IT	77.03	76.42	76.69
PT	74.89	74.53	74.67
SV	79.46	74.77	76.87
UK	75.13	63.20	67.91
ZH	53.75	24.00	31.50
Multi	74.41	67.86	70.15

Table 9. Comparing the new results with the previous ones, we can see that for Bangla and Hindi (two of the four imbalanced datasets), the macro-averaged F1 score (the score according to which the organizers evaluated and ranked the systems) obtained for the fine-grained tagset increased by more than 30 points. Next, considerable changes appeared for German, Farsi, Ukrainian, and Multilingual datasets, with an increase in macro-averaged F1 score between 12 and 15 points. For English, Spanish, French, Italian, Portuguese, Swedish, and Chinese, there were improvements, but smaller, of 3, 5, or 9 points in the macro-averaged F1 score.

In conclusion, we can confirm that increasing the training dataset really helps to improve the model and implicitly to obtain better results. Besides this, another important factor is the number of times that the learning algorithm works through the entire training dataset (the number of epochs), which in this case led to visible improvements in results.

4.2 Evaluation

We were able to get results for all languages in the practice phase, however, simulated errors were added in the test dataset (in 30% of the set for the following languages: English, Spanish, French, Italian, Portuguese, Swedish, and Chinese), in the evaluation phase, and our model could not handle them properly. Character-level corruption strategies were enforced for Chinese, where characters were replaced with visually or phonetically similar ones. Token-level corruption strategies, on the other hand, were devised for other languages, focusing on common typing mistakes made by humans. This involved randomly substituting a letter with a neighboring letter on the keyboard, taking into account the specific keyboard layouts of each language.

On a small scale (2-3 characters), we were able to deal with those problematic characters, but in languages that we were not familiar with, we had difficulty detecting them. Similarly to the practice phase, Tables 10 and 11 are the results we have achieved with our model during the evaluation phase for the languages where we could successfully handle the input.

Table 10. Macro-averaged results of evaluation phase for predicted **fine-grained tagset**, using the model trained on initial dataset, with 3 epochs

Lang.	Prec.	Recall	F1	Ranking	F1 (winning team)
EN	63.76	60.62	61.75	17/34	85.53
DE	57.11	55.92	55.86	13/17	88.09
ES	57.25	53.17	54.51	16/18	89.78
IT	58.85	55.99	56.36	14/15	89.79
SV	55.88	51.59	52.12	15/16	89.57

Table 11. Macro-averaged results of evaluation phase for predicted **coarse-grained tagset**, using the model trained on initial dataset, with 3 epochs

Lang.	Prec.	Recall	F1
EN	75.88	74.30	75.05
DE	72.55	69.57	70.95
ES	70.32	65.00	67.47
IT	73.19	68.01	70.39
SV	72.21	62.75	66.81

As far as rankings are concerned, in the evaluation phase we have managed to get moderate results in the English track (we ranked 17^{th} out of 34), while our results in the other tracks could be further improved in the future: 13^{th} out of 17 for German, 16^{th} out of 18 for Spanish, 14^{th} out of 15 for Italian and 15^{th} out of 16 for Swedish. In the post-evaluation stage of the competition, we managed to achieve better results (see Table 12) with our improved system. A notable difference between the system we used during the evaluation phase and the current one is the dataset on which we trained the model. The first model was trained on the dataset that can be seen in Table 1, which contains 166,413 unique instances and 2,671,439 tokens, while the last one was trained on the 25k dataset that has 298,388 unique instances, with a total number of 7,472,249 tokens (see Table 7). With

the increase of the dataset, we also increased the number of epochs, if the first model was trained for only 3 epochs, the last one was trained for 7 epochs (the rest of the hyperparameters remained the same). The training process for the model used in the evaluation phase took over 2 hours, while it took almost 18 hours for the current one.

Table 12. Macro-averaged results after the evaluation phase for predicted **fine-grained tagset**, using the model trained on 25k dataset, with 7 epochs

Lang.	Prec.	Recall	F1	Ranking
EN	74.71	74.61	74.66	7/34
DE	75.69	74.96	75.33	7/17
ES	68.35	67.58	67.96	9/18
IT	77.96	76.64	77.30	4/15
SV	72.58	70.46	71.51	9/16

Comparing the macro-averaged F1 scores achieved for the fine-grained tagset in the post-evaluation stage (scores from Table 12) with the ones obtained during the evaluation phase (scores from Table 10), we can notice a huge improvement of almost 21 points for Italian. For German and Swedish, the score increased by over 19 points, while for English and Spanish, with approximately 13 points. With the new results obtained, that's how we would place ourselves on the leaderboard: 7^{th} place out of 34 for English, 7^{th} place out of 17 for German, 9^{th} place out of 18 for Spanish, 4^{th} place out of 15 for Italian, and 9^{th} place out of 16 for Swedish.

5 Conclusions

In this thesis, we got the opportunity to explore a transformer model's capabilities at dealing with NLP tasks – identification of complex (fine-grained) named entities in multiple languages, in our case – and how to handle task-specific input. More specifically, we put the classic BERT model to the test and found it to live up to its reputation as a general-purpose transformer model by managing moderate results. Moreover,

our experiments showed that we could improve the performance of a model for named entity recognition using a larger training corpus. Taking into account the fact that this is not always possible, methods that integrate additional relevant knowledge (additional context information) into transformer models may overcome this insufficiency. We have learned more about the workings of the transformer model and now have a better understanding of what tackling such a task entails with regard to approaches and resource management. Therefore, we can say that a robustly optimized pre-trained approach of BERT, such as XLM-RoBERTa, which is a retrained BERT model with improved training methodology, more data and compute power, would outperform the results we achieved with BERT.

As an overall conclusion, the fine-grained level performance was inspected by the competition organizers, and it was observed that, although the coarse classes are usually easy to identify, for example, the PER class, distinguishing the fine-grained tags poses a greater challenge due to their high ambiguity [8]. In this scenario, it was observed that pre-trained transformer models often confuse entities of the Scientist class with entities from the Artist or Politician classes. This is because these models possess a higher level of pre-trained knowledge related to Artist and Politician entities compared to Scientist entities. Therefore, the problem still remains open.

As for the future directions that could improve the results of this particular model, another important thing would surely be a more versatile module for handling input test data. Contrary to expectation, we should have put more focus on this part of the system. Apart from that, parallelization of the system could have potentially made it available to us to harness more powerful transformer models. In addition to the aforementioned significance of external data, another essential element for achieving strong performance would be the usage of ensemble learning strategies: training multiple models and combining them in an ensemble to generate the final predictions.

References

[1] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 260–270. *CoRR, abs/1603.01360*, no. 1603.01360. DOI: 10.18653/v1/N16-1030.

[2] L. Zhang, X. Nie, M. Zhang, M. Gu, V. Geissen, C.J. Ritsema, D. Niu, and H. Zhang, "Lexicon and attention-based named entity recognition for kiwifruit diseases and pests: A Deep learning approach," *Frontiers in Plant Science*, vol. 13, 2022, Article ID: 1053449. DOI: https://doi.org/10.3389/fpls.2022.1053449.

[3] A. Iftene, D. Trandabăț, M. Toader, and M. Corîci, "Named Entity Recognition for Romanian," in *Proceedings of the 3th Conference on Knowledge Engineering: Principles and Techniques Conference (KEPT2011),* Studia Universitatis, Babes Bolyai, vol. 2, 2011, pp. 19–24.

[4] Y. Shao, C. Hardmeier, and J. Nivre, "Multilingual named entity recognition using hybrid neural networks," in *The Sixth Swedish Language Technology Conference (SLTC)*, 2016, https://api.semanticscholar.org/CorpusID:57588814.

[5] D. Gifu and G. Vasilache, "A language-independent named entity recognition system," in *Proceedings of The 10th International Conference Linguistic Resources and Tools for Processing The Romanian Language, ConsILR-2014*, Alexandru Ioan Cuza" University Publishing House, Iaşi, 2014, pp. 181–188.

[6] D. Cristea, D. Gifu, I. Pistol, D. Sfirnaciuc, and M. Niculiţă, "A mixed approach in recognising geographical entities in texts," in *Linguistic Linked Open Data: 12th EUROLAN 2015 Summer School and RUMOUR 2015 Workshop*, (Sibiu, Romania, July 13-25, 2015), Revised Selected Papers 1, Springer, 2016, pp. 49–63.

[7] A. Iftene, "Identifying Geographical Entities in Users' Queries," in *CLEF 2009, LNCS 6241, Part I (Multilingual Information Access Evaluation, Vol. I, Text Retrieval Experiments)*, C. Peters et al., Eds. Heidelberg: Springer, 2010, pp. 535–538.

[8] B. Fetahu, S. Kar, Z. Chen, O. Rokhlenko, and S. Malmasi, "SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2)," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), ACL*, 2023.

[9] B. Fetahu, Z. Chen, S. Kar, O. Rokhlenko, and S. Malmasi, "MultiCoNER v2: a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition," arXiv:2208.14536, 2023.

[10] M. Mitrofan and V. Pais, "Improving Romanian BioNER Using a Biologically Inspired System," in *Proceedings of the 21st Workshop on Biomedical Language Processing*, Association for Computational Linguistics, 2022, pp. 316–322.

[11] S. Ashwini and J. D. Choi, "Targetable Named Entity Recognition in Social Media," arXiv:1408.0782, 2014.

[12] A. Iftene and A. Balahur-Dobrescu, "Named Entity Relation Mining Using Wikipedia," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, (28-30 May, Marrakech, Morocco), 2008, http://www.lrec-conf.org/proceedings/lrec2008/pdf/192_paper.pdf.

[13] I. Augenstein, L. Derczynski, and K. Bontcheva, "Generalisation in named entity recognition: A quantitative analysis," *Computer Speech & Language*, vol. 44, pp. 61–83, 2017.

[14] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," arXiv:2010.11934, 2020.

[15] M. Arkhipov, M. Trofimova, Y. Kuratov, and A. Sorokin, "Tuning Multilingual Transformers for Language-Specific Named Entity

Recognition," in *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, Association for Computational Linguistics, 2019, pp. 89–93.

[16] S. Malmasi, A. Fang, B. Fetahu, S. Kar, and O. Rokhlenko, "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, 2022, pp. 1412–1437.

[17] X. Wang, Y. Shen, J. Cai, T. Wang, X. Wang, P. Xie, F. Huang, W. Lu, Y. Zhuang, and K. Tu, "Damo-nlp at Semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition," arXiv:2203.00545, 2022.

[18] B. Chen, Y.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," arXiv:2203.03216, 2022.

Viorica-Camelia Lupancu, Adrian Iftene Received December 08, 2023

Viorica-Camelia Lupancu
"Alexandru Ioan Cuza" University of Iasi, Romania, Faculty of Computer Science
General Berthelot, No. 16, Iasi, Romania
E–mail: lupancu_camelia_99v@yahoo.com

Adrian Iftene
ORCID: https://orcid.org/0000-0003-3564-8440
"Alexandru Ioan Cuza" University of Iasi, Romania, Faculty of Computer Science
General Berthelot, No. 16, Iasi, Romania
E–mail: adiftene@info.uaic.ro