FII_Better at SemEval-2023 Task 2: MultiCoNER II Multilingual Complex Named Entity Recognition

Viorica-Camelia Lupancu[1]
Alexandru-Gabriel Plătică[1]
Cristian-Mihai Roșu[1]
Daniela Gîfu[1,2]
Diana Trandabăț[1]

[1]Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
[2]Institute of Computer Science, Romanian Academy - Iasi Branch
{lupancu.camelia.99, platicaalex, cristian.rosu453}@gmail.com
daniela.gifu@iit.academiaromana-is.ro
diana.trandabat@gmail.com

Abstract

The "MultiCoNER II Multilingual Complex Named Entity Recognition" task[1] at SemEval 2023 competition focuses on identifying complex named entities (NEs), such as the titles on creative works (songs, books, movies), in several languages. In the context of SemEval, our team, FII_Better, presents an exploration of a base transformer model's capabilities regarding the task, focused more specifically on five languages (English, Spanish, Swedish, German, Italian). We take DistilBERT and BERT 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 learned a lot about working with transformers and, on top of that, we managed to get fair results on the chosen languages.

1 Introduction

Named entity recognition (NER) is a subtask of natural language processing (NLP), being still a crucial learning problem (Lample et al. (2016), Zhang et al. (2022)). NER is used especially for relevant information extraction from text data (Shao et al. (2016), Gifu and Vasilache (2014)). It is impressive that state-of-the-art NER systems rely heavily on hand-crafted features and domain-specific knowledge (Cristea et al., 2016).

The MultiCoNER II shared task (Fetahu et al., 2023b) 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.

NER helps one easily identify key elements in a text, like athletes, politicians, human settlements, clothing pieces, medications and more. Extracting the main entities in a text helps sorting unstructured data and detecting important information, which is crucial if one has to deal with large datasets.

The datasets mainly contain sentences from three domains: Wikipedia, web questions and user queries which are usually short and low-context sentences (Malmasi et al. (2022a), Fetahu et al. (2023a)). 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 (like an article or results of a web search) 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 multilingual context or not, section 3

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

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1107–1113
July 13-14, 2023 ©2023 Association for Computational Linguistics

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

Currently, processing complex named entities is still a challenging NLP task. There is relatively little work on recognizing other types of entities than the traditional ones (persons, locations, organizations, date, currency). Complex NEs, like chemicals, ingredients, diseases or active substances are not simple nouns and are much more difficult to recognize (Mitrofan and Pais, 2022). They can take the form of any linguistic constituent and have a very different surface from than traditional NEs. Their ambiguity makes it challenging to recognize them. Additionally, more and more people share nowadays information about various topics online, making NER for these non-traditional entities more important in the context of data collected from social media, where people's interests are directly expressed (Ashwini and Choi, 2014).

There has been made an attempt to investigate the ability of modern NER systems to generalize effectively over a variety of genres. This attempt also found out, as expected, that there is a strong correlation between the NER performance and the training corpus size, so by having a bigger corpus, the results may be more accurate (Augenstein et al., 2017). 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 transformer, which outperforms the previously released multilingual transformers (Xue et al., 2020).

Among those, BERT is one of the most powerful unsupervised models. A multilingual variant of it, trained 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 (Arkhipov et al., 2019). Lastly, the most successful approaches used in the previous MultiCoNER SemEval task were multilingual systems that revolved around improving upon the baseline model of XLM-RoBERTa, adapted to the multilingual track and then specialized for each monolingual track, as well as an enhanced BiLSTM network (Malmasi et al., 2022b).

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 over the dataset, as well as the steps we went through before choosing the BERT model and using the data for training.

3.1 Architecture

The architecture we used to train our model is rather simple and plays directly into BERT's functionalities.

The diagram below showcases the basic flow and steps, while the details are detailed in the following sections.

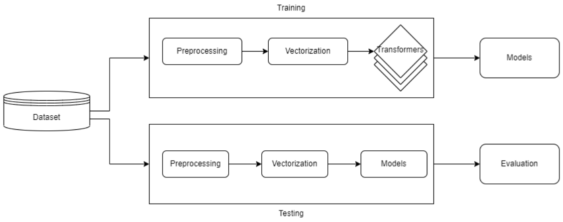


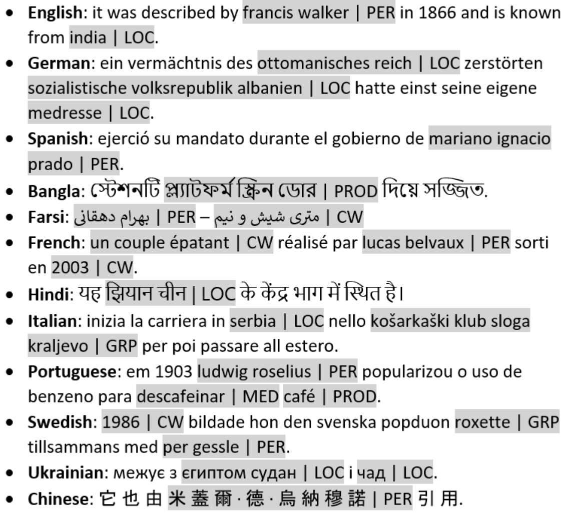
Figure 1: The architecture of our NER system

3.2 Dataset

The dataset that we are using, MultiCoNER II, is a large multilingual dataset used for NER, that covers domains like: Wiki sentences, questions and search queries across 12 languages. This dataset contains 26M tokens and it is assembled from public resources. MultiCoNER II defines the following NER tag-set, with 6 classes:

- Location (LOC): Facility, OtherLOC, HumanSettlement, Station
- Creative Work (CW): VisualWork, MusicalWork, WrittenWork, ArtWork, Software, OtherCW
- Group (GRP): MusicalGRP, PublicCORP, PrivateCORP, OtherCORP, AerospaceManufacturer, SportsGRP, CarManufacturer, TechCORP, 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

Figure 2 presents a snippet with annotated entities from the dataset:



- **English**: it was described by francis walker | PER in 1866 and is known from india | LOC.
- **German**: ein vermächtnis des ottomanisches reich | LOC zerstörten sozialistische volksrepublik albanien | LOC hatte einst seine eigene medresse | LOC.
- **Spanish**: ejerció su mandato durante el gobierno de mariano ignacio prado | PER.
- **Bangla**: স্টেশনটি প্ল্যাটফর্ম স্ক্রিন ডোর | PROD দিয়ে সজ্জিত.
- **Farsi**: بهرام دهقانی | PER – متری شیش و نیم | CW
- **French**: un couple épatant | CW réalisé par lucas belvaux | PER sorti en 2003 | CW.
- **Hindi**: यह झियान चीन | LOC के केंद्र भाग में स्थित है।
- **Italian**: inizia la carriera in serbia | LOC nello košarkaški klub sloga kraljevo | GRP per poi passare all estero.
- **Portuguese**: em 1903 ludwig roselius | PER popularizou o uso de benzeno para descafeinar | MED café | PROD.
- **Swedish**: 1986 | CW bildade hon den svenska popduon roxette | GRP tillsammans med per gessle | PER.
- **Ukrainian**: межує з єгиптом судан | LOC і чад | LOC.
- **Chinese**: 它 也 由 米 蓋 爾 · 德 · 烏 納 穆 諾 | PER 引 用.

Figure 2: Examples for all the languages existing in MultiCoNER II

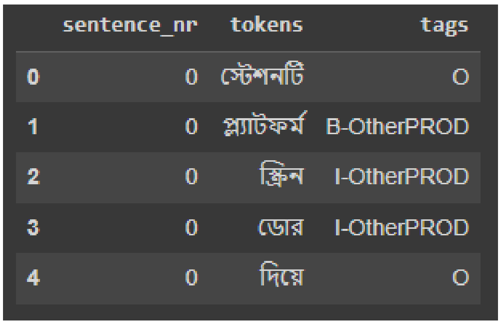
The dataset is split in three: (i) training dataset - used to train our model, (ii) development dataset - used for prediction in the practice phase and (iii) test dataset - used for prediction in evaluation phase. These three splits were made for each language. Table 1 shows some statistics of the dataset. For Latin languages, the number of instances is considerable higher over the three splits.

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

Table 1: Data statistics

3.2.1 Preprocessing

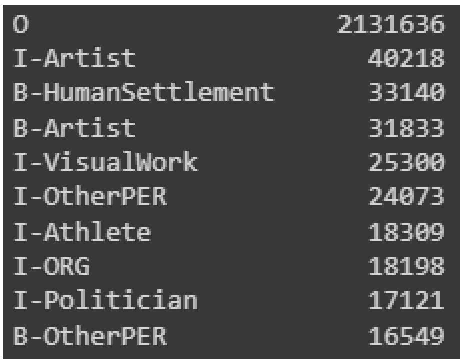
We have concatenated the training data from all of the languages into a single CONLL file. Then, to make reading and processing a bit easier, we have converted the data into CSV format, with the following structure (see Figure 3):



```
   sentence_nr  tokens          tags
0            0  স্টেশনটি            O
1            0  প্ল্যাটফর্ম   B-OtherPROD
2            0  স্ক্রিন       I-OtherPROD
3            0  ডোর         I-OtherPROD
4            0  দিয়ে               O
```

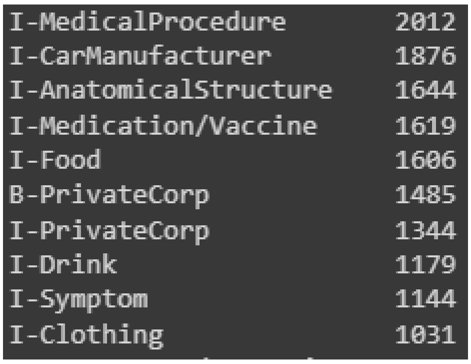
Figure 3: Initial CSV format of the training data

At this step, we took note of the number of 2,671,439 total entries. Spread between 67 NE tags, there was a clear discrepancy of distributions (highlighted in Figures 4 and 5).



```
O                    2131636
I-Artist               40218
B-HumanSettlement      33140
B-Artist               31833
I-VisualWork           25300
I-OtherPER             24073
I-Athlete              18309
I-ORG                  18198
I-Politician           17121
B-OtherPER             16549
```

Figure 4: Top 10 most frequent tags

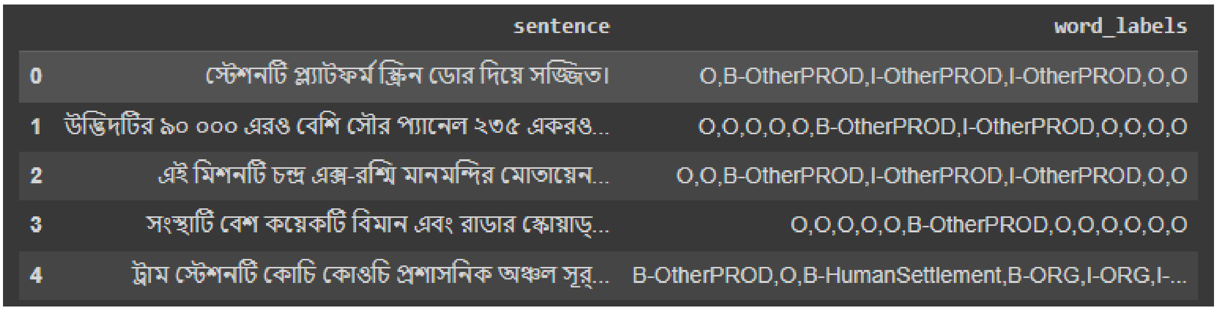


```
I-MedicalProcedure       2012
I-CarManufacturer        1876
I-AnatomicalStructure    1644
I-Medication/Vaccine     1619
I-Food                   1606
B-PrivateCorp            1485
I-PrivateCorp            1344
I-Drink                  1179
I-Symptom                1144
I-Clothing               1031
```

Figure 5: Bottom 10 least frequent tags

Finally, we have grouped the entries by sentence number and have used this format (Figure 6) of the data going forward with the training.

This dataset had a final size of 166,413 entries, or better said sentences.



	sentence	word_labels
0	স্টেশনটি প্ল্যাটফর্ম স্ক্রিন ডোর দিয়ে সজ্জিত।	O,B-OtherPROD,I-OtherPROD,I-OtherPROD,O,O
1	উদ্ভিদটির ৯০ ০০০ এরও বেশি সৌর প্যানেল ২৩৫ একরও...	O,O,O,O,O,B-OtherPROD,I-OtherPROD,O,O,O,O
2	এই মিশনটি চন্দ্র এক্স-রশ্মি মানমন্দির মোতায়েন...	O,O,B-OtherPROD,I-OtherPROD,I-OtherPROD,O,O
3	সংস্থাটি বেশ কয়েকটি বিমান এবং রাডার স্কোয়াড্...	O,O,O,O,O,B-OtherPROD,O,O,O,O,O,O
4	ট্রাম স্টেশনটি কোচি কোওচি প্রশাসনিক অঞ্চল সূর্...	B-OtherPROD,O,B-HumanSettlement,B-ORG,I-ORG,I-...

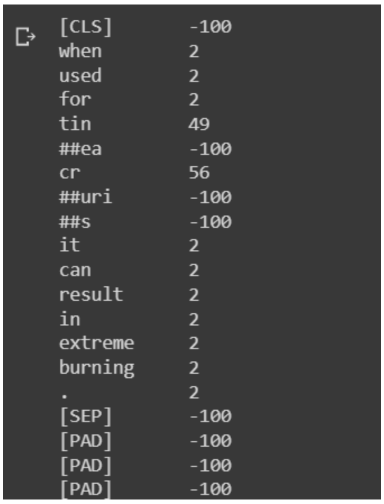
Figure 6: Final dataset format

3.2.2 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 0 to 66)
- ids_to_labels being the reverse map of the above

Then for each pair of (sentence, tags) in the dataset, we encode the sentence's words using a tokenizer with a padding of 128 and convert the tags 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 which are not in the first part of the word after tokenization, are attributed a custom value of -100. This can be better visualized in Figure 7.



```
[CLS]       -100
when        2
used        2
for         2
tin         49
##ea        -100
cr          56
##uri       -100
##s         -100
it          2
can         2
result      2
in          2
extreme     2
burning     2
.           2
[SEP]       -100
[PAD]       -100
[PAD]       -100
[PAD]       -100
```

Figure 7: Sentence encoding visualization

Considering the final transformed model we ended up using, the tokenizer we applied for this was, appropriately, the *bert-base-uncased* tokenizer.

The training set was turned into a DataLoader instance (from pytorch) and at this point was ready to be used.

3.3 Methods

3.3.1 Training

With a dataset of this size, we have run into difficulties trying to emulate the recommended baseline results with our own resources, as such we opted to try out different pretrained 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 on developing a model of our own based on the DistilBERT transformer.

Using it as a base, we have created a baseline English model that has been fine-tuned on the English training data and obtained decent enough results to begin building upon it. The results of this baseline are shown in Table 2. For the training parameters, we have used a learning rate of 1e-2, a batch size of 32, a number of epochs of 8 and a SGD (Stochastic Gradient Descent) optimizer.

Precision	Recall	f1	Accuracy
0.61	0.59	0.57	0.89

Table 2: Initial fine-tuned DistilBERT weighted results

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. 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.

Precision	Recall	f1	Accuracy
0.91	0.90	0.91	0.90

Table 3: Initial fine-tuned BERT weighted results

Further testing used the same hyper-parameters, with the only difference being the number of epochs we trained the model for.

4 Results

4.1 Analysis

During the practice phase of the competition, we have submitted to each track a file that contains only the predicted tag for every token.

it	_	_	O
originally	_	_	O
operated	_	_	O
seven	_	_	O
bus	_	_	O
routes	_	_	O
which	_	_	O
were	_	_	O
mainly	_	_	O
supermarket	_	_	O
routes	_	_	O
for	_	_	O
asda	_	_	B-CORP
and	_	_	O
tesco	_	_	B-CORP
.	_	_	O

Table 4: Example of dev sentence

In Table 4 there is an example from the en_dev file, and below we can find the predicted tags that we must submit in the CodaLab competition:

O O O O O O O O O O O O B-CORP O B-CORP O

The weighted results that we have achieved with our model on the dev files can be seen in Table 5.

Lang.	Precision	Recall	f1	Accuracy
EN	0.94	0.94	0.94	0.94
BN	0.87	0.88	0.86	0.88
DE	0.91	0.91	0.91	0.91
ES	0.93	0.93	0.93	0.93
FA	0.87	0.89	0.85	0.89
FR	0.93	0.93	0.93	0.93
HI	0.87	0.89	0.87	0.89
IT	0.93	0.93	0.93	0.93
PT	0.92	0.92	0.92	0.92
SV	0.92	0.92	0.92	0.92
UK	0.89	0.91	0.89	0.91
ZH	0.72	0.77	0.72	0.77

Table 5: Weighted-averaged results of practice phase for predicted tag

There are 2 scores that are noteworthy here, one regarding each token to its predicted tag (Table 6) and the other one regarding the tag predicted being in the correct tag-set. (Table 7)

Language	Precision	Recall	f1
EN	0.68	0.63	0.64
BN	0.62	0.34	0.40
DE	0.63	0.57	0.59
ES	0.66	0.60	0.63
FA	0.37	0.25	0.28
FR	0.67	0.61	0.62
HI	0.45	0.24	0.30
IT	0.68	0.63	0.65
PT	0.68	0.58	0.62
SV	0.63	0.55	0.57
UK	0.57	0.40	0.45
ZH	0.26	0.10	0.13
Multi	0.57	0.46	0.49

Table 6: Macro-averaged results of practice phase for predicted tag

We can observe that, compared to the prediction of the tags for each individual in all of the languages, the class in which each of the predicted tags can be found has increased scores. This means that, although it does not find the exact tag, it predicts another tag inside the same tag-set.

4.2 Evaluation

We were able to get results from all languages in the practice phase, however simulated errors were added in the datasets of the evaluation phase (except for DE test data) and our model could not

Language	Precision	Recall	f1
EN	0.80	0.78	0.79
BN	0.67	0.43	0.50
DE	0.73	0.70	0.71
ES	0.75	0.70	0.72
FA	0.56	0.37	0.42
FR	0.75	0.73	0.74
HI	0.57	0.29	0.37
IT	0.78	0.74	0.76
PT	0.75	0.68	0.71
SV	0.76	0.65	0.69
UK	0.75	0.51	0.58
ZH	0.41	0.15	0.20
Multi	0.69	0.56	0.60

Table 7: Macro-averaged results of practice phase for predicted tag-set

handle them properly. On a small scale (2-3 characters), we were able to resolve those problematic characters, but in languages that we were not familiar with, we had difficulty in detecting them.

Similarly to the practice phase, in Tables 8 and 9 are the results we have achieved with our model during the evaluation phase for the languages where we could successfully handle the input.

Language	Precision	Recall	f1
EN	0.63	0.60	0.61
DE	0.57	0.55	0.55
ES	0.57	0.53	0.54
IT	0.58	0.55	0.56
SV	0.55	0.51	0.52

Table 8: Macro-averaged results of evaluation phase for predicted tag

Language	Precision	Recall	f1
EN	0.75	0.74	0.75
DE	0.72	0.69	0.70
ES	0.70	0.65	0.67
IT	0.73	0.68	0.70
SV	0.72	0.62	0.66

Table 9: Macro-averaged results of evaluation phase for predicted tag-set

Looking at similar approaches from the previous year (SemEval-2022), we noticed a few papers that each add a specific component that elevate the results above the baseline:

- In Ma et al. (2022), before feeding the pretrained BERT model to build their NER system, they concatenated input text with entity information from the LUKE dictionary using string matching. Having participated only in the English track, they achieved an average F1-score of 0.7837, compared to our 0.61.

- In Boros et al. (2022), while using the same BERT encoder with a transformer layer and a CRF head for classification, they similarly preprocess the test text by adding to it a similar sentence from the training set chosen using SentenceBERT beforehand. They participated in all language tracks from last year with the best results in German (average F1-score of 0.7723 against our 0.55). The other shared languages between us are English (average F1-score of 0.7196 against our 0.61) and Spanish (average F1-score of 0.6893 against our 0.54)

- In Pandey et al. (2022), they tested multiple classification heads apart from the CRF layer and found that for English the best approach was a linear layer (average F1-score of 0.7174 against our 0.61), while for other languages like Spanish they found the best approach to be pretraining BERT using the Whole Word Masking learning objective over Wikipedia data and the CRF layer (average F1-score of 0.612 against our 0.54).

Even though Italian and Swedish were not part of last year's task, we understand from these papers that it is a difficult endeavor to have a single model capable of achieving great results across this spectrum of language. We limited ourselves by over-relying on a single model in this case.

As far as rankings are concerned, 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).

5 Conclusion

In this paper, we got the opportunity to explore a transformer model's capabilities at dealing with NLP tasks - in this case complex NER - 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 of general-purpose transformer model by managing moderate results.

We have learned a lot about the workings of the transformer model and now have a better understanding of what tackling such a task entails with regards to approaches and resource management.

Among the things that could improve the results of this particular model, one 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.

References

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Sandeep Ashwini and Jinho D. Choi. 2014. Targetable named entity recognition in social media. *CoRR*, abs/1408.0782.

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

Emanuela Boros, Carlos-Emiliano González-Gallardo, Jose Moreno, and Antoine Doucet. 2022. L3i at SemEval-2022 task 11: Straightforward additional context for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1630–1638, Seattle, United States. Association for Computational Linguistics.

Dan Cristea, Daniela Gifu, Ionuţ Pistol, Daniel Sfirnaciuc, and Mihai Niculiţă. 2016. 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*, pages 49–63. Springer.

Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. MultiCoNER v2: a Large Multilingual dataset for Fine-grained and Noisy Named Entity Recognition.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023b. SemEval-2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Daniela Gifu and Gabriela Vasilache. 2014. A language independent named entity recognition system. *Alexandru Ioan Cuza" University Publishing House, Iaşi*, pages 181–188.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Long Ma, Xiaorong Jian, and Xuan Li. 2022. PAI at SemEval-2022 task 11: Name entity recognition with contextualized entity representations and robust loss functions. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1665–1670, Seattle, United States. 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. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.

Maria Mitrofan and Vasile Pais. 2022. Improving Romanian BioNER using a biologically inspired system. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 316–322, Dublin, Ireland. Association for Computational Linguistics.

Amit Pandey, Swayatta Daw, Narendra Unnam, and Vikram Pudi. 2022. Multilinguals at SemEval-2022 task 11: Complex NER in semantically ambiguous settings for low resource languages. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1469–1476, Seattle, United States. Association for Computational Linguistics.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2016. Multilingual named entity recognition using hybrid neural networks. In *The sixth Swedish language technology conference (SLTC)*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Lilin Zhang, Xiaolin Nie, Mingmei Zhang, Mingyang Gu, Violette Geissen, Coen J Ritsema, Dangdang Niu, and Hongming Zhang. 2022. Lexicon and attention-based named entity recognition for kiwifruit diseases and pests: A deep learning approach. *Frontiers in Plant Science*, 13.