SemEval 2023: MultiCoNER II Final Report

Asal Shavandi Chahyon Ku Josh Spitzer-Resnick London Lowmanstone IV
University of Minnesota
{shava006,ku000045,spitz123,lowma016}@umn.edu

Abstract

The MultiCoNER II shared task aims to detect complex named entities for multiple languages. Inspired by Multilingual Data Augmentation technique (MulDA) (Liu et al., 2021), in this project we introduced 6 translation techniques to augment our training dataset hoping to improve the performance of the baseline and address challenges such as sensitivity to noise and lower performance on out-of-knowledge-base named entities. Then, we compare the performance to the previous state-of-the-art and best-performing work, KB-NER (Wang et al., 2022) and show that our technique achieves not only better performance than the not augmented baseline but also better or comparable results to the information retrieval based augmentation. Our code can be found in https://github.com/chahyon-ku/polygots.

1 Introduction

The named entity recognition task is a critical part of information extraction in which every word in a sentence is classified into named entity types such as names of people, organization, location, etc. (Nadeau and Sekine, 2007). Since it was first organized in 1996 at the Sixth Message Understanding Conference, many mono- and multilingual tasks, such as the CoNLL 2003 (Sang and Meulder, 2003), Ontonotes corpus v5 (Pradhan et al., 2013), and WNUT 2017 Emerging Entities (Derczynski et al., 2017) were organized to tackle its challenges.

Among named entities, complex named entities are the more syntactically complex named entities–often names of creative works–that existing systems have a hard time recognizing (Ashwini and Choi, 2014). Complex named entities are more challenging to detect than traditional ones, because they are rarer in data, newly created more frequently, and have more diverse syntactic structures.

Even though knowledge-retrieval-based systems achieved great results at the MultiCoNER shared task from SemEval 2022 (Malmasi et al., 2022), they are sensitive to noisy and out-of-domain entities. The MultiCoNER II shared task proposes new tasks to address the shortcomings of top performing models from the MultiCoNER shared task.

Our group, Team Polygots, attempt to propose improvements to the baseline NER model by trying various data augmentations on the training data.

2 Related Works

2.1 Named Entity Recognition

Named Entity Recognition (NER) is a core natural language processing (NLP) task (Chen et al., 2022) that has a lot of applications in academia, marketing, medical and security domains. Transformer-based pretrained language models have achieved great success in almost every NLP task (Kalyan et al., 2021) including NER. These models learn universal language representations from large volumes of text data using self-supervised learning and transfer this knowledge to downstream tasks (Kalyan et al., 2021). Multilingual BERT (M-BERT), released by (Devlin et al., 2019) as a single language model pre-trained from monolingual corpora in 104 languages, is shown to be very good at cross-lingual model transfer (Pires et al., 2019). XLM-RoBERTa (Conneau et al., 2019) is another pretrained multilingual language model at scale that has led to significant performance gain for a wide range of cross-lingual transfer tasks.

2.2 Multilingual Language Models

Fine-tuning pretrained contextual embedding is a useful and effective approach to many NLP tasks (Wang et al., 2022) and recently many researchers have put their effort into training fine-tuned multilingual models such as mBERT and XLM-RoBERTa to improve their model's performance. (Malmasi et al., 2022) designed a NER system using XLM-RoBERTa on MultiCoNER I

dataset that computes a representation for each token which was then used to predict the token tag using a Conditional Random Field (CRF) classification layer (Sutton and McCallum, 2010). Their system resulted in a F1 score of 0.478.

2.3 Data Augmentation for Natural Language Processing

There are four main categories of data augmentation methods: translation, substitution, generation, and mix-up. Translation-based methods, such as MulDA (Liu et al., 2021), translates a sentence to another language and often back to the source language to introduce variance. Substitution-based methods, such as MELM (Zhou et al., 2022), replaces characters, words, or phrases based on heuristics or language models. Generation-based methods, such as DAGA (Ding et al., 2020) trains a language model on the training data and randomly sample from the language model to generate new data. Mix-up-based methods, such as SeqMix (Zhang et al., 2020), linearly interpolate between pairs of samples to generate novel sentences.

2.4 Best Models from MultiCoNER I

The first ranked team from MultiCoNER I shared task, DAMO-NLP (Wang et al., 2022), which got the highest F1 score for the multilingual task, took a different approach and introduced a knowledge-based NER system which used (Malmasi et al., 2022) system as their baseline and added a knowledge retrieval module to enhance their performance. The knowledge retrieval module takes an input sentence as a query and retrieves top-k related paragraphs from Wikipedia which will be then concatenated and fed into the NER module. The output of the NER module which is a token representation of the input sentence will be fed into a linear-chain CRF to produce the label predictions. This method has shown an F1 score of 0.853.

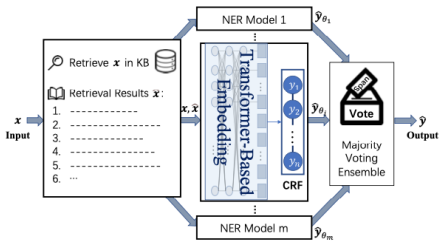

Figure 1: The architecture of DAMO-NLP knowledge-based NER system

The second-ranked team from MultiCoNER I, USTC-NELSLIP (Chen et al., 2022), also used the same concept of a knowledge retrieval system but instead of retrieving top-related paragraphs from Wikipedia, they introduced a gazetteer-adapted integration network (GAIN). This system first adapts the representation of the gazetteer network built from Wikidata to the XLM-RoBERTa model by minimizing KL divergence between them. After adaptation, these networks will be integrated for backend-supervised NER training. This method has also shown an F1 score of 0.853.

3 Problem Formulation

3.1 Task

MultiCoNER II is a Multilingual Complex Named Entity Recognition shared task, offered as part of SemEval-2023, The 17th International Workshop on Semantic Evaluation. Given a sentence, the task is to detect and categorize all named entities in the sentence. To be more concrete, each word in the sentence is tagged with Beginning-Inside-Outside tags. The beginning tag denotes the first word in a named entity, inside denotes other words in a named entity, and outside means the word is not a part of a named entity. Then, each beginning and inside tags are categorized into one of 36 fine-grained labels, which are organized into 6 categories according to the WNUT 2017 (Derczynski et al., 2017) taxonomy entity types: person, group, corporation, location, product, and creative work. The dataset consists of 12 languages, where each language has between 9k-16k training sentences and 500-900 development sentences.

3.2 Dataset

MultiCoNER II is divided into 12 languages, including 7 of the same languages from MultiCoNER I (Bangla, Chinese, English, Farsi, German, Hindi, and Spanish) and 5 new languages (French, Italian, Portuguese, Swedish, and Ukrainian). The second iteration of this task leaves out 4 previous languages (Dutch, Korean, Russian, and Turkish) as well as the multilingual and code-mixed tasks of the previous competition. For our efforts, we have simplified our exploration to look at the English and French tasks.

Rather than the 6 coarse categories present in MultiCoNER I common of standard NER systems, there are instead 36 defined granular labels which are then organized into 6 categories (see figure

Figure 2: Example sentences and labeled named entities from MultiCoNER I (Malmasi et al., 2022) and MultiCoNER II

3). For each language in the dataset in both train and development sets however, OtherCW, OtherCORP, and TechCORP never occur, thus limiting the dataset to 33 observed labels. Notably, MultiCoNER II introduces a new medical (MED) category, and lumps the previous year's corporation (CORP) category into the group (GRP) category. Examples of the difference in granularity of labeling between the two tasks can be seen in figure 2.

label	description
CORP /	Corporation
CW	Creative Work
GRP	Group
LOC	Location
PER	Person
PROD	Product

category	labels
Creative Work (CW)	ArtWork, MusicalWork, ~~OtherCW~~, Software, VisualWork, WrittenWork
Group (GRP)	AerospaceManufacturer, CarManufacturer, MusicalGRP, ORG, ~~OtherCORP~~, PrivateCORP, PublicCORP, SportsGRP, ~~TechCORP~~
Location (LOC)	Facility, HumanSettlement, OtherLOC, Station
Medical (MED) //	AnatomicalStructure, Disease, MedicalProcedure, Medication/Vaccine, Symptom
Person (PER)	Artist, Athlete, Cleric, OtherPER, Politician, Scientist, SportsManager
Product (PROD)	Clothing, Drink, Food, OtherPROD, Vehicle

Figure 3: Coarse labels present in MultiCoNER I (top) as compared to granular labels in MultiCoNER II (bottom)

The data itself is stored in the CoNLL format, where each token has a label of 'O' for outside if it is not part of an entity, 'B-<label>' if it begins an entity, and 'I-<label>' if it is inside of an entity after it begins. By modeling the samples in this sequential fashion, there is no issue in working with languages that read left to right or right to left. For languages that have a concept of capitalization, all samples have been made lowercase, which removes the ability to use capitalization as an aid in identifying entities.

The dataset was not released until late October, which limited our time to explore and analyze it. In contrast to the stated aims of the task to explore limitations of previous methods which were brittle to out-of-knowledge base entities and noise such as misspellings and typos, none of those challenges were introduced, thus leaving the novelty in the new dataset to a new set of languages and a finer-grained label set. However, as the dataset was created with weak supervision, we observe that not all of the annotations may be accurate.

While MultiCoNER I had a consistent 15,300 training and 800 development samples per language (with the multilingual task a simple combination of the 11 languages), MultiCoNER II has more variability in the exact number of training and development samples per language, however, the ratios remained proportional across languages. Each language has between 9k-16k training sentences and 500-900 development sentences, though there are notably fewer samples for Bangla, Chinese, German, and Hindi than there are for English, Farsi, French, Italian, Portuguese, Spanish, Swedish, and Ukrainian. Across each language and dataset, there are roughly 1.25-1.5 entities per sentence on average. The samples and number of entities in each dataset can be seen in figure 10.

It is worth noting the imbalanced distribution of the named entity types as shown in figure 9.

4 Proposed Methods

4.1 MulDA Translations

MulDA (Liu et al., 2021) is a data augmentation technique that focuses on multilingual NER. MulDA uses off-the-shelf Google Cloud API as its translation tool which supports more than 100 languages. MulDA introduced a 3-step translation method that replaces named entities with contextual placeholders before sentence translation and then after translation, it replaces placeholders in translated sequences with the corresponding translated entities. See Figure 4 for a detailed example of how this is done.

In addition to translation, the MulDA paper goes on to use a linearization technique introduced by DAGA (Ding et al., 2020) that adds entity types before sequence tokens after the translation. It then trains an LSTM-based language model based on linearized sequences. This augmentation technique helps to increase diversity by generating syn-

Labeled sentence in the source language:
[PER Jamie Valentine] was born in [LOC London].

1. Translate sentence with placeholders:
src: PER0 was born in LOC1.
tgt: PER0 nació en LOC1.

2. Translate entities with context:
PER0
src: [Jamie Valentine] was born in London.
tgt: [Jamie Valentine] nació en Londres.

LOC1
src: Jamie Valentine was born in [London].
tgt: Jamie Valentine nació en [Londres].

3. Replace placeholders with translated entities:
[PER Jamie Valentine] nació en [LOC Londres].

Figure 4: MulDA's labeled sentence translation where **src** and **tgt** are the source and target languages respectively

thetic labeled data in multiple languages (Liu et al., 2021). While we imitated the linearization aspect of DAGA in our translations, we do not attempt to do additional language generation. Thus, when referring to MulDA in the rest of the paper, we are referring only to section 4.1.

4.1.1 Full and Partial

For our final project, we fully implemented the MulDA algorithm as written in the paper, replacing named entities with the entity names and then translating each entity in brackets individually. In the MulDA code, it seems as though they make language-specific additions that are not mentioned in the paper. We have only implemented what was described in the paper.

We then attempted a version of MulDA which, instead of replacing the original entities with their translations, replaces them with their original forms. This version is called "MulDA Partial", while the version that replaces the entities with their their translated forms is called "MulDA Full". The rationale behind doing this is that it will increase diversity in the entities found in the dataset since now entities must be recognized even if they are from a language different than the rest of the text. In these names, "Partial" refers to the fact that the sentence is only partially translated (everything but the entities) when it is added as an example to the dataset.

	Fully Translated	Linearized	Stabilized
MulDA Partial			
MulDA Full	X		
Stabilized Partial			X
Stabilized Full	X		X
Linearized Partial		X	
Linearized Full	X	X	

Table 1: This figure shows which techniques are used for our translation techniques

4.2 Our Translations

On top of the MulDA translation technique, we experimented with variants and subsets of the process to identify the importance of each step and eliminate unnecessary steps if possible. To specify, we have the "linearized" variant, which uses a non-bracket translation scheme from DAGA (Ding et al., 2020), the "stabilized" variant, which uses brackets and discards "unstable" translations, and other replacing Google Translate with other pretrained translation models like T5 and Helsinki.

4.2.1 Linearized

We were introduced to DAGA (Ding et al., 2020) through MulDA (Liu et al., 2021) paper, which is an augmentation method with language models trained on linearized labeled sentences. Linearization is the process of inserting entity tags before the corresponding word as shown in figure 5 (Ding et al., 2020).

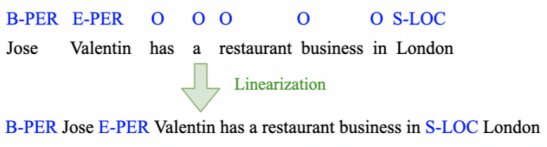

Figure 5: An example of labeled sentence linearization

We took step 3 of MulDA and DAGA's linearization idea and made our translations based on that. For each training sample, we built a linearized sequence, more specifically we used brackets to mark

the span of each entity and then translated the sequence to the target language. Figure 6 displays a linearized sequence with brackets that will be sent to the Google cloud for translation. This linearization method helped us to debug the translated texts and find the corresponding words easier. In order to avoid having tags translated, we replaced them with "UNK" tokens and save the tags and their corresponding words in a dictionary and retrieve them after translation.

src:
the county seat is [B-HumanSettlement saint-georges]
What is sent to Google Cloud:
the county seat is [UNK saint-georges]
tgt:
le chef-lieu est B-HumanSettlement saint-georges

Figure 6: An example of using brackets in a linearized sequence

We attempted an additional technique that does not translate the entities. For this technique, we first linearized the text, but instead of putting brackets around the linearized text, we instead put quotes and additional span tags surrounding the text which indicated that the entities should not be translated by Google Translate. For an example of how this was done, see Figure 7.

src: heron was born in [HumanSettlement welwyn garden city] in 1949.

sent to Google Translate:
heron was born in "B-HumanSettlement welwyn" "I-HumanSettlement garden" "I-HumanSettlement city" in 1949.

tgt: heron est né à "B-HumanSettlement welwyn" "I-HumanSettlement garden" "I-HumanSettlement city" en 1949.

output: heron est né à [HumanSettlement welwyn garden city] en 1949.

Figure 7: How the partial method translates text

4.2.2 Stabilized

Stabilized is the closest to the original MulDA paper out of our translation techniques and generates two augmented datasets (Stabilized Full and Stabilized Partial).

In the Stabilized data augmentation method, we do not complete Step 1 of MulDA (illustrated in Figure 4). Instead, we begin with Step 2, putting brackets around entities and translating via Google Translate. Then, after translating the full sentence once for each entity, we check to see if the translations match each other. If not, we discard the example, and don't include it in either of the output datasets. We theorize that this "stabilizes" Google Translate, leading to better performance than Full or Partial. That is, if Google Translate gives two different translations just due to brackets around different words in the source text, this likely indicates that Google Translate is not particularly stable or proficient at translating that sentence. Thus, by removing results where Google Translate is not stable, we increase the quality of the dataset.

Overall, stabilization had a significant effect on the number of examples introduced to the two Stabilized datasets. Out of 16778 examples, a full 4149 examples were dropped due to Google Translate not translating the examples in a stable fashion. (An additional 72 sentences were dropped due to invalid bracketing before translation, and 52 sentences were dropped due to brackets not being found after translation, leading to a total of 4273 dropped examples.) Since the dropped examples are exactly the same between Stabilized Full and Stabilized partial, the two datasets provide a direct contrast to one another; in the Stabilized Full dataset, entities are translated, whereas in the Stabilized Partial, they are not.

4.2.3 Bracket Choice

We also ran a small-scale experiment to determine which type of brackets led to the most stabilization. We iterated through square brackets, curly braces, double quotes, angular brackets, and parentheses. Ultimately, the experiment determined that using double quotes (" ", akin to Partial in Section 4.2.1) worked the best for stabilization, with only 3481 examples needing to be dropped due to stability in comparison to square quotes' 4149 examples. However, we decided to use square brackets for our full report because 1. It more closely matched the MulDA technique (which uses square brackets) and 2. When translating from English to French, Google Translate modified the quotes in 1823 examples, often times converting them into French quotes known as "les guillemets": « ». Since the quotes were modified, the algorithm could not find the boundaries between the translated entities and the rest of the text, and therefore these examples needed to be dropped as well. Thus, overall, trans-

lating using square brackets led to more examples being added to the datasets, despite the decrease in stability.

4.2.4 T5 and Helsinki

Training samples of both English and French datasets were also translated by pretrained language models using Huggingface transformers library. For English to French translation, we used T5-small model (Raffel et al., 2020). We also used Helsinki-NLP/opus-mt-fr-en model (Tiedemann, 2020; Tiedemann and Thottingal, 2020) that was specifically trained for French to English translation since T5-small model only supports one-way translation from English to French, German, and Romanian.

4.3 Knowledge Base (KB-NER)

We compare our perforamnce with Knowledge Base-Named Entity Recognition (KB-NER) (Wang et al., 2022) to compare with the previous state-of-the-art named entity recognition model. KB-NER, the winner of MultiCoNER I, uses knowledge-base augmentation to append relevant context to the input sentence to allow the NER model to attend to "external" knowledge.

5 Experiments and Results

We fine-tune a pre-trained named entity recognition model on various sets of data. The named entity recognition model consists of a pre-trained XLM-RoBERTa-base (Conneau et al., 2019) model with a conditional random field classier on top. This setup is derived directly from the baseline model from MultiCoNER 1 and was utilized by all top-performing teams. Per convention, we trained the model using the AdamW optimizer with a learning rate of 1e-5. We trained each model for 20 epochs, which took about 2.5 hours with a single A40 GPU on the Minnesota Supercomputing Institute's Agate cluster. We used macro-averaged validation F1 score as the main evaluation metric for comparing the performance of models trained on various datasets.

To allow for deeper analysis of the performance and error, we restrict our set of languages to a single pair, English and French, out of 13 languages from the MultiCoNER II task. As the test data has not been released, we report performance on the development set, which was not seen during training.

5.1 Experiment 1: Mulda Translation

	F1	P	R
EN	0.7798	0.7741	0.7855
EN-M-P	**0.8063**	0.7986	**0.8140**
EN-M-F	0.7898	**0.8046**	0.7755
FR	0.8214	0.8145	0.8284
FR-M-P	**0.8298**	0.8196	**0.8402**
FR-M-F	0.8224	**0.8261**	0.8188

Table 2: Validation F1 score, precision, and recall for MulDA translations.

Both MulDA translation techniques, full and partial, improved the macro averaged F1 scores on the development set. However, it was surprising to see that our variant, MulDA-Partial, which does not translate the entity from the source language, performed better than MulDA's original technique, MulDA-Full. For both languages, the partial version performed 0.05 - 0.15 points better than their full counterpart. This suggests that step 3 of MulDA's translation technique which translates the entities to the target language, despite being intuitive, adds unfavorable bias to the dataset. Furthermore, for both languages, the precision is higher for the full version while the recall is higher for the partial version.

5.2 Experiment 2: Our Translation Techniques

	F1	P	R
EN	0.7798	0.7741	0.7855
EN-S-P	0.7804	0.7822	0.7785
EN-S-F	**0.7953**	**0.7889**	0.8017
EN-L-P	0.7841	0.7850	0.7832
EN-L-F	0.7939	0.7692	**0.8202**
EN-H	0.7203	0.7125	0.7284
FR	0.8214	0.8145	0.8284
FR-S-P	**0.8229**	**0.8160**	**0.8299**
FR-S-F	0.8070	0.8006	0.8136
FR-L-P	0.8073	0.8121	0.8025
FR-L-F	0.8098	0.8150	0.8047
FR-T	0.7868	0.7780	0.7959

Table 3: Validation F1 score, precision, and recall for our translations.

Our attempts to replace or take the subset of the MulDA translation generally worsened performance compared to MulDA-Partial, the best

performing MulDA. To specify, the best performing technique for English, Stable-Full, was 0.011 points lower in F1 score, while the best performing technique for French, Stable-Partial, was 0.055 points lower in F1 score. It is also noting that the Stable-Partial, Linear-Partial, and Linear-Full of French and the pretrained translation for both languages actually performed worse than the not augmented baseline, showing that naively translating the text is not enough to improve performance. We hypothesize that this is due to inconsistencies in the alignment between the source and the target text causing misplaced named entity tags for the translated results. We have observed many of these cases from our translation techniques and to the best of our efforts attempted to filter out those inconsistencies, but failed to improve the F1 scores above this point.

5.3 Experiment 3: Knowledge Base Augmentation

	F1	P	R
EN	0.7798	0.7741	0.7855
EN-KB	0.7849	0.7922	0.7778
EN-M-P-KB	0.7748	0.7826	0.7671
EN-M-F-KB	0.7710	0.7695	0.7725
FR	0.8214	0.8145	0.8284
FR-KB	0.8302	0.8146	0.8465
FR-M-P-KB	0.8268	0.8434	0.8109
FR-M-F-KB	0.8262	0.8367	0.8160

Table 4: Validation F1 score, precision, and recall for top performing translations + knowledge base augmentation.

This experiment compares the efficacy of the translation techniques to the previous state-of-the-art best-performing work, KB-NER (Wang et al., 2022). To specify, as KB-NER merely appends the relevant context after the original input, it can also be applied to translation-augmented text. For the training data, we append the relevant context based on the ground truth entity spans provided in the data. For the development data, to simulate a situation where we do not know the ground truth entities, we first predict the entities using our NER model trained on not augmented data and use those predicted spans to generate and append the context.

We successfully show that the performance of MulDA translation is better than that of knowledge-base-based augmentation for English by 0.022 F1 score and comparable for French (worse by 0.0004). However, despite our attempt to seamlessly combine these two techniques, we could not improve the results over information retrieval augmentation. To specify, knowledge-augmenting the translation-augmented dataset worsened the F1 score by 0.005 - 0.01 points.

6 Error Analysis

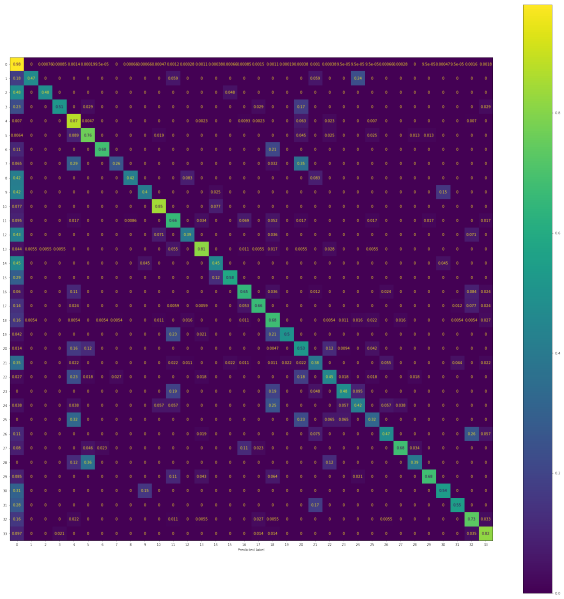

Figure 8: Normalized confusion matrix between 34 classes, disregarding the difference between beginning and intermediate tokens. We see that 'O' is commonly incorrectly predicted across entity types, and Other<type> are commonly confused with other more specific versions of their broader category.

Through all of the techniques tried in this report, common themes emerged. Overall, the most common source of error was predicting an entity type when there was none or failing to predict any entity type. This also makes sense, due to the vast imbalance of token without an entity type label. Looking at the top 10 most common errors across augmentation techniques, the most common difficulties across both English and French in either failing to recognize an entity or predict one where there isn't occurred with products ('OtherPROD'), creative works ('VisualWork', 'MusicalWork', 'WrittenWork'), and groups ('ORG'). In English, occasionally difficult would occur with the medical category ('Disease'). In French, locations ('Facility', 'HumanSettlement'), creative works ('ArtWork') and people ('Artist') occasionally appeared. This may be due to the difficulties in distinguishing be-

tween longer complex names and regular writing.

In terms of confusing entity types for another, the most common errors were confusing the person types of 'Artist', 'Athlete', 'Politician', and 'OtherPER'. The next most common included confusing the creative work types of 'Software' and 'VisualWork', and the group types of 'PublicCorp' and 'ORG'. In French, 'SportsManager' was also occasionally confused with 'Athlete'. This points to the difficulty of distinguishing between similar names without outside knowledge about their role in society, and that based on text alone it is difficult to determine the characteristics of a catch-all category label.

When retaining the distinction between tokens that begin an entity (starting with a 'B-') and those that follow it (starting with a 'I-'), it is noticeable that the most common errors do not include confusing correct entity types for their respective beginning or intermediate version. While both beginning and intermediate entity types were occasionally confused for the absence of an entity, this suggests that the differences between fundamental types of entities is much more difficult of a problem than understanding weather a token starts or is a continuing part of an entity.

A visualization of the confusion matrix when distinction between beginning and intermediate token labels are removed (normalized for the percentage those in the class, as the class labels are heavily skewed) can be seen in figure 8.

7 Limitations

7.1 Google Translate

We used Google Cloud API as our translation tool but it posses a few limitations. The first limitation that we found was that the tags such as "MED-Symptom" were also getting translated that we had to store and replace them before translation and then restore them for the final sequence. second limitation was that sometimes the order of the tokens changed after translation and in this case we used back-translation to find the corresponding tag for entities. Other limitations such as capitalization of tokens, dropping plural 's' and duplicates were also observed that were handled after translation.

7.2 Dataset Annotations

We later realized that some samples in the training sets don't have correct annotations. For example in the sentence "he holds wins over tito ortiz masakatsu funaki yuki kondo semmy schilt and minoru suzuki," all entities are athletes and should be considered athletes in context, but for some reason "yuki kondo" is left out as "OtherPER." This error seemed most common among tags of people.

We brought this up to the competition organizers, who unfortunately informed us that "the dataset is created with weak supervision. So things like these are expected. The annotations are not 100% accurate all the time." We have requested an estimate for the number of inaccurate examples, but have yet to receive a concrete reply.

8 Ethics

8.1 Environmental Risks and Cost

Data augmentation increases the size of the dataset and consequently the training time, therefore it results to higher development cost and CO_2 emissions (Feng et al., 2021). Google Cloud API offers free translation for the first 500,000 characters, after that you will be charged monthly so the translation technique with this API can't be free forever.

8.2 Bias

Organizations such as Amazon and Meta by releasing tasks with datasets such as this project, inject their own bias and values into the models and systems. For example, in this project tags such as PER-Artist or OtherPER are used for entities corresponding to people but how and what sources these organizations have used for annotation is unknown. Building models for such datasets can enforce bias or even misinformation to users.

Role Assignments

1. Asal Shavandi: MulDA implementation, Translation with pretrained language models, Full Linearized Labeled Sequence Translation, Limitations, and Ethics

2. Chahyon Ku: NER model impl. Translation (partial). MELM. Model training and hyperparameter tuning. Result and error analysis. Future work.

3. Josh Spitzer-Resnick: Broader impact. Dataset analysis. Error analysis.

4. London Lowmanstone IV: Translation (stable full and stable partial)

References

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

Beiduo Chen, Jun-Yu Ma, Jiajun Qi, Wu Guo, Zhen-Hua Ling, and Quan Liu. 2022. Ustc-nelslip at semeval-2022 task 11: Gazetteer-adapted integration network for multilingual complex named entity recognition.

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.

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, Copenhagen, Denmark. 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.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus : A survey of transformer-based pretrained models in natural language processing.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5834–5846, Online. Association for Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. 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.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert?

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. SeqMix: Augmenting active sequence labeling via sequence mixup. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8566–8579, Online. Association for Computational Linguistics.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.

Appendix

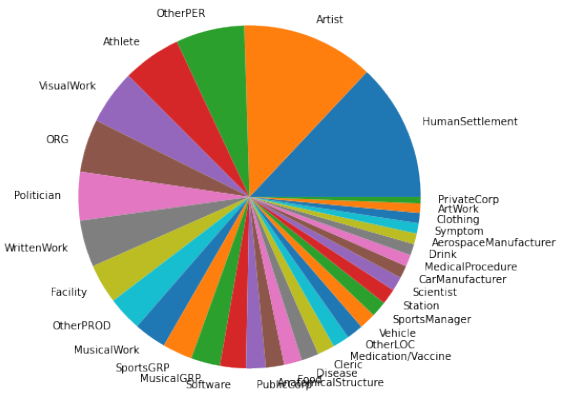

Figure 9: Distribution of label classes in MultiCoNER II across all 12 languages including the train and dev sets

language	MultiCoNER I train	dev	test	MultiCoNER II train	dev	test
Bangla (bn)	15,300	800	133,119	9,708	507	...
Chinese (zh)	15,300	800	151,661	9,759	506	...
Dutch (nl) *I*	15,300	800	217,337			
English (en)	15,300	800	217,818	16,778	871	...
Farsi (fa)	15,300	800	165,702	16,321	855	...
French (fr) *II*				16,548	857	...
German (de)	15,300	800	217,824	9,785	512	...
Hindi (hi)	15,300	800	141,565	9,632	514	...
Italian (it) *II*				16,579	858	...
Korean (ko) *I*	15,300	800	178,249			
Portuguese (pt) *II*				16,469	854	...
Russian (ru) *I*	15,300	800	217,501			
Spanish (es)	15,300	800	217,887	16,453	854	...
Swedish (sv) *II*				16,363	856	...
Turkish (tr) *I*	15,300	800	136,935			
Ukrainian (uk) *II*				16,429	851	...
Multilingual (multi) *I*	168,300	8,800	471,911			
Code mixed (mix) *I*	1,500	500	100,000			

language	MultiCoNER I train	dev	test	MultiCoNER II train	dev	test
Bangla (bn)	15,305	800	135,634	13,223	676	...
Chinese (zh)	23,831	1,281	173,183	15,228	773	...
Dutch (nl) *I*	22,331	1,157	265,942			
English (en)	23,553	1,230	272,922	25,449	1,296	...
Farsi (fa)	22,794	1,213	192,194	23,662	1,236	...
French (fr) *II*				26,377	1,352	...
German (de)	23,126	1,239	271,979	15,951	841	...
Hindi (hi)	15,956	828	144,940	12,870	684	...
Italian (it) *II*				26,443	1,398	...
Korean (ko) *I*	24,643	1,302	217,281			
Portuguese (pt) *II*				24,439	1,292	...
Russian (ru) *I*	19,840	1,042	249,458			
Spanish (es)	22,528	1,176	265,748	23,907	1,231	...
Swedish (sv) *II*				25,414	1,391	...
Turkish (tr) *I*	23,305	1,245	149,369			
Ukrainian (uk) *II*				21,956	1,135	...
Multilingual (multi) *I*	237,212	12,513	570,611			
Code mixed (mix) *I*	1,777	610	117,830			

Figure 10: Samples (top) and entities (bottom) in each dataset in MultiCoNER I and II