SeqL at SemEval-2022 Task 11: An Ensemble of Transformer Based Models for Complex Named Entity Recognition Task

Fadi Hassan[1], **Wondimagegnhue Tufa**[2], **Guillem Collell**[1],
Piek Vossen[2], **Lisa Beinborn**[2], **Adrian Flanagan**[1], and **Kuan Eeik Tan**[1]

[1]Helsinki Research Center, Europe Cloud Service Competence Center
Huawei Technologies Oy (Finland) Co. Ltd., Helsinki, Finland
{firstname.lastname}@huawei.com
[2]Faculty of Humanities, Vrije Universiteit Amsterdam
{w.t.tufa, p.t.j.m.vossen, l.beinborn}@vu.nl

Abstract

In this paper, we present a system for detecting complex named entities in multilingual and code-mix settings. We discuss the results obtained in task 11 (MultiCoNER) of the SemEval 2022 competition. The model is an ensemble of various transformer-based language models combined with a Conditional Random Field (CRF) layer. Our model ranks fourth in track 12 (multilingual track) and fifth in track 13 (code-mixed track). We describe the details of our model implementation and discuss the effect of different aggregation methods. Finally, we conduct additional analyses to understand the performance differences between languages.

1 Introduction

Named Entity Recognition (NER) is the task of identifying proper names in a text and categorizing them into predefined entity types such as person (PER), location (LOC), or creative work (CW). For example, given a sentence *"Michael Jeffrey Jordan was born in Brooklyn, New York"* the goal is to label entities correctly with their corresponding category using the BIO-scheme:

> *"Michael [B-PER], Jeffrey [I-PER], Jordan [I-PER] was born in Brooklyn [B-LOC], New [I-LOC] York [I-LOC]"*

Existing systems are often trained on standard news text and strongly rely on surface form features such as capitalization and punctuation. These approaches do not scale well to user-generated content because of the increased variation in language and context in an ever-expanding domain (Meng et al., 2021; Fetahu et al., 2021; Augenstein et al., 2017).

Examples of challenging scenarios for named entity recognition are: (a) Entities in very short text inputs such as search queries with limited or no context (Meng et al., 2021) (b) Structurally complex entities such as movie or book titles ranging from complex noun phrases to full clauses (c) Recognizing named entities in dynamically evolving contexts in which novel entities emerge (Augenstein et al., 2017; Aguilar et al., 2019).

Although many named entities are shared between languages, named entity recognition systems usually rely on language-specific cues (e.g. capitalization of nouns in German, compounding phenomena in agglutinative languages such as Korean, Japanese and Turkish (Agerri and Rigau, 2016)). Such kind of detection cues do not scale well to other languages. As a consequence, most models need to be fine-tuned for each language separately on manually annotated high-quality training data which is a costly process.

Task 12 of SemEval 2022 provides a test bench for more robust systems which can detect complex named entities in 11 languages (Malmasi et al., 2022b). The dataset intentionally contains semantically ambiguous entities with limited contexts. We focus on the multilingual tracks of the competition which requires the prediction of named entities in all 11 languages by a single model (track 12). As an additional challenge, the model is also evaluated on code-mixed data (track 13).

We summarize the main finding of our analysis as follows:

- In Sec 4.2, we show that the choice of the tagging scheme affects model performance. We observe that BILOU Tagging is more effective than BIO Tagging in our experiment albeit the total number of training is reduced when annotation is changed to BILOU. We hypothesize that this is due to explicit distinction between single and multi-token entities in BILOU which might help model perfor-

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1583 - 1592
July 14-15, 2022 ©2022 Association for Computational Linguistics

mance.

- In Sec 6.1, we compare the performance differences across languages and find a surprisingly large difference between German and Russian (10-point difference in f1). This is contrary to what is expected since the size of the training data and the label distribution across languages are comparable. We hypothesize that linguistic factors such as script and typology or factors relating to the pre-trained phase contribute to this difference.

- In Sec 6.3, we further expand on this hypothesis and experiment with a zero-shot model to examine patterns of transfer between pairs of languages. We observe higher transfer between English, Dutch and German. These are also the languages for which the model yields the highest scores individually.

In the rest of the paper, we discuss related work, experimental setup and model training, and extensive error analysis.

2 Related Work

Named entity recognition can be modeled as a sequence labeling problem. Deep-learning based approaches learn suitable representations in an end-to-end fashion and outperform rule-based and handcrafted feature-based approaches (Akbik et al., 2018; Wang et al., 2020).

(Huang et al., 2015) proposed a BiLSTM-CRF architecture for sequence tagging which is used by most state-of-the-art models. These models combine long short-term memory layers in a bidirectional fashion to use both past and future input and predict named entity sequences using a conditional random field layer. Significant performance gains have been obtained by initializing the model with pre-trained contextual embedding models such as BERT (Devlin et al., 2019), Flair (Akbik et al., 2018) and LUKE (Yamada et al., 2020).

Subsequent works explore some of the limitation of using a vanilla transformer. (Guo et al., 2019) show that a transformer architecture is less effective for modeling sequence labeling that strongly relies on left and right context and long-range dependencies which is the case for named entity recognition. LUKE (Yamada et al., 2020) is the state of the art in the CoNLL-03 NER dataset. It is pre-trained by contextualized representations based on bi-directional transformers on entity-annotated corpus of words and entities.

The complexity of the task and its multilingual nature are the main factors in choosing our modeling approach. The task complexity entails that our approach should rely on token context and in capturing relationships between labels since the surface form cues (e.g capitalization) are normalized in the training data. The multilingual aspect entails choosing a crosslingual pre-trained model which can handle the target languages. We choose `XLM-RoBERTa-large` (Conneau et al., 2019) and `Microsoft/infoxlm-large` (Chi et al., 2020). `XLM-RoBERTa-large` model is a cross-lingual version of RoBERTa. XLM-RoBERTa has outperformed cross-lingual BERT and it is the state of the art on many cross-lingual tasks including Named Entity Recognition. `Microsoft/infoxlm-large` is similarly a multilingual pre-trained model for over 100 languages with a new cross-lingual pre-training task named cross-lingual contrast (XLCO).

A comparison of the two pre-trained models shows both to be competitive on tasks such as cross-lingual natural language inference (XNLI) and `Microsoft/infoxlm-large` to be significantly better on cross-lingual question answering (MLQA) and cross-lingual sentence retrieval on the Tatoeba dataset. We provide a direct comparison of these two models for named entity recognition (which was previously missing in the literature) and explore an ensemble of the two models.

3 Data analysis

We use the training dataset provided as part of SemEval 2022 Task 11 MultiCoNER: Track 12 (Multilingual) and Track 13 (Code-mixed) (Malmasi et al., 2022a). The dataset consists of training and development data in 11 languages annotated with six named entity types. Table 1 provides statistical characteristics of the dataset. We observe that overall 18% of the tokens are labeled as a named entity which is comparable to other datasets. In terms of entity types, Person (PER), Group (GRP) and Creative Works (CW) occur more frequently across languages. We notice that the absolute number of entity tokens is twice as high for Chinese as for the other languages. This can be explained by the character-level tokenization of the Chinese texts. As a consequence, 98% of all Chinese entities are multi-token entities compared to 55% for Korean and 85% for English. Figure 1 visualizes the entity density across languages showing a large difference for Chinese but only small variations for the other languages. This difference may be smoothed by subtoken representations of the language models. The Chinese characters can not be broken any further, whereas the other language tokens can.

	Multilingual	Code-Mixed
NER Entity(#)	6	6
Language (#)	11	N/A
Sentences (#)	168.3 K	1.5 K
Tokens (#)	2750.9 K	17.5 K
Part of Entity (%)	18	30
Outside of Entity (%)	82	70

Table 1: Summary of training data statistics

4 System Description

The system that we proposed for both track 12 and track 13 is based on an ensemble of two pre-trained transformer models (PTMs). The first one is the `XLM-RoBERTa-large` model (Liu et al., 2020; Conneau et al., 2019) which is a cross-lingual version of RoBERTa.The second one is `Microsoft/infoxlm-large` (Chi et al., 2020) which is also multilingual pre-trained model that supports over 100 languages and includes a new cross-lingual pre-training.

4.1 Fine-tuning

Both the selected pre-trained model takes as input a sequence of tokens and encodes them to the embedding space. During fine-tuning. These embeddings are passed to a dense layer that predicts class scores. On top of the class scores, we used a CRF layer.

4.2 Tagging Schemes

Several NER tagging schemes have been used in the literature. However, choosing the ideal scheme is a complex problem (Konkol and Konopík, 2015). The two most popular NER tagging schemes are BIO and BILOU. In BIO, sometimes referred to as IOB (Sang and Buchholz, 2000), a different tag is assigned to each word in the text depending on whether it is the beginning $(B-y)$, inside $(I-y)$, or outside (O) a named entity phrase y. In case of BILOU, in addition to the previous $(B-y)$, $(I-y)$ and (O) tags, words at the end of an entity phrase get an end tag $(E-y)$ and single-token entities get a unit-length tag $(U-y)$. BILOU annotations increase the amount of information related to the boundaries of named entities compared to BIO but reduce the amount of training cases per tag.

4.3 Ensemble

In our experiments on the development set, we get the best performance using an ensemble of seven models. Four of them are based on `XLM-RoBERTa-large`, and the other three are based on `Microsoft/infoxlm-large`. Furthermore, to make the set of models more diverse, we used a different random seed to initialize their weights and while we kept the same set of hyper-parameters as defined in Table 2. Finally, we used two different ensemble techniques, explained in more detail in the following sections. We provide additional information about the ensemble models in Appendix A.

4.4 Voting and Score Fusion

A hard voting ensemble involves summing the votes for crisp (discretized) class labels from our models and predicting the class with the most votes. While *soft voting* is an ensemble that involves summing the predicted probabilities for class labels and predicting the class label with the largest sum probability. To consider the context of the labels, we employed a CRF layer on top of the aggregated scores. See Figure 2.

5 Experiment and Result

All our models were implemented with PyTorch (Paszke et al., 2019), on top of the pre-trained transformer models provided by HuggingFace (Wolf et al., 2019). For the PTMs, the output of the last attention layer was used as input for the classifier layer. The CRF classifier was implemented using the AllenNLP library (Gardner et al., 2017). Adam optimizer (Kingma and Ba, 2014) was used to update model parameters. Finally, cosine annealing decay with $T\_max = 20$ and $eta\_min = 1.0e-8$ was applied for the learning rate after (early_stopping_patience / 2) consecutive epochs without improving (Loshchilov and Hutter, 2016).

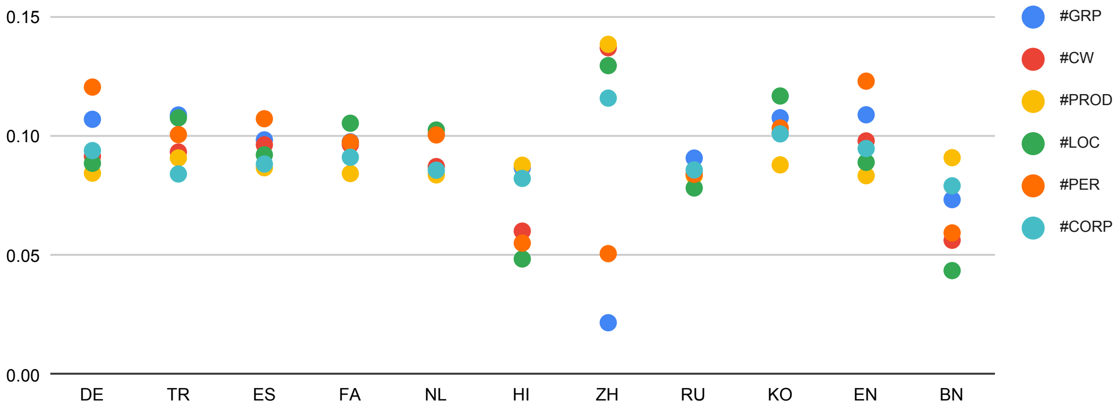


Figure 1: Named Entities Phrase Density

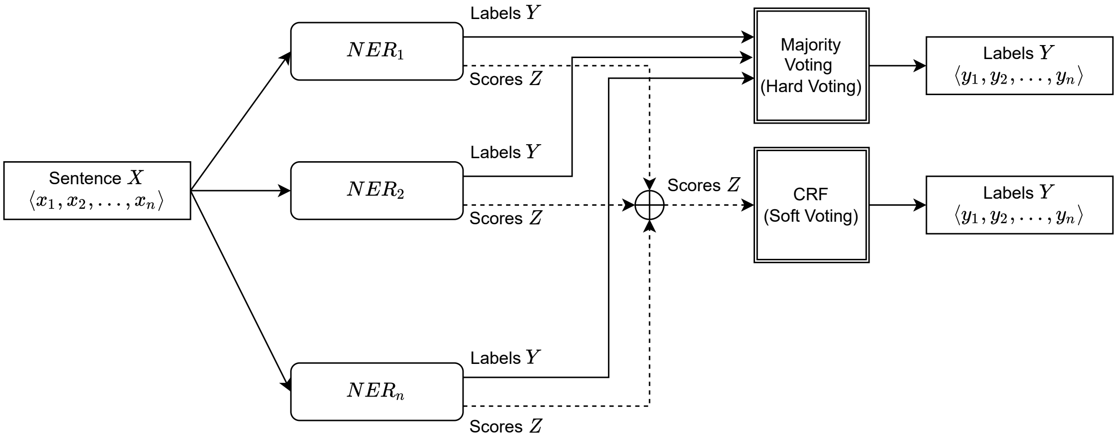

Sentence X $\langle x_1, x_2, \ldots, x_n \rangle$
NER_1
NER_2
NER_n
Labels Y
Scores Z
Labels Y
Scores Z
Labels Y
Scores Z
Scores Z
Majority Voting (Hard Voting)
CRF (Soft Voting)
Labels Y $\langle y_1, y_2, \ldots, y_n \rangle$
Labels Y $\langle y_1, y_2, \ldots, y_n \rangle$

Figure 2: Voting

Hyperparameter	value
Attention output	Last layer
Curriculum learning	Sorting by # of tokens
Padding	Batch padding
Tagging scheme	BILOU
Max sequence length	128
Batch size	32
Learning rate	5.0e-6
Learning rate decay	cosine annealing
Early stopping patience	6
Early stopping metric	Macro span-F1
Optimizer	Adam
Loss	Viterbi

Table 2: Optimizer and hyperparameters used to fine-tune our model

5.1 Model Training and Evaluation

We train our model using only the official training data. We used the development data to evaluate the performance of the models. Models were evaluated at the end of every epoch. Early stopping and cosine annealing decay were determined using the macro average span-F1 score. In the evaluation phase, task organizers provided an unlabelled test dataset. We used the pre-trained model to make the predictions without retraining and uploaded it to Codalab. The ensemble of models with score fusion provides the best results on the development and test datasets. See tables 3 and 4.

Hyperparameter selection impact During the training phase, we tested several combinations of the hyperparameters, and we used a greedy approach to select the best individual hyperparameters. Table 2 shows the most important parameters that have a significant impact on the performance

Track → Model ↓	Track 12 (Multilingual) P	 R	 F1	Track 13 (Code-Mixed) P	 R	 F1
Baseline	64.5	65.6	64.2	60.0	61.7	59.0
Best infoxlm-large	86.4	87.1	86.8	76.9	76.5	76.7
Best XLM-R-large	86.4	86.4	86.4	78.2	76.8	77.3
Ensemble voting	87.7	**87.4**	87.6	79.4	78.1	78.7
Ensemble fusion	**88.6**	87.0	**87.8**	**81.8**	**78.5**	**80.0**

Table 3: NER results on the development datasets in span-level precision (P), recall (R) and F1 in %.

of our models.

A crucial first step was the application of curriculum learning which reduced the training time by 50%. That was crucial since training our model on such a big dataset takes about one hour per epoch.

The tagging scheme is one of the most critical parameters that impact the performance of our model. For example, using BILOU scheme improved the performance about 1.5% on span-F1 score compared with the BIO scheme.

Max sequence length and batch size played a primary role in the training speed and performance. At the same time, a small value for the learning rate prevented the model from over-fitting rapidly. Finally, learning rate decay made the model convergence smoother before the early stopping occurrence.

PTM Impact As shown in Table 3, `Microsoft/infoxlm-large` and `XLM-RoBERTa-large` have almost the same performance. To the best of our knowledge, they share the same structure and are pre-trained on the same data. However, their pre-training objective functions differ. On the other hand, model size impact can be clearly seen by comparing the above large models with the `baseline` model which is based on `XLM-RoBERTa-base`.

Ensemble Impact The *score fusion* (Sect. 4.3) ensemble outperforms the individual models and the *vote-based* ensemble on almost all metrics. This improvement was due to the soft score aggregation, which gives the model better control to select the correct class than the crisp *vote-based* class aggregation.

CRF Impact Applying CRF on model scores gives better results than using argmax only. However, it was a bit hard to apply it in the score fusion ensemble model. In this type of ensemble, we aggregated the scores, not the output of the CRF layer. There were two options to solve this problem, i) take the CRF layer of one of the ensemble models and use it directly on top of the aggregated scores without fine-tuning, or ii) fine-tune a new CRF layer on the aggregated scores. We tried both solutions, and our finding was that the second option gives better performance, about 0.1% improvement in span-F1 score compared with the first option. See last row in Table 3.

6 Analysis and Conclusion

In this section, we analyze our model output for the multilingual task and the code-mixed task on the development dataset because the gold labels for the test data were not released. Table 5 shows the performance averaged over six entity types ranked by language.

We see that the model performance varies strongly between languages. The best result is obtained for German and is 10 percentage points higher than the lowest result which is obtained for Russian. When we compare the different entity types, we observe the highest variance for Chinese.

The large differences are surprising as the training data size is equal for all languages and the entity types are roughly evenly distributed (with the exception of Chinese). We therefore analyse these differences further below.

6.1 Variation Across Language

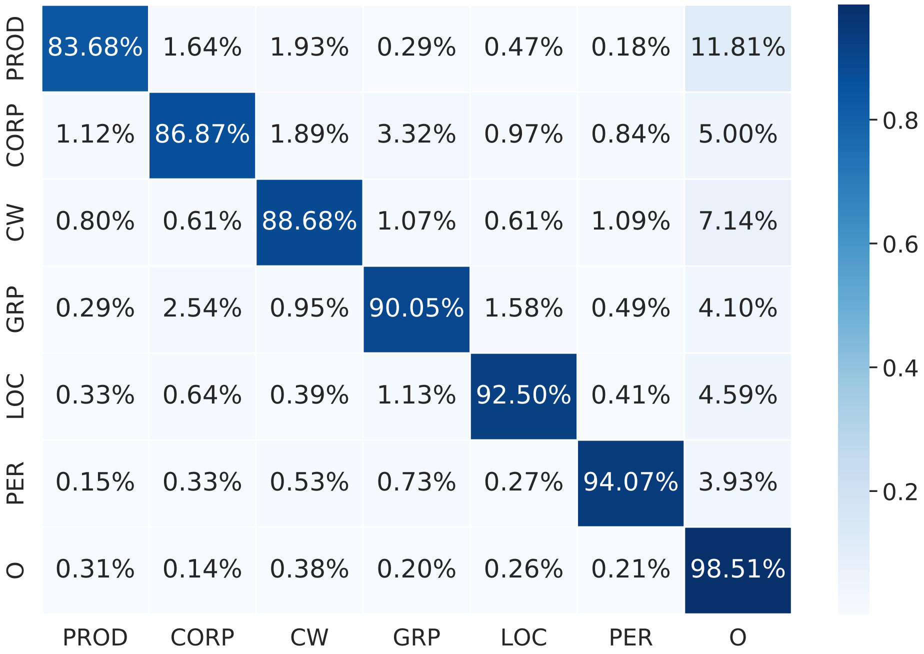
We first clustered languages into three groups based on their model performance: Group-1 has a score of 0.9 or higher and includes German, Dutch and English. Group-2 has a score between 0.85 and 0.9 and includes Turkish, Chinese, Spanish, Korean, Hindi and Bangla. Group-3 has a score lower than 0.85 and includes Farsi and Russian. With a single language model and a comparable

Track →	Track 12 (Multilingual)			Track 13 (Code-Mixed)		
Model ↓	P	R	F1	P	R	F1
Ensemble voting	74.64	75.84	74.92	79.72	79.58	79.57
Ensemble fusion	75.96	75.78	75.49	81.10	79.72	80.29

Table 4: NER results on the test datasets in span-level precision (P), recall (R) and F1 in %.

Language	DE	NL	EN	TR	ZH	ES	KO	HI	BN	FA	RU
Macro-F1	92.28	91.43	90.26	88.71	88.11	87.62	86.47	86.19	86.11	84.54	82.92

Table 5: Macro-F1 (in %) averaged over NEs (evaluated on development dataset)



	PROD	CORP	CW	GRP	LOC	PER	O
PROD	83.68%	1.64%	1.93%	0.29%	0.47%	0.18%	11.81%
CORP	1.12%	86.87%	1.89%	3.32%	0.97%	0.84%	5.00%
CW	0.80%	0.61%	88.68%	1.07%	0.61%	1.09%	7.14%
GRP	0.29%	2.54%	0.95%	90.05%	1.58%	0.49%	4.10%
LOC	0.33%	0.64%	0.39%	1.13%	92.50%	0.41%	4.59%
PER	0.15%	0.33%	0.53%	0.73%	0.27%	94.07%	3.93%
O	0.31%	0.14%	0.38%	0.20%	0.26%	0.21%	98.51%

Figure 3: Confusion Matrices on the development datasets (Multilingual + Code-Mixed)

fine-tuning dataset across languages, the disparity in the result could possibly be attributed to
(a) Representational quality: differences in language representations in XLM-R, as some languages are represented better than others
(b) Typological properties: typological difference between languages as languages that are related tend to take advantage of transfer during fine-tuning
(c) Script characteristics: languages with similar scripts tend to take advantage of shared subtokens during fine-tuning.

We observe that all group-1 languages are commonly categorized as high-resource languages for which `XLM-RoBERTa-base` performs well on downstream tasks (Conneau et al., 2019; Joshi et al., 2020). Liu et al. (2020); Hu et al. (2020) give evidence for this effect in downstream tasks.

In terms of typological properties, although group-1 and group-3 languages are members of the same language family, they differ in their scripts which to some extent negatively influence transfer at least during the fine-tuning phase (Muller et al., 2021).

6.2 Confusion Matrix

Figure 3, shows a confusion matrix for coarse-grained entity types. From these results, we observe a stronger ambiguity between the O label and the entity types than across entity types. A possible explanation can be lexical overlap in the training data. The highest confusion can be observed between Creative Work (CW) and the Out Label (O). CW often consists of titles of movies and other creative work that include words that also occur in regular expressions annotated as O.

Frequency Analysis As a follow-up, we carried out a more-detailed frequency analysis of the tokens that are annotated as entity tokens and O. Frequency analysis on the token shows a long tail distribution with more than 90% of the errors occurring only once. Table 6 shows the most frequently misclassified tokens, which are the English and Dutch definite determiner and the Chinese symbol for *Sri Lanka*.

Token	Frequency
De	36
the	22
斯	22

Table 6: Tokens which are misclassified most frequently

From the misclassified tokens, we observed substantial overlap between false negative and false positive tokens. Among these, determiners (articles such as ‘a’, ‘the’, ‘de’) and words that stand for the type of products are common ("movie", "series", "municipal"). These words are typically expected at the border of named entity expressions. There may be two explanations for these cases:

Inconsistent Annotations where a token is sometimes included in the named entity expression and sometimes it is not. This issue most likely occurs on border labels. To show this we take annotation examples from the training data e.g., *the oculus quest* as the name of a product can be annotated as [O, B-PROD, I-PROD] or [B-PROD, I-PROD, I-PROD] where *the* is annotated as outside of entity type in the first case and inside on the second case which creates inconsistency on *the* token.

Variable Contexts where the same token truly occurs both within entity phrases in some context and outside as the context change. For example in *the communist party of great britain* is annotated as [O, B-GRP, I-GRP, I-GRP, I-GRP, I-GRP] where the token *great* is annotated as GRP and in a different context - *the great sum of 1,000 pounds* it is annotated as outside of entity. Both cases are difficult to resolve for a model.

Token	Annotated Labels	Label Distribution
his	I-CW, O, I-PER	[0.01, 0.99, 0.00]
songs	I-CW, O, B-CW	[0.05, 0.83, 0.12]
since	I-CW, O, B-CW	[0.01, 0.99, 0.01]

Table 7: Examples for label variation

We explored the first cause where the issue might arise from an inconsistent annotation in the training data. We first extract tokens from the training data with multiple labels along with their corresponding label proportion. In Table 7 some examples are given, where "songs" also tend to occur at the beginning of a CW. We then use this proportion to create a post processing filter where we "correct" the model output at the borders of entity phrases for tokens with overlapping False Positive and False Negative cases in case the model output deviates from the bias.

We experimented with different thresholds for the bias to apply where the maximum value represents a bias value and the minimum value represents the exception value. Although this approach did not result in a performance gain or loss, we observe that closed class words such as articles but also proper names for locations, product names, proper nouns and symbols are often affected by this filter. A possible explanation for lack of effect could be that the same annotation inconsistency also applies to the test data.

6.3 Transferability

To analyze how transfer plays out between the group-1 languages, we fine-tuned `XLM-RoBERTa-base` on the English dataset and evaluated it in a zero-shot setting on the rest of the languages. Figure 4 shows the result of this experiment. Though other factors might play a role, we can infer from this result that group-1 languages (German and Dutch) have a more positive transfer from English than the other languages, although Spanish also benefits from English.

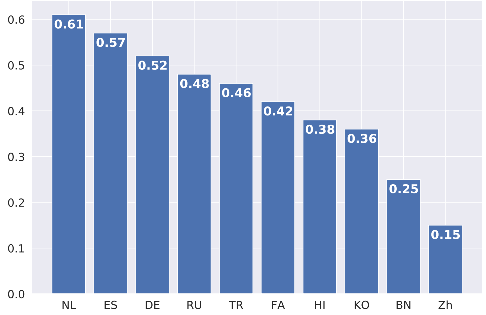


Figure 4: English Zero shot performance across languages

	NL	NL-DE	NL-ZH	NL-Rand ZH	Den-ZH	Den-DE
GRP	0.82	0.8	0.76	0.76	0.02	0.11
CW	0.72	0.68	0.67	0.71	0.14	0.09
PROD	0.69	0.62	0.61	0.69	0.14	0.08
LOC	0.87	0.88	0.84	0.86	0.13	0.09
PER	0.88	0.82	0.86	0.84	0.05	0.12
CORP	0.84	0.78	0.77	0.76	0.12	0.09

Table 8: Bilingual Models evaluated on Dutch Development Dataset (Macro F1)
The last two Column (Den-ZH and Den-DE) shows Named entity density measures

Next to zero-shot transferability, we also experimented with bilingual transfer. We selected one language as the target language, in this case, Dutch, and we measured the contribution of all other languages as training data in addition to half of the Dutch training data. We combine half of the dutch training data with half of German (Column NL-DE), half of Chinese(Column NL-ZH) and randomized Chinese (Column NL-Rand ZH) where we randomize Chinese tokens with a token from XLM vocabulary. The first four column shows bilingual models evaluated on Dutch development set measured in macro-averaged F1. The total set of training sentences was kept the same across all experiments.

We observe that contrary to the zero-shot results, Chinese contributes overall only just a bit lower than German when tested on the Dutch test set. This is remarkable because German and Dutch are typologically very close and use the same script. Apparently, the observed density of entities for Chinese is a factor that may compensate for the difference in script and language typology. We can see in Table 8 that the contributions of Chinese lag behind when the density is lower than German (GRP) but is almost the same when it is higher (CW, PROD, LOC, CORP). The only exception is PER which has the lowest density for Chinese but still a higher contribution. To test the assumption that just the label density plays a role, we even replaced the Chinese tokens with random tokens. The results show that even partially randomized Chinese training data outperforms the German contribution on most entity types.

6.4 Conclusion

In this paper, we proposed a single named entity recognition system that can process multilingual and code-mixed text based on an ensemble of transformer-based models. We have accomplished fourth and fifth positions in the test phase for track 12 (Multilingual) and track 13 (Code-Mixed). Even though the proposed system performs pretty well on the development dataset, there is a considerable performance drop in track 12 on the test dataset. Further study needs to be done to address that performance change.

Summarising the results from the error analysis and the statistics on the training data, we can conclude that there are four factors that play a role in the cross-lingual performance of this task, given that an equal amount of training data is available for fine-tuning in all languages. We provided evidence that transfer from the XLM pre-training, typological relatedness, and shared scripts can be factors that contribute to transfer but on the other hand the density of the entities in the training data is another factor.

Our system does not include external gazetteers or targeted unsupervised learning on difficult entity types such as products and creative works. In future work, we would like to include them, which could help to improve the performance due to the extra information they include.

Acknowledgements

The research by the authors of affiliation 1 and by Wondimagegnhue Tufa was funded by Huawei Finland.

References

Rodrigo Agerri and German Rigau. 2016. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, 238:63–82.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2019. A multi-task approach for named entity recognition in social media data. *arXiv preprint arXiv:1906.04135*.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *CoRR*, abs/1701.02877.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

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

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.

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

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. 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 1315–1325, Minneapolis, Minnesota. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Michal Konkol and Miloslav Konopík. 2015. Segment representations in named entity recognition. In *International Conference on Text, Speech, and Dialogue*, pages 61–70. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

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

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

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

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward

Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Erik F Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. More embeddings, better sequence labelers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3992–4006, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

A Ensemble Models

In Table 9 and Table 10, we provide score distribution of all models that form our ensemble on the development datasets.

#	PTM	Random Seed	P	R	F1
1	microsoft/infoxlm-large	102	87.6	85.6	86.6
2	microsoft/infoxlm-large	2022	86.4	87.1	86.8
3	microsoft/infoxlm-large	2033	87.5	85.7	86.6
4	xlm-roberta-large	2044	85.8	86.5	86.2
5	xlm-roberta-large	2055	86.4	86.4	86.4
6	xlm-roberta-large	2066	85.9	86.8	86.4
7	xlm-roberta-large	2077	86.2	86.4	86.3

Table 9: Ensemble models - Multilingual

#	PTM	Random Seed	P	R	F1
1	microsoft/infoxlm-large	102	77.6	75.4	76.5
2	microsoft/infoxlm-large	2022	76.9	76.5	76.7
3	microsoft/infoxlm-large	2033	78.1	74.9	76.5
4	xlm-roberta-large	2044	78.2	76.8	77.3
5	xlm-roberta-large	2055	78.2	76.8	77.3
6	xlm-roberta-large	2066	77.3	77.1	77.2
7	xlm-roberta-large	2077	77.1	76.1	76.6

Table 10: Ensemble models - Code-Mixed