DRU at WojoodNER 2024: ICL LLM for Arabic NER

Nancy Hamdan and **Hadi Hamoud** and **Chadi Abou Chakra**
and **Osama Rakan Al Mraikhat** and **Doha Albared** and **Fadi A. Zaraket**
Arab Center for Research and Policy Studies, Doha
{nhamdan,hhamoud,cabouchakr,oalmraikhat,dal007,fzaraket}@dohainstitute.edu.qa

Abstract

This paper details our submission to the WojoodNER Shared Task 2024, leveraging in-context learning with large language models for Arabic Named Entity Recognition. We utilized the Command R model, to perform fine-grained NER on the Wojood-Fine corpus. Our primary approach achieved an F1 score of 0.737 and a recall of 0.756. Post-processing the generated predictions to correct format inconsistencies resulted in an increased recall of 0.759, and a similar F1 score of 0.735. A multi-level prompting method and aggregation of outputs resulted in a lower F1 score of 0.637. Our results demonstrate the potential of ICL for Arabic NER while highlighting challenges related to LLM output consistency.

1 Introduction

Named Entity Recognition (NER) is an essential component of information extraction that involves identifying and classifying named entities in a given text into predefined categories such as names of organizations, people, and locations. NER acts as an important pre-processing step for a variety of downstream tasks including text summarization and question answering (Sang and Meulder, 2003). Figure 1 shows an Arabic NER example. Flat NER identifies non-overlapping entities in a text, while nested NER recognizes entities within other entities such as 'Strip' in the third row which is a geopolitical entity nested within the event entity.

Datasets and methodologies differ for each NER type, with specific corpora and annotation techniques for each one. The Wojood corpus (Jarrar et al., 2022) is a significant contribution for Arabic NER, containing about 550K tokens in Modern Standard Arabic and various dialects, annotated with 21 entity types such as person, organization, and location. The Wojood-Fine corpus (Liqreina et al., 2023) extends this with 31 subtypes for categories such as geopolitical entities and facilities. Both corpora include flat and nested versions.

Translation		Nested		Flat
إعادة*ʾiʿādh*	Re-	O	O	O
استيطان*āstyṭān*	Settlement	B-EVENT	B-EVENT	B-EVENT
قطاع*qṭāʿ*	Strip	I-EVENT	B-GPE B-S-O-P	I-EVENT
غزة*ġzh*	Gaza	I-EVENT	I-GPE I-S-O-P	I-EVENT

Figure 1: Arabic NER nested and flat example. **S-O-P:** STATE-OR-PROVINCE.

The WojoodNER shared task series (Jarrar et al., 2023) aims to advance Arabic NER research using Wojood and Wojood-Fine. The 2024 shared task (Jarrar et al., 2024) introduced new subtasks, including an "open track" subtask (Subtask 3) encouraging innovative systems using external datasets and tools, focusing on NER's real-world utility through generative models. This paper presents our submission to the open track for the flat NER subtask (Subtask 3A).

In-context learning (ICL) with large language models (LLMs) has emerged as a promising approach for NER. ICL enables LLMs to perform tasks by utilizing prompts with a few demonstrative examples per category.

Our method proceeds as follows:

- We created a detailed *novel* system prompt based on Wojood-Fine annotation guidelines.
- We instructed an LLM model that we have selected with the prompt and predicted fine-grained flat NER.
- We post-processed the output resolving LLM inconsistency format issues with detailed steps to extract consistent predictions.
- We evaluated the results considering (i) full compliance with the specified output format, and (ii) after post-processing, allowing lenient format.

2 Related Work

Recent studies explored different techniques and frameworks to enhance ICL for NER. GPT-NER transforms sequence labeling into a generation task, classifying entities with corresponding tokens. To

Proceedings of The Second Arabic Natural Language Processing Conference, pages 885–893
August 16, 2024 ©2024 Association for Computational Linguistics

Table 1: Subtask 3 dataset statistics.

	#Sentences	#Tokens	#Ent Flat	#Ent Nested
Train	1,928	50,144	16,096	18,111
Dev	372	10,049	3,348	3,807

Table 2: Subtype entity counts in Subtask 3.

		Flat		Nested	
Tag	**Sub-type Tag**	**Train**	**Dev**	**Train**	**Dev**
GPE	COUNTRY	187	37	232	39
	STATE-OR-PROVINCE	1,002	151	1,056	172
	TOWN	235	35	271	50
	NEIGHBORHOOD	6	0	6	0
	CAMP	16	0	16	0
	GPE_ORG	81	19	86	20
LOC	CONTINENT	0	0	2	0
	CLUSTER	30	6	31	6
	BOUNDARY	18	11	18	11
	WATER-BODY	6	4	6	4
	LAND-REGION-NATURAL	7	2	15	2
	REGION-GENERAL	187	54	128	39
	REGION-INTERNATIONAL	4	0	8	0
ORG	GOV	1,107	280	1,146	293
	COM	187	40	189	40
	EDU	27	0	27	0
	NONGOV	846	204	872	209
	MED	266	61	266	61
	SCI	112	28	112	28
	ORG_FAC	118	28	118	28
FAC	AIRPORT	3	0	3	0
	BUILDING-OR-GROUNDS	162	32	162	32
	Total	**4,607**	**992**	**4,770**	**1,034**

mitigate hallucination, self-verification prompts the LLM to confirm if the extracted entities belong to a label (Wang et al., 2023).

C-ICL enhances entity extraction by using correct and incorrect samples in prompts, improving the model's accuracy through reasoning and error correction (Mo et al., 2024). Similarly, P-ICL uses point entities as auxiliary information in prompts, enhancing the LLM's ability to recognize each entity type (Jiang et al., 2024). For few-shot nested NER, researchers developed a framework that includes a prompt with task instructions, demonstrations, and possible labels, and developed an example selection mechanism called EnDe retriever to optimize prompt effectiveness (Zhang et al., 2024).

Researchers evaluated GPT-3.5 and GPT-4 for Arabic NLP tasks (Abdelali et al., 2024), including NER, using zero-shot learning and further evaluated GPT-4 using few-shot learning on datasets such as ANERcorp (Benajiba et al., 2007), Aqmar (Schneider et al., 2012), and QASR (Mubarak et al., 2021). The study revealed patterns of errors for sequence tagging tasks like NER and POS tagging that included format deviations and incorrect token generation which led to performance drops.

3 Dataset

The NER shared task dataset for Subtask 3 includes data from five news domains related to the war on Gaza. The subtask provides training and development sets that were manually annotated with fine-grain named entities using the annotation guidelines described in (Liqreina et al., 2023). The dataset includes both flat and nested versions. The training set contains 1,928 sentences with a total of 50,144 tokens, while the development set has 372 sentences with 10,049 tokens, as shown in Table 1. Table 2 details the counts of the different entities for each split of the dataset.

4 LLM Selection

After evaluating several open source LLMs, we selected Cohere's Command R model (Command R Team, 2024) as our chosen LLM. Command R is an open-weights generative model specifically optimized for long-context tasks such as retrieval-augmented generation (RAG). This model contains 35 billion parameters and supports a context length of up to 128,000 tokens, making it well-suited for extensive text processing tasks.

Command R is proficient in handling 10 major languages: English, French, Spanish, Italian, German, Portuguese, Japanese, Korean, Arabic, and Chinese. Notably, it ranks as the second-best model on The Open Arabic LLM Leaderboard (Elfilali et al., 2024), which assesses the performance of Arabic LLMs. The evaluation datasets include native Arabic benchmarks such as AlGhafa (Almazrouei et al., 2023) and ACVA (Huang et al., 2023), focusing on reasoning, language understanding, and commonsense tasks.

Despite its smaller parameter size, Command R achieved an average score of 54.43, closely trailing the Llama3 model with 70 billion parameters, which scored 59.86. This performance underscores Command R's efficiency and capability. For our experiments, we employed the 8-bit quantized version of the Command R model, which balances performance and resource utilization effectively.

5 Prompt Design

To explain the task to the model, we devised a detailed system prompt. It outlines the steps for tagging tokens according to the Wojood-Fine annotation guidelines (Liqreina et al., 2023). The prompt helps the model predict different tag levels for each token and specifies the required output format, ensuring correct parsing for evaluation. The full prompt is listed in Appendix A.

The initial sentences in the prompt instruct the

Table 3: Evaluation results considering the ‘O’ in All, and excluding it in ’Organizers’. **CRPrompt**: strictly following the system prompt format. **CRPrompt6**: after post-processing the six cases. **Multi-level**: the multi-level prompt method.

	All			Organizers		
	P	**R**	**F1**	**P**	**R**	**F1**
RPrompt	0.719	0.756	0.737	0.265	0.330	0.294
CRPrompt6	0.713	0.759	0.735	0.264	0.355	0.305
Multi-level	0.634	0.639	0.637	0.136	0.209	0.165

LLM to perform highly accurate NER for Arabic text. The model is expected to understand Arabic thoroughly and predict the appropriate named entity tags for each token in a provided list, following structured steps.

To tag the tokens correctly, the model needs to predict up to three levels of tags. The first level is the high-level tag from Wojood’s original 21 tags. The second level is a subtype, applicable only for GPE, ORG, FAC, and LOC tags. The third level is needed only if the second level tag is GPE_ORG or ORG_FAC, identifying the specific subtypes from GPE and ORG, or ORG and FAC.

To simplify the problem, the prompt instructs the model to predict only two tag levels: L1 (high-level tag) and L2 (subtype). It lists possible L2 tags based on L1 (GPE, ORG, FAC, or LOC) and their definitions. The model is instructed to output predictions in CSV format: token in the first column, L1 tags in the second, and L2 tags in the third, with the most probable tags listed first.

To handle cases where GPE_ORG or ORG_FAC are subtypes, these tags are removed from the L2 list. Instead, the model is instructed that L1 tags can include both GPE and ORG, or ORG and FAC. Few-shot examples demonstrate this, e.g., L1 could be “B-GPE B-ORG” and L2 “B-COUNTRY B-GOV,” or L1 “B-ORG B-FAC” and L2 “B-EDU B-BUILDING-OR-GROUNDS.” We reconstruct the correct subtype tag in post-processing. The prompt also instructs the model to follow the BIO schema, with “B-” marking the beginning and “I-” marking the inside of an entity.

We added 27 illustrative examples of input sentences and output predictions to the prompt. These sentences were chosen manually, preferring shorter sentences with diverse examples and more non-“O” tags. The format of the few-shot examples is shown in Appendix B.

5.1 Prompt by AI Agency

We experimented with ChatGPT to automatically create prompts that generate NE tags according to the guidelines. However, the experiments were not successful.

6 Output Quality Issues

After inspecting Command R’s generated outputs, we identified several quality issues. Firstly, the model sometimes produced extra tokens or missed tokens for a given input sentence. Although there are 10,049 tokens in the development set, the model outputted only 9,915 tokens because some predicted sentence outputs were either shorter or longer than their corresponding ground truth sentences. Secondly, the model tagged some tokens with tags it had not encountered in the prompt before, resulting in 14 different hallucinated tags.

Thirdly, only 9,416 out of the 9,915 token predictions generated by the model strictly conformed to the specified output format. For the remaining tokens, there were several instances of incorrect output formats. We observed a total of six distinct cases of wrong output format that are amenable to postprocessing as shown in Appendix C.

Although the formats were syntactically incorrect, their tags semantically made sense upon inspection. We handled the six cases using a post processing step and we report results with and without post-processing.

6.1 Post-processing

To handle the above mentioned issues, we post-processed generated output that did not match the expected format. For every ground truth token without a corresponding predicted token, we set the predicted tag to “O", considering the order of tokens and the possibility of duplicate tokens in a sentence. If the original predicted tag was a hallucinated one, we also set the predicted tag to “O". For the remaining cases of output format issues, we converted the different cases to the expected output format as outlined in the prompt by handling them as detailed in Appendix C.

7 Split Multi-level Prompt Method

We also experimented with smaller prompts targeting subsets of named entity types and aggregated the results. First, we created separate prompts, prompt1 (11 Wojood NER enitities) and prompt2 (remaining 10). They instruct the model to tag only using the provided subset. For the output of the first

Table 4: Performance for non-zero, non-‘O’ top and bottom entities. **S-O-P:** STATE-OR-PROVINCE, **B-O-G:** BUILDING-OR-GROUNDS, **R-G:** REGION-GENERAL.

CRPrompt				**CRPrompt6**				**Multi-level**			
Label	**Precision**	**Recall**	**F1-Score**	**Label**	**Precision**	**Recall**	**F1-Score**	**Label**	**Precision**	**Recall**	**F1-Score**
LAW	0.66	0.64	0.65	LAW	0.64	0.64	0.64	PERCENT	0.82	0.40	0.54
GPE	0.48	0.79	0.60	GPE	0.45	0.85	0.59	PERS	0.33	0.73	0.45
PERS	0.46	0.81	0.59	PERS	0.45	0.81	0.58	S-O-P	0.39	0.50	0.44
ORG	0.57	0.47	0.51	ORG	0.56	0.52	0.54	TOWN	0.46	0.41	0.44
WEBSITE	0.57	0.44	0.5	WEBSITE	0.57	0.44	0.5	ORG	0.46	0.32	0.38
SCI	0.17	0.13	0.15	BOUNDARY	0.33	0.12	0.18	NORP	0.14	0.13	0.13
B-O-G	0.09	0.11	0.10	NONGOV	0.41	0.11	0.18	ORDINAL	0.33	0.07	0.11
R-G	0.06	0.05	0.06	R-G	0.16	0.17	0.16	EVENT	0.07	0.18	0.10
EVENT	0.12	0.02	0.04	EVENT	0.11	0.02	0.04	LOC	0.03	0.06	0.04
QUANTITY	0.04	0.04	0.04	QUANTITY	0.04	0.04	0.04	COUNTRY	0.02	0.06	0.03

subset, we collected the tokens classified with tags that have subtypes (GPE, ORG, FAC, and LOC) and asked the LLM to classify the extracted entities into the corresponding subtypes. We created four different ICL prompts for that. We reconstructed the predicted tags from both levels. If a sub-tag was GPE_ORG or ORG_FAC, we report the third-level tags from the results of the corresponding prompts (GPE and ORG prompts or from ORG and FAC).

8 Results

As shown in Table 3, the evaluation before post processing (CRPrompt) resulted in a precision of 0.719, recall of 0.756, and an F1 score of 0.737. The post processing of the six cases (CRPrompt6) improved recall slightly to 0.759 with a slightly lower precision of 0.713, and a similar F1 score of 0.735. Using the split multi-level prompt method (Multi-level) resulted in a decreased F1 score of 0.637, recall of 0.639, and precision of 0.634. The ‘Organizers’ column shows the Wojood Shared Task evaluation results for our different methods. Table 4 shows the top and bottom five performing tags with the different methods without considering the “O” tag.

9 Discussion

We discuss several key insights and challenges using Command R model for Arabic NER. The 0.737 F1 and 0.756 recall primary scores indicate an effective performance. Post-processing resulted in slight recall improvement with a similar F1 score. This suggests that while format corrections helped recover some missed entities, it also impacted the precision. Adherence to output formats may be better with finetuning approaches to avoid unnecessary performance penalties.

The multi-level method resulted in a lower F1 score of 0.637. We hypothesized that shorter, more specific prompts help the model focus better. However, separating entities into distinct prompts might have disrupted the LLM contextual understanding. It seems LLMs work better when performing multiple tasks that feed into each others. The potential for error propagation in aggregating multiple prompts may have introduced errors as well.

10 Conclusion

In this paper, we explored using LLMs for Arabic NER through ICL with the Command R model. Our experiments indicate that while the model has potential for Arabic NER tasks when provided with high-quality prompts, it occasionally deviates from the specified output format, necessitating additional post-processing.

Future work could focus on better example retrieval for prompts, as the selection of the few-shot examples could significantly impact in-context learning performance (Min et al., 2022). Additionally, developing more robust prompting strategies to enforce format consistency and comparing different LLMs to understand their strengths and limitations in Arabic NER could yield valuable insights. These improvements could enhance the efficiency and scalability of LLMs for NER tasks.

11 Limitations

This study has several limitations. Our system’s reliance on post-processing to correct format inconsistencies is an important consideration for anyone using our prompt and setup, as it necessitates additional steps to achieve fully compliant outputs. Furthermore, the system prompt was specifically designed for Wojood-Fine annotation guidelines, potentially limiting generalizability to other annotation frameworks within Arabic NER tasks.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Ebtesam Almazrouei, Ruxandra Cojocaru, Michele Baldo, Quentin Malartic, Hamza Alobeidli, Daniele Mazzotta, Guilherme Penedo, Giulia Campesan, Mugariya Farooq, Maitha Alhammadi, Julien Launay, and Badreddine Noune. 2023. AlGhafa evaluation benchmark for Arabic language models. In *Proceedings of ArabicNLP 2023*, pages 244–275, Singapore (Hybrid). Association for Computational Linguistics.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Command R Team. 2024. Command R documentation. `https://docs.cohere.com/docs/command-r`. Last accessed: 2024-05-15.

Ali Elfilali, Hamza Alobeidli, Clémentine Fourrier, Basma El Amel Boussaha, Ruxandra Cojocaru, Nathan Habib, and Hakim Hacid. 2024. Open arabic llm leaderboard. `https://huggingface.co/spaces/OALL/Open-Arabic-LLM-Leaderboard`.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023. Acegpt, localizing large language models in arabic. *Preprint*, arXiv:2309.12053.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa' Omar. 2023. WojoodNER 2023: The first Arabic named entity recognition shared task. In *Proceedings of ArabicNLP 2023*, pages 748–758, Singapore (Hybrid). Association for Computational Linguistics.

Mustafa Jarrar, Nagham Hamad, Mohammed Khalilia, Bashar Talafha, and Muhammad Elmadany, AbdelRahim Abdul-Mageed. 2024. WojoodNER 2024: The Second Arabic Named Entity Recognition Shared Task. In *Proceedings of the 2nd Arabic Natural Language Processing Conference (Arabic-NLP), Part of the ACL 2024*. Association for Computational Linguistics.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. *Preprint*, arXiv:2205.09651.

Guochao Jiang, Zepeng Ding, Yuchen Shi, and Deqing Yang. 2024. P-icl: Point in-context learning for named entity recognition with large language models. *Preprint*, arXiv:2405.04960.

Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. Arabic fine-grained entity recognition. *Preprint*, arXiv:2310.17333.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

Ying Mo, Jian Yang, Jiahao Liu, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-icl: Contrastive in-context learning for information extraction. *Preprint*, arXiv:2402.11254.

Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. 2021. QASR: QCRI aljazeera speech resource a large scale annotated Arabic speech corpus. 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 2274–2285, Online. Association for Computational Linguistics.

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

Nathan Schneider, Behrang Mohit, Kemal Oflazer, and Noah A. Smith. 2012. Coarse lexical semantic annotation with supersenses: An Arabic case study. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 253–258, Jeju Island, Korea. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *Preprint*, arXiv:2304.10428.

Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. 2024. In-context learning for few-shot nested named entity recognition. *Preprint*, arXiv:2402.01182.

A Detailed system prompt

You are a highly accurate model for named entity recognition (NER) from Arabic text. You understand the Arabic language very well. Given a list of Arabic tokens, predict the correct named entity tags for each token. Make your predictions according to the following steps.
1. Each token may have multiple tags up to two levels of tags (L1 and L2).
2. L1 has the following possible tags
"PERS": People names, including first, middle, last, and nicknames. Titles are not included except for Prophets, kings, etc.
"NORP": Group of people.
"OCC": Occupation or professional title.
"ORG": Legal or social bodies like institutions, companies, agencies, teams, parties, armies, and governments.
...
"O" for other or no tag.
3. L1 can have more than one tag only if the tags in it are GPE and ORG, or ORG and FAC.
4. Tokens with GPE tags for L1 might have detailed L2 tags from the following set:
"COUNTRY": Taggable mentions of the entireties of any nation.
...
"SPORT": Athletes, Sports Teams.
5. Tokens with LOC tags for L1 might have detailed L2 tags from the following set:
"CONTINENT": Taggable mentions of the entireties of any of the seven continents.
...
"REGION-INTERNATIONAL": Taggable locations that cross national borders.
6. Tokens with ORG tags for L1 might have detailed L2 tags from the following set:
"GOV": Government organizations.
...
"SPO": Sports organizations that are primarily concerned with participating in or governing organized sporting events.
7. Tokens with FAC tags for L1 might have detailed L2 tags from the following set:
"PLANT": One or more buildings used and/or designed solely for industrial
...
"PATH": Streets, canals, and bridges.
8. List the most probable tags first and start always with L1 tags.
9. Use CSV format in the output where the actual token is the first column, L1 tags in the second column separated by spaces, and L2 tags in the third column separated by spaces.
10. Follow the BIO schema where B- precedes the tag name for the start token of the named entity, and I- precedes the inside tokens of the named entity.

Example: محطة، B-ORG B-FAC, B-COM B-BUILDINGS-OR-GROUNDS

or قطارات, I-ORG I-FAC, I-COM I-BUILDING-OR-GROUNDS
The following are examples of input you will be given and output you should respond with.

B Examples of the few-shot instances given in the system prompt

Input:
بلدية
البيرة
لقرية
مزارع
النوباني

Output:
بلدية,B-ORG,B-GOV
البيرة,I-ORG,I-GOV
لقرية,B-GPE B-ORG,B-TOWN B-GOV
مزارع,I-GPE I-ORG,I-TOWN I-GOV
النوباني,I-GPE I-ORG,I-TOWN I-GOV

Input:
مدارس
الأونروا
في
مخيم
البداوي
للاجئين
في
لبنان

Output:
مدارس,O,
الأونروا,B-ORG,B-NONGOV
في,O,
مخيم,B-GPE,B-CAMP
البداوي,I-GPE,I-CAMP
للاجئين,I-GPE,I-CAMP
في,O,
لبنان,B-GPE,B-COUNTRY

C Detailed post-processing of incorrect output format cases

Case 1: Generated three columns instead of two for L1 and L2 and prediction is the special case of GPE_ORG or ORG_FAC. First and second columns represent L1, and third column represents L2.

Example predictions:
B-ORG,B-FAC,B-COM B-BUILDING-OR-GROUNDS
B-FAC,B-ORG,B-SCI

Post processing: Join the tags in the first and second columns with a space, making them the first column, and set the last column as the second column.

Case 2: Generated three columns instead of two for L1 and L2 and it is not Case 1. L1 is represented by first column only.

Example predictions:
B-ORG,B-COM,B-GOV

Post processing: Disregard the third column, assuming that B-COM is a more confident prediction than B-GOV, as the model was instructed to list the more probable tags first.

Case 3: Generated four columns instead of two for L1 and L2 and prediction is the special case of GPE_ORG or ORG_FAC. First and second columns represent L1, third and fourth columns represent L2.

Example predictions:
B-FAC,B-ORG,B-SCI,B-BUILDING-OR-GROUNDS

Post processing: Join the tags in the first and second columns with a space, making them the first column, and join the second and third tags with a space, making them the second column.

Case 4: Same as Case 3 but tags after the first tag in the first column are space separated instead of comma separated.

Example predictions:
B-FAC,B-ORG B-MED B-BUILDING-OR-GROUNDS

Post processing: Join the tags in the second column with a comma, treat the case as Case 3.

Case 5: Same as Case 3 but all tags are space separated instead of comma separated.

Example predictions: I-ORG I-FAC I-COM I-BUILDING-OR-GROUNDS

Post processing: Join the tags by comma and treat case as Case 3.

Case 6: The prediction is always the first string before the "|".

Example predictions:
O | O | O
B-FAC,B-ORG,B-SCI,B-BUILDING-OR-GROUNDS | O | O

Post processing: Extract the string before the first "|" and handle it like any of the above cases based on its length and the character it can be split by (comma or whitespace).

D Zero-performance tags in different methods

D.1 CRPrompt

GPE_ORG
LAND-REGION-NATURAL
FAC
PRODUCT
TIME
CLUSTER

D.2 CRPrompt6

GPE_ORG
LAND-REGION-NATURAL
FAC
PRODUCT
TIME
CLUSTER

D.3 Multi-level

GPE_ORG
ORG_FAC
FAC
BUILDING-OR-GROUNDS
LAND-REGION-NATURAL
REGION-GENERAL
CLUSTER
SCI
WEBSITE