MultiCoNER: A Large-scale Multilingual dataset for Complex Named Entity Recognition

Shervin Malmasi, Anjie Fang[*]**, Besnik Fetahu**[*]**, Sudipta Kar**[*]**, Oleg Rokhlenko**
Amazon.com,
Seattle, WA, USA
{malmasi,njfn,besnikf,sudipkar,olegro}@amazon.com

Abstract

We present MULTICONER, a large multilingual dataset for Named Entity Recognition that covers 3 domains (Wiki sentences, questions, and search queries) across 11 languages, as well as multilingual and code-mixing subsets. This dataset is designed to represent contemporary challenges in NER, including low-context scenarios (short and uncased text), syntactically complex entities like movie titles, and long-tail entity distributions. The 26M token dataset is compiled from public resources using techniques such as heuristic-based sentence sampling, template extraction and slotting, and machine translation. We applied two NER models on our dataset: a baseline XLM-RoBERTa model, and a state-of-the-art GEMNET model that leverages gazetteers. The baseline achieves moderate performance (macro-F1=54%), highlighting the difficulty of our data. GEMNET, which uses gazetteers, improvement significantly (average improvement of macro-F1=+30%). MULTICONER poses challenges even for large pre-trained language models, and we believe that it can help further research in building robust NER systems.

MULTICONER is publicly available,[1] and we hope that this resource will help advance research in various aspects of NER.

1 Introduction

Named Entity Recognition (NER) is a core task in Natural Language Processing which involves identifying entities in text, and recognizing their types (e.g., classifying entities as a person or location). Recently, the development of Transformer-based NER approach have results in new state-of-the-art (SOTA) results on well-known benchmark datasets like CoNLL03 and OntoNotes (Devlin et al., 2019). Despite these strong results,

Figure 1: Some examples for all the languages incorporated in MULTICONER.

there remain a number of practical challenges that may not be represented by these existing datasets.

As noted by Augenstein et al. (2017), increasingly higher scores on these datasets are driven by several factors:

- Well-formed data, with punctuation and capitalized nouns, makes the NER task easier, providing the model with additional contextual cues.

- Texts from articles and the news domain contain long sentences with rich context around entities, providing valuable signals about the boundaries and types of entities.

- Data from the news domain[2] contains "easy" entities such as country, city, and person names, allowing pre-trained models to perform well due to their existing knowledge of such tokens.

- Memorization effects, due to large overlap of entities between the train and test sets also increases performance.

Accordingly, models trained on existing benchmark datasets such as CoNLL03 tend to perform significantly worse on unseen entities or noisy text (Meng et al., 2021).

---

[*]These authors contributed equally in this work.

[1] https://registry.opendata.aws/multiconer/

[2]e.g. CoNLL03 (Sang and De Meulder, 2003)

1.1 Contemporary Challenges in NER

There are many challenging scenarios for NER outside of the news domain. We categorize the challenges typically encountered in NER according to several dimensions: (i) available context around entities, (ii) named entity surface form complexity, (iii) frequency distribution of named entity types, (iv) dealing with multilingual and code-mixed textual snippets, and (v) out-of-domain adaptability.

Context News domain text often features long sentences that reference multiple entities. In other applications, such as voice input to digital assistants or search queries issued by users, the input length is constrained and the context is less informative. Datasets featuring such low context settings are needed to assess model performance.

Additionally, capitalization and punctuation features are large drivers of success in NER (Mayhew et al., 2019). However, inputs such as search queries from users, or voice commands transcribed using ASR, lack these surface features. To understand model performance in such cases, an uncased dataset is needed.

Entity Complexity Datasets in existing NER benchmarks are often dominated by entities representing persons, locations, and organizations. Such entities are often composed of proper nouns, or have names with simple syntactic structures. However, not all entities are so simple in structure: some entity types (e.g., creative works) can be linguistically complex. They can be complex noun phrases (`Eternal Sunshine of the Spotless Mind`), gerunds (`Saving Private Ryan`), infinitives (`To Kill a Mockingbird`), or full clauses (`Mr. Smith Goes to Washington`). Syntactic parsing of such nouns is hard, and most current parsers and NER systems fail to recognize them. The top system from WNUT 2017 achieved 8% recall for creative work entities (Aguilar et al., 2017). Corpora including such challenging entities are needed for evaluation of model performance in such cases.

Entity Distributions In many domains, entities have a large long-tail distribution, with millions of possible values (e.g., location names). This makes it hard to build representative training data, as a small dataset can only cover a portion of the potentially infinite entity space. A very large test set is required for comprehensive evaluation.

Furthermore, some domains have entity spaces that are continuously growing. While all entity types are open classes (i.e., new ones are added), some groups have a faster growth rate, e.g., new books, songs, and movies are released weekly. Assessing true model generalization requires test sets with many entities that are unseen in the training set, in order to mimic an open-world setting.

Multilinguality and Code-Mixing The recent success of multilingual models have greatly boosted task performance in languages with fewer resources, by leveraging transfer learning from high resource languages. However, there are limits to what can be achieved with cross-lingual transfer, and training data in additional languages is necessary to make progress in this field. The availability of a NER dataset that addresses all the above challenges across many languages will enable new research directions in multilingual model evaluation, as well as for few- and zero-shot cross-lingual transfer scenarios.

Code-mixing, where entities and the main text may be in different languages, is another related research area in multilingual NER where additional resources can help. Code-mixed data is increasing online, especially in social media platform where multiple languages are used in a single post. Such a dataset is needed to study this area and evaluate truly multilingual NER systems.

Domain Adaptation A robust NER model is expected to perform effectively in several domains, such as well written sentences, questions, web search queries, etc. While well written sentences can be easy for NER, shorter questions and queries can be challenging. It is important to study how to adapt existing NER into newly emerging domains. However, most of the existing NER benchmarks only focus on data in a single domain limiting its usage for studying domain adaptation.

MULTICONER We address the aforementioned challenges by presenting MULTICONER, a multilingual dataset that features a large number of entities (including complex ones) in three distinct domains that represent different challenges. Some key facts about MULTICONER:

- Textual snippets in MULTICONER are low in context, allowing to assess NER model's capability in detecting ambiguous named entities;

Source	Gold Sentence
English – Wiki	[heat vision and jack]$_{CW}$, a 1999 television pilot
Spanish – Wiki	reina consorte de [escocia]$_{LOC}$ como esposa de [jacobo v]$_{PER}$.
English – QA	when was the [nokia 2.2]$_{PROD}$ released
English – Search Query	cast of [dr. devil and mr. hare]$_{CW}$
Russian – QA	где было [королевы крика]$_{CW}$ снято
Code-Mixed (KO/EN)	[symphony no. 7 in e major]$_{CW}$ 란 무엇입니까?

Table 1: Examples from MULTICONER: entities are in brackets, followed by their type.

- Named entities contain a highly diverse distribution from simple Location (LOC) to highly complex entities Creative Work (CW);

- Using open knowledge bases such as Wikipedia and Wikidata, we generate textual snippets that contain highly diverse named entities;

- Through a combination of localized versions of Wikipedia and Wikidata, and as well as automated text translation approaches, we generate NER data for 11 languages and 3 domains that can be used to test cross-lingual and cross domain NER performance. Some examples are presented in Figure 1.

2 MULTICONER Dataset Overview

The MULTICONER dataset was designed in order to address the NER challenges described in §1.1. It represents 3 domains (wiki sentences, questions, and search queries) and includes 11 languages, including multilingual and code-mixed subsets. MULTICONER was collected and released as part of the SemEval 2022 Task#11, serving more than 236 participants across all the different languages (Malmasi et al., 2022).

2.1 NER Taxonomy

MULTICONER leverages the WNUT 2017 (Derczynski et al., 2017) taxonomy entity types, which defines the following NER tag-set with 6 classes:

- PERSON (PER for short, names of people)
- LOCATION (LOC, locations/physical facilities)
- CORPORATION (CORP, corporations/businesses)
- GROUPS (GRP, all other groups)
- PRODUCT (PROD, consumer products)
- CREATIVE-WORK (CW, movie/song/book titles)

This taxonomy allows us to capture a wide array of entities, including those with more complex entity structure, such as creative works.

2.2 Languages and Subsets

Bangla	(BN)	Hindi	(HI)	German	(DE)
Chinese	(ZH)	Korean	(KO)	Turkish	(TR)
Dutch	(NL)	Russian	(RU)	Farsi	(FA)
English	(EN)	Spanish	(ES)		

Table 2: The languages included in MULTICONER, along with their 2-letter codes.

There are 11 languages included in MULTICONER (cf. Table 2). These languages were chosen to include a diverse typology of languages and writing systems, and range from well-resourced (e.g., EN) to low-resourced ones (e.g., FA).

MULTICONER contains 13 different subsets: 11 monolingual subsets for the above languages, a multilingual subset (denoted as MULTI), and a code-mixed one (MIX).

Monolingual Subsets Each of the 11 languages has their own subset with data from all domains.

Multilingual Subset This contains randomly sampled data from all the languages mixed into a single subset. This subset is designed for evaluating multilingual models, and should ideally be used under the assumption that the language for each sentence is unknown. A more detailed description of the multilingual train/dev/test set construction is provided in §3.

Code-mixing Subset This subset contains code-mixed instances, where the entity is from one language and the rest of the text is written in another language. Like the multilingual subset, this subset should also be used under the assumption that the languages present in an instance are unknown.

2.3 Domains and Data Sources

The three domains[3] of MULTICONER are listed below. Details on the construction of the different subsets are provided in §3.

Wikipedia Sentences (LOWNER) This subset of MULTICONER, which we call Low-context Wikipedia NER (LOWNER) set, is built by sampling sentences from Wikipedia and using heuristics to identify ones that represent the NER challenges we target. More details on how we select sentence from Wikipedia are provided in §3.2.

---

[3] Domain can have ambiguous interpretations (van der Wees et al., 2015), in our case it reflects a combination of provenance and text genre.

Questions (MSQ-NER) The MSQ-NER subset of MULTICONER represents NER in the QA domain. It is composed of a set of natural language questions, based on the MS-MARCO QnA corpus (V2.1) (Bajaj et al., 2016).

Search Queries (ORCAS-NER) The ORCAS-NER subset of MULTICONER represents the search query domain. To build this data, we utilize 10 million Bing user queries from the ORCAS dataset (Craswell et al., 2020).

2.4 Data Splits

To ensure that obtained NER results on this dataset are *reproducible*, we create three predefined sets for training, development and testing. The entity classes within each set are approximately uniformly distributed. Table 3 shows detailed statistics for each of the 13 subtasks and data splits.

Training Data For the training data split, we limit the size to be 15, 300 sentences. The number of instances was chosen to be comparable to well-known NER datasets such as CoNLL03 (Sang and De Meulder, 2003). Majority of the instances come from the LOWNER domain, with a small sample of 100 instances from the MSQ-NER and ORCAS-NER domains. These instances represent out-of-domain adaptation. The out-of-domain instances are limited in order to be able to realistically assess the out-of-domain performance of models trained on the MULTICONER dataset.

Note that in the case of the Multilingual subset, the training split contains all the instances from the individual language splits. For Code-Mixed on the other hand, we constructed only a small training split, in this way we allow for NER models to better model this task, rather than having abundance of code-mixed instances. The Code-Mixed instances are constructed by first sampling instances from the language specific training splits, and then at random replacing the original entity surface forms into their corresponding surface forms in another language present in our dataset.

Development Data We randomly sample 800 instances per subset from the LOWNER domain (5% of the training set size), a reasonable amount of data for assessing model generalizability.

The only difference in the development data is for the Multilingual and Code-Mixed subtastks, where the development splits are constructed similar as for the training splits (see above).

Test Data Finally, the testing set represents the remaining instances that are not part of the training or development set. To avoid exceedingly large test sets, we limit the number of instances in the test set to be around 215k sentences at most (cf. Table 3). The only exception is for the Multilingual and Code-Mixed subsets. The Multilingual test split was generated from the language specific test splits, and was downsampled to contain only 471k instances. On the other hand, for the Code-Mixed subset, we sample test sentences from the language specific test splits, and replace the original entity surface forms with the surface form of the entity in another language, picked at random.

The larger size of test sets are done for two reasons: (1) to assess the generalizability of models on unseen and complex entities; and (2) assess cross-domain adaptation performance. Table 4 provides a breakdown of the number of instances for the different domains across the different subtasks.

Finally, the overlap of NEs between the test and train set is fairly small, with an overlap of 5.6% across all NE classes and languages. Such a small NE overlap ensures that the test dataset is suitable for measuring NER model generalization.

2.5 License, Availability, and File Format

The dataset is released under a CC BY-SA 4.0 license, which allows adapting the data. Details about the license are available on the Creative Commons website.[4] The data is distributed using the commonly used BIO tagging scheme in CoNLL03 format (Sang and De Meulder, 2003). The complete dataset is available for download.[5]

3 Dataset Construction

In this section, we provide a detailed description of the methods used to generate our dataset.

3.1 Entity Gazetteer Data

We require a large, multilingual, and reliable source of known entities for generating our dataset. To this end we leverage the Wikidata to obtain entity information. Instead of using all entities in the KB, or collecting entities from web sources (Khashabi et al., 2018), we instead focus on entities that map to our taxonomy.

---

[4] https://creativecommons.org/licenses/by-sa/4.0
[5] https://registry.opendata.aws/multiconer/

class	split	EN	DE	ES	RU	NL	KO	FA	ZH	HI	TR	BN	MULTI	MIX
PER	train	5,397	5,288	4,706	3,683	4,408	4,536	4,270	2,225	2,418	4,414	2,606	43,951	296
	dev	290	296	247	192	212	267	201	129	133	231	144	2,342	96
	test	55,682	55,757	51,497	44,687	49,042	39,237	35,140	26,382	25,351	26,876	24,601	111,346	19,313
LOC	train	4,799	4,778	4,968	4,219	5,529	6,299	5,683	6,986	2,614	5,804	2,351	54,030	325
	dev	234	296	274	221	299	323	324	378	131	351	101	2,932	108
	test	59,082	59,231	58,742	54,945	63,317	52,573	45,043	43,289	31,546	34,609	29,628	141,013	23,111
GRP	train	3,571	3,509	3,226	2,976	3,306	3,530	3,199	713	2,843	3,568	2,405	32,846	248
	dev	190	160	168	151	163	183	164	26	148	167	118	1,638	75
	test	41,156	40,689	38,395	37,621	39,255	31,423	27,487	18,983	22,136	21,951	19,177	77,328	16,357
CORP	train	3,111	3,083	2,898	2,817	2,813	3,313	2,991	3,805	2,700	2,761	2,598	32,890	294
	dev	193	165	141	159	163	156	160	192	134	148	127	1,738	112
	test	37,435	37,686	36,769	35,725	35,998	30,417	27,091	25,758	21,713	21,137	20,066	75,764	18,478
CW	train	3,752	3,507	3,690	3,224	3,340	3,883	3,693	5,248	2,304	3,574	2,157	38,372	298
	dev	176	189	192	168	182	196	207	282	113	190	120	2,015	102
	test	42,781	42,133	43,563	39,947	41,366	33,880	30,822	30,713	21,781	23,408	21,280	89,273	20,313
PROD	train	2,923	2,961	3,040	2,921	2,935	3,082	2,955	4,854	3,077	3,184	3,188	35,120	316
	dev	147	133	154	151	138	177	157	274	169	158	190	1,848	117
	test	36,786	36,483	36,782	36,533	36,964	29,751	26,590	28,058	22,393	21,388	20,878	75,871	20,255
#instances	train	15,300	15,300	15,300	15,300	15,300	15,300	15,300	15,300	15,300	15,300	15,300	168,300	1,500
	dev	800	800	800	800	800	800	800	800	800	800	800	8,800	500
	test	217,818	217,824	217,887	217,501	217,337	178,249	165,702	151,661	141,565	136,935	133,119	471,911	100,000

Table 3: MULTICONER dataset statistics for the different languages for the train/dev/test splits. For each NER class we show the total number of entity instances per class on the different data splits. The bottom three rows show the total number of sentences for each language.

lang	LOWNER	ORCAS-NER	MSQ-NER	Total
EN	100,000	100,000	17,818	217,818
DE	100,000	100,000	17,824	217,824
ES	100,000	100,000	17,887	217,887
RU	100,000	100,000	17,501	217,501
NL	100,000	100,000	17,337	217,337
KO	60,425	100,000	17,824	178,249
FA	48,792	100,000	16,910	165,702
ZH	33,776	100,000	17,885	151,661
HI	24,807	100,000	16,758	141,565
TR	19,581	100,000	17,354	136,935
BN	15,698	100,000	17,421	133,119
MULTI	200,000	200,000	71,911	471,911
MIX	42,168	15,667	42,165	100,000

Table 4: Test data statistics per domain.

We map Wikidata entities to our NER taxonomy (§2.1). This is done by traversing Wikidata's class and instance relations, and manually mapping them to our NER classes, e.g., Wikidata's human class maps to PER in our taxonomy, song to CW, and so on. Alternative names (aliases) for entities are included. The distribution of these entities is shown in Table 9 in §A.1.

3.2 Wiki Sentences

LOWNER, the Wiki sentences component of MULTICONER is obtained by parsing Wikipedia articles into sentences, and selecting suitable candidates. Figure 2 shows a diagram of the basic data processing steps, which are described below.

This process is performed for the following lan-

guages: NL, EN, FA, KO, RU, ES, TR. For the other languages (BN, ZH, DE, HI), we apply Machine Translation to obtain the data, as described in §3.4.

Wikipedia Parsing (A) We start by downloading the complete Wikipedia dumps for our target languages.[6] The files are parsed to first extract individual articles, which are then each parsed to remove markup and extract individual sentences. This process yields a set of sentences[7] with the interlinks intact, along with the IDs of the original article they were extracted from.

Interlink Parsing (B) In the next step, we parse the sentences to identify interlinks (links to other Wikipedia articles). We then map the interlinks in each sentence to an entity in the Wikidata KB. This mapping is provided in the KB, which links entities to their Wikipedia page names, which can be used to map linked pages to an entity ID. The identified entities are finally resolved to our NER taxonomy (using the same approach that was described in §3.1). Some interlinks point to inexisting Wikipedia articles, or the linked Wikipedia article cannot be joined to a corresponding Wikidata entry. We mark such cases as unresolvable.

Sentence Filtering (C) Next, we filtered sentences using several strategies and heuristics.

---

[6]Dumps are available here: https://dumps.wikimedia.org/backup-index.html
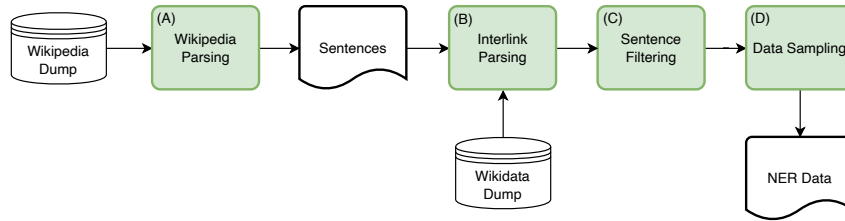
[7]e.g., > 180 million sentences for English.

Figure 2: An overview of the different steps involved in extracting the MULTICONER data from Wikipedia dumps.

- Length filtering: short sentences (< 28 characters) and long ones (> 180 characters) are removed.
- Interlink filtering: sentences without interlinks, or with unresolvable links, are dropped. Sentences with interlinks that did not map to our taxonomy are dropped.
- Capitalization heuristic filtering: for languages that capitalize proper nouns or entities, a heuristic is used to filter out sentences that contain capitalized tokens that are not part of an interlink. This removes sentences containing potential nouns that cannot be tagged as entities by our method since they are not linked to a known entity whose type can be determined.

This filtering process removes long and high-context sentences that contain references to many entities. This step discards over 90% of the sentences retrieved in the prior steps, e.g., resulting in approx. 14 million candidate sentences for EN. Finally, to assess the NER label quality, for a small random sample of 400 sentences, we assessed the accuracy of NER gold labels, which was was measured at 94% accuracy for EN-LOWNER.

This process is very effective at yielding short, low-context sentences. Example English sentences are shown in Table 5. The sentences contain some context, but they are much shorter than the average Wikipedia sentence, and usually only contain a single entity, making them more aligned with the challenges listed in Section 1.1.

```
The design is considered a forerunner to the modern
[food processor].
The regional capital is [Oranjestad, Sint
Eustatius].
The most frequently claimed loss was an [iPad].
An [HP TouchPad] was prominently displayed in an
episode of the sixth season.
The incumbent island governor is [Jonathan G. A.
Johnson].
A revised edition of the book was released in 2017
as an [Amazon Kindle] book.
```

Table 5: Sample sentences extracted from Wikipedia. Resolved entities are in brackets.

MSQ-NER	ORCAS-NER
average retail price of \<PROD\>	\<CW\> imdb
where was \<CW\> filmed	best hotels \<LOC\>
how many miles from \<LOC\> to \<LOC\>	\<PER\> parents
how many kids does \<PER\> have	\<PROD\> price
when did \<GRP\> start	\<GRP\> website
when will \<CORP\> report earnings	\<CORP\> customer service

Table 6: Sample templates used to generate data for the MSQ-NER and ORCAS-NER domain. Slots are in angle brackets.

Data Sampling (D) We downsample the collected data to construct a smaller subset. Given that some of the NE classes are more prevalent (e.g. PER and LOC, account for more than 60% of named entities) , similar to stratified sampling, we sample at NE class level, with the only difference, that the number of instances per class is fixed. In this way, we create a dataset that has more uniform representation of the different NE classes. Furthermore, retaining all sentences is impractical, given the total amount of data.

Finally, we lowercase all the selected sentences to increase the difficulty of the NER task. The final stats per subset and split are shown in Table 3.

3.3 Questions and Search Queries

We apply a template-based process to generate data in the Questions and Search Query domains. This involves two broad steps: template extraction, and template slotting.

The same steps are applied to two data sources to generate the NER datasets. This process is visualized in Figure 3, and the individual steps are detailed below.

Running Named Entity Recognition (A) Similar to the work of Wu et al. (2020), we aim to templatize the input questions and search queries by first applying NER to extract entities, which are then mapped to our taxonomy.

Specifically, we apply the spaCy NER pipeline[8]

---

[8]We use the `en_core_we_lg` pre-trained pipeline in
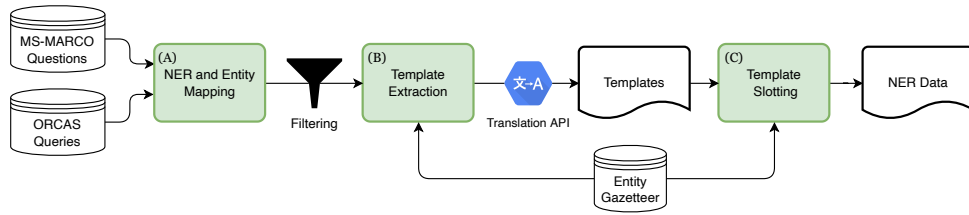
Figure 3: An overview of the data processing steps in our template-based approach to generating the NER data for the MSQ-NER and ORCAS-NER domains.

to identify entities. While this pre-trained NER system cannot correctly identify all the entities in the data, it does identify many correct ones. This process yields a sufficient amount of data for us to extract common patterns from the original input. Recognized entities are then mapped to entries in our gazetteer via string matching. Input texts that have entities that could not be mapped, or have no recognized entities are then filtered out. This process yields a set of sentences, with recognized entities that exist in our gazetteer.

Template Extraction (B) Next, we replace identified entities with their types to create templates, e.g., "when did [iphone] come out" is transformed to "when did <PROD> come out". The templates are then grouped together in order to merge all inputs having the same template, and sorted by frequency.

To minimize noise in the data, we apply frequency-based filtering, and only templates appearing $>= 5$ times are included. This process results in $3,445$ unique question templates, and $97,324$ unique search query templates. There are a wide range of question shapes and entity types. Examples are listed in Table 6.

Since these templates are all in English, we apply Machine Translation to translate them into the other 10 languages of our dataset.

Template Slotting (C) In the last step we generate the NER data by slotting the templates with random entities from the Wikipedia KB with the same class. For example, "when did <PROD> come out" can be slotted as "when did [xbox 360] come out" or "when did [Sony Alpha DSLR-A77 II] come out".

To maintain the same relative distribution as the original data, each template is slotted the same number of times it appeared (i.e., the template frequency) using different entities. Templates are

```
EN: average cost of living in <p translate=no> <LOC> </p>
ZH: <p translate=no> <LOC> </p>的平均生活成本
DE: durchschnittliche Lebenshaltungskosten in <p translate=no> <LOC> </p>
HI: रहनए कइ अउसत लअअ <p translate=no> <LOC> </p>
```

Table 7: Examples of template translations.

slotted with entities from the same language, i.e., a DE template is slotted with DE entities. The slotted texts are lowercased to simulate the low-context challenges outlined in §1.1, which increases the difficulty of the task. This yields two domains of MULTICONER: MSQ-NER and ORCAS-NER.

3.4 Dataset Translation

We apply automatic translation to generate two portions of our data. LOWNER sentences for four languages (Bangla, Chinese, German, Hindi) are translated from English Wiki sentences. The NER templates used for MSQ-NER and ORCAS-NER are also translated from the English templates. We do not translate any of our gazetteer entities.

We use the Google Translation API[9] to perform our translations. The input texts may contain known entity spans or slots. To prevent these spans from being translated, we leveraged the notranslate attribute to mark these spans and prevent them from being translated. Table 7 shows examples of template translations.

The translation quality of LOWNER, ORCAS-NER and MSQ-NER in the different languages such as German, Chinese, Bangla, and Hindi is very high, with over 90% translation accuracy (i.e., accuracy as measured by human annotators in terms of the translated sentence retaining the semantic meaning and as well have a correct syntactic structure in the target language).

3.5 Code-mixed Data Generation

We generate code-mixed data by sampling instances from the respective languages, and replacing the NE surface forms from the *source* language

---

spaCy where the NER model is trained using OntoNotes 5.

[9] https://cloud.google.com/translate

to a *target* language, chosen at random from any of the languages in Table 2. This results in a dataset, where the instances contain up to two languages, where the non-NE tokens are in a language that is different from the NE tokens. Note that, in some cases, some of the NE surface forms may remain in the source language if we do not possess the NE's surface form in one of the other languages from Table 2.[10]

4 NER Model Performance

To evaluate whether our new dataset poses real-world challenges (cf. §1.1), we train and test two existing NER systems: (1) XLM-RoBERTa (Conneau et al., 2020), a large multilingual Transformer model; and (2) GEMNET (Meng et al., 2021; Fetahu et al., 2021, 2022), a state of the art model that integrates gazetteers into XLM-RoBERTa.

4.1 Evaluation Metrics

We evaluate the different NER models using standard performance metrics, such as P/R/F1. We measure the performance at the class level, where we distinguish between *micro/macro* averages. The difference between *micro* and *macro* average P/R/F1, is that for unbalanced distribution *micro* performance is skewed towards the more prominent NE classes. Additionally, we consider *Mention Detection* (MD), which corresponds to the ability of models to detect NE boundaries, without taking into consideration their actual NER class.

4.2 Results

Table 8 shows the results obtained for both XLM-RoBERTa (baseline, denoted as B), and a state of the art model, GEMNET (denoted as GM). The results are shown only for the F1 score achieved on the individual NER classes, and finally the micro, macro F1 and MD scores are shown. Table 10 in §A.2 shows a detailed performance for each sub-task and domain.

Baseline. For the baseline approach, we simply fine-tune XLM-RoBERTa on the language specific training data, and test on the corresponding test splits. We note that overall, across all subsets, the baseline achieves the highest performance of micro-F1=0.646 for DE, and lowest of micro-F1=0.397 for BN. This result is expected, given that the test data contains highly complex entities

---

[10]For ZH, the tokenization is done at the character level.

that are out-of-domain, and additionally are not seen in the training data.

GEMNET. The state of the art approach, GEMNET, makes use of external gazetteers, constructed from Wikidata for the task of NER. For each token GEMNET computes two representations: (i) textual representation based on XLM-RoBERTa, and (ii) gazetteer encoding, which uses a gazetteer to map to tokens to gazetteer entries, which correspondingly maps them to their NER tags. The two representations are combined using a Mixture-of-Experts (MoE) gating mechanism (Shazeer et al., 2017), which allows the model depending on the context to either assign higher weight to its textual representation or the gazetteer computed representation.

GEMNET provides a highly significant improvement over the baseline, with an average improvement of micro-F1=30%. The highest improvement is shown for languages that are considered to be low-resource, such as TR with micro-F1=+41.5%, and KO with micro-F1=+33%.

The obtained results in Table 8 show that GEMNET is highly flexible in detecting unseen entities during the training phase. Depending on its gazetteer coverage, if a named entity is matched by its gazetteers, this will allow GEMNET to correctly identify the named entity. In more detail, internally, GEMNET dynamically weighs both representation of a token (i.e., textual and gazetteer representations), to correctly determine the correct tag for a token. Note that, the gazetteers may contain noisy labels for a named entity (e.g. "Bank of America" can match to CORP and LOC), hence, GEMNET needs to additionally leverage the token context to determine the correct tag.

5 Conclusions and Future Work

We presented MULTICONER, a new large-scale dataset that represents a number of current challenges in NER. Results obtained on our data showed that our dataset is challenging. A XLMR based model achieves only approx. 50% F1 in average while GEMNET improves F1 performance by more than 30% using gazetteers.

These results demonstrate that MULTICONER represents challenging scenarios where even large pre-trained language models fail to achieve good performance without external resources. It is our hope that this resource will help further research for building better NER systems. This dataset can

	PER		LOC		GRP		CORP		CW		PROD		Micro F1		Macro F1		MD	
	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM
EN	0.807	0.939	0.664	0.848	0.599	0.876	0.567	0.889	0.474	0.806	0.563	0.873	0.627	0.873	0.612	0.872	0.731	0.892
DE	0.797	0.968	0.679	0.921	0.591	0.940	0.588	0.951	0.516	0.897	0.633	0.940	0.646	0.936	0.634	0.936	0.767	0.951
ES	0.750	0.941	0.589	0.854	0.531	0.884	0.564	0.893	0.496	0.811	0.515	0.840	0.587	0.872	0.574	0.870	0.689	0.888
RU	0.666	0.839	0.632	0.780	0.539	0.818	0.600	0.870	0.534	0.803	0.576	0.805	0.597	0.817	0.591	0.819	0.699	0.834
NL	0.766	0.949	0.658	0.889	0.586	0.893	0.599	0.905	0.514	0.854	0.574	0.871	0.626	0.895	0.616	0.894	0.731	0.911
KO	0.595	0.900	0.650	0.865	0.513	0.910	0.560	0.923	0.439	0.846	0.517	0.905	0.558	0.888	0.546	0.891	0.666	0.896
FA	0.634	0.870	0.588	0.792	0.573	0.867	0.473	0.805	0.362	0.688	0.480	0.797	0.523	0.801	0.518	0.803	0.638	0.823
TR	0.549	0.894	0.497	0.860	0.404	0.896	0.480	0.897	0.374	0.849	0.441	0.914	0.468	0.883	0.457	0.885	0.614	0.893
ZH	0.532	0.884	0.627	0.889	0.371	0.866	0.552	0.902	0.434	0.818	0.552	0.861	0.531	0.870	0.511	0.870	0.664	0.895
HI	0.578	0.883	0.536	0.846	0.485	0.869	0.502	0.851	0.298	0.760	0.418	0.839	0.478	0.843	0.469	0.841	0.640	0.877
BN	0.529	0.895	0.420	0.850	0.322	0.883	0.428	0.889	0.243	0.747	0.406	0.865	0.397	0.856	0.391	0.855	0.570	0.888
MULTI	0.679	0.810	0.556	0.743	0.496	0.721	0.563	0.746	0.428	0.644	0.523	0.697	0.550	0.732	0.541	0.727	0.674	0.810
MIX	0.709	0.835	0.621	0.765	0.532	0.714	0.581	0.748	0.481	0.604	0.560	0.735	0.585	0.731	0.581	0.733	0.752	0.847
Avg.	0.661	0.893	0.594	0.839	0.503	0.857	0.543	0.867	0.430	0.779	0.520	0.842	0.552	0.846	0.542	0.846	0.680	0.877

Table 8: XLM-RoBERTa (B) baseline and GEMNET (GM) results as measured by the F1 score for the different NER tags. In the last three columns are shown the *micro*, *macro*, and *mention detection* – MD F1 performance.

serve as a good benchmark for evaluating different methods of infusing external entity knowledge into language models.

The extension of MULTICONER to additional languages is the most straightforward direction for future work. The addition of completely new domains is something we will also consider, along with the the expansion of the existing domains to include additional topics and templates.

References

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

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

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. Orcas: 18 million clicked query-document pairs for analyzing search. *arXiv preprint arXiv:2006.05324*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*. 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.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2022. Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2777–2790. Association for Computational Linguistics.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev-Arie Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai, Subhro Roy, Stephen Mayhew, Zhili Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, and Dan Roth. 2018. Cogcompnlp: Your swiss army knife for NLP. In *LREC*. European Language Resources Association (ELRA).

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@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 1412–1437. Association for Computational Linguistics.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. In *EMNLP/IJCNLP (1)*, pages 6255–6260. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1499–1512. Association for Computational Linguistics.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Marlies van der Wees, Arianna Bisazza, Wouter Weerkamp, and Christof Monz. 2015. What's in a domain? analyzing genre and topic differences in statistical machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 560–566. The Association for Computer Linguistics.

Tongshuang Wu, Kanit Wongsuphasawat, Donghao Ren, Kayur Patel, and Chris DuBois. 2020. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

A Appendix

A.1 Gazetteer Statistics

Table 9 shows the number of entries for NE class and language. The entries are extracted from Wikidata.

lang	PER	LOC	CORP	GRP	PROD	CW
BN	42,970	31,336	1,072	8,691	990	12,152
ZH	388,910	346,879	30,323	64,031	15,919	120,831
NL	1,321,741	738,609	27,589	79,566	21,105	204,130
EN	1,797,558	1,117,951	72,105	227,822	67,113	490,523
FA	224,265	233,962	8,641	14,346	11,802	60,857
DE	1,308,532	533,551	42,321	99,468	38,735	219,801
HI	22,279	18,480	1,160	2,382	1,044	7,826
KO	148,367	72,153	9,625	23,209	8,385	55,624
RU	984,093	495,059	21,609	68,834	21,571	148,003
ES	1,389,698	480,310	29,465	113,197	25,658	228,369
TR	171,133	141,225	6,099	19,388	6,718	43,029

Table 9: Gazetteer entity statistics per class for our target languages.

A.2 Cross-Domain Evaluation Results

Table 10 shows for the different subtasks, the cross-domain evaluation results for the baseline and GEMNET approaches. We note that in all cases the GEMNET approach shows strong gains in terms of macro-F1 score across all subtasks. This is especially the case for the domains MSQ-NER and ORCAS-NER, where the models contain very little knowledge about these domains[11], hence, showing the generalizability of models in out-of-domain scenarios.

Finally, we note that in the case of the LOWNER domain, which is an in-domain evaluation scenario[12], in terms of MD, the gap between the baseline and GEMNET approach shrinks. For the `Multi`, the gap is only 4.1%. This shows that the baseline models for in-domain scenarios is able to correctly identify entity boundaries, even though its NER accuracy may not be optimal. For instance, for `Multi` the gap in terms of macro-F1 is 12.7%. showing, that models that leverage external knowledge like GEMNET, are more accurate in terms of NER accuracy and as well have higher coverage in spotting entity boundaries.

---

[11]The MultiCoNER training set for each of the subtasks, contains 50 instances from the MSQ-NER and ORCAS-NER domains

[12]The MultiCoNER training set consists nearly of only LOWNER instances.

	PER		LOC		GRP		CORP		CW		PROD		Micro F1		Macro F1		MD	
	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM	B	GM
Domain – LOWNER																		
EN	0.921	0.971	0.855	0.938	0.766	0.925	0.756	0.939	0.681	0.862	0.656	0.849	0.789	0.920	0.773	0.914	0.851	0.932
DE	0.913	0.978	0.871	0.957	0.781	0.948	0.776	0.952	0.706	0.913	0.772	0.921	0.816	0.949	0.803	0.945	0.903	0.965
ES	0.897	0.944	0.797	0.866	0.725	0.871	0.792	0.910	0.667	0.802	0.627	0.761	0.759	0.864	0.751	0.859	0.821	0.883
RU	0.734	0.794	0.702	0.757	0.695	0.794	0.745	0.855	0.687	0.793	0.647	0.753	0.702	0.788	0.702	0.791	0.752	0.809
NL	0.904	0.949	0.878	0.926	0.797	0.900	0.801	0.898	0.732	0.840	0.715	0.810	0.816	0.894	0.805	0.887	0.871	0.913
KO	0.774	0.896	0.817	0.885	0.734	0.882	0.760	0.910	0.710	0.850	0.714	0.852	0.761	0.880	0.751	0.879	0.810	0.890
TR	0.813	0.897	0.825	0.875	0.807	0.906	0.798	0.906	0.684	0.831	0.640	0.805	0.768	0.871	0.761	0.870	0.818	0.884
ZH	0.869	0.917	0.855	0.924	0.534	0.795	0.740	0.859	0.659	0.816	0.655	0.834	0.743	0.868	0.719	0.858	0.811	0.901
HI	0.792	0.837	0.732	0.813	0.710	0.757	0.651	0.713	0.487	0.578	0.524	0.634	0.649	0.722	0.649	0.722	0.765	0.813
BN	0.822	0.853	0.769	0.823	0.701	0.780	0.666	0.725	0.569	0.663	0.576	0.679	0.680	0.752	0.684	0.754	0.814	0.859
MULTI	0.855	0.916	0.808	0.882	0.717	0.868	0.733	0.884	0.664	0.825	0.648	0.808	0.741	0.868	0.737	0.864	0.852	0.893
MIX	0.855	0.862	0.808	0.809	0.717	0.737	0.733	0.757	0.664	0.616	0.648	0.719	0.741	0.749	0.737	0.750	0.852	0.850
Domain – MSQ-NER																		
EN	0.781	0.921	0.613	0.823	0.366	0.788	0.408	0.801	0.348	0.795	0.355	0.852	0.598	0.842	0.479	0.830	0.733	0.860
DE	0.758	0.984	0.708	0.958	0.317	0.939	0.397	0.964	0.415	0.909	0.346	0.948	0.643	0.959	0.490	0.950	0.783	0.970
ES	0.700	0.979	0.526	0.879	0.216	0.857	0.403	0.924	0.388	0.856	0.335	0.885	0.529	0.901	0.428	0.897	0.643	0.912
RU	0.692	0.961	0.652	0.864	0.317	0.904	0.436	0.842	0.435	0.915	0.280	0.806	0.601	0.891	0.469	0.882	0.726	0.904
NL	0.745	0.980	0.511	0.895	0.273	0.831	0.450	0.947	0.436	0.922	0.342	0.937	0.546	0.919	0.460	0.919	0.680	0.932
KO	0.547	0.864	0.644	0.917	0.255	0.903	0.370	0.947	0.288	0.907	0.235	0.924	0.531	0.904	0.390	0.910	0.669	0.908
FA	0.674	0.914	0.512	0.789	0.533	0.829	0.413	0.805	0.236	0.740	0.331	0.762	0.499	0.814	0.450	0.807	0.615	0.840
TR	0.597	0.881	0.568	0.905	0.246	0.875	0.389	0.956	0.357	0.890	0.211	0.873	0.517	0.897	0.395	0.897	0.647	0.908
ZH	0.534	0.947	0.709	0.957	0.401	0.907	0.432	0.941	0.390	0.920	0.283	0.843	0.588	0.945	0.458	0.919	0.743	0.961
HI	0.725	0.955	0.715	0.925	0.464	0.925	0.572	0.929	0.360	0.899	0.280	0.827	0.656	0.928	0.519	0.910	0.802	0.952
BN	0.589	0.950	0.468	0.879	0.000	0.000	0.433	0.942	0.298	0.821	0.239	0.793	0.465	0.891	0.338	0.731	0.643	0.915
MULTI	0.628	0.775	0.571	0.751	0.271	0.503	0.401	0.602	0.323	0.539	0.306	0.463	0.531	0.712	0.417	0.605	0.688	0.817
MIX	0.629	0.857	0.477	0.764	0.445	0.733	0.521	0.786	0.349	0.666	0.532	0.777	0.496	0.763	0.492	0.764	0.738	0.891
Domain – ORCAS-NER																		
EN	0.588	0.886	0.340	0.719	0.313	0.811	0.342	0.834	0.191	0.736	0.430	0.902	0.365	0.813	0.367	0.815	0.530	0.841
DE	0.581	0.943	0.355	0.839	0.313	0.929	0.388	0.949	0.266	0.868	0.454	0.959	0.392	0.913	0.393	0.914	0.564	0.926
ES	0.524	0.927	0.296	0.815	0.260	0.902	0.323	0.875	0.229	0.811	0.333	0.929	0.331	0.876	0.327	0.876	0.490	0.888
RU	0.535	0.871	0.470	0.770	0.327	0.845	0.417	0.887	0.313	0.791	0.472	0.864	0.428	0.838	0.422	0.838	0.597	0.850
NL	0.543	0.939	0.292	0.815	0.274	0.887	0.366	0.912	0.265	0.865	0.409	0.943	0.360	0.892	0.358	0.893	0.536	0.905
KO	0.445	0.916	0.401	0.812	0.321	0.938	0.403	0.935	0.220	0.835	0.362	0.945	0.359	0.896	0.359	0.897	0.529	0.900
FA	0.498	0.870	0.386	0.759	0.450	0.884	0.327	0.788	0.202	0.641	0.399	0.822	0.361	0.790	0.377	0.794	0.535	0.816
TR	0.459	0.900	0.338	0.823	0.295	0.898	0.403	0.892	0.274	0.849	0.376	0.944	0.361	0.884	0.358	0.884	0.538	0.893
ZH	0.396	0.854	0.398	0.821	0.368	0.872	0.468	0.920	0.291	0.816	0.473	0.878	0.397	0.860	0.399	0.860	0.555	0.880
HI	0.492	0.875	0.390	0.810	0.410	0.905	0.460	0.886	0.266	0.791	0.387	0.902	0.401	0.861	0.401	0.861	0.578	0.882
BN	0.459	0.886	0.334	0.838	0.265	0.898	0.372	0.913	0.192	0.752	0.365	0.906	0.331	0.867	0.331	0.866	0.506	0.888
MULTI	0.479	0.645	0.322	0.516	0.305	0.533	0.401	0.589	0.240	0.443	0.411	0.567	0.356	0.545	0.360	0.549	0.543	0.689
MIX	0.517	0.792	0.308	0.685	0.324	0.687	0.387	0.722	0.235	0.563	0.421	0.739	0.364	0.695	0.365	0.698	0.577	0.828

Table 10: XLM-RoBERTa (B) baseline and GEMNET (G) domain results as measured by the F1 score for the different NER tags. The last three columns show the *micro*, *macro*, and *mention detection* – MD F1 performance.