ACLM: A Selective-Denoising based Generative Data Augmentation Approach for Low-Resource Complex NER

Sreyan Ghosh[♠*] **Utkarsh Tyagi**[♠*] **Manan Suri**[♣] **Sonal Kumar**[♠]
S Ramaneswaran[♥] **Dinesh Manocha**[♠]
[♠]University of Maryland, College Park, USA,
[♣]NSUT Delhi, India, [♥]NVIDIA, Bangalore, India
{sreyang, utkarsht, sonalkum, dmanocha}@umd.edu
manansuri27@gmail.com, ramanr@nvidia.com

Abstract

Complex Named Entity Recognition (NER) is the task of detecting linguistically complex named entities in low-context text. In this paper, we present ACLM (**A**ttention-map aware keyword selection for **C**onditional **L**anguage **M**odel fine-tuning), a novel data augmentation approach, based on conditional generation, to address the data scarcity problem in low-resource complex NER. ACLM alleviates the context-entity mismatch issue, a problem existing NER data augmentation techniques suffer from and often generates incoherent augmentations by placing complex named entities in the wrong context. ACLM builds on BART and is optimized on a novel text reconstruction or denoising task - we use *selective masking* (aided by attention maps) to retain the named entities and certain *keywords* in the input sentence that provide contextually relevant additional knowledge or hints about the named entities. Compared with other data augmentation strategies, ACLM can generate more diverse and coherent augmentations preserving the true word sense of complex entities in the sentence. We demonstrate the effectiveness of ACLM both qualitatively and quantitatively on monolingual, cross-lingual, and multilingual complex NER across various low-resource settings. ACLM outperforms all our neural baselines by a significant margin (1%-36%). In addition, we demonstrate the application of ACLM to other domains that suffer from data scarcity (e.g., biomedical). In practice, ACLM generates more effective and factual augmentations for these domains than prior methods.[1]

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that aims to detect various types of named entities (NEs) from text. Recently, there has been considerable progress in NER using neural learning methods that achieve state-of-the-art (SOTA) performance (Wang et al., 2021; Zhou and Chen, 2021) on well-known benchmark datasets, including CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes (Schwartz et al., 2012). However, these datasets are designed to evaluate the performance on detecting "relatively easy" NEs like *proper names* (e.g., people such as "Barack Obama," locations such as "New York," or organizations such as "IBM") in well-formed, context-rich text that comes from news articles (Augenstein et al., 2017). On the other hand, complex NER benchmarks like MultiCoNER (Malmasi et al., 2022) present several contemporary challenges in NER, including short low-context texts with emerging and semantically ambiguous complex entities (e.g., movie names in online comments) that reduce the performance of SOTA methods previously evaluated only on the existing NER benchmark datasets. Our experiments reveal that the performance of the current SOTA NER method (Zhou and Chen, 2021) (previously evaluated only on the CoNLL 2003 dataset) drops by 23% when evaluated on MultiCoNER and 31.8% when evaluated on a low-resource setting with just 500 training samples (more details in Table 8). Thus, we emphasize that research on building systems that can effectively detect complex NEs in the text is currently understudied in the field of NLP.

In the past, researchers have made several attempts at building supervised approaches to detect complex and compositional noun phrase entities in sentences (Doddington et al., 2004; Biggio et al., 2010; Magnolini et al., 2019). However, the scarcity of annotated training data for building effective systems has always been a challenge. Data augmentation has been shown to be an effective solution for low-resource NER (Ding et al., 2020; Liu et al., 2021; Zhou et al., 2022). In practice, though these systems perform well and generate

[1]Code: https://github.com/Sreyan88/ACLM
[*]These authors contributed equally to this work.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 104–125
July 9-14, 2023

coherent augmentations on common NER benchmark datasets with easy proper noun NEs, they fail to be effective for complex NER, often generating incoherent augmentations. We first argue that certain types of complex NEs follow specific linguistic patterns and appear only in specific contexts (examples in Appendix 4), and augmentations that do not follow these patterns impede a NER model from learning such patterns effectively. This sometimes also leads to augmentations with context-entity mismatch, further hurting the learning process. For e.g., unlike proper names, substituting complex NEs from other sentences in the corpus or replacing them with synonyms (Dai and Adel, 2020a) often leads to augmentations where the NE does not fit into the new context (e.g., swapping proper names across sentences might still keep the sentence coherent *but* swapping the name of a book with a movie (both *creative work* entity) *or* the name of a football team with a political party (both *group* entity) makes it incoherent). Fine-tuning pretrained language models (PLMs), similar to prior-work (Ding et al., 2020; Liu et al., 2021; Zhou et al., 2022), fail to generate new context around complex NEs or completely new NEs with the desired linguistic patterns due to low-context sentences and the lack of existing knowledge of such linguistically complex NEs (examples in Fig. 3). This leads to in-coherent augmentations and poses a severe problem in knowledge-intensive tasks like biomedical NER, where non-factual augmentations severely hurt learning. Our experiments also reveal that introducing new context patterns around NEs proves to be a more effective data augmentation technique for complex NER than diversifying NEs (ACLM vs. MELM in Table 1).

Main Results: To overcome the aforesaid problems, we formulate data augmentation as a conditional generation task and propose ACLM, a conditional text generation model that generates augmentation samples by introducing new and diverse context patterns around a NE. ACLM builds on BART (Lewis et al., 2020) and is fine-tuned on a modification of the text reconstruction from corrupted text task, a common denoising-based PLM pre-training objective. In contrast to other PLM pre-training strategies, which randomly mask a portion of the text for corruption, our modified objective is based on *selective masking*, wherein we mask all other words in the sentence except the NEs and a small percentage of *keywords* related to the NEs. We refer to this corrupted sentence as a *template*, and it serves as input to the model for both the training and generation phases. These keywords are other non-NE tokens in the sentence that provide contextually relevant additional knowledge or hints to BART about the complex NEs without the need of retrieving knowledge from any external sources. We select these keywords using attention maps obtained from a transformer model fine-tuned on the NER task, and they help the PLM overcome the problem where it might not possess enough knowledge about a semantically ambiguous complex NE (example in Fig. 3). Training ACLM on this modified objective allows us to generate diverse, coherent, factual, and high-quality augmentations given templates. We also propose *mixner*, a novel algorithm that mixes two templates during the augmentation generation phase and boosts the diversity of augmentations. Our primary contributions are as follows:

- We propose ACLM, a novel data augmentation framework specially designed for low-resource complex NER. Compared with previous methods in the literature, ACLM effectively alleviates the context-entity mismatch problem by preserving the true sense of semantically ambiguous NEs in augmentations. Additionally, to accompany ACLM, we propose *mixner*, which boosts the diversity of ACLM generations.
- We qualitatively and quantitively show the benefits of ACLM for monolingual, cross-lingual, and multilingual complex NER across various low-resource settings on the MultiCoNER dataset. Our proposed ACLM outperforms all other baselines in literature by a significant margin (1%-36%) and generates more diverse, coherent, and high-quality augmentations compared to them.
- We perform extensive experiments to study the application of ACLM in three other domains, including science and medicine. ACLM outperforms all our baselines in these domains (absolute gains in the range of 1%-11%) and generates more factual augmentations.

2 Background and Related Work

Complex NER Background: Complex NER is a relatively understudied task in the field of NLP.

Building on insights from Augenstein et al. (2017), we discuss key reasons behind high performance on common NER benchmark datasets and try to understand why modern SOTA NER algorithms do not work well on complex NER benchmarks: (1) **Context**: Most of the common benchmark datasets are curated from articles in the news domain. This gives them several advantages, including rich context and surface features like proper punctuation and capitalized nouns, all of which are major drivers of success in these datasets (Mayhew et al., 2019). In contrast, for entity recognition beyond news text, like search queries or voice commands, the context is less informative and lacks surface features (Guo et al., 2009; Carmel et al., 2014); (2) **Entity Complexity**: Data from news articles contain *proper names* or "easy" entities with simple syntactic structures, thus allowing pretrained models to perform well due to their existing knowledge of such entities. On the other hand, complex NEs like movie names are syntactically ambiguous and linguistically complex and which makes Complex NER a difficult task (Ashwini and Choi, 2014). Examples of such entities include noun phrases (e.g., Eternal Sunshine of the Spotless Mind), gerunds (e.g., Saving Private Ryan), infinitives (e.g., To Kill a Mockingbird), or full clauses (e.g., Mr. Smith Goes to Washington); (3) **Entity Overlap**: Models trained on these common benchmark datasets suffer from memorization effects due to the large overlap of entities between the train and test sets. Unseen and emerging entities pose a huge challenge to complex NER (Bernier-Colborne and Langlais, 2020).

Complex NER: Prior work has mostly focused on solving the entity complexity problem by learning to detect complex nominal entities in sentences (Magnolini et al., 2019; Meng et al., 2021; Fetahu et al., 2022; Chen et al., 2022). Researchers have often explored integrating external knowledge in the form of gazetteers for this task. Gazetteers have also proven to be effective for low-resource NER (Rijhwani et al., 2020). GemNet (Meng et al., 2021), the current SOTA system for complex NER, conditionally combines the contextual and gazetteer features using a Mixture-of-Experts (MoE) gating mechanism. However, gazetteers are difficult to build and maintain and prove to be ineffective for complex NER due to their limited entity coverage and the nature of unseen and emerging entities in complex NER.

Data Augmentation for Low-Resource NER: Data Augmentation to handle data scarcity for low-resource NLP is a well-studied problem in the literature and is built on word-level modifications, including simple synonym replacement strategies (Wei and Zou, 2019), or more sophisticated learning techniques like LSTM-based language models (Kobayashi, 2018), Masked Language Modeling (MLM) using PLMs (Kumar et al., 2020), auto-regressive PLMs (Kumar et al., 2020), or constituent-based tagging schemes (Zhou et al., 2019). However, most of these methods, though effective for classification tasks, suffer from token-label misalignment when applied to token-level tasks such as NER and might require complex pre-processing steps (Bari et al., 2020; Zhong and Cambria, 2021). One of the first works to explore effective data augmentation for NER replaces NEs with existing NEs of the same type or replaces tokens in the sentence with one of their synonyms retrieved from WordNet (Dai and Adel, 2020b). Following this, many neural learning systems were proposed that either modify the Masked Language Modelling (MLM) training objective using PLMs (Zhou et al., 2022; Liu et al.) or use generative language modeling with LSTM LMs (Ding et al., 2020) or mBART (Liu et al., 2021), to produce entirely new sentences from scratch. However, all these systems were designed for low-resource NER on common benchmark datasets and failed to generate effective augmentations for low-resource complex NER with semantically ambiguous and complex entities.

3 Methodology

In this section, we give an overview of our approach. Fig. 1 represents the entire workflow of our ACLM data augmentation framework. A sentence is first passed through a fine-tuned XLM-RoBERTa fine-tuned on only gold data to generate the attention map for each token in the sentence. This attention map is then used to selectively mask the sentence and create a template. This template is then used as an input to optimize the model on the text reconstruction objective for fine-tuning ACLM: the model is asked to reconstruct the entire original sentence from only the content in the template. While generating augmentations, ACLM follows the same template generation process in addition to adding two templates through *mixner*, which we discuss in detail in Section 3.3.

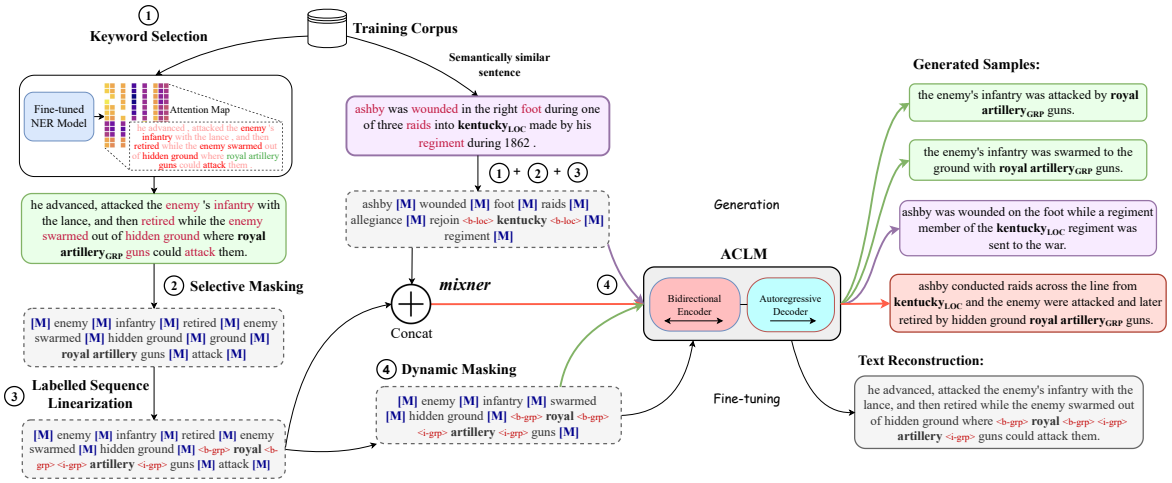


Figure 1: **Overview of ACLM:** ACLM follows a 4-step template creation process, which serves as an input to the model during fine-tuning and generation. ① **Keyword Selection:** The most important keywords (in red) associated with the NEs (in bold) in the sentence is first extracted using attention maps obtained from a fine-tuned NER model. ② **Selective Masking:** All words except the NEs and the keywords obtained from the previous step is replaced with mask tokens [**M**]. ③ **Labeled Sequence Linearization:** Label tokens are added before and after each entity in the sentence. ④ **Dynamic Masking:** The template goes through further masking where a small portion of the keywords are dynamically masked at each training iteration. While generation we also apply *mixner*, which randomly joins two templates after ③ and before ④. Post generating augmentations with ACLM, the generated augmentations are concatenated with the gold data and used to fine-tune our final NER model.

3.1 Template Creation

To corrupt a sentence and create a template, we follow a 4-step process described below:

1. **Keyword Selection**: For each sentence in our training corpus, we first obtain a set of non-NE tokens in the sentence that are most attended to by its NEs. We call these tokens *keywords*. For our research, we consider a non-NE token as a keyword if the NEs in the sentence contextually depend on them the most. We measure contextual dependency between NE and non-NE tokens using attention scores from attention maps extracted from a transformer-based NER model fine-tuned only on gold data. We hypothesize that attention heads in a transformer when fine-tuned for NER, formulated as a token-level tagging task, tend to pay the highest attention to the most contextually relevant tokens around it. Thus, formally put, consider a sentence with a total of T tokens comprised of t_{other} non-NE and t_{entity} NE tokens. Our primary aim is to find the top $p\%$ of t_{other} tokens, which we call keywords. To calculate the total attention score that each token in the sentence assigns to each other token, we sum up the attention scores across each of the heads in the transformer network and across the last a layers (a = 4 in our case). Different heads in different layers tend to capture different properties of language, and taking the average attention scores across the last 4 layers ensures that diverse linguistic relations are taken into account while choosing the keywords (e.g., syntactic, semantic, etc.). This also makes the keyword selection process more robust, as in low-resource conditions the attention maps may be noisy, and the NEs might not be focusing on the right context always. Additionally, the choice of just the last four layers is inspired by the fact that the lower layers have very broad attention and spend at most 10% of their attention mass on a single token (Clark et al., 2019). Note t_{entity} might be comprised of (1) multiple contiguous tokens forming an individual NE and (2) multiple such individual NEs. To handle the first case, inspired from Clark et al. (2019), we sum up the attention scores over all the individual tokens in the NE. For the second case, we find t_{attn} for each individual NE and take a set union of tokens in these t_{attn}. Thus, as an extra pre-processing step, to improve robustness, we also ignore punctuations, stop words, and other NEs from the top $p\%$ of t_{other} tokens to obtain our final keywords. We provide examples of templates in Appendix C.

2. **Selective Masking**: After selecting the top $p\%$ of t_{other} tokens in the sentence as keywords, we now have K non-NE keyword tokens and E entity tokens. To create the template, we now substitute each non-NE token not belonging to the K with the mask token and remove contiguous mask tokens.

3. **Labeled Sequence Linearization**: After we have our initial template, inspired by Zhou et al. (2022), we perform labeled sequence linearization to explicitly take label information into consideration during fine-tuning and augmentation generation. Similar to Zhou et al. (2022), as shown in Figure 1, we add label tokens before and after each entity token and treat them as the normal context in the sentence. Additionally, these label tokens before and after each NE provide boundary supervision for NEs with multiple tokens.

4. **Dynamic Masking**: Post labeled sequence linearization, our template goes through further masking wherein we dynamically mask a small portion of the K keywords during each iteration of training and generation. To be precise, we first sample a dynamic masking rate ε from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where the Gaussian variance σ is set to $1/K$. Next, we randomly sample tokens from the K keywords in the sentence according to the masking rate ε and replace this with mask tokens, followed by removing consecutive mask tokens. At every round of generation, dynamic masking helps boost 1) context diversity by conditioning ACLM generation on different templates with a different set of keywords and 2) length diversity by asking ACLM to infill a different number of mask tokens.

3.2 Fine-tuning ACLM

As discussed earlier, ACLM is fine-tuned on a novel text reconstruction from corrupted text task wherein the created templates serve as our corrupted text and ACLM learns to recover the original text from the template. Text reconstruction from the corrupted text is a common denoising objective that PLMs like BART and BERT are pre-trained on. For this work, we use it as our fine-tuning objective and differ from other existing pre-training objectives by our *selective masking* strategy for creating templates.

3.3 Data Generation

Post fine-tuning on the text reconstruction task, we utilize ACLM to generate synthetic data for data augmentation. For each sentence in the training dataset, we apply steps 1-4 in the Template Creation pipeline for R rounds to randomly corrupt the sentence and obtain a template which is then passed through the fine-tuned ACLM model to generate a total of $R\times$ augmented training samples. Additionally, to boost diversity, during auto-regressive generation, we randomly sample the next word from the *top-k* most probable words and choose the most probable sequence with beam search.

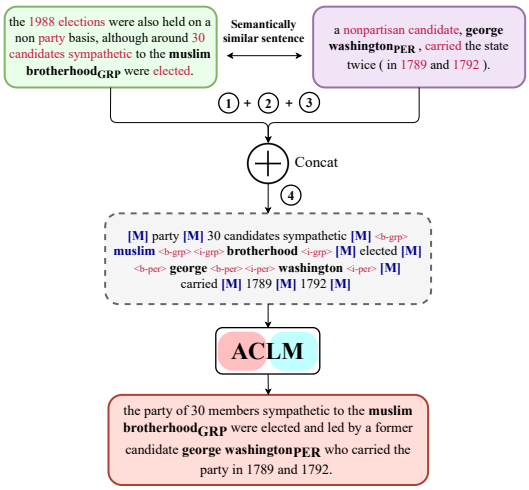


Figure 2: **Overview of *mixner*:** During the augmentation generation process, for a particular sentence in the training dataset, we retrieve another semantically similar sentence and concatenate them before step ④ of the template creation process. This merged template is then passed through ACLM to generate diverse augmentations that incorporate semantics and NEs from both sentences.

***mixner*:** During the R rounds of augmentation on our training dataset, we propose the use of *mixner*, a novel template mixing algorithm that helps ACLM generate diverse sentences with new context and multiple NEs in the sentence. More specifically, given the template for any arbitrary sentence a in the training set in step 3 of the template creation process, we retrieve the template for another sentence b that is semantically similar to a and join both the templates before passing on the template to step 4. We show examples of sentences generated with *mixner* in Fig. 3 and Section D.1. Note that we apply *mixner* only in the generation step and not during fine-tuning.

As mentioned earlier, to retrieve b from the training set, we randomly sample a sentence from the *top-k* sentences with the highest semantic similarity to a. To calculate semantic similarity between each sentence in the training set, we first take the embedding e for each sentence from a multi-lingual Sentence-BERT (Reimers and Gurevych, 2019) and then calculate semantic similarity by:

$$\text{sim}(e_i, e_j) = \frac{e_i \cdot e_j}{\|e_i\| \, \|e_j\|} \quad (1)$$

where $\text{sim}(.)$ is the cosine similarity between two embeddings , and $i, j \in N$ where $i \neq j$, and N

is the size if the training set. Additionally, we don't apply *mixner* on all rounds R but sample a probability γ from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ and only apply *mixner* if γ crosses a set threshold β.

3.3.1 Post-Processing

As a post-processing step, we remove augmentations similar to the original sentence and also the extra label tokens added in the labeled sequence linearization step. Finally, we concatenate the augmented data with the original data to fine-tune our NER model.

4 Experiments and Results

4.1 Dataset

All our experiments were conducted on the MultiCoNER dataset (Malmasi et al., 2022), a large multilingual dataset for complex NER. MultiCoNER covers 3 domains, including Wiki sentences, questions, and search queries, across 11 distinct languages. The dataset represents contemporary challenges in NER discussed in Section 2 and is labeled with six distinct types of entities: **person**, **location**, **corporation**, **groups** (political party names such as *indian national congress*), **product** (consumer products such as *apple iPhone 6*), and **creative work** (movie/song/book titles such as *on the beach*). We conduct experiments on a set of 10 languages $\mathbb{L}$ where $\mathbb{L}$ = {English (**En**), Bengali (**Bn**, Hindi (**Hi**), German (**De**), Spanish (**Es**), Korean (**Ko**), Dutch (**Nl**), Russian (**Ru**), Turkish (**Tr**), Chinese (**Zh**)}. Language-wise dataset statistics can be found in Table 12. We would also like to highlight that the number of sentences in MultiCoNER test sets ranges from **133,119 - 217,887**, which is much higher than test sets of other existing NER datasets. For more details on the dataset, we refer our readers to Malmasi et al. (2022). For monolingual and cross-lingual low-resource experiments, we perform iterative stratified sampling over all the sentences by using the entity classes in a sample as its target label across four low-resource settings (100, 200, 500, and 1000). We downsample the development set accordingly. For multi-lingual experiments, we combine all the data sampled for our monolingual settings. We evaluate all our systems and baselines on the original MultiCoNER test sets. We report micro-averaged F1 scores averaged across 3 runs for 3 different random seeds.

Algorithm 1 ACLM: Our proposed augmentation framework

Given training set $\mathbb{D}_{\text{train}}$, and PLM $\mathcal{L}$
$\mathbb{D}_{masked} \leftarrow \varnothing, \mathbb{D}_{aug} \leftarrow \varnothing$
for $\{X, Y\} \in \mathbb{D}_{train}$ **do** ▷ Training Loop
  $t_{other}, t_{entity} \leftarrow X$
  $K \leftarrow Top\ p\%\ of$ AttnMap(t_{other}) ▷ Keyword Selection
  $\tilde{X} \leftarrow$ GenTemplate$(X, \{t_{other}\} - \{K\})$ ▷ Selective Masking
  $\tilde{X} \leftarrow$ Linearize$(\tilde{X}, Y)$ ▷ Labeled Sequence Linearization
  $\mathbb{D}_{\text{masked}} \leftarrow \mathbb{D}_{\text{masked}} \cup \{\tilde{X}\}$
end for
for $\{X, Y\} \in \mathbb{D}_{masked}$ **do**
  $\tilde{X} \leftarrow$ DynamicMask(X, η) ▷ Dynamic Masking
  $\mathcal{L}_{finetune} \leftarrow$ FineTune$(\mathcal{L}, \tilde{X})$ ▷ Fine-tune ACLM
end for
for $\{X, Y\} \in \mathbb{D}_{train}$ **do** ▷ Generation Loop
  repeat $\mathcal{R}$ **times**:
    $\tilde{X} \leftarrow$ GenTemplate$(X, \{t_{other}\} - \{K\})$ ▷ Selective masking
    $\tilde{X} \leftarrow$ Linearize$(\tilde{X}, Y)$ ▷ Labeled Sequence Linearization
    $\tilde{X} \leftarrow$ DynamicMask$(\tilde{X}, \mu)$ ▷ Dynamic Masking
    $X_{aug} \leftarrow$ GenAug$(\mathcal{L}_{finetune}(\tilde{X}))$, $if\ \gamma < \beta$
    $X_{augmix} \leftarrow$ Mixner$(\mathcal{L}_{finetune}(\tilde{X}))$, $if\ \gamma > \beta$
    $\mathbb{D}_{aug} \leftarrow \mathbb{D}_{aug} \cup \{X_{aug}\} \cup \{X_{augmix}\}$
end for
$\mathbb{D}_{aug} \leftarrow$ PostProcess$(\mathbb{D}_{aug})$ ▷ Post-processing
return $\mathbb{D}_{train} \cup \mathbb{D}_{aug}$

4.2 Experimental Setup

ACLM. We use mBart-50-large (Tang et al., 2020) with a condition generation head to fine-tune ACLM. We fine-tune ACLM for 10 epochs using Adam optimizer (Kingma and Ba, 2014) with a learning rate of $1e^{-5}$ and a batch size of 32.

NER. We use XLM-RoBERTa-large with a linear head as our NER model. Though the field of NER has grown enormously, in this paper, we adhere to the simplest formulation and treat the task as a token-level classification task with a BIO tagging scheme. We use the Adam optimizer to optimize our model, set the learning rate to $1e^{-2}$, and train with a batch size of 16. The NER model is trained for 100 epochs, and the model with the best performance on the dev set is used for testing.

Hyper-parameter Tuning. For template creation during fine-tuning and generation, we set the selection rate p and the Gaussian μ to be 0.3 and 0.5, respectively. The number of augmentation rounds R is set as 5. For *mixner* we set Gaussian μ and β to be 0.5 and 0.7, respectively. All hyper-parameters are tuned on the development set with grid search. More details can be found in Appendix A.

4.3 Baselines

To prove the effectiveness of our proposed ACLM, we compare it with several strong NER augmentation baselines in the literature. In this sub-section, we briefly describe each of these baselines. All baselines were run for R rounds.

Gold-Only. The NER model is trained using only gold data from the MultiCoNER dataset without any augmentation.

		Monolingual											Cross-Lingual				
#Gold	**Method**	**En**	**Bn**	**Hi**	**De**	**Es**	**Ko**	**Nl**	**Ru**	**Tr**	**Zh**	**Avg**	**En → Hi**	**En → Bn**	**En → De**	**En → Zh**	**Avg**
	Gold-only	29.36	14.49	18.80	37.04	36.30	12.76	38.78	23.89	24.13	14.18	24.97	16.36	12.15	29.71	0.31	14.63
	LwTR	48.60	20.25	29.95	48.38	44.08	35.09	43.00	39.22	30.58	27.70	36.68	32.36	**24.59**	46.05	2.11	26.28
	DAGA	16.24	5.87	10.40	32.44	27.78	19.28	15.44	11.14	16.17	10.33	16.51	4.54	3.28	14.21	0.13	5.54
100	MELM	40.12	6.22	27.84	43.94	37.45	34.10	37.82	32.38	20.13	25.11	30.51	26.37	20.33	34.32	2.71	20.93
	ACLM *only entity*	14.06	17.55	19.60	29.72	38.10	31.57	38.47	27.40	35.62	26.34	27.84	21.72	16.55	30.93	1.58	17.69
	ACLM *random*	43.59	20.13	28.04	45.83	42.27	33.64	41.82	38.20	36.79	25.99	35.63	29.68	21.64	45.27	3.05	24.91
	ACLM *(ours)*	**48.76**	**23.09**	**33.53**	**48.80**	**44.14**	**38.35**	**46.22**	**39.48**	**37.20**	**35.12**	**39.47**	**32.52**	23.91	**46.48**	**3.58**	**26.62**
	Gold-only	51.83	19.31	33.68	49.62	45.16	42.51	47.83	31.55	26.76	32.34	38.06	36.90	27.44	48.70	3.76	29.20
	LwTR	52.88	23.85	34.27	50.31	47.01	42.77	52.01	40.18	35.92	30.57	40.98	40.07	32.36	48.95	6.04	31.85
	DAGA	33.30	17.12	19.58	35.10	33.56	26.50	38.04	29.83	23.35	25.66	28.20	18.92	14.37	29.32	1.79	16.10
200	MELM	47.83	5.47	29.67	45.85	42.08	36.62	49.47	41.84	31.25	32.27	36.24	27.55	18.80	41.10	6.21	23.41
	ACLM *only entity*	50.06	25.58	37.78	50.95	48.21	43.39	48.46	34.87	34.92	28.20	40.24	30.76	22.53	44.17	6.50	25.99
	ACLM *random*	52.69	35.26	39.83	51.14	48.70	42.19	48.71	39.68	37.26	34.22	42.96	36.52	27.19	47.73	7.12	29.64
	ACLM *(ours)*	**54.99**	**38.39**	**40.55**	**53.36**	**49.57**	**44.32**	**53.19**	**43.97**	**39.71**	**39.31**	**45.74**	**45.22**	**36.64**	**54.51**	**8.55**	**36.23**
	Gold-only	55.51	34.6	38.66	55.95	51.52	48.57	50.97	45.14	38.83	38.84	45.86	35.93	25.64	50.13	7.23	29.73
	LwTR	56.97	35.42	37.83	55.91	54.74	49.36	56.10	46.82	39.00	38.55	47.07	43.14	34.60	51.61	11.40	35.19
	DAGA	44.62	22.36	24.30	43.02	42.77	36.23	47.11	30.94	30.84	33.79	35.60	26.50	21.52	37.89	4.82	22.68
500	MELM	52.57	9.46	31.57	53.57	46.40	45.01	51.90	46.73	38.26	39.64	41.51	34.97	27.17	44.31	7.31	28.44
	ACLM *only entity*	57.55	35.69	35.82	56.15	53.64	50.20	53.07	46.40	41.58	38.65	46.87	35.48	29.37	49.10	7.99	30.48
	ACLM *random*	57.92	38.24	39.33	57.14	53.24	49.81	55.06	48.27	42.22	40.55	48.18	41.72	32.16	52.27	13.63	34.95
	ACLM *(ours)*	**58.31**	**40.26**	**41.48**	**59.35**	**55.69**	**51.56**	**56.31**	**49.40**	**43.57**	**41.23**	**49.72**	**44.36**	**35.59**	**54.04**	**16.27**	**37.57**
	Gold-only	57.22	30.20	39.55	60.18	55.86	53.39	60.91	49.93	43.67	43.05	44.40	43.44	33.27	54.61	5.34	34.17
	LwTR	59.10	39.65	43.90	61.28	57.29	51.37	59.25	52.04	44.33	43.71	51.19	43.32	33.74	53.32	7.38	34.44
	DAGA	50.24	32.09	35.02	51.45	49.47	42.41	51.88	41.56	33.18	39.51	42.68	33.12	26.22	42.13	5.15	26.65
1000	MELM	53.48	6.88	37.02	58.69	52.43	50.50	56.25	48.99	36.83	38.88	44.00	35.23	25.64	46.50	8.22	28.90
	ACLM *only entity*	55.46	38.13	41.84	60.05	56.99	53.32	58.22	50.17	45.11	39.62	49.89	37.38	29.77	41.10	6.49	28.69
	ACLM *random*	58.87	41.00	46.27	61.19	57.29	53.61	59.52	52.77	45.01	43.60	51.91	43.96	34.14	53.37	7.25	34.68
	ACLM *(ours)*	**60.14**	**42.42**	**48.20**	**63.80**	**58.33**	**55.55**	**61.22**	**54.31**	**48.23**	**45.19**	**53.74**	**44.59**	**35.70**	**56.74**	**8.94**	**36.49**

Table 1: Results of monolingual (Left) and cross-lingual (Right) low-resource complex NER. For cross-lingual experiments, we take English as the source language. ACLM obtains absolute average gains in the range of 1% - 22% over our baselines.

Label-wise token replacement (LwTR).(Dai and Adel, 2020b) A token in a sentence is replaced with another token with the same label; the token is randomly selected from the training set.

DAGA.(Ding et al., 2020) Data Augmentation with a Generation Approach (DAGA) proposes to train a one-layer LSTM-based recurrent neural network language model (RNNLM) by maximizing the probability for the next token prediction with linearized sentences. During generation, they use random sampling to generate entirely new sentences with only the [**BOS**] token fed to the model.

MulDA.(Liu et al., 2021) Multilingual Data Augmentation Framework (MulDA) builds on DAGA and trains a pre-trained mBART model on next token prediction with linearized sentences for generation-based multilingual data augmentation. For a fair comparison, we replace mBART in MulDA with mBART-50.

MELM.(Zhou et al., 2022) Masked Entity Language Modeling (MELM) proposes fine-tuning a transformer-encoder-based PLM on linearized labeled sequences using masked language modeling. MELM outperforms all other baselines and prior-art on low-resource settings on the CoNLL 2003 NER dataset across four languages in mono-lingual, cross-lingual, and multi-lingual settings.

ACLM *random*. We train and infer ACLM with templates created with randomly sampled *keywords* instead of taking *keywords* with high attention scores. This baseline proves the effectiveness of our *keyword* selection algorithm which provides NEs in the template with rich context.

ACLM *only entity*. We train and infer ACLM with templates created with only linearized entities and no *keywords*. This baseline proves the effectiveness of additional context in our templates.

4.4 Experimental Results

Monolingual Complex NER. Table 1 compares the performance of all our baselines with ACLM on the MultiCoNER test sets under various low-resource settings for 10 languages. As clearly evident, ACLM outperforms all our baselines in all settings by consistently achieving the best results in all individual languages. Moreover, ACLM improves over our neural baselines (MELM and DAGA) by a significant margin (absolute gains in the range of 1.5% - 22% across individual languages). Although LwTR performs better than ACLM in rare instances, we emphasize that (1) LwTR generates nonsensical, incoherent augmentations, (discussed further in Section D.1) and (2) Based on a learning-based paradigm, ACLM shows bigger margins to LwTR at slightly higher gold training samples (200

#Gold	Method	En	Bn	Hi	De	Es	Ko	Nl	Ru	Tr	Zh	Avg
	Gold-Only	56.21	35.66	42.16	55.71	54.98	45.14	57.48	46.13	44.40	30.72	46.86
	LwTR	55.65	38.47	43.44	54.71	53.95	44.78	56.50	46.93	45.41	31.56	47.14
100 ×10	MulDA	46.87	29.25	34.52	45.92	45.55	33.91	48.21	38.65	35.56	27.33	38.58
	MELM	53.27	23.43	41.55	48.17	51.28	39.23	51.37	45.73	41.97	30.67	42.67
	ACLM *(ours)*	**58.74**	**41.00**	**46.22**	**59.13**	**56.93**	**51.22**	**60.30**	**50.26**	**49.32**	**40.93**	**51.40**
	Gold-Only	58.67	39.84	46.34	59.65	58.50	50.70	60.79	51.66	47.12	40.98	51.42
	LwTR	51.78	35.93	38.87	52.73	51.59	42.55	54.49	43.99	41.23	35.19	44.83
200 ×10	MulDA	48.89	31.45	36.76	48.41	48.30	39.78	51.09	42.01	35.98	31.65	41.43
	MELM	52.53	24.27	40.10	49.69	52.42	43.56	47.28	44.35	40.62	34.28	47.45
	ACLM *(ours)*	**59.75**	**42.61**	**48.52**	**61.49**	**59.05**	**53.46**	**61.59**	**53.34**	**49.96**	**44.72**	**53.45**
	Gold-Only	61.10	40.94	48.20	61.67	59.84	54.56	62.36	53.33	48.77	45.82	53.66
	LwTR	59.09	38.37	43.80	59.37	57.76	50.38	60.42	51.00	46.53	42.87	50.96
500 ×10	MulDA	51.79	30.67	35.79	51.87	50.92	43.08	53.95	44.61	38.86	36.72	43.83
	MELM	58.67	26.17	41.88	53.05	57.26	51.97	61.49	43.73	40.22	40.12	47.66
	ACLM *(ours)*	**62.32**	**43.79**	**50.32**	**63.94**	**62.05**	**56.82**	**64.41**	**55.09**	**51.83**	**48.44**	**55.90**
	Gold-Only	64.14	43.28	50.11	66.18	63.17	57.31	65.75	56.94	51.17	49.77	57.78
	LwTR	61.67	39.90	45.28	63.13	60.21	53.43	63.37	54.07	48.38	45.36	53.48
1000 ×10	MulDA	56.35	33.73	40.71	56.90	55.35	48.42	58.39	49.25	42.06	40.19	48.14
	MELM	61.55	30.27	42.61	61.05	61.87	55.71	63.17	53.00	48.48	44.71	52.24
	ACLM *(ours)*	**64.50**	**46.59**	**52.14**	**67.65**	**64.02**	**59.09**	**67.03**	**57.82**	**53.25**	**50.60**	**58.27**

Table 2: Results of multi-lingual low-resource complex NER. ACLM obtains absolute gains in the range of 1% - 21%.

and 500) which we acknowledge is a reasonable size in real-world conditions.

Cross-lingual Complex NER. We also study the cross-lingual transferability of a NER model trained on a combination of gold and generated augmentations. Thus, we evaluated a model, trained on **En**, on 4 other languages, including **Hi**, **Bn**, **De**, and **Zh** in a zero-shot setting. ACLM outperforms our neural baselines by a significant margin (absolute gains in the range of 1% - 21%). None of these systems perform well in cross-lingual transfer to **Zh** which was also observed by (Hu et al., 2021).

Multi-lingual Complex NER. Table 2 compares the performance of all our baselines with ACLM on the MultiCoNER test sets under various multilingual low-resource settings. As clearly evident, ACLM outperforms all our baselines by a significant margin (absolute gains in the range of 1%-21% across individual languages). All our baselines, including our Gold-Only baseline, also perform better than their monolingual counterparts which demonstrates the effectiveness of multi-lingual fine-tuning for low-resource complex NER.

5 Further Analysis

5.1 Generation Quality

Quantitative Analysis. Table 3 compares augmentations from various systems on the quantitative measures of perplexity and diversity. Perplexity (Jelinek et al., 1977) is a common measure of text fluency, and we measure it using GPT2 (Radford et al., 2019). We calculate 3 types of diversity metrics: for Diversity-E and Diversity-N, we calculate the average percentage of new NE and non-NE words in the generated samples compared with the original samples, respectively. For Diversity-L, we

#Gold	Method	Perplexity(↓)	Diversity-E(↑)	Diversity-N(↑)	Diversity-L(↑)
	LwTR	137.01	30.72	16.46	0.0
200	MELM	83.21	**94.85**	0.0	0.0
	ACLM *(ours)*	**80.77**	35.64	**22.48**	**5.67**
	LwTR	129.349	30.07	16.22	0.0
500	MELM	82.31	**94.37**	0.0	0.0
	ACLM *(ours)*	**57.68**	44.12	**41.16**	**5.82**
	LwTR	131.20	29.85	16.55	0.0
1000	MELM	82.64	**95.13**	0.0	0.0
	ACLM *(ours)*	**62.00**	50.10	**34.84**	**5.40**

Table 3: Quantitative evaluation of generation quality from various systems on the measures of perplexity and diversity. Diversity-E, N, and L stand for Entity, Non-Entity, and Length, respectively.

calculate the average absolute difference between the number of tokens in generated samples and the original samples. ACLM achieves the lowest perplexity and highest non-NE and length diversity compared with other baselines. NE diversity in ACLM is achieved with *mixner* where ACLM fairs well compared to MELM which just replaces NEs. LwTR achieves the highest perplexity, thereby reaffirming that it generates incoherent augmentations.

Qualtitative Analysis. Fig. 3 illustrates the superiority of augmentations generated by ACLM when compared with our other baselines. As clearly evident, while MELM generates just minor changes in NEs, augmentations produced by LwTR often tend to be nonsensical and incoherent. On the other hand, ACLM generates meaningful and diverse sentences around NEs, which is further boosted with *mixner*. We provide examples in Appendix D.1.

5.2 Application to other domains

To evaluate the transferability of ACLM to other domains, we evaluate ACLM on 4 more datasets beyond MultiCoNER. These datasets include CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) (news), BC2GM (Smith et al., 2008) (bio-medical), NCBI Disease (Doğan et al., 2014) (bio-medical) and TDMSci (Hou et al., 2021) (science). Table 4 compares our baselines with ACLM across 2 low-resource settings on all 4 datasets. ACLM outperforms all our baselines on all settings except LwTR on CoNLL 2003. This occurs because LwTR generates a large variety of effective augmentations with NE replacement on easy entities in CoNLL 2003. The results demonstrate the effectiveness of ACLM over diverse domains, including domains with an acute scarcity of data (bio-medical). Additionally, we also emphasize that ACLM produces more factual augmentations and, unlike our other baselines, avoids context-entity mismatch, which makes the NER model store wrong knowledge in

Original	it was developed by a team led by former **[blizzard entertainment]**$_{CORP}$ employees, some of whom had overseen the creation of the **[diablo]**$_{CW}$ series.	✓ The original sentence describes the employees of an organization and provides details about them.
LwTR	it was developed by a makers led by, **[blizzard entertainment]**$_{CORP}$., some of whom had elevation the serving of the **[diablo]**$_{CW}$ 12th.	✗ LwTR replaces random words in the sentence, which makes it incoherent.
MELM	it was developed by a team led by former **[blizzago games]**$_{CORP}$ employees, some of whom had overseen the creation of the **[hablo]**$_{CW}$ series.	✗ MELM keeps the sentence coherent but generates new NEs that do not correspond to real-world entities.
ACLM *w/o mixner*	**[blizzard entertainment]**$_{CORP}$ employees have overseen the production of the animated films, including the production of the **[diablo]**$_{CW}$ series.	✓ ACLM generates new context patterns around the NE, keeping the sentence coherent and avoiding context-entity mismatch.
ACLM *w/ mixner*	the team of the **[blizzard entertainment]**$_{CORP}$ had overseen the creation of the game **[diablo]**$_{CW}$ and many of its workers founded **[pyro studios]**$_{CORP}$ in the early 1960s.	✓ mixner boosts ACLM diversity and still keeps the sentence coherent. It adds a NE in the sentence and augments the sentence with extra details about the employees of the organization.

Original	The control group consisted of 40 consecutive **[FMF]**$_{DISEASE}$ patients, who arrived at the **[FMF]**$_{DISEASE}$ clinic for their regular follow-up visit and were 40 years of age or older at the time of the examination.	✓ The original sentence describes an occasion where a group of 40 patients diagnosed with a certain kind of disease visited a clinic, and the sentence provides us with information on the age statistics of the patients.
LwTR	The control, consisted of 40 consecutive **[fragile]**$_{DISEASE}$ patients, who arrived at the **[FMF]**$_{DISEASE}$ status for their regular follow - up and were 40 years of age or older at the time of the examination analyzed	✗ LwTR replaces "FMF" in the sentence with "fragile" and the phrase "fragile patients" does not make sense. It also adds an extra word, "analyzed", at the end of the sentence.
MELM	The control group consisted of 40 consecutive **[FMR]**$_{DISEASE}$ patients, who arrived at the **[PDA]**$_{DISEASE}$ clinic for their regular follow-up visit and were 40 years of age or older at the time of the examination.	✗ MELM replaces the 1st occurrence of "FMF" in the sentence with "FMR" and the second occurrence with "PDA". "FMR" is not the name of a disease and is closest to "FMR1", which is the name of a gene. "PDA" stands for "Patent ductus arteriosus." Thus, the entire sentence does not make much sense.
ACLM *w/o mixner*	The sample consisted of four consecutive **[FMF]**$_{DISEASE}$ patients who arrived at the **[FMF]**$_{DISEASE}$ clinic for a visit of examination. Only one of the 4 remaining patients had **[FMF]**$_{DISEASE}$.	✓ ACLM introduces a new context pattern around the sentence. The entire sentence is coherent.
ACLM *w/ mixner*	Of 4000 (40%) patients with onset **[FMF]**$_{DISEASE}$, patients with **[FRDA]**$_{DISEASE}$ had no tendon reflexes at all.	✓ mixner boosts ACLM diversity and still keeps the sentence coherent. "FRDA" (Friedreich's ataxia) is a genetic disease that causes difficulty in walking and a loss of sensation in the arms and legs.

Figure 3: Examples of augmentations generated with different methods (Left) and explanation (Right). Words in red are Named Entities, and words underlined in the *Original* sentence are identified ACLM *keywords*. ACLM generates much more diverse, detailed, and coherent augmentations, which maintain factuality and also prove to be more effective. Generation diversity is further amplified with *mixner*.

#Gold	Method	CoNLL	BC2GM	NCBI	TDMSci	Avg
200	Gold-Only	79.11	50.01	72.92	47.20	62.31
	LwTR	**82.33**	52.78	72.15	51.65	64.73
	DAGA	76.23	47.67	71.14	48.03	60.77
	MELM	77.10	54.05	70.12	46.07	61.83
	ACLM *(ours)*	82.14	**58.48**	**74.27**	**56.83**	**67.93**
500	Gold-Only	84.82	55.56	75.75	47.04	65.79
	LwTR	**85.08**	60.46	78.97	60.74	71.31
	DAGA	81.82	51.23	78.09	57.66	67.20
	MELM	83.51	56.83	75.11	57.80	68.31
	ACLM *(ours)*	84.26	**62.37**	**80.57**	**61.77**	**72.24**

Table 4: Comparison of NER results on datasets from various different domains including news, science, and bio-medical.

data-sensitive domains. We show samples of generated augmentations in Fig. 3 and Appendix D.1.

6 Conclusion

In this paper, we propose ACLM, a novel data augmentation framework for low-resource complex NER. ACLM is fine-tuned on a novel text reconstruction task and is able to generate diverse augmentations while preserving the NEs in the sentence and their original word sense. ACLM effectively alleviates the context-entity mismatch problem and generates diverse, coherent, and high-quality augmentations that prove to be extremely effective for low-resource complex NER. Additionally, we also show that ACLM can be used as an effective data augmentation technique for low-resource NER in the domains of medicine and science due to its ability to generate extremely reliable augmentations.

Limitations

We list down some potential limitations of ACLM: 1) PLMs are restricted by their knowledge to generate entirely new complex entities due to their syntactically ambiguous nature. Adding to this, substituting complex NEs in existing sentences leads to context-entity mismatch. Thus, as part of future work, we would like to explore if integrating external knowledge into ACLM can help generate sentences with new complex entities in diverse contexts. 2) We do not conduct experiments in the language Farsi from the MultiCoNER dataset as neither mBart-50-large nor XLM-RoBERTa-large was pre-trained on this language. 3) The use of mBart-50-large for generation also restricts ACLM from being transferred to code-switched settings, and we would like to explore this as part of future work.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Sandeep Ashwini and Jinho D Choi. 2014. Targetable named entity recognition in social media. *arXiv preprint arXiv:1408.0782*.

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

M Saiful Bari, Tasnim Mohiuddin, and Shafiq Joty. 2020. Uxla: A robust unsupervised data augmentation framework for zero-resource cross-lingual nlp. *arXiv preprint arXiv:2004.13240*.

Gabriel Bernier-Colborne and Philippe Langlais. 2020. Hardeval: Focusing on challenging tokens to assess robustness of ner. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1704–1711.

Silvana Marianela Bernaola Biggio, Manuela Speranza, and Roberto Zanoli. 2010. Entity mention detection using a combination of redundancy-driven classifiers. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Hsu, and Kuansan Wang. 2014. Erd'14: entity recognition and disambiguation challenge. In *Acm Sigir Forum*, volume 48, pages 63–77. Acm New York, NY, USA.

Beiduo Chen, Jun-Yu Ma, Jiajun Qi, Wu Guo, Zhen-Hua Ling, and Quan Liu. 2022. Ustc-nelslip at semeval-2022 task 11: Gazetteer-adapted integration network for multilingual complex named entity recognition. *arXiv preprint arXiv:2203.03216*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

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*.

Xiang Dai and Heike Adel. 2020a. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Xiang Dai and Heike Adel. 2020b. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. Daga: Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv:2011.01549*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

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*, pages 2777–2790, Seattle, United States. Association for Computational Linguistics.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2021. Tdmsci: A specialized corpus for scientific literature entity tagging of tasks datasets and metrics. In *Proceedings of the the 16th conference of the European Chapter of the Association for Computational Linguistics,* Online, 19–23 April 2021.

Hai Hu, He Zhou, Zuoyu Tian, Yiwen Zhang, Yina Ma, Yanting Li, Yixin Nie, and Kyle Richardson. 2021. Investigating transfer learning in multilingual pre-trained language models through chinese natural language inference. *arXiv preprint arXiv:2106.03983*.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

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

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. Low-resource ner by data augmentation with prompting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4252–4258.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. Mulda: A multilingual data augmentation framework for low-resource cross-lingual ner. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5834–5846.

Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. 2019. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. Multiconer: a large-scale multilingual dataset for complex named entity recognition. *arXiv preprint arXiv:2208.14536*.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. *arXiv preprint arXiv:1903.11222*.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime Carbonell. 2020. Soft gazetteers for low-resource named entity recognition. *arXiv preprint arXiv:2005.01866*.

H Andrew Schwartz, Fernando Gomez, and Lyle Ungar. 2012. Improving supervised sense disambiguation with web-scale selectors. In *Proceedings of COLING 2012*, pages 2423–2440.

Larry Smith, Lorraine K Tanabe, Cheng-Ju Kuo, I Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, Manabu Torii, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):1–19.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Erik F. 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 - Volume 4*, CONLL '03, page 142–147, USA. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated Concatenation of Embeddings for Structured Prediction. In *the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (**ACL-IJCNLP 2021**)*. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Xiaoshi Zhong and Erik Cambria. 2021. *Time Expression and Named Entity Recognition*. Springer.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471, Florence, Italy. Association for Computational Linguistics.

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

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. *arXiv preprint arXiv:2104.08656*.

A Hyperparameter Tuning

All hyperparameters were originally tuned with grid search on the development set. In this section, we show performance on the test set for better analysis.

Keyword Selection rate p: The keywords in our template provide the model with contextually relevant additional knowledge about the NEs during training and generation. However, we are faced with the question: *How much context is good context?*. Too less context, like our ACLM *only entity* baseline with only linearized NEs in the template, might make it difficult for the model to know the appropriate context of the syntactically ambiguous complex NE and thus might lead to sentences generated with a context-entity mismatch (for e.g. *sam is reading on the Beach* where *on the beach* might be a name of a movie). On the contrary, retaining too many words from the original sentence in our template might lead to a drop in the diversity of generated sentences as the model needs to *infill* only a small portion of the words. To determine the optimal value of p we experiment on 2 low-resource settings on the English sub-set of MultiCoNER and report the micro F1 results on the test-set for $p \in \{0, 0.1, 0.2, 0.3. 0.4, 0.5, 0.6, 0.7\}$. All other hyperparameters are kept constant. As shown in Table 5, $p = 0.3$ gives us the best test-set performance, and the performance decreases after 0.4.

#Gold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
200	50.06	51.82	53.99	**54.99**	51.05	54.28	52.16	54.34
500	57.55	56.12	57.93	**58.31**	57.55	56.88	56.60	58.10

Table 5: Test set F1 for various Keyword Selection rates.

Augmentation rounds R**:** Augmenting the training dataset with several augmentation rounds R proves effective until a saturation point is reached. Continuing to add more augmented data to the gold dataset starts introducing noise to the combined data. Additionally, with an increase in R, the chances of auto-regressive generation with *top-k* sampling generating similar sentences increase. To determine the optimal value of R, we experiment on 2 low-resource settings on the English sub-set of MultCoNER and report the micro F1 results on the test-set for R ranging from 1 to 7. All other hyperparameters are kept constant. As shown in Table 5, $R = 5$ gives us the best test-set performance, and the performance decreases after 5 rounds.

Attention layers a **for Keyword Selection:** Selecting the right keywords for creating a template is integral to the success of ACLM. A clear example of this can be seen in Table 1, where ACLM outperforms ACLM *random* (which chooses random tokens as keywords for template creation) by a significant margin. Transformer encoders consist of multiple layers, and each layer consists of multiple attention heads. While all heads in the same layer tend to behave similarly, different layers generally encode different semantic and syntactic information (Clark et al., 2019). Thus we experiment with different values of α, or different combinations of transformer encoder layers which are used for calculating the attention scores for keyword selection. As mentioned in Section 3.1, by default, we average attention scores across all tokens, all heads, and the last α layers. For all our low-resource experiments, we use attention maps from a 24-layer XLMRoBERTa-large fine-tuned on the low-resource gold dataset for that particular setting. Table 7 compares the performance of 3 settings of α on 2 low-resource settings on the English sub-set of MultCoNER: **1.** Only last layer **2.** Last 4 layers. **3.** All 24 layers. As clearly evident, though setting 2 achieves the best performance, the difference in performance among different values of α is not too high. As part of future work, we would like to explore better ways to search for the optimal α.

#Gold	1	2	3	4	5	6	7
200	52.37	53.96	52.40	50.05	**54.99**	53.46	53.75
500	58.24	58.17	57.90	58.11	**58.31**	57.20	57.40

Table 6: Test set F1 for the number of augmentation rounds.

#Gold	1	2	3
200	52.43	**54.99**	54.13
500	58.09	**58.31**	58.15

Table 7: Test set F1 for various settings of α

B Additional Results

Current state of state-of-the-art: Most current state-of-the-art systems are built and evaluated on common NER benchmarks like CoNLL 2003 and OntoNotes v5.0. As discussed in Section 2, these benchmarks do not represent contemporary challenges in NER and contain sentences with easy entities and rich context. Table 8 compares the performance of a simple XLM-R (Conneau et al., 2019), and Co-regularized LUKE (Zhou and Chen,

2021) (SOTA NER system) on 2 common NER and 1 complex NER benchmarks in both low- and high-resource settings. As we can clearly see, both systems achieve remarkable performance on both CoNLL 2003 and OntoNotes v5.0 but struggle on MultiCoNER. Additionally, the gap widens in low-resource settings.

#Gold	Method	XLM-R	Co-regularized LUKE
500	CoNLL 2003	84.82	86.92
	OntoNotes	65.48	64.92
	MultiCoNER	55.51	55.12
All	CoNLL 2003	92.21	92.56
	OntoNotes	85.07	87.57
	MultiCoNER	70.31	69.58

Table 8: Performance comparison of XLM-RoBERTa (Conneau et al., 2019) and Co-regularized LUKE (Zhou and Chen, 2021) on two common benchmark NER datasets and MultiCoNER (Malmasi et al., 2022) (complex NER benchmark) in both high- and low-resource settings. Co-regularized LUKE is the current SOTA NER system on both CoNLL 2003 and OntoNotes v5.0. *Complex NER remains a difficult NLP task in both low- and high-resource labeled data settings.*

Training on the entire dataset: Beyond just evaluating ACLM performance on low-resource settings, we also compare ACLM with all our baselines on the entire MultiCoNER dataset (each language split contains ≈ 15300 sentences). Similar to low-resource settings, ACLM outperforms all our baselines across all languages and achieves an absolute average gain of 1.58% over our best baseline.

Method	En	Bn	Hi	De	Es	Ko	Nl	Ru	Tr	Zh	Avg
Gold-only	71.25	59.10	61.59	75.33	67.71	65.29	71.55	68.76	62.44	60.56	66.36
LwTR	71.22	58.86	60.72	75.50	70.06	65.80	72.94	68.26	62.70	58.74	66.48
DAGA	64.30	47.93	53.03	67.70	62.07	59.84	65.37	60.72	52.45	55.32	58.87
MELM	66.27	56.27	61.04	71.25	65.56	63.71	70.43	66.28	60.74	57.72	63.93
ACLM *(ours)*	**72.69**	**60.13**	**62.58**	**77.26**	**70.89**	**67.01**	**73.28**	**69.90**	**65.24**	**61.63**	**68.06**

Table 9: Result comparison Complex NER. **Avg** is the average result across all languages. ACLM outperforms all our baselines.

Entity-wise Performance Analysis: Previous to MultiCoNER, common benchmark datasets like CoNLL 2003 had only "easy entities" like names of Persons, Locations, and Organizations. The MultiCoNER dataset has 3 additional types of NEs, namely Products (**PROD**), Groups (**GRP**), and Creative Work (**CW**). These entities are syntactically ambiguous, which makes it challenging to recognize them based on their context. The top system from WNUT 2017 achieved 8% recall for creative work entities. Table 10 compares the entity-wise performance of ACLM with our various baselines on two low-resource settings on the MultiCoNER dataset. All results are averaged across all 10 languages. ACLM outperforms all our baselines on all individual entities, including PROD, GRP, and CW, which re-affirms ACLM's ability to generate effective augmentation for complex NER.

#Gold	Method	PER	LOC	PROD	GRP	CORP	CW
200	Gold-Only	56.35	42.32	30.10	31.36	33.83	23.30
	LwTR	56.13	41.78	34.87	36.52	39.30	27.46
	DAGA	45.19	35.40	19.96	21.92	19.60	14.33
	MELM	52.16	41.16	30.24	28.61	34.13	22.77
	ACLM *(ours)*	**64.42**	**48.92**	**41.76**	**37.31**	**44.08**	**30.61**
500	Gold-Only	63.05	48.48	42.75	37.55	45.10	31.34
	LwTR	64.80	**54.17**	45.70	**44.06**	50.80	35.10
	DAGA	51.82	41.11	28.58	30.50	34.10	21.61
	MELM	58.41	45.64	37.04	34.11	40.42	28.33
	ACLM *(ours)*	**66.49**	51.24	**48.87**	42.00	**51.55**	**35.18**

Table 10: Entity-wise performance comparison of different augmentation methods. Results are averaged across all languages.

Length-wise Performance Analysis: As mentioned in Section 2, low-context is a major problem in complex NER, and an effective complex NER system should be able to detect NEs in sentences with both low and high context (by context we refer to the number of words around the NEs in the sentence). By the nature of its fine-tuning pipeline, ACLM is able to generate augmentations of variable length, and our dynamic masking step further boosts the length diversity of generated augmentations. Adding to this, we acknowledge that effective augmentations for syntactically complex entity types should enable a model to learn to detect these entities in even low-context. Table 11 compares the entity-wise performance of ACLM with our various baselines on two low-resource settings on the MultiCoNER dataset. All results are averaged across all 10 languages. ACLM outperforms all our baselines across all length settings, which re-affirms ACLM's ability to generate effective augmentation for complex NER. To be specific, ACLM improves over our best baseline by 8.8% and 7.4% for 200 and 3.2% and 6.7% for 500 for low- and high-context sentences, respectively.

#Gold	Method	len < 5	5 ≤ len < 10	10 ≤ len
200	LwTR	26.35	34.38	43.56
	DAGA	18.20	29.53	39.49
	MELM	23.27	38.81	50.29
	ACLM *(ours)*	**35.10**	**47.25**	**57.72**
500	LwTR	34.04	42.74	56.47
	DAGA	23.00	38.18	51.09
	MELM	27.46	44.74	57.91
	ACLM *(ours)*	**37.42**	**52.23**	**63.13**

Table 11: Length-wise performance comparison of different augmentation methods. Results are averaged across all languages. ACLM outperforms all our baselines across all settings.

		Linguistically coherent entities	Context-Entity Match
Original	she became opposed to abortion in 1992 while attending a **bible** study and has since spoken out about how abortion has negatively impacted her life.		
LwTR	she became average to abortion in guitar while attending a **bible** study and has since spoken out about how academy has negatively impacted her life.	✗	✓
MELM	she became opposed to abortion in 1992 while attending a **vegetable** study and has since spoken out about how abortion has negatively impacted her life.	✗	✗
ACLM $w/o\ mixner$	the **bible** warned against abortion and said abortion had negatively impacted the welfare state.	✓	✓
ACLM $w/\ mixner$	while attending the **bible** seminar in 1964 at the **university of pittsburgh** he earned a master of science degree in biology.	✓	✓

Figure 4: Analysis and comparison of augmentations generated by our baselines with ACLM. Words **underlined** are the NEs. Context entity mismatch occurs when the generated NEs do not fit the surrounding context. Linguistic incoherence refers to cases where a generated NE does not follow the linguistic pattern for that particular type of NE or context.

C Templates and Attention Maps

Creating templates with *keywords* that effectively provides the PLM with additional knowledge about the NEs in the sentence is an integral part of ACLM. Fig. 11, 12, 13, 14, 15 shows examples of templates created for our sentences in MultiCoNER English subset, Spanish subset, Hindi subset NCBI Disease and TDMSci datasets, respectively. Additionally, we provide examples of attention maps used to create templates in Fig. 16f.

D Qualitative Analysis of Augmentations

D.1 Augmentation Examples

MultiCoNER Dataset: We provide additional examples of augmentations generated by ACLM and all our baselines in Fig. 9 and Fig. 10 for Hindi and English subsets of MultiCoNER dataset respectively.

Extra Datasets: Fig 5, 6, 7 and 8 illustrate augmetation examples for CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) (news), BC2GM (Smith et al., 2008) (bio-medical), NCBI Disease (Doğan et al., 2014) (bio-medical) and TDMSci (Hou et al., 2021) (science) datasets respectively. Except for on CoNLL 2003 datasets, both our baselines, LwTR and MELM, generate incoherent and unreliable training samples for the other 2 datasets. We only compare ACLM with LwTR and MELM as these methods don't generate augmentations from scratch and modify existing sentences. We define unreliable sentences as sentences generated with an entity-context mismatch (eg. a NE describing a disease prone to cows is placed in the context of humans or vice-versa). Generating unreliable augmentations prove fatal in data-sensitive domains like bio-medical as it may make the model store wrongful knowledge. Our detailed analysis of generated augmentations shows that: **(1)** LwTR is prone to generating such incoherent sentences because it randomly samples entities from the corpus with the same tag for replacement. **(2)** MELM on the other hand, fine-tuned on a transformer-encoder-based PLM, gets to see the entire context of the sentence for generating a new NE. However, it does not learn to focus on particular keywords and tends to generate a new NE based on the broader context of the sentence (e.g., it does not learn to differentiate between human and cow diseases and generates a new NE based on the broader context of the sentence). **(3)** ACLM generates highly reliable samples by conditioning on templates with keywords related to the NE. We illustrate examples of such templates in Fig. 14 and 15.

E Additional Details

Model Parameters: XLM-RoBERTa-large has ≈ 355M parameters with 24-layers of encoder, 1027-hidden-state, 4096 feed-forward hidden-state and 16-heads. mBART-50-large ≈ has 680M parameters with 12 layers of encoder, 12 layers of decoder, 1024-hidden-state, and 16-heads.

Compute Infrastructure: All our experiments are conducted on a single NVIDIA A100 GPU. An entire ACLM training pipeline takes ≈ 40 minutes.

Dataset Details: We use 5 datasets in total for our experiments: MultiCoNER [2] (Malmasi et al.,

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

Original	The **[European Commission]**$_{ORG}$ said on Thursday it disagreed with **[German]**$_{MISC}$ advice to consumers to shun **[British]**$_{MISC}$ lamb until scientists determine whether mad cow disease can be transmitted to sheep.
LwTR	The **[European Sox]**$_{ORG}$ seed on Thursday it disagreed with **[German]**$_{MISC}$ advice to consumers to shun **[British]**$_{MISC}$ regarding until scientists determine whether mad 70 disease can be -- to sheep 1
MELM	**[France]**$_{LOC}$'s **[Aquaculture Committee]**$_{ORG}$ suggested on Wednesday that consumers avoid eating meat from **[German]**$_{MISC}$ sheep until scientists determine whether mad cow disease can be transmitted to the animals.
ACLM *w/o mixner*	The **[European Commission]**$_{ORG}$ said on Thursday that consumers should shun **[British]**$_{MISC}$ lamb until scientists determine whether the disease can be transmitted to humans.
ACLM *w/ mixner*	The **[European Commission]**$_{ORG}$ has a scientific and multidisciplinary group of veterinary scientists who disagreed with the consumers on Thursday and decided to shun them out until scientists determine whether the **[Bovine Spongiform Encephalopathy]**$_{MISC}$ (BSE) -- mad cow disease can be transmitted.

Figure 5: Augmentation examples of the CoNLL 2003 dataset from the news domain. All generations are produced in a low-resource setting (500 training examples).

Original	To determine the genetic basis for the differences between the cardiac and **[brain AE3 variants]**$_{GENE}$, we isolated and characterized the rat gene.
LwTR	To determine the genetic basis for the differences between the cardiac and **[IgA AE3 related]**$_{GENE}$, we isolated and characterized the rat immunodeficiency increased
MELM	To determine the genetic basis for the differences between the cardiac and **[mouse EFR varianter]**$_{GENE}$, we isolated and characterized the rat gene.
ACLM *w/o mixner*	The genetic basis for the cardiac **[brain AE3 variants]**$_{GENE}$ in the rat population is unknown.
ACLM *w/ mixner*	On basis of the differences in both **[brain AE3 variants]**$_{GENE}$ and **[estrogen receptors]**$_{GENE}$ we isolated the mechanisms that govern the variations in mouse and human genes.

Figure 6: Augmentation examples of BC2GM from the bio-medical domain. All generations are produced in a low-resource setting (500 training examples).

Original	In order to understand the genetic and phenotypic basis for **[DPD deficiency]**$_{DISEASE}$, we have reviewed 17 families presenting 22 patients with complete **[deficiency of DPD]**$_{DISEASE}$.
LwTR	In order to understand the genetic and phenotypic basis for **[DPD deficiency]**$_{DISEASE}$, we have pathology 17 families presenting transcription patients with 292 deficiency of **[DPD constructed]**$_{DISEASE}$.
MELM	In order to understand the genetic and phenotypic basis for **[DDA eficiendency]**$_{DISEASE}$, we have reviewed 17 families presenting 22 patients with complete **[confferency cardiac disorderF]**$_{DISEASE}$.
ACLM *w/o mixner*	To determine the phenotypic basis of this **[DPD deficiency]**$_{DISEASE}$ gene, we reviewed the gene in 22 patients with an unusual **[deficiency of DPD]**$_{DISEASE}$.
ACLM *w/ mixner*	We examined the phenotypic basis of **[DPD deficiency]**$_{DISEASE}$ in four families with patients suffering from **[deficiency of DPD]**$_{DISEASE}$ (Twenty - eight patients with a **[protein S deficiency]**$_{DISEASE}$ and **[PROS1 gene defect]**$_{DISEASE}$).

Figure 7: Augmentation examples of NCBI dataset from the bio-medical domain. All generations are produced in a low-resource setting (500 training examples).

Original	These data show that if we are ever to fully master **[natural language generation]**$_{TASK}$, especially for the genres of news and narrative, researchers will need to devote more attention to understanding how to generate descriptive, and not just distinctive, referring expressions.
LwTR	These data show that if we are ever to runs focus **[Urdu/generation]**$_{TASK}$, proposed for the genres of + and narrative, researchers will need to devote more attention to understanding how to Fixed descriptive, and not just raw transformed morphological supervised corpora.
MELM	These data show that if we are ever to fully master **[the text interpretation]**$_{TASK}$, especially for the genres of news and narrative, researchers will need to devote more attention to understanding how to generate descriptive, and not just distinctive, referring expressions.
ACLM *w/o mixner*	These results show that in the **[natural language generation]**$_{TASK}$ of news text, researchers are able to generate descriptive text with distinctive language expressions.
ACLM *w/ mixner*	These data show that if we are ever to fully master **[natural language generation]**$_{TASK}$ for genres other than narrative, researchers will be able to generate descriptive and distinctive meaning by referring to them. We propose a holistic approach to **[image description generation]**$_{TASK}$ that is noisy and challenging.

Figure 8: Augmentation examples of TDMSci from the science domain. All generations are produced in a low-resource setting (500 training examples).

Original	**[हँसेल और ग्रेटल]**$_{CW}$, एक परी कथा जिसमें नामांकित पात्र ब्रेडक्रंब का निशान छोड़ते हैं
LwTR	**[ओपनऑफिस और ग्रेटल]**$_{CW}$, एक किया तक जिसमें रेटिंग पात्र ब्रेडक्रंब का में किया। हैं
MELM	**[ी के जूरा]**$_{CW}$, एक परी कथा जिसमें नामांकित पात्र ब्रेडक्रंब का निशान छोड़ते हैं
ACLM *w/o mixner*	**[हँसेल और ग्रेटल]**$_{CW}$ की कथा को १९९९ में नामांकित किया गया था।
ACLM *w/ mixner*	**[हँसेल और ग्रेटल]**$_{CW}$ की परी कथा को नामांकित किया गया था , जिसे **[निलेसातो]**$_{GRP}$ सैटेलाइट नेटवर्क द्वारा प्रसारित किया जाता है।

Original	उन्होंने १९०० में **[हार्वर्ड विश्वविद्यालय]**$_{GRP}$ से मास्टर डिग्री और १९०४ में डॉक्टरेट की उपाधि प्राप्त की।
LwTR	उन्होंने १९०० है। **[हार्वर्ड विश्वविद्यालय]**$_{GRP}$ से १९९३ डिग्री और १९०४ में डॉक्टरेट की उपाधि प्राप्त की।
MELM	उन्होंने १९०० में **[बॉल्ड कॉलेज]**$_{GRP}$ से मास्टर डिग्री और १९०४ में डॉक्टरेट की उपाधि प्राप्त की।
ACLM *w/o mixner*	उन्होंने १९०० में **[हार्वर्ड विश्वविद्यालय]**$_{GRP}$ से आर्किटेक्चर की डिग्री प्राप्त की।
ACLM *w/ mixner*	वह **[हार्वर्ड विश्वविद्यालय]**$_{GRP}$ से स्नातक की डिग्री प्राप्त करने के बाद डॉक्टरेट की उपाधि प्राप्त करने के लिए **[डिजाइन के हार्वर्ड ग्रेजुएट स्कूल]**$_{GRP}$ में आर्किटेक्चर इंजीनियर बन गए।

Figure 9: Augmentation examples on the Hindi subset of the MultiCoNER dataset. All generations are produced in a low-resource setting (500 training examples).

Original	gibson was educated at **[harrow school]**$_{GRP}$, where he played in the cricket team, and at **[trinity college]**$_{LOC}$.
LwTR	gibson was early at **[real pictures]**$_{GRP}$, where he played in the cricket team seventh and at **[trinity college]**$_{LOC}$.
MELM	gibson was educated at **[harford schools]**$_{GRP}$, where he played in the cricket team, and at **[is college]**$_{LOC}$.
ACLM *w/o mixner*	gibson was educated at **[harrow school]**$_{GRP}$ and played on the football team at **[trinity college]**$_{LOC}$.
ACLM *w/ mixner*	gibson was educated at **[harrow school]**$_{GRP}$, then at **[trinity college]**$_{LOC}$ and then at the missionary college of **[stavanger]**$_{LOC}$ from which he graduated in 1946.

Original	in previous years he had worked with **[alex cox]**$_{PER}$ on the soundtracks of his films **[sid and nancy]**$_{CW}$ and **[walker]**$_{CW}$ in 1986 and 1987.
LwTR	in previous years he had worked with **[alex pauwels]**$_{PER}$ on the soundtracks of his actor **[illegal and nancy]**$_{CW}$ and **[family]**$_{CW}$ in 1986 and 1987.
MELM	in previous years he had worked with **[roux wilsmith]**$_{PER}$ on the soundtracks of his films **[du, the ware]**$_{CW}$ and walkaway in 1986 and 1987.
ACLM *w/o mixner*	**[alex cox]**$_{PER}$ wrote the soundtracks for his films **[sid and nancy walker]**$_{CW}$ in 1987 .
ACLM *w/ mixner*	**[alex cox]**$_{PER}$ wrote the soundtracks for his film **[sid and nancy and walker]**$_{CW}$ and appeared in many of his films, including **[powder]**$_{CW}$, **[simply irresistible]**$_{CW}$ and **[d-tox]**$_{CW}$.

Figure 10: Augmentation examples on the English subset of the MultiCoNER dataset. All generations are produced in a low-resource setting (500 training examples).

Original	***Template***
speech pathologist **[lionel logue]**$_{PER}$ taught at the school from 1910 to 1911.	[M] **speech pathologist** [M] <B-PER> **lionel** <B-PER> <I-PER> **logue** <I-PER> [M] **taught** [M] **school** [M]
they were designed for interim use until the **[m73 machine gun]**$_{PROD}$ could be fielded.	[M] **designed** [M] **interim** [M] <B-PROD> **m73** <B-PROD> <I-PROD> **machine** <I-PROD> <I-PROD> **gun** <I-PROD> [M] **fielded** [M]
its aircraft and crews operate for its partly owned leisure subsidiary **[holiday europe]**$_{CORP}$.	[M] **aircraft** [M] **owned leisure subsidiary** <B-CORP> **holiday** <B-CORP> <I-CORP> **europe** <I-CORP> [M]

Figure 11: Examples of templates created for sentences taken from the English subset of the MultiCoNER dataset. All templates shown are created in a low-resource setting (500 training examples). Words underlined are identified *keywords*.

Original	*Template*
además fue lanzado como sencillo en algunos países, junto con la Segunda cancíon del álbum, **[waiting for the sun]**$_{CW}$.	[M] **lanzado** [M] **sencillo** [M] **países** [M] **cancíon** [M] **álbum** [M] <B-CW> **waiting** <I-CW> <I-CW> **for** <I-CW> <I-CW> **the** <I-CW> **sun** <I-CW> [M]
La revista **[time]**$_{CW}$ la agregó en una lista de las veinticinco mejores películas de animación	[M] **revista** [M] <B-CW> **time** <B-CW> [M] **lista** [M] **veinticinco mejores películas** [M]
En 2003, **[ebro foods]**$_{CORP}$, prpietaria de la factoría, decidió cesar la actividad.	[M] <B-CORP> **ebro** <B-CORP> <I-CORP> **foods** <I-CORP> [M] **prpietaria** [M] **factoría** [M] **decidió cesar** [M] **actividad** [M]

Figure 12: Examples of templates created for sentences taken from the Spanish subset of the MultiCoNER dataset. All templates shown are created in a low-resource setting (500 training examples). Words underlined are identified *keywords*.

Original	*Template*
आधिकारिक तौर पर बैंड समाप्त हो गया, लेकिन २००१ में अपने एल्बम **[जीवन की साँसे]**$_{CW}$ के साथ वापसी की।	[M] **आधिकारिक** [M] **बैंड** [M] **२००१** [M] **एल्बम** [M] <B-CW> **जीवन** <B-CW> <I-CW> **की** <I-CW> <I-CW> **साँसे** <I-CW> [M] **वापसी की** [M]
अगले सफल वर्षों में इसका विस्तार हुआ और **[मेट्रो मनिला]**$_{LOC}$ क्षेत्र में नए परिसरों की स्थापना हुई ।	[M] <B-LOC> **मेट्रो** <B-LOC> <I-LOC> **मनिला** <I-LOC> **क्षेत्र** [M] **नए परिसरों** [M] **स्थापना हुई** [M]
पास्ता को **[मेज़]**$_{PROD}$ क्षुधावर्धक के रूप में भी परोसा जा सकता है।	[M] **पास्ता** [M] <B-PROD> **मेज़** <B-PROD> [M] **परोसा** [M]

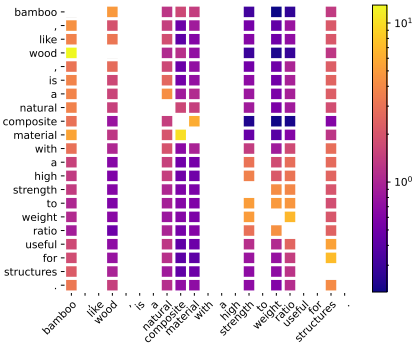
Figure 13: Examples of templates created for sentences taken from the Hindi subset of the MultiCoNER dataset. All templates shown are created in a low-resource setting (500 training examples). Words underlined are identified *keywords*.

Original	*Template*
Within the kidney, **[VHL]**$_{DISEASE}$ mRNA was differentially expressed within renal tubules suggesting that the **[VHL]**$_{DISEASE}$ gene product may have a specific role in kidney development.	[M] **kidney** [M] <B-DISEASE> **VHL** <I-DISEASE> [M] **mRNA** [M] **differentially expressed** [M] **renal tubules** [M] <B-DISEASE> **VHL** <I-DISEASE> [M] **gene product** [M] **kidney development** [M]
In conclusion , we demonstrated that a point mutation in a lariat branchpoint consensus sequence causes a null allele in a patient with **[FED]**$_{DISEASE}$.	[M] **demonstrated** [M] **mutation** [M] **branchpoint consensus sequence** [M] **allele** [M] **patient** [M] <B-DISEASE> **FED** <B-DISEASE> [M]
Mutations associated with variant phenotypes in **[ataxia-telangiectasia]**$_{DISEASE}$.	[M] **Mutations** [M] **variant phenotypes** [M] <B-DISEASE> **ataxia** <B-DISEASE> <B-DISEASE> **-** <B-DISEASE> <B-DISEASE> **telangiectasia** <B-DISEASE> [M]

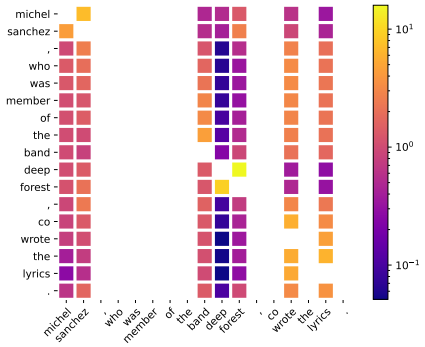
Figure 14: Examples of templates created for sentences taken from the NCBI Disease dataset. All templates shown are created in a low-resource setting (500 training examples). Words underlined are identified *keywords*.

Original	*Template*
Statistical approaches to **[machine translation (SMT)]**$_{TASK}$ use sentence-aligned, parallel corpora to learn translation rules along with their probabilities.	[M] **Statistical approaches** [M] <B-TASK> **machine** <B-TASK> <I-TASK> **translation** <I-TASK> <I-TASK> **(** <I-TASK> <I-TASK> **SMT** <I-TASK> **)** <I-TASK> [M] **sentence** [M] **aligned** [M] **corpora** [M] **translation** [M]
The goal of fully unsupervised **[word segmentation]**$_{TASK}$, then, is to recover the correct boundaries for arbitrary natural language corpora without explicit human parameterization.	[M] **goal** [M] **unsupervised** <B-TASK> **word** <B-TASK> <I-TASK> **segmentation** <I-TASK> [M] **correct boundaries** [M] **language corpora** [M] **human parameterization** [M]
In particular , for **[question classification]**$_{TASK}$, no labeled question corpus is available for French, so this paper studies the possibility to use existing English corpora and transfer a classification by translating the question and their labels .	[M] <B-TASK> **question** <B-TASK> <I-TASK>**classification** <I-TASK> [M] **labeled question corpus** [M] **French** [M] **paper** [M] **existing English corpora** [M] **classification** [M] **translating** [M] **question** [M] **labels** [M]

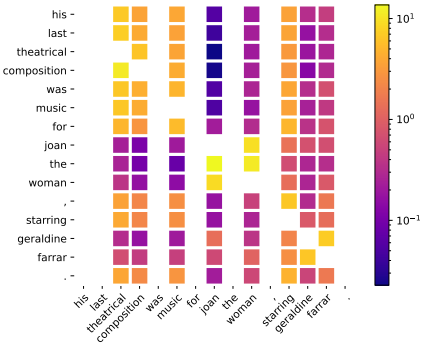
Figure 15: Examples of templates created for sentences taken from the TDMSci dataset. All templates shown are created in a low-resource setting (500 training examples). Words underlined are identified *keywords*.



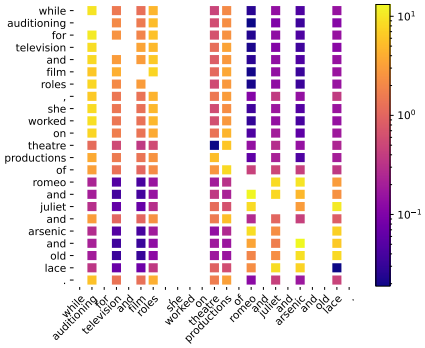
(a) Sentence: bamboo, like **[wood]**$_{\text{PROD}}$ is a natural **[composite material]**$_{\text{PROD}}$ with a high strength to weight ratio useful for structures.



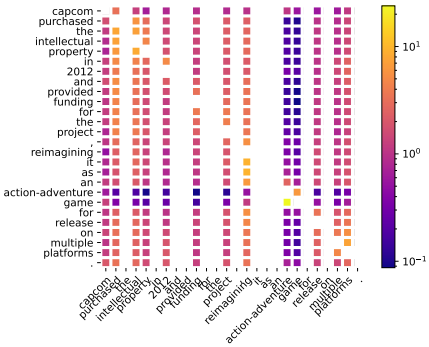
(b) Sentence: **[michael sanchez]**$_{\text{PER}}$, who was member of the band **[deep forest]**$_{\text{GRP}}$, co wrote the lyrics.



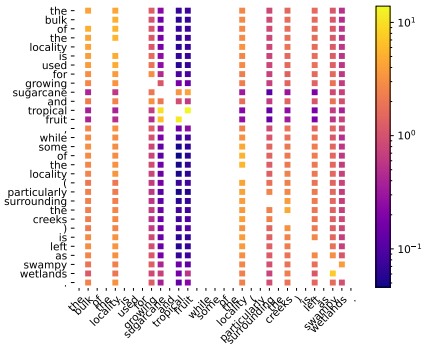
(c) Sentence: his last theatrical composition was music for **[joan the woman]**$_{\text{CW}}$ starring **[geraldine farrar]**$_{\text{PER}}$.



(d) Sentence: while auditioning for television and film roles, she worked on **[theatre]**$_{\text{GRP}}$ productions of **[romeo and juliet]**$_{\text{CW}}$ and **[arsenic and old lace]**$_{\text{CW}}$.



(e) Sentence: **[capcom]**$_{\text{CORP}}$ purchased the intellectual property in 2012 and provided funding for the project again,reimagining it as an **[action-adventure game]**$_{\text{CW}}$ for release on multiple platforms.



(f) Sentence: the bulk of the locality is used for growing **[sugarcane]**$_{\text{PROD}}$ and tropical fruit, while some of the locality,
particularly surrounding the creeks is left as swampy wetlands.

Figure 16: Attention maps for different sentences from the MULTICONER dataset. All the sentences are picked from a low-resource setting (1000 training examples).

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 12: MultiCoNER dataset statistics for the different languages for the train/dev/test splits. The bottom three rows show the total number of sentences for each language.

2022) (CC BY 4.0 licensed), CoNLL 2003 [3] (Tjong Kim Sang and De Meulder, 2003) (Apache License 2.0), BC2GM [4] (Smith et al., 2008) (MIT License), NCBI Disease [5] (Doğan et al., 2014) (Apache License 2.0) and TDMSci [6] (Hou et al., 2021) (Apache License 2.0). All the datasets are available to use for research purposes, and for our work, we use all these datasets intended for their original purpose, i.e., NER. MultiCoNER has data in 11 languages, including code-mixed and multilingual subsets. We experiment with 10 monolingual subsets discussed in Section 4.1 with appropriate reason for not experimenting on Farsi in our Limitations Section. According to the original papers of all 5 datasets used in the research, none of them contains any information that names or uniquely identifies individual people or offensive content.

Data statistics (train/test/dev splits): Detailed dataset statistics for MultiCoNER, CoNLL 2003, BC2GM, NCBI Disease and TDMSci can be found in Table 12 (language codes in Table 13), 14, 16, 17 and 15 respectively.

Implementation Software and Packages: We implement all our models in PyTorch [7] and use the HuggingFace [8] implementations of mBART50 and XLM-RoBERTA (base and large). We use the FLAIR toolkit (Akbik et al., 2019) to fine-tune all our NER models.

Potential Risks: Conditional Language Models used for Natural Language Generation often tend to *hallucinate* (Ji et al., 2022) and potentially generate nonsensical, unfaithful or harmful sentences to the provided source input that it is conditioned on.

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

Table 13: The languages included in MULTICONER, along with their 2-letter codes.

English data	Articles	Sentences	Tokens
Training set	946	14,987	203,621
Development set	216	3,466	51,362
Test set	231	3,684	46,435

Table 14: CoNLL Dataset Stats

	Train	Test
# Sentences	1500	500
# Task	1219	396
# Dataset	420	192
# Metric	536	174

Table 15: TDMSci dataset statistics for the train/test splits.

	Train	**Dev**	**Test**
# Sentences	15197	3061	6325

Table 16: BC2GM Dataset Train/Dev/Test Split

[3] https://huggingface.co/datasets/conll2003
[4] https://github.com/spyysalo/bc2gm-corpus
[5] https://huggingface.co/datasets/ncbidisease
[6] https://github.com/IBM/science-result-extractor
[7] https://pytorch.org/
[8] https://huggingface.co/

Corpus characteristics	Training set	Development set	Test set	Whole corpus
PubMed citations	593	100	100	793
Total disease mentions	5145	787	960	6892
Unique disease mentions	1710	368	427	2136
Unique concept ID	670	176	203	790

Table 17: NCBI disease dataset statistics for the train/dev/test splits.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations Section (After conclusion and before citations).

☑ A2. Did you discuss any potential risks of your work?
Section E.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1 and Abstract.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1.

☑ B1. Did you cite the creators of artifacts you used?
Citations.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section E.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section E.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section E.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section E.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section E.

C ☑ Did you run computational experiments?

Section E.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section E.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.2. Section A.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.1.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section E.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.