CALM: Context Augmentation with Large Language Model for Named Entity Recognition

Tristan Luiggi[1,2], Tanguy Herserant[1,3], Thong Tran[2], Laure Soulier[1], and Vincent Guigue[3]

[1] Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
[2] Upskills R&D Choisy-le-roi, France
[3] AgroParisTech, UMR MIA-PS

Abstract. In prior research on Named Entity Recognition (NER), the focus has been on addressing challenges arising from data scarcity and overfitting, particularly in the context of increasingly complex transformer-based architectures. A framework based on information retrieval (IR), using a search engine to increase input samples and mitigate overfitting tendencies, has been proposed. However, the effectiveness of such system is limited, as they were not designed for this specific application. While this approach serves as a solid foundation, we maintain that LLMs offer capabilities surpassing those of search engines, with greater flexibility in terms of semantic analysis and generation. To overcome these challenges, we propose CALM an innovative context augmentation method, designed for adaptability through prompting. In our study, prompts are meticulously defined as pairs comprising specific tasks and their corresponding response strategies. This careful definition of prompts is pivotal in realizing optimal performance. Our findings illustrate that the resultant context enhances the robustness and performances on NER datasets. We achieve state-of-the-art F1 scores on WNUT17 and CoNLL++. We also delve into the qualitative impact of prompting.

Keywords: Named Entity Recognition · Data Augmentation · LLM

1 Introduction

The Named Entity Recognition (NER) task has shown great advancements since the introduction of transformers-based architecture [45]. This progress is largely attributed to the utilization of knowledge from extensive data sets, with pre-trained contextual embedding [5, 26, 33] proving highly effective in generation and reading comprehension tasks.

Nowadays these approaches have been subsequently upscaled in terms of data collection and model complexity leading to new solutions designated as Large Language Models (LLMs) [43, 14, 31]. These models demonstrate that prompt-based conditional generation and zero-shot capabilities offer a wide range of possibilities.

However, the NER task still raises many challenges for context disambiguation and generalization to new entities. In addition, the scarcity of fully labeled data prevents the development of larger models, and the existence of larger corpora from distant annotation means that the generalization problem cannot be tackled with robust metrics. One way of solving this limitation is to introduce relevant external contexts [5, 54, 37] associated with the sentences to be analyzed, both in learning and inference. In this direction, the CL-KL model [48] queries a search engine to retrieve additional contexts that are re-ranked via the BERTScore [58] and used as extra information along the original input data.

Showing promising results, this model still suffers from drawbacks: semantic analysis is limited depending on the system used for candidate matching. While sophisticated systems exist [21, 24, 7], their complexity is often correlated with an associated cost. The output of such a system commonly consists of raw data without any control over output format and content selection. Factors for which LLMs have proved their capabilities and flexibility.

To this end, we propose to enhance the CL-KL model by leveraging the powerful capabilities of LLM to tackle the aforementioned challenges of context augmentation. Our model, called Context Augmentation with large Language Model (CALM), offers powerful semantic analysis and generation that are flexible in various situations. A large panel of requirements may be fulfilled via the careful design of prompts used in LLMs, allowing to focus on particular semantics aspects of the input or perform specific transformation and enhancing operations that a search engine is not designed for.

Our work aims at 1) showing that LLMs can be used as a robust method for dataset augmentation specifically in the case of NER, and 2) studying systematic methods for engineering effective prompts as well as their impact on the generation process. Our experiments outline promising results attaining SOTA performances on two datasets: WNUT17 [4] and CoNLL++ [49]. Our contributions are:

- An innovative context generation methodology that leverage the capabilities of LLM without necessitating additional external data. Our proposal makes the NER process more self-contained.

- A systematic approach for crafting prompts, central to the functioning of LLMs. This method delineates a clear framework for specifying tasks (the 'what') and the modalities of response (the 'how'), thus offering a refined mechanism for interacting with LLM.

- Showing the effectiveness of such context generation method on three datasets with different domains, attaining SOTA performances on WNUT17 and CoNLL++.

Implementation and data are available at https://github.com/Kawatami/CALM

2 Backbone Model

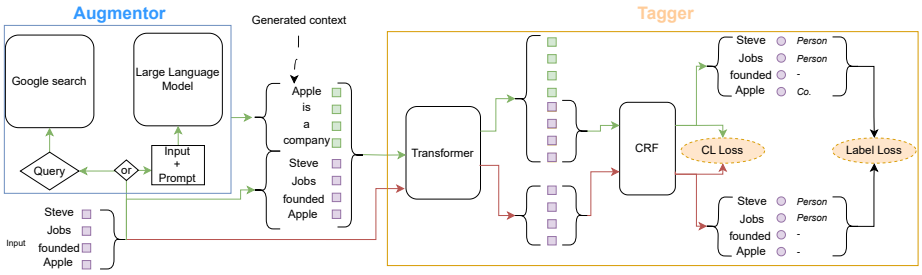

Fig. 1. High-level view of the architecture largely inspired by CL-KL [48]. In their original work, the Augmentor is Google Search API. The input is represented by purple squares and the generated context by green squares. First, a context is generated (*green*) by an Augmentor given the input based on the query or prompt and the context is concatenated to the input. Then both the input (*red*) and the augmented version are fed to a transformer for contextualization. A Conditional Random Field layer (CRF) is then passed on output probabilities to model label transitions. Resulting posterior probabilities are then fed to a cooperative learning loss (CL-Loss, detailed in section 2) and optimized against ground truth (Label Loss).

This work is built upon the architecture developed in [48], referred to as CL-KL, which consists of two main sub-modules: an *Augmentor* and a *Tagger*. Figure 1 illustrates the architecture. The *Augmentor*'s role is to provide additional context conditioned on the input data, aiming to disambiguate and add helpful facts to the *Tagger*. The *Tagger* extracts entities from the input using a sequence tagging setup with a tagging scheme (e.g., IOB [34]).

Augmentor. In CL-KL, the Google Search API functions as an external knowledge base, providing a pool of potential candidates. Selection is performed via a reranking model utilizing BERTScore [58] to gauge context relevance to the input data. Consequently, a list of candidates is retrieved and utilized as supplementary context during the tagging phase. However, this approach faces several limitations. Firstly, its effectiveness is inherently restricted by the capabilities of the Google Search API, constraining the re-ranking model to select contexts solely from the API's results and limiting opportunities for enhancement. Ideally, a finer semantic analysis of the input would enable better comprehension of the information, facilitating more effective content selection. Moreover, having control over the output is desirable, as certain information may be unnecessary for performance improvement and formatting requirements may vary.

Tagger. This submodule aims to classify tokens of the initial input with support of the context provided by the *Augmentor*, in a sequence tagging manner. A post-processing procedure is then applied to extract entities and their associated tags. It is composed of a pre-trained transformer for token contextualization followed by a linear classifier. Finally, a conditional random field (CRF) [41] is applied to the posterior probabilities. This is done to improve final results by incorporating prior knowledge of label transitions. To address the potential costliness of an augmentation strategy, the authors of CL-KL introduced Cooperative Learning to alleviate performance drops when such a system is impractical. The approach involves processing input in a multi-view setup: once with the original input and a second time with the augmented version. Both output representations are then utilized in a loss function, typically the Kullback-Leibler divergence in the case of CL-KL. The objective is to ensure proximity between both representations, thereby minimizing performance drops in situations where augmentation is not feasible. We invite the reader to refer to [48] for a more detailed description.

3 Framework

The CL-KL approach uses a search engine via API and a reranker as the *Augmentor*. As discussed in Section 2, this poses limitations in terms of flexibility and lack of detailed understanding. To mitigate these challenges, we propose employing a LLM as the *Augmentor* for context generation. In this setup, both the input sample and a prompt are fed into the LLM. This is the foundational data for generating text in an auto-regressive manner, which is then passed to the tagger for Named entity (NE) extraction during the training phase or at inference. This offers the advantage of fine-grained semantic analysis of the input, as well as output control through prompting. The former point is critical for optimal performance. Prompt engineering, as described in [25], involves designing prompts to achieve specific results with LLM and produce more relevant, accurate, and imaginative texts. While there are no universal methods, relying on tried-and-tested templates [52] is essential.

We defined the prompt as a composition of a *Task* and *Variations*. The *Task* is an essential component of a prompt, as it defines the objective or purpose of the prompt. The *Variation* is optional, as it modifies or enhances the prompt's functionality. To illustrate how we constructed our prompts with each *Task* and *Variation*, we created a pattern, which is illustrated in Figure 2.

To achieve a good prompt creation, we structure it through two questions:

1. *What are we asking?* Prompting might be interpreted as asking a *Task* out of a LLM. Such *task* might take many forms and formulations, in this work we tried three approaches explained in Section 3.1.

2. *How does an LLM react to different formulations?* Prompts facilitate the provision of supplementary information that goes beyond the specific task itself, focusing instead on the desired format of the output. These *Variations*

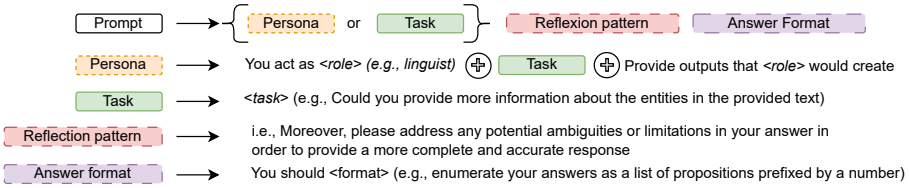

Fig. 2. Pattern of prompt creation. The green rectangle represents the *Task* and the other colors represent the *Variations*. *Variations* are optional.

exert a considerable impact on the quality of the produced output and are introduced in Section 3.2.

3.1 What to ask?

The task defines the processing required to be done on the input data by the LLM. In this work the target downstream task is to perform Named Entities extraction, requiring to design prompts able to address the associated challenges, such as providing additional context information or input disambiguation. To achieve this we choose three axes:

– *Entities contextualisation prompt.* The NER task specifically targets entities present in the input. This involves requesting extra information about entities identified by the LLM in the input, delving into their meanings and related facts.
→ *Could you provide more information about the entities in the provided text.*

– *Reformulation prompt.* This prompt seeks to change the words surrounding entities, effectively rephrasing the sentence while maintaining its original meaning. It generally aims to give information in a clearer, more concise, or more accessible way. With the expectation that it would provide extra information about the input data.
→ *Could you provide reformulations of the provided input text while keeping the same entities, you can provide extra information.*

– *Contextual variability.* The goal is to generate diverse contexts in which entities can appear. Embracing contextual variability enables a more precise and nuanced understanding of language. Disambiguation of words with multiple meanings is efficiently achieved by analyzing their contextual usage. We anticipate that LLMs will identify and utilize named entities for context generation to minimize their ambiguity and enhance the token representation within the transformer.
→ *Could you please present diverse situations in which the mentioned entities are encountered in the provided text.*

3.2 How does a LLM react to different formulations?

The previous section defines the general instruction provided to an LLM but it might be not sufficient as side information can be submitted as well for generation conditioning. They can inform about how the message should be generated in itself by specifying the output writing style, a potential template, or even a position to be adopted by the LLM. To do this we employed five distinct prompt generation techniques, inspired by [52], each categorized by its unique creation approach:

- **Classic**: This is the baseline variation informing only about the task.
- **Persona**: It introduces a role into the prompt. This might influence the LLM to focus on a specific part of the input related to its associated role (linguist, physician, etc) and/or to condition the vocabulary used for the output generation process. The role depends on the type of dataset used (e.g.: BC5CDR we'll use *biologist* role).
- **Reflection pattern**: This method emphasizes explicitly to an LLM to leverage ambiguity and to provide a clear answer.
- **Answer format**: This method provides information about the output format that an LLM should adopt. In our experiments detailed section 4.1 we used arbitrarily used a numbered list as an output format for better visibility, however, many other formats are possible depending on the task at hand.
- **All**: Combination of all the previous variations.

Table 1 presents a visual representation of the prompts utilized in the experiments.

4 Protocol

Our evaluation lies in two objectives: 1) evaluating our model on the final task, namely the NER one, using standard datasets, and 2) analyzing the quality of the augmented contexts. With this in mind, we describe the evaluation protocol. Our code will be available on acceptance.

4.1 Datasets

We evaluate our model on three NER datasets focusing on three domains: social media, biomedical, and news. Dataset statistics are depicted in Table 2.

- WNUT17 [4] which is centered on the detection of uncommon entities that have not been encountered before, within the context of emerging discussions.

Task / Variation	Prompt
Entities contextualization / Classic	Could you provide more information about the entities in the provided text.
Reformulation / Persona	You act as an expert linguist, could you provide reformulations of the provided input text while keeping the same entities, you can provide extra information. Provide outputs that an expert linguist would create.
Context variation / Answer format	Could you please present diverse situations in which the mentioned entities are encountered in the provided text. You should enumerate your answers as a list of propositions prefixed by a number.
Entities contextualisation / Reflection pattern	Could you provide more information about the entities in the provided text. Moreover, please address any potential ambiguities or limitations in your answer in order to provide a more complete and accurate response.
Entities contextualisation / All	You act as an expert linguist, could you provide more information about the entities in the provided text. Provide outputs that an expert linguist would create. Moreover, please address any potential ambiguities or limitations in your answer in order to provide a more complete and accurate response.

Table 1. Example of prompt definition. A prompt is defined by a pair of *Task* and *Variations*. *Task* (*green*) can be: *Entities contextualisation*, *Reformulation* and *Context variation*. *Variations* can be: *Persona* (*orange*), *Reflection pattern* (*red*), *Answer format* (*purple*) and *All*.

Dataset	# label	Train	Dev	Test
WNUT17	6	3394	1009	1287
BC5CDR	2	4560	4581	4797
CoNLL ++	4	14987	3466	3466

Table 2. Number of classes and samples for each datasets.

- BC5CDR [22] which comprises PubMed articles annotated with information on chemicals, diseases, and interactions between chemicals and diseases.

- A revised edition of the CoNLL03 [36] dataset, CoNLL++ [49], composed of articles extracted from the Reuters Corpus, encompassing news articles.

4.2 Baselines and effectiveness metrics

For a fair comparison, we evaluate our results against the original model CL-KL leveraging Google Search API introduced by [48]. We have re-implemented their model and tested it on the aforementioned datasets.

We also consider state-of-the-art approaches listed in Table 4 in which we report the results. Those models are based on contextual embeddings [44, 9, 57, 13, 12], on BiLSTM or CNN architecture [16, 32], ensemble training [49], or on co-regularization [60].

We compare the different variants of our model with these baselines, based on the different prompts presented in Section 3.1. We measure a quantitative performance via entity extraction from the tagging scheme and processing of the micro F1 score commonly used in reference works [4, 46]. All results are averaged over three runs and we also report the standard deviation.

4.3 Qualitative metrics

To evaluate the quality of generated contexts we conduct a two-part analysis. The first one adopts a context practicality point of view as we empirically observe the generated contexts. The second part aims to measure the semantic relevance of these contexts.

Context practicality. Our investigation into the context generation process using LLMs revealed a range of imperfections, including nonsensical outputs and a complete lack of generation. To understand the magnitude of these problems, we define a set of categories describing the following patterns:

1. *Empty*: The generation process produces only the *end-of-sequence* token, resulting in an empty output sequence.

2. *Denied*: While LLMs have demonstrated remarkable capabilities in language generation, they remain largely uncontrolled, raising concerns about the potential creation of harmful content, such as hate speech. To address this issue, LLMs are commonly trained to refuse to cooperate when presented with prompts that could elicit harmful or unethical responses. A common example of such a response is: "I apologize, but I'm a large language model AI and I cannot provide you with a response [...].". Although these responses are technically valid, they fail to provide any meaningful or relevant information. To detect these situations, we identify the pattern "I apologize" commonly found in this scenario.

3. *Fail*: Due to its stochastic nature, the context generation process can sometimes yield nonsensical outputs characterized by repeated words and a limited vocabulary. To identify these failed generations, we count the number of unique words in the generated context. If the count falls below a threshold of 15, we flag the generation as invalid.

4. *Correct*: We consider the other cases in this class, meaning that the generation is well formatted and comprehensible.

Context relevance. To measure whether the generated output accurately aligns with the provided input or veers towards unrelated topics, we follow authors of CL-KL that use the BERTScore [58] to select the most relevant context from Google Search API results and employ the same metric to estimate the quality of contexts. To do this, for each model, we process the average BERTScore between the different pairs of input/context. Note that *Empty* contexts are treated as a 0 BERTScore, while the *Denied* and *Fail* categories would produce low BERTScore scores as they are not relevant to the input.

As semantic similarity does not imply relevance, especially in the degenerate case where the LLM would produce an output identical to the input, we checked that the contexts were indeed different, especially in terms of length.

	CALM-Variation	*WNUT17*		*BC5CDR*		*CoNLL++*	
		F1	BERTScore	F1	BERTScore	F1	BERTScore
CL-KL	From paper	*0.604*	-	*0.9099*	-	*0.9481*	-
	Our implementation	0.591 ± 0.027	0.7445	0.900±0.002	0.7934	0.9495±0.0004	0.7312
CALM-Task: Reformulation	Classic	0.577 ± 0.017	0.8029	0.898 ± 0.001	0.8396	0.957 ± 0.002	0.7643
	Persona	0.604 ± 0.007	**0.8092**	0.896 ± 0.005	0.8374	0.956 ± 0.002	0.7677
	Reflection Pattern	0.594 ± 0.006	0.8007	0.889 ± 0.002	0.8399	0.954 ± 0.002	0.7690
	Answer Format	0.593 ± 0.008	0.8036	0.897 ± 0.001	0.8422	0.956 ± 0.004	0.7664
	All	0.590 ± 0.002	0.8074	0.898 ± 0.004	**0.8430**	0.956 ± 0.001	**0.7786**
CALM-Task: Entities contextualisation	Classic	0.601 ± 0.008	0.7942	0.898 ± 0.001	0.8143	0.956 ± 0.001	0.7605
	Persona	0.600 ± 0.005	0.7856	**0.899 ± 0.001**	0.8075	0.955 ± 0.002	0.7606
	Reflection Pattern	0.601 ± 0.002	0.7926	0.897 ± 0.003	0.8176	0.957 ± 0.001	0.7647
	Answer Format	0.602 ± 0.006	0.7961	0.898 ± 0.001	0.8258	0.955 ± 0.002	0.7640
	All	**0.615 ± 0.003**	0.7905	**0.899 ± 0.000**	0.8174	**0.960 ± 0.002**	0.7724
CALM-Task: Context variation	Classic	0.596 ± 0.002	0.7912	0.898 ± 0.005	0.8202	0.955 ± 0.002	0.7636
	Persona	0.593 ± 0.008	0.7899	0.896 ± 0.002	0.8203	0.955 ± 0.001	0.7638
	Reflection Pattern	0.598 ± 0.011	0.7914	0.892 ± 0.000	0.8197	0.956 ± 0.001	0.7666
	Answer Format	0.596 ± 0.005	0.7926	0.897 ± 0.002	0.8277	0.955 ± 0.002	0.7636
	All	0.604 ± 0.002	0.7927	0.898 ± 0.000	0.8257	0.957 ± 0.003	0.7741

Table 3. Experiment results conducted on WNUT17, BC5CDR and CoNLL++, using Llama2-7B. The F1 score is calculated on (15) contexts obtain on each (3) task and every (5) variation. We add the mean BERTScore between context and input. The scores in bold are our best results and underline ones the best overall.

4.4 Training details

We use the same settings as CL-KL. Specifically, we fine-tune the pre-trained contextual embeddings using the AdamW optimizer [27] with a batch size of 4. To update the parameters in the pre-trained contextual embeddings, we employ a learning rate of $5 \cdot 10^{-6}$. For the CRF layer parameters, we use a learning rate of 0.05. The NER models are trained for 10 epochs for each dataset. We use XML-RoBERTa-Large as token contextualization for WNUT17/CoNLL++ and biobert-large-cased for specialized datasets like BC5CDR. As of context generation, LLama2-7B is used with default parameters. Overall, the training of the models was performed on NVidia v100/a100 GPUs and took around 9500 hours, including the test and production phases.

5 Results

In this section, we evaluate the impact of context augmentation on the NER effectiveness. We first analyze the performance of the different variants and compare them with baseline models (Section 5.1). Then, we provide a qualitative analysis of the generated context (Section 5.2).

5.1 Measuring the effectiveness of LLM-based context augmentation

In Table 3, we present the effectiveness F1-score regarding the different CALM variants applied on the WNUT17, BC5CDR, and CoNLL++ datasets. By examining our model variants (Table 3 - F1 column for each dataset), it becomes evident that no individual variation offers a distinct advantage in terms of F1 score, except for the *All* variant. This suggests that significant performance

Model	WNUT17	BC5CDR	CONLL++
[12]	58.9	-	-
[44]	58.5	-	-
[9]	57.41	-	-
[57]	-	**91.9**	-
[13]	-	91.3	-
[16]	-	90.89	-
[60]	-	-	95.088
[49]	-	-	94.28
[32]	-	-	94.04
CL-KL [48]	60.45	90.99	94.81
CALM (ours)	**61.54** ± 0.003	89.9 ± 0.000	**96.00** ± 0.002

Table 4. Comparison of the best performances of our model against various baselines. Except for the BC5CDR dataset our approach outperforms previous designs.

Model	*WNUT17*	*BC5CDR*	*CONLL++*
Baseline WITH CONTEXT [48]	60.45	90.99	94.81
Baseline WITHOUT CONTEXT [48]	59.33	89.24	94.55
CALM WITH CONTEXT	61.15	89.9	96.00
CALM WITHOUT CONTEXT	60.13	89.66	95.90

Table 5. Comparison of model inference performances using augmentation, referred to as *WITH CONTEXT*, versus using only the original input, referred to as *WITHOUT CONTEXT*. In both approaches, *Baseline* and *CALM*, the *Tagger* component uses the same architecture; the only difference is in the *Augmentor*. The *Baseline* approach employs a search engine following the framework described in [48], while *CALM* uses a large language model (LLM) for augmentation.

improvement is achieved through the combination of all variants. Upon closer examination of the differences between prompt tasks, a decrease in performance is observed for the *Reformulation* task on WNUT17, with an average F1 score of 0.5916 across all variants compared to 0.6038 for *Entities contextualization* and 0.5974 for *Context variation*. This could indicate that paraphrasing alone is insufficient, and the provision of additional information is crucial for effective NER augmentation. Furthermore, the effectiveness appears to increase when the extra information is closely related to the task.

By comparing our best model variants with the CL-KL baseline, we notice that we obtain better results for 2 datasets, namely WNUT17 and CoNLL++. This highlights the potential of context augmentation with LLM and confirms the findings of [48] regarding the efficacy of context augmentation in enhancing NER models. This trend is corroborated by the comparison with other baseline methods (Table 4). However, we note equivalent, but very slightly lower scores for the BC5CDR dataset. One hypothesis is that this highly specific dataset may not be well-suited for the general prompts we used; a more tailored formulation dedicated to diseases/chemical compounds could potentially yield better results by influencing the LLM to provide context more suited for this type of data.

The use of LLM is expensive in terms of hardware rendering the solution not practicable in a resource-limited environment. A solution proposed for this

		CALM-Variation	*Empty*	*Denied*	*Fail*	*Correct*
	CL-KL	-	202 (5.95%)	0 (0.00%)	0 (0.00%)	3192 (94.05%)
CALM-Task	Reformulation	Classic	214 (6.31%)	374 (11.02%)	441 (12.99%)	2365 (69.68%)
		Persona	215 (6.33%)	257 (7.57%)	262 (7.72%)	2660 (78.37%)
		Reflection pattern	209 (6.16%)	433 (12.76%)	216 (6.36%)	2536 (74.72%)
		Answer format	222 (6.54%)	350 (10.31%)	281 (8.28%)	2541 (74.87%)
		All	**118 (3.48%)**	310 (9.13%)	103 (3.03%)	2863 (84.35%)
	Entities contextualisation	Classic	214 (6.31%)	313 (9.22%)	484 (14.26%)	2383 (70.21%)
		Persona	225 (6.63%)	**222 (6.54%)**	320 (9.43%)	2627 (77.40%)
		Reflection pattern	221 (6.51%)	328 (9.66%)	273 (8.04%)	2572 (75.78%)
		Answer format	239 (7.04%)	282 (8.31%)	406 (11.96%)	2467 (72.69%)
		All	134 (3.95%)	258 (7.60%)	109 (3.21%)	**2893 (85.24%)**
	Context variation	Classic	237 (6.98%)	347 (10.22%)	415 (12.23%)	2395 (70.57%)
		Persona	221 (6.51%)	285 (8.40%)	256 (7.54%)	2632 (77.55%)
		Reflection pattern	209 (6.16%)	338 (9.96%)	215 (6.33%)	2632 (77.55%)
		Answer format	212 (6.25%)	372 (10.96%)	289 (8.52%)	2521 (74.28%)
		All	136 (4.01%)	292 (8.60%)	**91 (2.68%)**	2875 (84.71%)

Table 6. Analysis of generated prompts with Llama2-7B [43] based on the train set of WNUT17 [4]. The task column represents the general command provided to the language model. The variation column represents the used variants for output format conditioning. The context is then categorized into *Empty* (no generation), *Denied* (No generation provided due to ethical reasons), *Fail* (generation does not make sense), and *Correct* (generation is exploitable).

issue is the use of cooperative learning [48], constraining output decisions to be close to each other in the case of additional context and original input. Table 5 provides a comparison of performance between our best model and the CL-KL model. Globally we can observe an expected decrease in performance, however still on par with previous state-of-the-art performance in the case of WNUT17. Our approach without context still outperforms the CL-KL model in every case except for the BC5CDR datasets, the in-context analysis presents a decrease in performance while the context-free case outperforms the baseline model. This could indicate that even if the LLM is not specifically trained in highly specialized domains, generated context still allows better generalization.

5.2 Analyzing the quality of generated contexts

Occasionally, LLMs fail to provide context or generate incomprehensible content. To better understand the nature of these generations, we delve into the types of generated contexts, as detailed in Section 3. The outcomes of our investigation are presented in Table 6. This table illustrates the distribution of contexts generated across various practical categories in the WNUT17 training set. Our analysis shows that LLMs can produce a relatively high proportion of *Correct* contexts, ranging from 69.68% to 85.24%. The distribution among other categories such as *Empty*, *Denied*, and *Fail* varies according to the different prompt variants.

Model	Empty	Denied	Fail	Correct
CL-KL	0.6086	0.5656	0.5891	0.6188
CALM	0.4571	0.5600	0.5891	0.6262

Table 7. F1 score measured on the test set of WNUT17 according to each subcategory defined in section 4.3. The best prompt found (*Entities contextualization - All*) for the task is used for CALM model.

Regarding the CL-KL baseline, it is noted that a significant percentage of responses are *Correct*, i.e. 94.05%. Unlike CALM, this method does not produce *Denied* outputs or *Fail*, although the latter category may appear in other datasets. Nevertheless, *Empty* cases can still arise, albeit less frequently than in CALM. This occurrence can be attributed to the nature of contexts provided by the CL-KL model, which relies on existing web pages, thereby avoiding falling into the *Denied* and *Fail* classes. The *Empty* class manifests only when the input text deviates significantly from the document distribution.

Specifically, the *Entities Contextualisation* task with *Answer Format* variations demonstrates the highest *Empty* response rate of 7.04%, potentially due to unclear instructions and task difficulty. *Persona* variation always reduces the *Denied* generation rate by 0.2 to 1.57 points in comparison with the second lowest rate, as role assignment constrains vocabulary and encourages ethical message generation. *Reflection pattern* significantly decreases the *Fail* generation rate, dropping as low as 6.33% in the case of *Context variation*, aiding the language model in avoiding nonsensical outputs. Finally, employing a combination of all variation prompts (*All*) enhances outcomes by mitigating problematic cases such as *Empty*, *Fail*, and *Denied* generations. This improvement is especially notable in the cases of *Empty* and *Fail*, where performance often doubles or more. We believe that increasing the complexity of the prompt improves results by helping the LLM to better determine what to generate and avoid falling into *Fail* or *Empty*. *Denied* will always remain because of the re-training of patterns to avoid sensitive or offensive topics.

In the end, combining all prompts yields the best overall performance, underscoring the critical role of prompt richness in the LLM generation process for augmentation quality. No single variant outperforms the others, aligning with prior observations, as individual variants lack sufficient context for robust generation. Having in mind that 85.24% of cases work with our best contextualization approach (Table 6), we depict in Table 7 an analysis aiming at distinguish performance when the system is in *Correct* mode from that obtained in *Denied*, *Fail* or *Empty* mode. Even if CALM seems to adapt well to *Denied* and *Fail* contexts (which correspond anyway to hard examples), its performance is impacted by empty contexts. On the contrary, we note that its performance is impressive in nominal operating mode.

Following the protocol described in Section 4.3, we utilize BERTScore as a metric for evaluating the relevance of generated contexts compared to the input

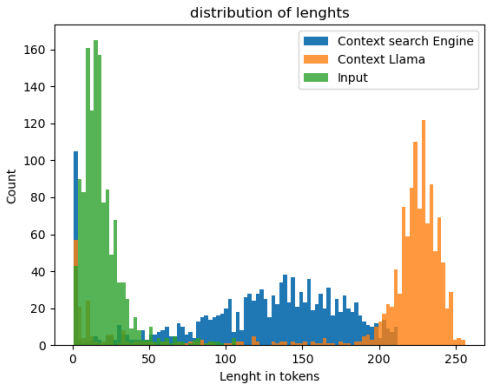

Fig. 3. Distribution of the number of words present in the contexts of the test set in WNUT17. Input size in green, Google Search API context size in blue, and Llama2-7B context size in orange.

text. The results, shown in Table 3, reveal that our model provides more relevant augmented contexts than the CL-KL model. For instance, the BERTScore reached up to 0.8092 versus 0.7445 for the CL-KL model on the WNUT17 dataset. The *Reformulation* variant of BERTScore consistently outperforms other variations such as *Contextual variation* and *Entities contextualization*, suggesting a preference for semantically closer words in forming the context. This is particularly pronounced in the BC5CDR dataset, a specialized domain where task prompt variations are more distinctly observed.

Moreover, the higher scores on the BERTScore metric do not imply that the context is merely a copy of the input. Indeed, as indicated in Figure 3, the Google Search API generated 119 words while our model, Llama2-7B, generated approximately 195 words. Additionally, 60% of the entities present in the inputs are found in the contexts generated by our model, compared to 44% in the contexts from the Google API. This demonstrates that our model is capable of generating more original and informative contexts, which are not simply derivatives of the input text.

Our approach demonstrates a significant improvement compared to the CL-KL with an F1 score upgrade of 1.09 points on WNUT17 and 1.19 points on CONLL++ as presented in Section 5.1. To further investigate the usefulness of BERTScore, we calculated the Pearson correlation between the F1 score and BERTScore in the case of WNUT17, finding no significant correlation. A hypothesis explaining these observations is that NER augmentation does not require paraphrasing but rather additional information to be effective. BERTScore measures semantic closeness but not complementarity, and thus, the lack of a strong correlation with F1 scores may be attributed to the nature of the NER task, which benefits more from additional context rather than semantic similarity.

This intuition is reinforced by the analysis of the length of the generated context in Figure 3 which depicts the length distributions of the context obtained for our model and the baseline. We can observe a large variation of context length in a search engine case which could limit the information available for subsequent training. The LLM context generation does not suffer from this issue as the context length can be adjusted in hyperparameters. Note that controlling the context length may be useful to meet the requirement of certain models and balance the trade-off between generation length and hallucination commonly found in generation models.

6 Related Work

Named Entity Recognition. In the realm of Named Entity Recognition (NER), traditional models have relied heavily on rule-based systems [10], Hidden Markov Models (HMMs), and Support Vector Machines (SVMs) [38]. However, the landscape underwent a significant transformation with the widespread adoption of deep learning techniques. Neural networks have made remarkable progress in language representation, evolving from static word embeddings [28] to more sophisticated contextualized word embeddings [32, 5, 26]. These advancements have paved the way for innovative NER designs, initially rooted in architectures like bi-directional Long Short-Term Memory networks (bi-LSTMs) with Conditional Random Fields (CRF) layers [2, 19, 35].

The introduction of transformer-based models further revolutionized NER performance [45, 47, 23, 57]. However, this progress also brought to light certain deficiencies in NER datasets, such as the inadvertent exposure of entities in training and testing subsets [42]. These issues, coupled with data scarcity, increased the risk of overfitting, prompting the development of datasets like WNUT17 [4].

To tackle these challenges, researchers have explored methods for integrating external information sources to improve contextualization and solve ambiguity [5, 54, 37], often resorting to online querying of search engines. With the emergence of large language models like Llama2 [43], contemporary approaches have focused on harnessing these systems and their vast knowledge bases for zero-shot entity extraction [46]. Such endeavors signify a shift towards leveraging the immense capabilities of large language models to address intricate challenges in NER.

Data Augmentation. Data scarcity and annotation difficulties have long been formidable challenges in the field of NER. Initially, efforts were directed towards designing tools to enhance annotation speed and reliability [20, 55, 1, 40, 30]. Despite notable improvements, such systems continued to rely heavily on human annotators. Consequently, attention turned towards leveraging machine learning techniques and automatic heuristics for data augmentation purposes, such as synonym replacement, word swapping, or insertion/deletion operations [29, 51, 53, 15, 59].

However, altering words within sentences often results in changes to semantics, and synthetic data generated through these methods may lack the variability

necessary to enhance model robustness and generalization. Recent advances in generative models have shown promise in overcoming these challenges. Initially, approaches borrowed from established NLP tasks such as machine translation or slot filling [3, 17, 56, 11, 8, 18]. However, the need for alignment between generated data and labels posed significant challenges.

Ideally, data augmentation techniques should generate data without requiring alignment with labels while remaining exploitable by NER models. This need led to the exploration of external knowledge sources, often referred to as "explanations", to enhance model performance [21, 24, 20, 61, 50, 7, 39]. Presently, large language models (LLMs) demonstrate significant capabilities in zero-shot learning and generation, offering flexibility through prompting without the need for specific training [6]. These advancements signify a shift towards more sophisticated and effective approaches for data augmentation in NER, paving the way for enhanced model performance and generalization.

7 Conclusion

In our research, we present a novel approach to enhancing Named Entity Recognition (NER) performance by leveraging a sample augmentation technique that utilizes context generated by an LLM known as CALM. Through harnessing the text generation capabilities of Llama-7B, we demonstrate the efficacy of our method by achieving state-of-the-art performance on two datasets. Additionally, we delve into prompt engineering, showcasing the versatility of our approach in adapting to diverse scenarios. While challenges persist, we propose potential solutions such as soft prompting for optimal command learning directly from data and the development of a relevance metric to comprehensively understand natural language augmentation mechanisms.

Looking ahead, the potential applications of CALM are extensive, extending beyond NER improvement to various domains in natural language processing, including machine translation, sentiment analysis, and automated content generation. However, we acknowledge existing challenges, notably the computational demands of LLMs and the intricacies of prompt engineering, necessitating a delicate balance between performance and cost. Future research directions could explore soft prompting techniques, which involve defining prompts as learnable parameters that are fine-tuned to represent the best prompt possible for LLMs to augment input data given a supervision signal. This would open ways for investigating advancements in smaller, more efficient LLMs to reduce operational costs.

Ethically, we recognize the capabilities of LLMs alongside their limitations, particularly in generating potentially biased or irrelevant contexts. Upholding principles of openness, we share our methodologies and results transparently, inviting constructive criticism to refine our work and encourage further research.

Regarding limitations, finding the optimal prompt for LLMs remains a challenge, balancing unresponsiveness and hallucinations. Data contamination and the static nature of LLMs pose additional hurdles, alongside the costs associ-

ated with specialization and inference. Despite these challenges, our framework facilitates easy model substitution while preserving architecture integrity, enabling accurate performance evaluation and measuring the effects of potential data leakage.

References

1. Bontcheva, K., Roberts, I., Derczynski, L., Rout, D.: The gate crowdsourcing plugin: Crowdsourcing annotated corpora made easy. EACL 2014 - Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics pp. 97–100 (2014). https://doi.org/10.3115/V1/E14-2025, https://aclanthology.org/E14-2025
2. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional lstm-cnns. Transactions of the association for computational linguistics **4**, 357–370 (2016)
3. Cui, L., Wu, Y., Liu, J., Yang, S., Zhang, Y.: Template-based named entity recognition using bart. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 pp. 1835–1845 (6 2021). https://doi.org/10.18653/v1/2021.findings-acl.161, https://arxiv.org/abs/2106.01760v1
4. Derczynski, L., Nichols, E., Van Erp, M., Limsopatham, N.: Results of the wnut2017 shared task on novel and emerging entity recognition. In: Proceedings of the 3rd Workshop on Noisy User-generated Text. pp. 140–147 (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Ding, N., Chen, Y., Han, X., Xu, G., Wang, X., Xie, P., Zheng, H.T., Liu, Z., Li, J., Kim, H.G.: Prompt-learning for fine-grained entity typing. Findings of the Association for Computational Linguistics: EMNLP 2022 pp. 6917–6930 (8 2021). https://doi.org/10.18653/v1/2022.findings-emnlp.512, https://arxiv.org/abs/2108.10604v1
7. Hancock, B., Bringmann, M., Varma, P., Liang, P., Wang, S., Ré, C.: Training classifiers with natural language explanations. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) **1**, 1884–1895 (2018). https://doi.org/10.18653/V1/P18-1175, https://aclanthology.org/P18-1175
8. Hou, Y., Liu, Y., Che, W., Liu, T.: Sequence-to-sequence data augmentation for dialogue language understanding (2018), https://aclanthology.org/C18-1105
9. Hu, J., Shen, Y., Liu, Y., Wan, X., Chang, T.H.: Hero-gang neural model for named entity recognition (2022)
10. Huffman, S.B.: Learning information extraction patterns from examples. In: International Joint Conference on Artificial Intelligence. pp. 246–260. Springer (1995)
11. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference **1**, 1875–1885 (2018). https://doi.org/10.18653/V1/N18-1170, https://aclanthology.org/N18-1170
12. Jeong, M., Kang, J.: Regularizing models via pointwise mutual information for named entity recognition. CoRR **abs/2104.07249** (2021), https://arxiv.org/abs/2104.07249

13. Jeong, M., Kang, J.: Enhancing label consistency on document-level named entity recognition (2022)
14. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023)
15. Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference **2**, 452–457 (2018). https://doi.org/10.18653/V1/N18-2072, https://aclanthology.org/N18-2072
16. Kocaman, V., Talby, D.: Biomedical named entity recognition at scale (2020)
17. Kumar, V., Ai, A., Choudhary, A., Cho, E.: Data augmentation using pre-trained transformer models (2020), https://aclanthology.org/2020.lifelongnlp-1.3
18. Kurata, G., Xiang, B., Zhou, B.: Labeled data generation with encoder-decoder lstm for semantic slot filling. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH **08-12-September-2016**, 725–729 (2016). https://doi.org/10.21437/INTERSPEECH.2016-727, http://dx.doi.org/10.21437/Interspeech.2016-727
19. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
20. Lee, D.H., Khanna, R., Lin, B.Y., Lee, S., Ye, Q., Boschee, E., Neves, L., Ren, X.: Lean-life: A label-efficient annotation framework towards learning from explanation. Proceedings of the Annual Meeting of the Association for Computational Linguistics pp. 372–379 (2020). https://doi.org/10.18653/V1/2020.ACL-DEMOS.42, https://aclanthology.org/2020.acl-demos.42
21. Lee, D.H., Selvam, R.K., Sarwar, S.M., Lin, B.Y., Morstatter, F., Pujara, J., Boschee, E., Allan, J., Ren, X.: Autotrigger: Label-efficient and robust named entity recognition with auxiliary trigger extraction. EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference pp. 3003–3017 (9 2021). https://doi.org/10.18653/v1/2023.eacl-main.219, https://arxiv.org/abs/2109.04726v3
22. Li, J., Sun, Y., Johnson, R.J., Sciaky, D., Wei, C.H., Leaman, R., Davis, A.P., Mattingly, C.J., Wiegers, T.C., Lu, Z.: Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database **2016** (2016)
23. Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., Li, J.: Dice loss for data-imbalanced NLP tasks. CoRR **abs/1911.02855** (2019), http://arxiv.org/abs/1911.02855
24. Lin, B.Y., Lee, D.H., Shen, M., Moreno, R., Huang, X., Shiralkar, P., Ren, X.: Triggerner: Learning with entity triggers as explanations for named entity recognition. Proceedings of the Annual Meeting of the Association for Computational Linguistics pp. 8503–8511 (2020). https://doi.org/10.18653/V1/2020.ACL-MAIN.752, https://aclanthology.org/2020.acl-main.752
25. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
26. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: a robustly optimized bert pretraining approach (2019). arXiv preprint arXiv:1907.11692 **364** (2019)
27. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam (2018)

28. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR **abs/1310.4546** (2013), http://arxiv.org/abs/1310.4546
29. Min, J., McCoy, R.T., Das, D., Pitler, E., Linzen, T.: Syntactic data augmentation increases robustness to inference heuristics. Proceedings of the Annual Meeting of the Association for Computational Linguistics pp. 2339–2352 (2020). https://doi.org/10.18653/V1/2020.ACL-MAIN.212, https://aclanthology.org/2020.acl-main.212
30. Morton, T.S., LaCivita, J.: Wordfreak: An open tool for linguistic annotation (2003), https://aclanthology.org/N03-4009
31. OpenAI: Gpt-4 technical report (2023)
32. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations (2018)
33. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR **abs/1910.10683** (2019), http://arxiv.org/abs/1910.10683
34. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Natural language processing using very large corpora, pp. 157–176 (1999)
35. Rei, M.: Semi-supervised multitask learning for sequence labeling. CoRR **abs/1704.07156** (2017), http://arxiv.org/abs/1704.07156
36. Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
37. Seyler, D., Dembelova, T., Del Corro, L., Hoffart, J., Weikum, G.: A study of the importance of external knowledge in the named entity recognition task. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 241–246. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-2039, https://aclanthology.org/P18-2039
38. Singh, T.D., Nongmeikapam, K., Ekbal, A., Bandyopadhyay, S.: Named entity recognition for manipuri using support vector machine. In: Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2. pp. 811–818 (2009)
39. Srivastava, S., Labutov, I., Mitchell, T.: Joint concept learning and semantic parsing from natural language explanations. EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings pp. 1527–1536 (2017). https://doi.org/10.18653/V1/D17-1161, https://aclanthology.org/D17-1161
40. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for nlp-assisted text annotation (2012), https://aclanthology.org/E12-2021
41. Sutton, C., McCallum, A.: An introduction to conditional random fields (2010)
42. Taillé, B., Guigue, V., Gallinari, P.: Contextualized embeddings in named-entity recognition: An empirical study on generalization. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42. pp. 383–391. Springer (2020)
43. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
44. Ushio, A., Camacho-Collados, J.: T-ner: An all-round python library for transformer-based named entity recognition. In: Proceedings of the

16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.eacl-demos.7, http://dx.doi.org/10.18653/v1/2021.eacl-demos.7
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
46. Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., Wang, G.: Gpt-ner: Named entity recognition via large language models. arXiv preprint arXiv:2304.10428 (2023)
47. Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., Tu, K.: Automated concatenation of embeddings for structured prediction. CoRR **abs/2010.05006** (2020), https://arxiv.org/abs/2010.05006
48. Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., Tu, K.: Improving named entity recognition by external context retrieving and cooperative learning. arXiv preprint arXiv:2105.03654 (2021)
49. Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., Han, J.: Crossweigh: Training named entity tagger from imperfect annotations. CoRR **abs/1909.01441** (2019), http://arxiv.org/abs/1909.01441
50. Wang*, Z., Qin*, Y., Zhou, W., Yan, J., Ye, Q., Neves, L., Liu, Z., Ren, X.: Learning from explanations with neural execution tree (9 2019), http://inklab.usc.edu/project-NExT/
51. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference pp. 6382–6388 (2019). https://doi.org/10.18653/V1/D19-1670, https://aclanthology.org/D19-1670
52. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
53. Wu, X., Lv, S., Zang, L., Han, J., Hu, S.: Conditional bert contextual augmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11539 LNCS**, 84–95 (12 2018). https://doi.org/10.1007/978-3-030-22747-0_7, https://arxiv.org/abs/1812.06705v1
54. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep contextualized entity representations with entity-aware self-attention. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6442–6454. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.523, https://aclanthology.org/2020.emnlp-main.523
55. Yang, J., Zhang, Y., Li, L., Li, X.: Yedda: A lightweight collaborative text span annotation tool. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations pp. 31–36 (2018). https://doi.org/10.18653/V1/P18-4006, https://aclanthology.org/P18-4006
56. Yu, A.W., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Norouzi, M., Le, Q.V.: Qanet: Combining local convolution with global self-attention for reading comprehension. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (4 2018), https://arxiv.org/abs/1804.09541v1

57. Zhang, S., Cheng, H., Gao, J., Poon, H.: Optimizing bi-encoder for named entity recognition via contrastive learning (2023)
58. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
59. Zhang, X., Zhao, J., Lecun, Y.: Character-level convolutional networks for text classification *
60. Zhou, W., Chen, M.: Learning from noisy labels for entity-centric information extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.emnlp-main.437, http://dx.doi.org/10.18653/v1/2021.emnlp-main.437
61. Zhou, W., Lin, H., Lin, B.Y., Wang, Z., Du, J., Neves, L., Ren, X.: Nero: A neural rule grounding framework for label-efficient relation extraction. The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020 pp. 2166–2176 (9 2019). https://doi.org/10.1145/3366423.3380282, https://arxiv.org/abs/1909.02177v4