Zero- and Few-Shots Knowledge Graph Triplet Extraction with Large Language Models

Andrea Papaluca[1], Daniel Krefl[2], Sergio J. Rodríguez Méndez[1], Artem Lensky[3,4],
Hanna Suominen[1,5,6]

[1]School of Computing, The Australian National University, Canberra, ACT, Australia, [2]Independent,

[3]School of Engineering and Technology, The University of New South Wales, ACT, Australia,

[4]School of Biomedical Engineering, The University of Sydney, NSW, Australia,

[5]School of Medicine and Psychology, The Australian National University, Canberra, ACT, Australia,

[6]Department of Computing, University of Turku, Turku, Finland

Correspondence: andrea.papaluca@anu.edu.au

Abstract

In this work, we tested the Triplet Extraction (TE) capabilities of a variety of Large Language Models (LLMs) of different sizes in the Zero- and Few-Shots settings. In detail, we proposed a pipeline that dynamically gathers contextual information from a Knowledge Base (KB), both in the form of context triplets and of (sentence, triplets) pairs as examples, and provides it to the LLM through a prompt. The additional context allowed the LLMs to be competitive with all the older fully trained baselines based on the Bidirectional Long Short-Term Memory (BiLSTM) Network architecture. We further conducted a detailed analysis of the quality of the gathered KB context, finding it to be strongly correlated with the final TE performance of the model. In contrast, the size of the model appeared to only logarithmically improve the TE capabilities of the LLMs. We release the code on GitHub [1] for reproducibility.

1 Introduction

The task of Triplet Extraction (TE) (Nayak et al., 2021) is of fundamental importance for Natural Language Processing (NLP). This is because the core meaning of a sentence is usually carried by a set of $(subject, predicate, object)$ triplets. Therefore, the capability to identify such triplets is a key ingredient for being able to understand the sentence.

Currently, the State-Of-The-Art (SOTA) for TE is achieved by models that approach the TE task in an *end-to-end* fashion (Zheng et al., 2017; Zeng et al., 2018; Fu et al., 2019; Zeng et al., 2019; Tang et al., 2022). That is, they are trained to perform all the TE sub-tasks, namely, Named Entity Recognition (NER (Yadav and Bethard, 2018)), Entity Linking (EL (Alam et al., 2022)), and Relation Extraction (RE (Detroja et al., 2023)), together. These

SOTA models follow the classic NLP paradigm, *i.e.*, they are trained by supervision on specific TE datasets. However, this dependence on labeled data restricts their generality and, therefore, limits the applicability of such models to the real world.

While several labeled datasets for the TE task are publicly available (Riedel et al., 2010; Gardent et al., 2017), these cover only part of the spectrum of possible entities and relations. This means that a supervised model trained on these public data will be restricted to the closed set of entities and relations seen during training, implying that it may lack generalization capabilities. Producing a tailored dataset for training a model for particular applications, is, however, in general expensive (Johnson et al., 2018).

For this reason, the recent language understanding and reasoning capabilities demonstrated by Large Language Models (LLMs), such as the Generative Pre-trained Transformer 4 (GPT-4) (OpenAI, 2023), LLM Meta AI (LLaMA) (Touvron et al., 2023), and Falcon (Penedo et al., 2023) to name a few, have led researchers (Chia et al., 2022; Kim et al., 2023; Wadhwa et al., 2023; Wei et al., 2023b; Zhu et al., 2023) to investigate whether they represent a viable option to overcome the limitations imposed by supervised models for TE. In detail, the new approach being that at inference time the LLMs are prompted to extract the triplets contained in a sentence, while being provided with only a few labeled examples (or no example at all in the Zero-Shot setting). This LLM approach largely limits the amount of data needed to perform the task, and, in particular, lifts the restriction of adhering to a predefined closed set of relations. However, the investigations so far indicated that the Zero and Few-Shots performance of the LLMs appears to be often underwhelming compared to the classic fully trained NLP models (Wadhwa et al., 2023; Wei et al., 2023b; Zhu et al., 2023).

In order to enhance the abilities of LLMs in the

---

[1]https://github.com/BrunoLiegiBastonLiegi/
KG-TE-with-LLMs

TE task, we propose in this work to aid them with the addition of a Knowledge Base (KB). We demonstrate that augmenting LLMs with KB information, *i.e.*, dynamically gathering contextual information from the KB, largely improves their TE capabilities, thereby making them more competitive with classic NLP baselines. In particular, we show that when the retrieved information is presented to the LLMs in the form of complete TE examples relevant to the input sentence, their performance gets closer to the fully trained SOTA models.

2 Related Work

Classical *end-to-end* fully supervised models currently hold the best performance in the TE task. Starting from the older baseline, Zheng et al. (2017), which also introduced the revised version of the WebNLG dataset for Natural Language Generation (NLG) that is commonly used, several other architectures based on the bidirectional Recurrent Neural Networks (RNNs) (Zeng et al., 2018; Fu et al., 2019; Zeng et al., 2019) have steadily improved the SOTA over the years. More recently, Transformer-based models achieved a big leap forward in performance, with the recent UniRel model being the current SOTA (Tang et al., 2022) in the datasets we consider. A further class of fully supervised models, such as Huguet Cabot and Navigli (2021) and Josifoski et al. (2022), treats the TE problem as a *sequence-to-sequence* generation task, which is more similar to the LLM approach adopted here, but still requires some training or finetuning.

With the advent of LLMs, Chia et al. (2022) and Kim et al. (2023) tested the use of such models for those TE cases where the availability of examples to train on is low. The first work proposed to use a LLM to generate training examples to finetune a *Relation Extractor* model to recognize relations for which labels were not available. The latter work, instead, suggested using relation templates of the form ⟨X⟩ *relation* ⟨Y⟩ and finetune a LLM to fill out ⟨X⟩ and ⟨Y⟩ with the entities appearing in the sentence. Wadhwa et al. (2023),Wei et al. (2023b), and Zhu et al. (2023) investigated the general TE task in both Zero- and Few-Shots settings. These studies proposed different approaches based on LLM prompting. The first work tested the Few-Shots performance of GPT-3 (Brown et al., 2020a) and Text-to-Text Transfer Transformer (T5) (Raffel et al., 2023) under the inclusion of manually-crafted and dataset-dependent contextual information in the prompt. The second work proposed to perform TE by sequentially prompting ChatGPT in two stages: asking to individuate the possible relation types first and then extracting the entities participating in each relation. The procedure demonstrated better results than a one-stage approach where the model is prompted to extract the triplet directly. Finally, the third work, evaluated GPT-3 (Brown et al., 2020b) and GPT-4 (OpenAI, 2023) on some standard benchmarks in the Zero- and One-Shot settings. However, classical fine-tuned models proved to be superior in the majority of the cases.

In our study, we similarly test the Zero- and Few-Shots capabilities of LLMs in two standard TE datasets that have not been covered by these previous works. In contrast to Wadhwa et al. (2023) that manually crafted static dataset-specific context to be fed to the LLM, we propose here to dynamically gather contextual information useful for extracting the triplets from a KB. This makes our approach more flexible and less data-dependent, as the KB does not require any manual operation and can be easily switched depending on the need. Also, in contrast to other works, we investigate a wide range of Language Models with varying sizes. This allows us to provide an in-depth analysis of the scaling of the performance, both, from the perspective of the model chosen, and the quality of the contextual KB information included in the prompt.

3 The Pipeline

In this section, we provide a detailed illustration of the pipeline used to test the TE capabilities of LLMs.

3.1 Task Formulation

Given a sentence composed of tokens $(t_1, t_2, \cdots, t_N)$, the TE task consists of identifying all the relations expressed in it and extracting them in the form of triplets (s, p, o). Here, $s = (t_i, \cdots, t_{i+n_s})$ and $o = (t_k, \cdots, t_{k+n_o})$ represent a subject and an object of length n_s and n_o tokens, and p is the predicate. Usually, the task is related to a specific KB, *i.e.*, a graph of the form $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, composed of entities $e \in V$ as vertices and relations $r \in \mathcal{E}$ as directed edges. Therefore, s and o of the sentence correspond to vertices $e_s, e_o \in \mathcal{V}$. The predicate p is mapped to a relation included in the closed set of possible edge types of the KB.
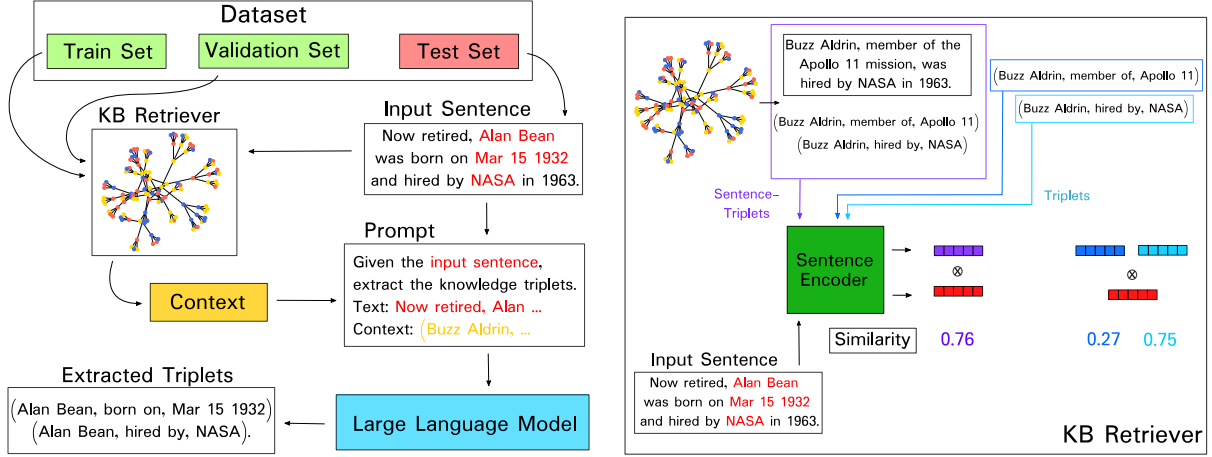
Figure 1: The TE pipeline. Left: illustration of the pipeline. A KB is constructed from the training and validation splits of a given dataset. For each test sentence, the relevant contextual information is retrieved from the KB and included in the prompt for a LLM-based TE. Right: summary of information retrieval from the KB. Either the sentence-triplets pairs or the single triplets alone are encoded by a sentence encoder and compared to the encoding of the input sentence by cosine similarity.

3.2 LLMs as Triplet Generators

In order to perform TE, we can prompt LLMs to generate, for a given input sentence, a sequence of tokens corresponding to the set of entity-relation triplets $\left\{(e_s^i, r_p^i, e_o^i)\right\}_{i=1}^n$. As demonstrated by Wadhwa et al. (2023), Wei et al. (2023b), and Zhu et al. (2023), LLMs are, in principle, able to extract the knowledge triplets contained in a text without a need for task-specific training, under a suitable choice of prompt. In general, successful LLM prompts follow a fixed schema that provides a detailed explanation of what the task consists of, a clear indication of the sentence to process, and some hints or examples for the desired result.

In this work, we tested the use of three different prompts: a simple baseline and two slight variations of it. However, preliminary testing in TE showed no significant difference in the F1 scores among them. Therefore, we opted for using only the base prompt reported in Figure 2 in the main experiments. The details of the prompts tested and their results can be found in Appendix A.3.

3.3 KB-aided Triplet Extraction

In order to support LLMs in the TE task, we propose the pipeline illustrated in Figure 1. The pipeline augments the LLM with external KB information. In detail, for each input sentence, relevant context information contained in the KB is retrieved and attached to the LLM prompt described above. The context-enriched prompt is then fed to the LLM for the knowledge triplet generation.

We prepare the information coming from the KB in two different forms: either as simple context triplets

$$T_c = \left\{(e_s^i, r_p^i, e_o^i)\right\}_{i=1}^{N_{KB}} \in \mathcal{G}, \qquad (1)$$

or as sentence-triplets pairs

$$E_c = \left\{(S_c^i, T_c^i)\right\}_{i=1}^{N_{KB}}. \qquad (2)$$

The latter provides factual examples of triplets to be extracted for specific sentences. Note that we indicate with N_{KB} the number of triplets, respectively,

Triplet Extraction Prompt

Some text is provided below. Extract up to {max_triplets} knowledge triplets in the form (subject, predicate, object) from the text.

Examples:

Text: Abilene, Texas is in the United States.
Triplets:
(abilene texas, country, united states)

Text: The United States includes the ethnic group of African Americans and is the birthplace of Abrahm A Ribicoff who is married to Casey Ribicoff.
Triplets:
(abrahm a. ribicoff, spouse, casey ribicoff)
(abrahm a. ribicoff, birth places, united states)
(united states, ethnic group, african americans)

Text: {text}
Triplets:

Figure 2: The base prompt we experimented with. At inference time the $\{text\}$ and $\{max\_triplets\}$ variables are substituted with the sentence to process, respectively, the maximum number of triplets found in a sentence in the corresponding dataset.

14

sentence-triplets pairs retrieved from the KB. In the first case, augmentation is achieved by simply attaching the retrieved triplets T_c as an additional "Context Triplets" argument to the base prompt reported in Figure 2. For the second approach, instead, we substitute the two static examples provided in the base prompt, with the input relevant examples E_c retrieved from the KB.

The relevant context information to build the T_c triplets set for each input sentence is retrieved as follows. Given the KB, we isolate all the triplets $(e_s^i, r_p^i, e_o^i) \in \mathcal{G}$ contained therein, and store them in a node based vector store index (Liu, 2022). In detail, each node of this index corresponds to one and only one of the triplets and stores the embedding obtained by running a small-scale sentence encoder, MiniLM (Wang et al., 2020), on the corresponding $(subject, predicate, object)$ string. In the first approximation, this should be enough to provide a meaningful embedding for each triplet. During inference (*i.e.*, TE), we first encode the input sentence using the MiniLM. This is followed by comparing the obtained sentence embedding with all the triplet embeddings contained in the index to retrieve the top N_{KB} most similar triplets to the input sentence. Out of this N_{KB}-dimensional sample, we further select the first two triplets for each relation type present in the sample. This is done to obtain a more diverse set of context triplets with a more homogeneous distribution over the relations. In some cases, indeed, the risk of obtaining a highly biased distribution towards a specific relation type exists, which is sub-optimal for those sentences that contain several different relationships.

Note that a similar procedure can be followed to prepare the E_c examples set. However, in this case, the focus will be shifted to the example sentences we wish to include. Namely, each node of the vector store index is going to consist both of the example sentence and the KB triplets to be extracted from it. Then, the embedding vector is obtained by running the sentence encoder on either, the example sentence alone, or the sentence and triplets combined. As before, at inference time the top N_{KB} most similar (sentence, triplets) pairs to the input sentence are retrieved and included in the prompt as Few-Shots examples.

4 Experiments

In this section, we first provide details about the datasets and models we tested. This is followed by

	Train	Validation	Test	Relations	Max	Avg
WebNLG	5,019	500	703	171	7	2.29
NYT	56,195	5,000	5,000	24	22	1.72

Table 1: Statistics of the WebNLG and NYT datasets. The number of training, validation, and testing sentences is reported, together with the number of relations types in the dataset and the maximum and average number of triplets contained in a sentence.

	Parameters [B]	Context
GPT-2 (Radford et al., 2019)	0.1 | 1.5	1,024
Falcon (Penedo et al., 2023)	7 | 40	2,048
LLaMA (Touvron et al., 2023)	13 | 65	2,048

Table 2: The number of parameters (in billions [B]) and context window size of the selected LLMs.

the presentation of the main results for the TE task.

4.1 Datasets and Models

In order to test the TE capabilities of a selected set of LLMs (see Table 2 for their comparison), we experimented with two standard benchmarks for the TE task: the aforementioned WebNLG (Gardent et al., 2017) and the New York Times (NYT) (Riedel et al., 2010) dataset (see Table 1 for their basic statistics). The former was initially proposed as a benchmark for the NLG task, but has been successively adapted to the TE task and included in the WebNLG challenge (Castro Ferreira et al., 2020). As the revision provided by Zheng et al. (2017) appears to be the most widely used in the literature, we decided to run our tests on that particular version of WebNLG. The NYT benchmark is a dataset created by distant supervision, aligning more than 1.8 million articles from the NYT newspaper with the Freebase KB. For each dataset, we used the training and validation splits to build the corresponding KB following the procedure outlined in Section 3.3.

We selected the LLMs reported in Table 2 for testing. We ran locally all the models in their 8-bit quantized version provided by the Hugging-Face (Wolf et al., 2020) library. We tested the use of OpenAI models through their provided API as well. However, as their results were often inconsistent and given the limited access and control we had over them, we decided to exclude these models from the main report. All the experiments regarding them can be found in Appendix A.1. The temperature was set to $\tau = 0.1$ for all the experiments. We experimented with higher temperatures but observed that they were detrimental to the TE

Model	WebNLG	NYT
NovelTagging (Zheng et al., 2017)	0.283	0.420
CopyRE (Zeng et al., 2018),	0.371	0.587
GraphRel (Fu et al., 2019)	0.429	0.619
OrderCopyRE (Zeng et al., 2019)	0.616	0.721
UniRel (Tang et al., 2022)	**0.947**	**0.937**

Table 3: Micro-averaged F1 of some finetuned models selected from the literature.

Model		WebNLG		NYT	
		0-Shot	2-Shots	0-Shot	2-Shots
GPT-2	base	0.000	0.006	0.000	0.000
	xl	0.000	0.037	0.000	0.000
Falcon	7b	0.000	0.066	0.000	0.002
	40b	0.021	0.158	0.000	0.007
LLaMA	13b	0.006	0.129	0.000	0.002
	65b	**0.041**	**0.219**	0.000	**0.017**

Table 4: Zero and 2-Shots micro-averaged F1 performance of the LLMs tested with the prompt of Figure 2 and without any context coming from the KB.

Model		WebNLG		NYT	
		0.5-Shot	5-Shots	0.5-Shot	5-Shots
GPT-2	base	0.249	0.430	0.175	0.375
	xl	0.297	0.517	0.193	0.448
Falcon	7b	0.381	0.567	**0.250**	0.519
	40b	0.345	0.615	0.226	0.547
LLaMA	13b	0.374	0.609	0.247	0.582
	65b	0.377	**0.677**	0.243	**0.647**

Table 5: 0.5 and 5-Shots micro-averaged F1 performance of the LLMs tested with the prompt of Figure 2 augmented with $N_{KB} = 5$ triplets, respectively, sentence-triplets pairs retrieved from the KB.

performance of the model. For Falcon and LLaMA LLMs, we also explored their *instructed* counterparts, *i.e.*, models that were fine-tuned for chat applications, either through Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2023) or supervision from other LLMs (Taori et al., 2023). However, as the instructed models always performed on par, or worse, in our tests, we decided to present the base variants.

We made use of the LlamaIndex (Liu, 2022), LangChain (Chase, 2022) and HuggingFace transformers (Wolf et al., 2020) python libraries for the implementation of the pipeline.

4.2 Zero- and 2-Shots without the KB

As a baseline, we test the Zero- and 2-Shots capabilities of the LLMs without any additional information supplemented from a KB. As described in Section 3, we prompt the LLM with the base prompt of Figure 2 to extract all the triplets for a sentence in the form (subject, predicate, object). In particular, for the 2-Shots settings, two standard examples are included in the prompt but not changed over the different sentences (*c.f.* Figure 2).

In general, the LLMs queried by the base prompt do not seem capable of performing well in the TE task (Table 4). The two static examples included in the 2-Shots setting help to clarify the task and improve substantially the performance over the Zero-Shot. However, all models struggle to achieve the performance of the classical base-

line NLP models (Table 3). The sole exception is the LLaMA 65B model that achieves an F1 score close to the one obtained by Zheng et al. (2017) in the WebNLG dataset with 2-Shots. In particular, the NYT benchmark appears to be challenging for LLMs as they have difficulties even reaching a mere 1% F1 score. This discrepancy in performance between the datasets could potentially be explained as follows: In contrast to the WebNLG dataset, which features more linear and simple sentences, in NYT articles quite complex structures, with several subordinate clauses and implicit relations, are frequent. In particular, the triplet labels of the NYT dataset often cover only a subset of the actual relations found in the sentence. Therefore, without training examples available, LLMs cannot infer which relations are and are not supposed to be extracted.

4.3 Zero-shot with KB Triplets (0.5-Shots)

If we supplement the LLMs with context triplets retrieved from the KB, as described in Section 3.3 and illustrated in Figure 1, the performance of the LLM in the TE task increases substantially (see Table 5). We refer to this setting where only a set of context triplets, but no example sentence, is provided to the model as 0.5-Shots. The additional triplets hint at which relations and entities the LLM should expect, but they do not give any indication of which sentence pattern they could arise from.

In this case, the smallest model we tested, namely GPT-2 *base*, is competitive with the LLaMA 65B model without context triplets, both, for the WebNLG and the NYT dataset. Furthermore, the bigger models (Falcon, LLaMa) perform better or on par with some of the classical NLP baselines for the WebNLG dataset given in Table 3.

Even so for the NYT dataset a large improvement is obtained under the addition of context

triplets, all the LLMs are not able to reach scores competitive with the classical NLP models. The reason behind this might be related to the lower capability of the KB retriever to gather relevant context for NYT (*c.f.* Figure 3) discussed below and to the specific difficulties associated with the NYT dataset discussed in the previous section.

In general, it is interesting to observe that performance with the addition of the context triplets appears to be less dependent on the particular LLM used in case of 0.5-Shot setting. Quite remarkably, the small GPT-2 *xl* is able to retain most of the performance of the larger models. This is particularly evident for the NYT dataset, where all the LLMs are not able to perform better than a 25% F1 threshold. This could be seen as a symptom of the TE accuracy being mainly driven by the added context triplets in this case. Indeed, we also tested this KB triplets augmentation combined with the inclusion of the two static examples used in Section 4.2, but no significant differences were observed.

4.4 Few Shots with KB Sentence-Triplets Pairs

To further aid LLMs in the TE task, we experiment with inclusion in the prompt of input-specific (sentence, triplets) example pairs retrieved from the KB, as detailed in Section 3.3. Such updated prompts should provide a much stronger signal to the LLM as they not only suggest which entities and relations the LLM should expect, but also which kind of patterns in the sentence correspond to a specific relation. In particular, as it will be discussed in Section 4.5, the measured train-test overlapping seems to be large for both datasets (*c.f.* Figure 3) and, therefore, the updated prompts are likely to include examples of similar sentences. Therefore, performance improvements are expected, and in fact, looking at Table 5, we see that including 5 of these examples in the prompt makes the LLM competitive with most of the classical baselines reported in Table 3 (except the most recent SOTA from Tang et al. (2022)).

Interestingly, the performance gap between the two datasets narrowed under the updated prompt. In particular, the NYT corpus seems to have become far easier now for the LLMs. As discussed in Section 4.2, this dataset consists of sentences with a much more complex structure and more implicit relations. Therefore, having available examples of similarly constructed sentences might have helped the models to more easily identify the
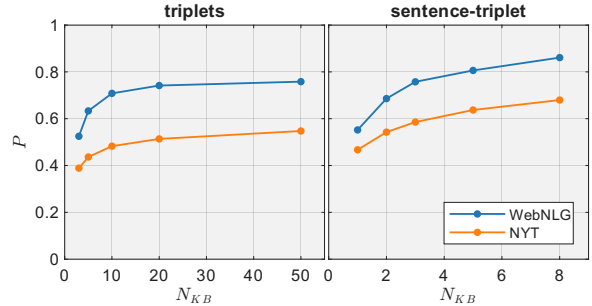

Figure 3: Probability that the correct triplet is present inside the retrieved KB context consisting of (left) triplets alone or (right) sentence-triplet example pairs, plotted against the amount of context gathered, N_{KB}.

correct triplets.

4.5 Quality of the KB Context

To evaluate the effectiveness of the KB retriever and the quality of the included KB context, we plot in Figure 3 the probability of finding the correct triplets with increasing N_{KB}, *i.e.*, the solution to the TE task, inside the gathered KB context. Namely, for each test sentence contained in the two datasets, we looped over every labeled triplet and counted the number of times it was contained inside the context provided by the retriever. We repeated this procedure for different values of N_{KB}.

Figure 3(left) suggests that $N_{KB} \sim 10 - 20$ retrieved triplets almost maximize the probability of retrieving a useful context already, as, beyond that, the improvement is only marginal. However, as few as five triplets worked the best in our tests. Probably, a greater number of context triplets retrieved leads to a marginally increased likelihood of including relevant information, but at the cost of a larger dilution. Conversely, as illustrated by Figure 3(right), for the sentence-triplets augmentation convergence is not reached with $N_{KB} = 8$ yet. However, in our experiments, the final TE performance only marginally improved going from 5 to 8 sentence-triplets examples included. Still, it is interesting to note that LLM performance increases with N_{KB} in this case, providing further evidence that the examples composed of sentence-triplets pairs are much more informative. Adding several of them does not lead to a dilution of useful information, but rather contributes to widening the spectrum of examples the LLM can take "inspiration" from.

In general, the probability of providing the correct triplet to the LLM through the context appears
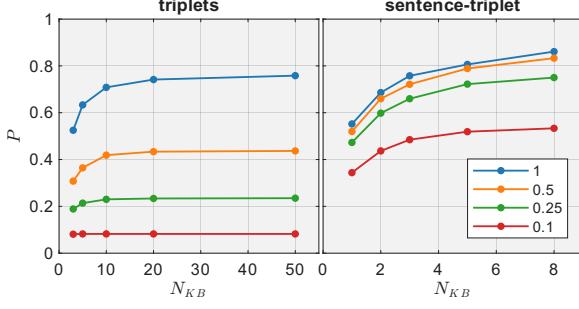
Figure 4: Probability that the correct triplet is present among the retrieved KB context for the WebNLG dataset, as in Figure 3, but with different scaled-down versions of the original KB.

to be large: greater than 50% in the majority of the cases, and even approaching the $70 \sim 80\%$ for the WebNLG dataset. This is symptomatic of substantial overlap that exists between the training, validation, and test splits for both datasets, to the point that even a stochastic model, that randomly sampled the triplets out of the KB context retrieved, was able achieve performance competitive with many of the LLMs and baselines of Table 3 in some cases (see Appendix A.2 for more details).

4.6 Ablation Study

To further investigate the impact of the additional knowledge retrieved from the KB, we revisit in this section the performance of one of our best performing LLMs, LLaMA-65b. In detail, we construct a scaled-down version of the KB via randomly sampling from the original training and validation splits, keeping only a fraction of the original sentences and triplets. For this reduced KB, the probability of having the correct triplet answer already within the retrieved information is reduced (*c.f.* Figure 4). This allows us to evaluate how the accuracy of the model is impacted by the quality of the retrieved data.

We decided to conduct this test on the WebNLG dataset. As $P(N_{KB})$ for the full-scale KB has been larger than for the NYT dataset, *c.f.* Figure 3, a wider range of values to be explored is allowed. Nonetheless, a preliminary test on the NYT dataset yielded similar results. In Figure 5a we report the variation of the final F1 score obtained by LLaMA-65b with prompts augmented by $N_{KB} = 5$ triplets and sentence-triplets pairs gathered from a KB of different scales $S = 0, 0.1, 0.25, 0.5, 1$. Here, the scale refers to the fraction of left-over data from the original KB. Note that $S = 0$ corresponds to the

original prompt without any additional information from the KB. The F1 score is plotted against the probability $P_S(N_{KB} = 5)$ of having the correct triplet inside the retrieved data with $N_{KB} = 5$ for the different KB sizes. This corresponds to the probability curves of Figure 4 evaluated at $N_{KB} = 5$. We observe that the performance degrades as the probability $P_S(N_{KB})$ shrinks with decreasing S, as expected. In particular, the relation appears to be linear: $F1_{triplets} \sim 0.25 \cdot P_s(N_{KB} = 5) + 0.21$. $F1_{sentence-triplets} \sim 0.55 \cdot P_s(N_{KB} = 5) + 0.21$. with measured determination coefficients $r^2 = 0.98$ and $r^2 = 0.96$, respectively. This suggests that there is a strong correlation between the TE capabilities of the model and the quality of the retrieved data.

Furthermore, we investigated how the final TE performance scales with the size of the model. In Figure 5b, the F1 score is plotted against the number of parameters N_{par} in log scale for all the models we tested. The plot includes the results obtained for both the WebNLG and NYT datasets, for all settings considered. We observe that for each of the three settings, the models' performance grows linearly in log scale with respect to their sizes. The scaling in the number of parameters N_{par} in log scale can be approximated by

$$F1_{norm} \sim m \cdot \log N_{par}. \quad (3)$$

The slope parameters of the linear fit for WebNLG are $m = 0.0456, 0.0304$, and 0.0871 for, respectively, 2-Shot, 0.5-Shot(KB), 5-Shots(KB) settings, and for the NYT the corresponding parameters are $m = 0.0028, 0.0257$ and 0.0906. The determination coefficients for the WebLNG and NYT datasets are, respectively, $r^2 = 0.67, 0.62$, and 0.97, and $r^2 = 0.18, 0.7$ and 0.90. Interestingly, the F1 score increase with the size of the model is steeper for the few-shots prompt (*c.f.* Figure 5b right). This suggests that larger models might be more capable in making use of several examples included inside of the prompt.

Therefore, the $F1$ score and thus the TE accuracy appears to scale linearly with the size of the KB (*c.f.* Figure 5a), but only logarithmically with the size of the model (*c.f.* Figure 5b). This suggests that it could be better to invest resources to improve the quality of the KB and its associated information retriever, rather than in training larger models.

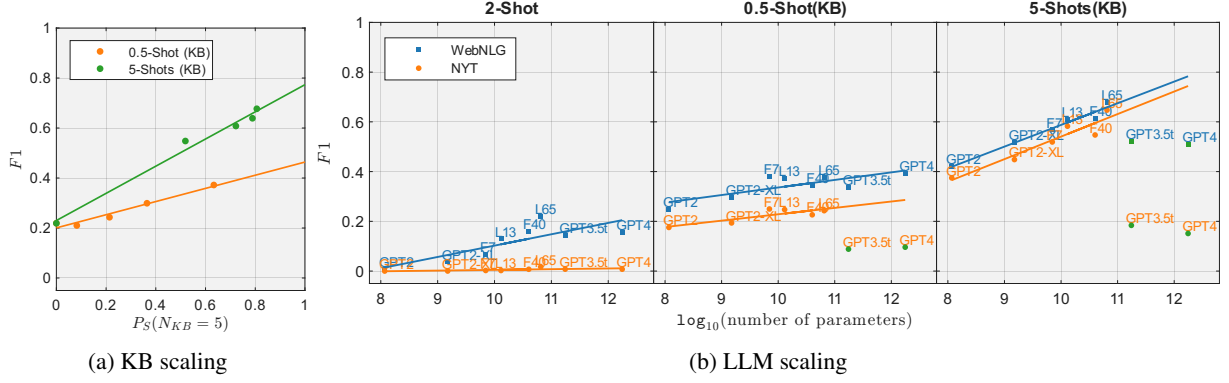| | (a) KB scaling | (b) LLM scaling |

Figure 5: (Left) Triplets (orange) and sentence-triplets (green) KB augmented performance of the LLaMA-65b model with different scaled-down versions of the KB built for the WebNLG, $S = 0, 0.1, 0.25, 0.5, 1$. The F1 score is plotted against the probability of retrieving the correct triplet with $N_{KB} = 5$ for each S (namely $P(N_{KB} = 5)$ for each curve of Figure 4). (Right) $F1$ score obtained by the tested models, plotted against their corresponding log of number of parameters, for WebNLG (blue) and NYT (orange) in the three settings: 2-shots, 0.5-shots with KB triplets ($N_{KB} = 5$), and 5-shots with KB sentence-triplets pairs ($N_{KB} = 5$). The outliers (GPT-4 and GPT-3.5 turbo) are shown in green.

5   Conclusion

In this work, a pipeline for Zero- and Few-Shots TE from sentences was presented and tested for various LLMs. We showed that the inclusion of KB information into the LLMs prompting can substantially improve the TE performance. In particular, small models were often able to outperform their bigger siblings without access to the additional KB information. Furthermore, with the information from the KB organized as sentence-triplets pair examples relevant to the input sentence, the accuracy of the LLMs improved further. In this setting, the larger LLMs were getting closer to the classical SOTA models and outperformed most of the older baselines. However, even for the largest models, TE remains a challenging task without any finetuning. LLMs were still no match for SOTA classical finetuned models in the two standard benchmark datasets we tested as part of our work, in agreement with Wadhwa et al. (2023); Wei et al. (2023b); Zhu et al. (2023).

Moreover, the performed investigation of the quality of the retrieved KB context showed that the solution to the TE task was often contained inside it already. This first indicated that a large overlapping between the train, validation and tests sets exists for both WebNLG and NYT, leading us to reconsider their generality for benchmarking TE capabilities and suggesting that a revision with better test isolation might be helpful. Secondly, it demonstrated that, while LLMs are capable of correctly individuating the relevant information in

the context, they do not shine, yet, in re-elaborating such information, generalizing and making use of it for different examples. Indeed, the investigation of the impact of the quality of the retrieved KB context, showed as the performance of the LLaMA-65b model linearly decreased with the probability of finding the solution of the task within the context already, indicating that the intrinsic incompleteness of KBs might represent a big limiting factor of this approach. Concurrently, we found that the TE performance improved only approximately logarithmically with the size of the model. This suggests that improving the quality of the KB and the associated information retriever might be more effective than increasing the modeling power of the LLM for TE.

Acknowledgements

References

Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Reforgiato Recupero, Harald Sack, Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, Chris Biemann, Mehwish Alam, Davide Buscaldi, Michael Cochez, Francesco Osborne, Diego Refogiato Recupero, and Harald Sack. 2022. Neural entity linking: A survey of models based on deep learning. *Semant. Web*, 13(3):527–570.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mc-Candlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mc-Candlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Harrison Chase. 2022. Langchain.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Kartik Detroja, C.K. Bhensdadia, and Brijesh S. Bhatt. 2023. A survey on relation extraction. *Intelligent Systems with Applications*, 19:200244.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mark Johnson, Peter Anderson, Mark Dras, and Mark Steedman. 2018. Predicting accuracy on large datasets from smaller pilot data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 450–455, Melbourne, Australia. Association for Computational Linguistics.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapa Nakashole, and Tom Mitchell. 2023. Zero-shot triplet extraction by template infilling. *Preprint*, arXiv:2212.10708.

Jerry Liu. 2022. LlamaIndex.

Tapas Nayak, Navonil Majumder, Pawan Goyal, and Soujanya Poria. 2021. Deep neural approaches to relation triplets extraction: a comprehensive survey. *Cognitive Computation*, 13(5):1215–1232.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. UniRel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting relation extraction in the era of large language models. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2023:15566–15589.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023a. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023b. Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. 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 367–377, Hong Kong, China. Association for Computational Linguistics.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *Preprint*, arXiv:2305.13168.

A Appendix

A.1 OpenAI Models results

We report here the results obtained by the OpenAI models listed in Table 6. We ran them remotely through the OpenAI API and always setting a temperature $T = 0.1$. We were not able to find any information regarding the parameter precision they used. Note that the GPT-3.5 and GPT-4 are instructed models. For some experiments we also tested the use of *text-davinci-002*, which is a non-instructed model based on GPT-3 and, apparently, the only base variant OpenAI provides through their API.

	Parameters [B]	Context
text-davinci-002 (Brown et al., 2020b)	175*	2,048
GPT-3.5 (Brown et al., 2020b)	175*	4,096
GPT-4 (OpenAI, 2023)	1,760*	8,192

Table 6: The number of parameters (in billions [B]) and context window size of the OpenAI LLMs. We indicate by * the numbers that are not officially confirmed.

Tables 7 and 8 report the results obtained by the OpenAI models on the WebNLG and NYT datasets in all the different settings. The comparison with the other models, *c.f.* Tables 4 and 5, shows them to be comparable to the Falcon 40B model in the majority of the cases. However, they recorded very underwhelming results on the NYT dataset both, in the 0.5-Shots and Few-Shots settings. Our manual inspection of the triplets they provided as an answer suggested that they were less keen to adhere to the entities and relations appearing in the provided KB context, often paraphrasing or reformulating them in a more prolix form that lowered the accuracy. This might be a consequence of the instructed training they had gone through, as discussed in Section 4.1. In contrast, the results provided by the non-instructed *text-davinci-002* model were more in line with all the other LLMs.

A.2   Random Model

In order to better understand the results obtained by the KB-augmented LLMs, we considered the following simple random TE model: first, we randomly select the number of triplets $n \in [1, max\_triplets]$ to extract, with $max\_triplets$ indicating the maximum number of triplets contained in a sentence of the dataset. Then, we uniformly sample n triplets out of the retrieved KB context. Surprisingly, the random model is very competitive with the KB-augmented LLM for small N_{KB} on the WebNLG dataset (see Figure 6), and similar results were observed for the NYT dataset. This can be explained by the hand of Figure 3. In detail, we infer from the figure that the KB-augmented prompt has a large probability of containing the correct triplets to extract already, therefore, even randomly selecting a subset of them yields a relatively high accuracy. This provides further confirmation that the TE performance is largely driven by the KB retriever.

However, the performance of the random model decreases polynomially with N_{KB}, as the probability of randomly sampling the correct triplets

Model		WebNLG		NYT	
		0-Shot	2-Shots	0-Shot	2-Shots
OpenAI	GPT-3.5	0.000	0.144	0.000	**0.008**
	GPT-4	**0.007**	**0.156**	0.000	0.007

Table 7: Zero and 2-Shots micro-averaged F1 performance of the LLMs tested with the prompt of Figure 2 and without any context coming from the KB.

Model		WebNLG		NYT	
		0.5-Shot	5-Shots	0.5-Shot	5-Shots
OpenAI	*text-davinci-002*	**0.403**	0.491	**0.144**	**0.418**
	GPT-3.5	0.336	**0.520**	0.088	0.184
	GPT-4	0.394	0.510	0.096	0.151

Table 8: 0.5 and 5-Shots micro-averaged F1 performance of the OpenAI LLMs tested with the prompt of Figure 2 augmented with $N_{KB} = 5$ triplets, respectively, sentence-triplets pairs retrieved from the KB.
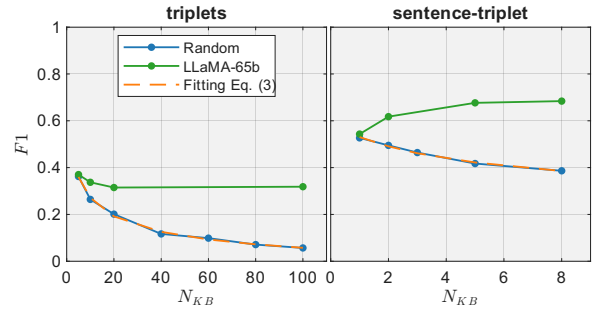

Figure 6: Degradation of the random model performance with the increase of the context information included, N_{KB}. The LLaMA-65b, instead, is able to retain most of its performance when more triplets are added (left panel), and sees a significant F1 rise with an increasing number of sentence-triplets pairs (right panel). For reference, we also report the fit of (4) as a dashed orange line.

follows the empirical scaling relation

$$F1_{rand}(N_{KB}) \sim \left(\frac{P(N_{KB})}{N_{KB}} \right)^n, \qquad (4)$$

with n number of triplets to extract and $P(N_{KB})$ probability of retrieving the correct triplet from the KB (Figure 3).

In contrast, the LLM is able to retain much of its original performance for a larger number of triplets provided (*c.f.* Figure 6(left)) or even improve under the inclusion of more sentence-triplets examples (*c.f.* Figure 6(right)).

A.3   Prompts

Here we report all the TE prompts that we tested. Figures 7 and 8 report two variations of the base prompt of Figure 2. The first one implements a Chain-of-Thought (Wei et al., 2023a) approach where multi-step reasoning is enforced. The second tries to provide the LLM with more information about the task, describing in more detail the role of each one of the core components of TE. In Table 9 the three prompts of Figures 2, 7, and 8 are compared for the WebNLG and NYT datasets

Chain-of-Thought Prompt

```
Some text is provided below. Procede step by step:
- Identify a predicate expressed in the text
- Identify the subject of that predicate
- Identify the object of that predicate
- Extract the corresponding (subject, predicate, object)
  knowledge triplet
- Repeat until all predicates contained in the text have been extracted,
  but no more than {max_triplets} times
----------------------------------------------------------------------------------------------------
Text: {text}
Triplets:
```

Figure 7: Prompt implementing the Chain-of-Thoughts approach (Wei et al., 2023a).

Documented Prompt

```
Some text is provided below. The text might contain one or more
predicates expressing a relation between a subject and an object.
The subject is the entity that takes or undergo the action expressed by
the predicate.
The object is the entity which is the factual object of the action.
The information provided by each predicate can be summarized as a
knowledge triplet of the form (subject, predicate, object).
Extract all the information contained in the text in the form of
knowledge triplets. Extract no more than {max_triplets} knowledge
triplets.
----------------------------------------------------------------------------------------------------
Text: {text}
Triplets:
```

Figure 8: Prompt providing more details about the core components of the TE task, namely, including definitions of subject, object, predicate, and triplet.

under the use of two different LLMs, GPT-2 *xl*, and LLaMA 65B. The three prompts yield similar micro-averaged F1 scores, with small deviations.

Figure 9 reports the prompt that we used in the 0.5-Shots setting. The prompt consists of a simple adaptation of the base prompt of Figure 2 to accommodate for the additional triplets retrieved from the KB.

	Prompt	WebNLG	NYT
GPT-2 xl	*base*	0.037	0.0002
	documented	0.034	0.0003
	chain-of-thought	0.039	0.0004
		0.002	0.0001
LLaMA-65b	*base*	0.219	0.017
	documented	0.213	0.012
	chain-of-thought	0.219	0.015
		0.003	0.002

Table 9: Comparison of the 2-Shots TE micro-averaged F1 performance with the three different prompts of Figures 2, 7, and 8. The standard deviation of the performance across the three prompts is reported below each column.

0.5-Shots Prompt

```
Some text and some context triplets in the form
(subject, predicate, object) are provided below.
Firstly, select the context triplets that are relevant to the input text.
Then, extract up to {max_triplets} knowledge triplets in the form
(subject, predicate, object) contained in the text taking inspiration
from the context triplets selected.
----------------------------------------------------------------------------------------------------
Text: {text}
Context Triplets:
{context_triplets}
Triplets:
```

Figure 9: Adaptation of the base prompt found in Figure 2 to the 0.5-Shots setting. An additional $\{context\_triplets\}$ argument is included to accommodate for the KB triplets retrieved from the KB.