Knowledge Base Completion for Long-Tail Entities

Lihu Chen[1] [*], **Simon Razniewski**[2], **Gerhard Weikum**[2]
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2] Max Planck Institute for Informatics, Saarbrücken, Germany
{lihu.chen}@telecom-paris.fr
{srazniew,weikum}@mpi-inf.mpg.de

Abstract

Despite their impressive scale, knowledge bases (KBs), such as Wikidata, still contain significant gaps. Language models (LMs) have been proposed as a source for filling these gaps. However, prior works have focused on prominent entities with rich coverage by LMs, neglecting the crucial case of long-tail entities. In this paper, we present a novel method for LM-based-KB completion that is specifically geared for facts about long-tail entities. The method leverages two different LMs in two stages: for candidate retrieval and for candidate verification and disambiguation. To evaluate our method and various baselines, we introduce a novel dataset, called MALT, rooted in Wikidata. Our method outperforms all baselines in F1, with major gains especially in recall.

1 Introduction

Motivation and Problem. Knowledge base completion (KBC) is crucial to continuously enhance the scope and scale of large knowledge graphs (KGs). It is often cast into a link prediction task: infer an O(bject) argument for a given S(ubject)-P(redicate) pair. However, the task is focused on the KG itself as the only input, and thus largely bound to predict SPO facts that are also derivable by simple logical rules for inverse predicates, transitive predicates etc. (Akrami et al., 2020; Sun et al., 2020). To obtain truly new facts, more recent methods tap into large language models (LMs) that are learned from huge text collections, including all Wikipedia articles, news articles and more. The most promising approaches to this end generate cloze questions for knowledge acquisition and ask LMs to generate answers (Petroni et al., 2019). The LM input is often augmented with carefully crafted short prompts (e.g., a relevant Wikipedia paragraph) (Shin et al., 2020; Jiang et al., 2020b; Qin and Eisner, 2021).

However, notwithstanding great success for question answering to humans, the LM-based approach falls short on meeting the high quality requirements for enriching a KG with crisp SPO facts. Even if most answers are correct, there is a non-negligible fraction of false or even "hallucinated" outputs by the LM, and large KGs, like Wikidata (Vrandečić and Krötzsch, 2014), cannot tolerate error rates above 10 percent. Moreover, even correct answers are not properly canonicalized: they are surface phrases and not unique entities in the KG. These problems are further aggravated when the to-be-inferred O arguments are *long-tail* entities, with very few facts in Wikidata. Here, we call an entity *long-tail* when it has less than 14 triples in Wikidata, because nearly 50% of the Wikidata entities have fewer than 14 triples. These are exactly the pain point that calls for KBC. This paper addresses this problem.

As an example, consider the late Canadian singer *Lhasa de Sela*. Wikidata solely covers basic biographic facts and selected awards, nothing about her music. However, text sources such as her Wikipedia article or other web pages provide expressive statements about her albums, songs, collaborations etc. For example, we would like to spot the facts that ⟨*Lhasa de Sela, collaboratedWith, Bratsch*⟩ and ⟨*Lhasa de Sela, performedSong, Anyone and Everyone*⟩. Note that capturing these as SPO facts faces the challenge of having to capture and disambiguate multi-word names (*"Lhasa de Sela"*) and common-noun phrases (*"anyone and everyone"*). When trying to extract such statements via cloze questions or more refined prompts to LMs such as GPT-3 (Brown et al., 2020) or chatGPT, the outputs would often be *"Lhasa"*, which is highly ambiguous, or *"everyone"*, which is incomplete and impossible to interpret.

Approach and Contribution. This paper devises a novel method for knowledge base completion (KBC), specifically geared to cope with long-tail

[*] Work done during an internship at Max Planck Institute for Informatics

Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023), pages 99–108

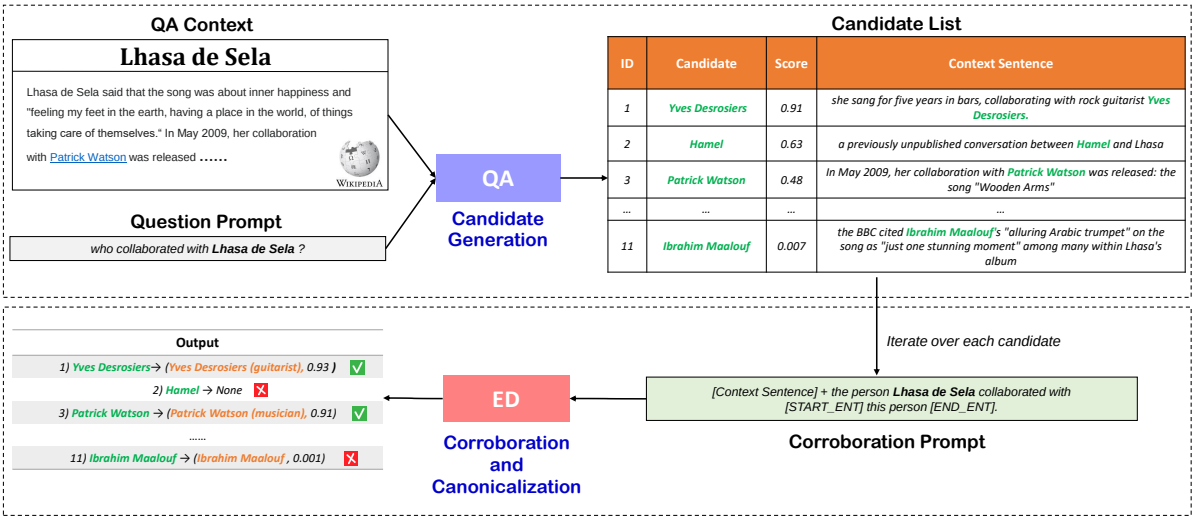

Figure 1: The framework of our two-stage KBC method.

entities. Although we will present experimental comparisons to prior works on relation extraction from text, we believe that ours is among the first works to successfully cope with the challenge of noise and ambiguity in the long tail.

Our method leverages Transformer-based language models in a new way. Most notably, we employ two different LMs in a two-stage pipeline, as shown in Figure 1. The first stage generates candidate answers to input prompts and gives cues to retrieve informative sentences from Wikipedia and other sources. The second stage validates (or falsifies) the candidates and disambiguates the retained answer strings onto entities in the underlying KG (e.g., mapping *"Lhasa"* to `Lhasa de Sela` , and "Bratsch" to `Bratsch (band)`).

The novel contributions of this work are the following:

- the first KBC method that leverages LMs to cope with long-tail entities;
- a new dataset, called MALT, to benchmark methods with long-tail entities;
- experimental comparisons with baselines, using the MALT data.

Our code and data are available at `https://github.com/tigerchen52/long_tail_kbc`.

2 Related Work

Knowledge Base Completion. This task, KBC for short, has mostly been tackled as a form of link prediction: given a head entity S and a relation P, predict the respective tail entity O, using the KG as sole input. A rich suite of methods have been developed for this task, typically based on latent embeddings computed via matrix or tensor factorization, neural auto-encoders, graph neural networks, and more (see, e.g., surveys (Chen et al., 2020; Ji et al., 2022) and original references given there). However, the premise of inferring missing facts from the KG itself is a fundamental limitation. Indeed, several studies have found that many facts predicted via the above KBC techniques are fairly obvious and could also be derived by simple rules for transitivity, inverse relations etc. (Akrami et al., 2020; Sun et al., 2020).

Language Models as Knowledge Bases. The LAMA project (Petroni et al., 2019) posed the hypothesis that probing LMs with cloze questions is a powerful way of extracting structured facts from the latently represented corpus on which the LM was trained. A suite of follow-up works pursued this theme further and devised improvements and extensions (e.g., (Heinzerling and Inui, 2021; Jiang et al., 2020a; Kassner and Schütze, 2020; Roberts et al., 2020; Shin et al., 2020; Zhong et al., 2021)). This gave rise to the notion of "prompt engineering" for all kinds of NLP tasks (Liu et al., 2021). In parallel, other works studied biases and limitations of the LM-as-KB paradigm (e.g., (Cao et al., 2021; Elazar et al., 2021; Razniewski et al., 2021; Jiang et al., 2020b)). In this work, we investigate the feasibility of leveraging LMs to complete real-world KBs, and mainly focus on long-tail facts.

3 Two-Stage KBC Method

We propose an unsupervised method for KBC that taps into LMs as latent source for facts that can-

Subject Type	Relation	Wikidata ID	Triples	multi-token (%)	ambiguous (%)	long-tail (%)
Business	founded by	P112	5720	97.3	21.1	91.2
MusicComposition	performer	P175	1876	91.1	62.0	47.3
	composer	P86	3016	98.2	59.8	88.5
Human	place of birth	P19	13416	23.6	81.6	99.3
	place of death	P20	7247	25.9	84.8	99.6
	employer	P108	3503	96.5	37.4	81.4
	educated at	P69	13386	99.6	38.7	72.2
	residence	P551	886	32.1	87.1	96.4
Micro-Avg	-	-	-	65.3	58.6	87.0

Table 1: Statistics for MALT dataset.

Dataset	SPO triples	Long-tail fraction
DocRED (2019)	63K	32.0 %
LAMA-TREx (2019)	34K	39.6 %
X-FACTR (2020a)	46K	49.6 %
MALT (Ours)	49K	87.0 %

Table 2: Estimated fractions of long-tail S entities across different datasets, where long-tail means at most 13 triples in Wikidata. The estimations are based on 200 samples across 8 relations.

not be inferred from the KG itself. Our method operates in two stages:

1. For a given S-P pair, generate candidate facts ⟨S,P,"O"⟩ where "O" is an entity name and possibly a multi-word phrase.
2. Corroborate the candidates, retaining the ones with high confidence of being correct, and disambiguate the "O" argument into a KG entity.

Candidate Generation. We devise a generic prompt template for cloze questions, in order to infer an "O" answer for a given S-P pair. This merely requires a simple verbalizer for the relation P:

"⟨S-type⟩ S ⟨P-verb⟩ which ⟨O-type⟩?"

(e.g., "the song ⟨S⟩ is performed by which person?" for the predicate `performer`). The S-type and O-type are easily available by the predicate type-signature from the KG schema. As additional context we feed a Wikipedia sentence from the S entity's article into the LM. This is repeated for all sentences in the respective Wikipedia article. Specifically, we employ the SpanBERT language model (Joshi et al., 2020), which is fine-tuned on on the SQuAD 2.0 (Rajpurkar et al., 2018) [1]. Note that all of this is completely unsupervised: there is no need for any fine-tuning of the LM, and there is no prompt engineering.

[1] https://huggingface.co/mrm8488/spanbert-large-finetuned-squadv2

Candidate Corroboration and Canonicalization. The first stage yields a scored list of candidates in the form of pairs ("O", s) with an entity name and a Wikipedia sentence s. In the corroboration stage, the candidates are fed into a second LM for re-ranking and pruning false positives. Specifically, we employ the generative entity disambiguation model GENRE (De Cao et al., 2020), which in turn is based on BART (Lewis et al., 2020) and fine-tuned on BLINK (Wu et al., 2020) and AIDA (Hoffart et al., 2011). We construct the input by the template:

"⟨S-type⟩ S ⟨P-verb⟩ [ENT] this ⟨O-type⟩ [ENT]"

(e.g., "the song Anyone and Everyone is performed by [ENT] this person [ENT]"), contextualized with the sentence s. GENRE generates a list of answer entities E, taken from an underlying KG, like Wikidata, that is, no longer just surface names. If the candidate name "O" approximately matches a generated E (considering alias names provided by the KG), then the entire fact, now properly canonicalized, is kept. Since we may still retain multiple facts for the same S-P input and cannot perfectly prevent false positives, the inferred facts are scored by an average of the scores from stage 1 and stage 2.

4 MALT: New Dataset for Benchmarking

Benchmarks for KBC and LM-as-KB cover facts for all kinds of entities, but tend to focus on prominent ones with frequent mentions. Likewise, benchmarks for relation extraction (RE) from text, most notably TACRED (Zhang et al., 2017), DocRED (Yao et al., 2019) and LAMA (Petroni et al., 2019) do not reflect the difficulty of coping with long-tail entities and the amplified issue of surface-name ambiguity (see Table 2. Therefore, we developed a new dataset with emphasis on the long-tail challenge, called MALT (for "**M**ulti-token, **A**mbiguous, **L**ong-**T**ailed facts").

Relation	ID	Candidate Generation	Corroboration and Canonicalization
founded by	P112	the business [x] is founded by which person?	the business [x] is founded by [ENT] this person [ENT]
performer	P175	the song [x] is performed by which person?	the song [x] is performed by [ENT] this person [ENT]
composer	P86	the song [x] is composed by which person?	the song [x] is composed by [ENT] this person [ENT]
place of birth	P19	the person [x] was born in which place?	the person [x] was born in [ENT] this place [ENT]
place of death	P20	the person [x] died in which place?	the person [x] died in [ENT] this place [ENT]
employer	P108	the person [x] worked in which place?	the person [x] worked in [ENT] this place [ENT]
educated at	P69	the person [x] graduated from which place?	the person [x] graduated from [ENT] this place [ENT]
residence	P551	the person [x] lived in which place?	the person [x] lived in [ENT] this place [ENT]

Table 3: Prompts for relations in MALT. [x] is a placeholder for the subject entity and [ENT] is a special token for the mention.

Relation	ID	NER + RC (CNN)			REBEL			KnowGL			GenIE			Ours		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
founded by	P112	13.5	21.2	16.5	42.8	27.3	33.3	0.0	0.0	0.0	59.1	7.9	13.9	57.0	44.5	50.0
performer	P175	5.2	10.1	6.9	25.3	28.1	26.6	0.0	0.0	0.0	47.3	19.1	27.2	42.7	15.6	22.9
composer	P86	17.3	20.5	18.8	37.9	27.7	32.0	37.6	25.7	30.6	70.0	16.6	26.8	67.3	65.6	66.4
place of birth	P19	4.7	4.7	4.7	49.3	20.5	28.9	49.4	23.4	31.7	64.1	9.2	16.1	47.9	61.4	53.8
place of death	P20	12.5	4.7	6.8	52.6	11.8	19.2	66.6	9.4	16.5	47.5	3.0	5.6	46.6	48.2	47.4
employer	P108	8.7	4.9	6.3	50.0	4.9	8.8	0.0	0.0	0.0	54.0	0.1	0.2	30.0	29.3	29.6
educated at	P69	8.9	8.4	7.7	15.4	1.1	2.1	22.2	1.1	2.2	46.7	0.1	0.2	42.9	39.5	41.2
residence	P551	0.0	0.0	0.0	33.3	8.3	13.3	33.3	8.3	13.3	44.4	0.2	0.4	19.2	41.7	26.3
Micro-Avg	-	26.7	13.7	13.7	38.3	16.2	20.6	26.2	8.5	11.8	52.2	6.9	11.2	44.2	43.2	42.2

Table 4: Performance comparison on MALT data.

To construct the dataset, we focus on three types of entities: Business, MusicComposition and Human, richly covered in Wikidata and often involving long-tail entities. We randomly select subjects from the respective relations in Wikidata, and keep all objects for them. We select a total of 8 predicates for the 3 types; Table 1 lists these and gives statistics.

The dataset contains 65.3% triple facts where the O entity is a multi-word phrase, and 58.6% ambiguous facts where the S or O entities share identical alias names in Wikidata. For example, the two ambiguous entities ,*"Birmingham, West Midlands (Q2256)"* and *"Birmingham, Alabama (Q79867)"*, have the same Label value *"Birmingham"*. In total, 87.0% of the sample facts have S entities in the long tail, where we define long-tail entities to have at most 13 Wikidata triples.

5 Experimental Evaluation

Baselines. To the best of our knowledge, there is no prior work on KBC or LM-as-KB that is specifically geared for coping with long-tail entities. As a proxy, we thus compare to several state-of-the-art methods for relation extraction (RE) from text. At test time, these methods receive the retrieved Wikipedia sentences for a ground-truth SPO fact and the SP pair as input, and are run to extract the withheld O argument (sentence-level extraction).

We compare to the following baselines:

- *NER + RC (CNN)* uses TNER (Ushio and Camacho-Collados, 2022) to recognize entity mentions in context sentences, followed by a CNN-based relation classifier Nguyen and Grishman (2015). The RC component is trained on REBEL (Cabot and Navigli, 2021).
- *REBEL* (Cabot and Navigli, 2021) is an end-to-end relation extraction for more than 200

different relation types in Wikidata.

- *KnowGL* (Rossiello et al., 2023) is an open-source system that can convert text into a set of Wikidata statements.
- *GenIE* (Josifoski et al., 2022) is an end-to-end closed triplet extraction model, which is trained on REBEL dataset (Cabot and Navigli, 2021). GenIE uses Wikidata as the target KB and can extract 5,891,959 entities and 857 relations.

Setup. There are two hyper-parameters for all competitors, the number of candidates k (or the "top-k" hyper-parameter for baseline models) and the threshold α for cutting off the extracted triples. For our framework, k is 20 for all competitors and the threshold α is learned by using a hold-out (20%) validation set. We report results for precision, recall and F1, with the original Wikidata triples as ground truth. Although MALT provides canonicalized entities, we consider the extracted O to be a correct prediction as long as it appears in the alias table because some baselines themselves cannot do disambiguation.

Our method is completely unsupervised, and the only additional cost is prompt. We manually design one template for each relation (as shown in Table 3).

Results. Table 4 shows the results from this experimental comparison. We observe that the GenIE baselines does well in terms of precision, but has very poor recall. In contrast, our two-stage method achieves both good precision and recall. Regarding precision, it is almost as good as GenIE (44% vs. 52%); regarding recall, it outperforms GenIE and the other baselines by a large margin (43% vs. 7%). Our method still leaves substantial room for further improvement, underlining the challenging nature of inferring facts for long-tail entities. We think of our method as a building block to aid a human curator by judicious suggestions for facts that would augment the KG.

Many of the inferred SPO facts are indeed completely missing in Wikidata; so they are also not in the withheld ground-truth samples for the above evaluation. To estimate how many facts we could potentially add to the KG and how good our automatically inferred predictions are, we picked 25 samples for each relation, a total of 250 fact candidates, and asked human annotators to assess their correctness. Over all relations, this achieved an average precision of 61%. For the relation `educated at`, our method even has 76% precision, and this is a case where the KG has enormous gaps: out of 10M sampled entities of type `Human`, only 65% have facts for this relation. For this case, our KBC method collected 1.2M candidate facts, showing the great potential towards closing these gaps.

6 Conclusion

We highlighted the challenge of knowledge base completion (KBC) for long-tail entities, introduced the MALT dataset for experimental comparisons and fostering further research, and presented a completely unsupervised method for augmenting knowledge bases with long-tail facts. Our method operates in two stages, candidate generation and candidate corroboration (incl. disambiguation), and leverages two different LMs in a complementary way. Experimental results show substantial gains over state-of-the-art baselines, and highlight the benefits of our two-stage design with two LMs complementing each other.

Limitations

Although our dataset presents a significant advancement over previous benchmarks, it is still limited in that it only contains entities already known to Wikidata. One could argue that the very long tail is what is even beyond Wikidata.

In the second stage, our method harnesses an LM pre-trained for entity disambiguation. Therefore, our methodology, in its current form, cannot predict objects that are not already known to that LM and its underlying KB.

Acknowledgements

This work was partially funded by ANR-20-CHIA-0012-01 ("NoRDF"). We thank Fabian M. Suchanek and Gaël Varoquaux for their helpful feedback.

References

Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1995–2010. ACM.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard H. Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Trans. Assoc. Comput. Linguistics*, 9:1012–1031.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 782–792.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-factr: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st workshop on*

vector space modeling for natural language processing, pages 39–48.

F Petroni, PSH Lewis, A Piktus, T Rocktäschel, Y Wu, AH Miller, and S Riedel. 2020. How context affects language models' factual predictions. AKBC.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5203–5212. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. 2021. Language models as or for knowledge bases. *arXiv preprint arXiv:2110.04888*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Nandana Mihindukulasooriya, Owen Cornec, and Alfio Gliozzo. 2023. Knowgl: Knowledge generation and linking from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha P. Talukdar, and Yiming Yang. 2020. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5516–5522. Association for Computational Linguistics.

Asahi Ushio and Jose Camacho-Collados. 2022. T-ner: An all-round python library for transformer-based named entity recognition. *arXiv preprint arXiv:2209.12616*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.

A Appendix

A.1 The Motivation of Our Two-stage KBC Method

In this section, we explain how we design the two-stage KBC method. Existing approaches use cloze-style prompts to query masked language models. However, they cannot cope with multi-token facts well and suffer from the long-tail issue. Therefore, we experiment with a series of prompts for querying LMs, and experiments can be categorized into two classes: *Context-Free* and *Context-Based*.

Context-Free experiments evaluate the capabilities of LMs to generate facts by only using prompt queries. We consider the following baselines.

Cloze: As prior methods, this baseline uses a cloze-style prompt to query masked LMs (the first frame in Figure A1). Here, two types of LMs are compared in this experiment. **Left-to-Right** LMs predict the upcoming words based on a sequence of words, and GPT-1 (Radford et al., 2018) and Transformer-xl (Dai et al., 2019) are used.

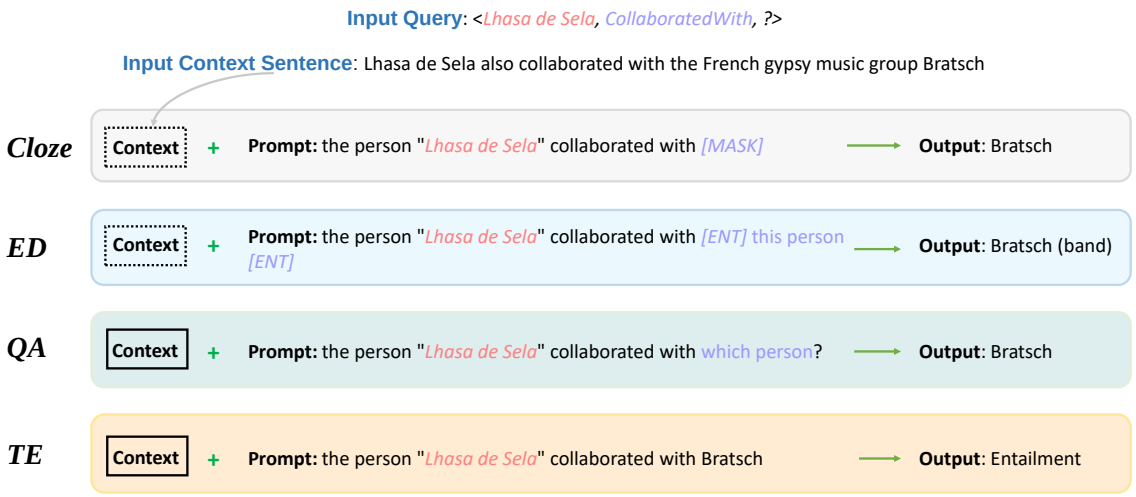

Figure A1: An illustration of different prompts for querying language models. The dashed lines mean the context sentence is optional.

Masked LMs aim to predict masked text pieces based on the surrounding context, and BERT-base and BERT-large (Devlin et al., 2019) are used. To enable BERT to handle multi-token facts, we also introduce the decoding strategy proposed in X-FACTR (Jiang et al., 2020a) for comparison.

ED: Because the Cloze-style prompt cannot generate multi-token facts directly, we propose to use Language Models with Entity Disambiguators as knowledge bases, i.e., LMED-as-KB. As shown in Figure A1, we can design such a prompt *"the person Lhasa de Sela collaborated with* `[ENT]` *this person* `[ENT]`*."*, where the mention is surrounded by special tokens `[ENT]` and `[ENT]`. After we use the prompt to query the generative disambiguation model, and it is able to disambiguate the mention *"this person"* and output the correct canonicalized entity *"Bratsch (band)"*, although the mention is not *"this band"*. The core benefit of introducing LMED is that it can output disambiguated entity names with multiple tokens. Here, we use the **Encoder-Decoder** entity disambiguation model GENRE (De Cao et al., 2020), which is fine-tuned on BLINK (Wu et al., 2020) and AIDA (Hoffart et al., 2011).

In Context-based experiments, prompts are combined with additional context information to better retrieve facts from LMs, which has been demonstrated to substantially improve the cloze-style performance of LMs (Petroni et al., 2020). Apart from Cloze and ED baselines, we introduce another two methods.

QA: Question-Answering models are able to extract answers to a question from a given document, and we adapt them to extract facts by designing question prompts. As shown in the third frame of Figure A1, given the input context and the question prompt *"the person Lhasa de Sela collaborated with which person?"*, a QA model successfully outputs the correct answer. For experiments, we use two LMs fine-tuned on the SQuAD 2.0 (Rajpurkar et al., 2018), RoBERTa-large (Liu et al., 2019) [2] and SpanBERT-large (Joshi et al., 2020) [3]. Besides, we use GPT3 (Brown et al., 2020) as another QA baseline.

TE: Textual Entailment models can judge whether a premise entails a hypothesis. To adapt TE for extracting facts from context, we first use a Named Entity Recognition model and then apply a textual entailment model to this entity and sentence for judging the entailment relation. For example, given the context *"Lhasa de Sela also appeared as a guest of the French gypsy music group Bratsch"*, the entity *"Bratsch"* is recognized and we use the prompt: *context* $\rightarrow$ *the person Lhasa de Sela collaborated with Bratsch*. If the premise entails the hypothesis, we can regard this as a correct tail entity. Here, we add type constraints for particular relations. Two LMs fine-tuned on TE datasets, RoBERTa-large (Liu et al., 2019)[4] and DeBERTa-

[2] https://huggingface.co/deepset/roberta-large-squad2

[3] https://huggingface.co/mrm8488/spanbert-finetuned-squadv2

[4] https://huggingface.co/ynie/

Model	Prompt	Size	Multi-token	Disambiguated	P	R	F1
GPT-1	*Cloze*	110M	✗	✗	0.3	3.2	0.7
Transformer-xl	*Cloze*	257M	✗	✗	2.4	3.9	2.9
BERT- base	*Cloze*	110M	✗	✗	7.1	4.9	4.2
w/ decoding	*Cloze*	110M	✓	✗	11.1	1.2	1.7
BERT-large	*Cloze*	340M	✗	✗	19.0	3.7	4.7
w/ decoding	*Cloze*	340M	✓	✗	8.7	2.1	2.4
GENRE	*ED*	406M	✓	✓	19.1	5.4	7.4

Table A1: Context-Free performances of different language models on MALT.

Model	Prompt	Size	Multi-token	Disambiguated	P	R	F1
BERT- base	*Cloze*	110M	✗	✗	11.1	12.4	11.7
BERT- large	*Cloze*	340M	✗	✗	11.8	14.4	12.3
RoBERTa-large	*QA*	355M	✓	✗	5.6	45.1	9.7
SpanBERT-large	*QA*	340M	✓	✗	1.2	66.2	2.4
GPT-3	*QA*	175B	✓	✗	10.9	11.5	7.9
RoBERTa-large	*TE*	355M	✓	✗	13.2	19.7	13.4
DeBERTa-large	*TE*	304M	✓	✗	13.5	22.2	14.6
GENRE	*ED*	406M	✓	✓	16.5	30.9	18.9

Table A2: Context-Based performances of different language models on MALT.

large (He et al., 2020)[5], are used in this experiment. The NER model is TNER (Ushio and Camacho-Collados, 2022).

Technically speaking, QA and TE are not LM-as-KB methods because they cannot generate facts without the help of context. However, these two methods have a unified pattern with Cloze and ED under the context-based setting, we, therefore, include them for comparison.

A.1.1 Can LMs Generate Facts?

In this context-free experiment, we aim to answer whether LMs can generate facts and various models are evaluated on MALT. The experimental results are shown in Table A1. We first observe all models perform poorly on MALT-Wikidata because it contains a large number of multi-token and long-tail entities. Left-to-Right and Masked LMs have difficulties in dealing with these facts, even with the introduction of multi-token decoding. Moreover, we observe that GENRE outperforms other baselines consistently and this confirms the feasibility of the usage of LMED-as-LM. Overall, a single query does not retrieve facts from LMs very effectively, and the reasons are twofold: 1) the capacity of LMs for storing world knowledge is limited by model size, i.e., LMs with tens or hundreds of billions of parameters can memorize all facts in Wikidata (Heinzerling and Inui, 2021); 2) proper prompts are needed for a better recall, e.g., by additional information or prompt engineering.

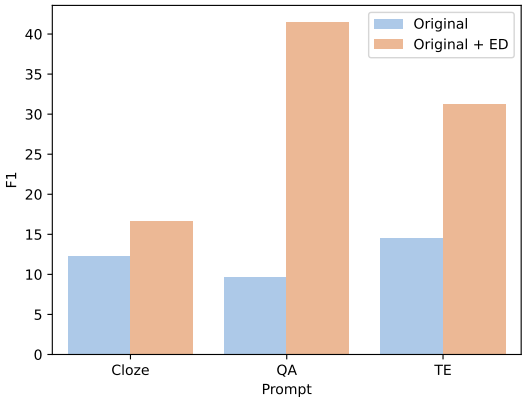

Figure A2: Improvements of adding the ED Prompt.

A.1.2 Can Context help?

In this context-based experiment, context sentences are introduced for assessing the capability of LMs to generate facts by exploiting context. Concretely, we traverse the sentences in Wikipedia for relevant entities and each context sentence is combined with a corresponding prompt to compose a new query. Next, facts are retrieved or extracted by using different LMs. For duplicated outputs, we merge them and average the score. The experimental results are shown in Table A2. We can see that adding context can remarkably improve the performances on MALT-Wikidata, e.g., BERT-large (4.7 → 12.3) and GENRE (7.4 → 18.9). GENRE consistently outperforms other baselines in terms of F1 while QA mode can obtain very high recalls. For TE methods, they are a workable approach while still lagging behind our framework.

A.1.3 Our Two-stage KBC Method

Based on the above analyses, we find that ED prompts can generate disambiguated and relatively high-quality facts while QA prompts have the highest recall. Hence, a question naturally appears: *"Can we synergize the two components to yield better facts?"*

To answer this question, we apply the ED prompt method to the facts generated by the other three methods, Cloze, QA, and TE. The post-processing step of ED prompt serves to verify and re-rank the candidates of the first step. The experimental results are shown in Figure A2. We observe the

roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli

[5]https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

combination can bring consistent improvements and the pipeline of "QA + ED" achieves the best score. Therefore, we leverage two different LMs in a two-stage pipeline. The first stage generates candidate answers by using a high-recall question-answering model. The second stage employs an entity disambiguation model for validating the candidates.