LM-KBC: Knowledge Base Construction from Pre-trained Language Models

Sneha Singhania, Tuan-Phong Nguyen and Simon Razniewski

Max Planck Institute for Informatics, Germany

Abstract

Pre-trained Language Models (LMs) have advanced a range of semantic tasks, and have also shown promise for factual knowledge extraction encoded in them. Although several works have explored this ability in the LM probing setting, viability of knowledge base construction from LMs has not yet been explored. In light of this, we hosted the *LM-KBC* challenge at the 21st International Semantic Web Conference (ISWC 2022). Participants were asked to build actual knowledge bases from LMs, for a given set of subjects and relations. In crucial difference to existing probing benchmarks like LAMA [1], we made no simplifying assumptions on relation cardinalities, i.e., a subject-entity could stand in relation with zero, one, or many object-entities. Furthermore, submitted systems were required to go beyond just ranking the predictions and materialize the outputs, which we evaluated using the established KB metrics of precision, recall, and F_1-score. The challenge had two tracks: (1) a BERT-type LM track with low computational requirements and (2) an open track, where participants could use any LM of their choice. In this first edition of the challenge, we received a total of five submissions, four for track 1 and one for track 2. We present the contributions and insights of our peer-reviewed submissions and lay out the possible paths for future work. The challenge website is https://lm-kbc.github.io.

Keywords

Language Models, Knowledge Base Construction, Prompt Learning, Language Model Probing

1. About LM-KBC

Background Large-scale LMs such as BERT [2] and GPT-3 [3] are optimized to either predict masked-out textual inputs or perform sentence completion and have notably advanced performances on a range of downstream NLP tasks like question answering and machine translation. Recently, LMs gained attention for their purported ability to yield structured pieces of knowledge directly from their parameters. This is promising as current knowledge bases (KBs) such as Wikidata [4], DBpedia [5], Yago [6] and ConceptNet [7] are part of the backbone of the Semantic Web ecosystem, yet are inherently incomplete. While constructing a KB, major challenges include relations being optional (e.g., `academic-degree`, `place-of-death`, or `parent-organization`) and presence of multiple correct object-entities per subject-relation pair (e.g., `shares-border`, `occupation`, or `speaks-language`). Additionally, KBs need materialization, i.e., deliberate decisions on which statements to include or exclude, for scrutability and consistent downstream usage.

LM-KBC: Knowledge Base Construction from Pre-trained Language Models (lm-kbc.github.io), Challenge at ISWC 2022

ssinghan@mpi-inf.mpg.de (S. Singhania); tuanphong@mpi-inf.mpg.de (T. Nguyen); srazniew@mpi-inf.mpg.de (S. Razniewski)

Previous approaches to KB construction utilized unstructured text [8, 9], crowdsourcing, or semi-structured resources [10, 5, 11]. In the seminal LAMA paper [1], Petroni et al. showed that LMs achieved encouraging results in masked knowledge ranking tasks—ranking candidate objects for a given subject-relation pair. Despite much follow-up work reporting further advancements [12, 13, 14, 15], as well as criticism [16, 17, 18, 19, 20], the prospect of using LMs for KB construction remains under-explored. The LAMA benchmark, and its variants, are not suited to investigate actual KB construction since they (i) evaluate on randomly sampled subject-object pairs, thus missing out on assessing per-subject recall, and on deciding whether a subject has objects at all, (ii) focus on single word object-entities due to the limitation of single masked token prediction specification of the underlying LM, and (iii) only evaluate a model's ranking abilities, but do not force it to make deliberate accept/reject decisions. Knowledge base construction is a task different from ranking—it requires challenging decisions on how to obtain recall in the long tail [21, 22] and how to decide acceptance thresholds.

In our challenge, we invited participants to present LM-based systems for actual KB construction, with three main challenges:

1. *Variance in the number of true objects per subject-relation pair.* For example, Germany shares borders with 9 countries, whereas Vietnam borders only 3 countries. Thus, systems had to make decisions on how many objects to retain.
2. *Instances without any true object.* For example, Apple has no parent organization, while Google is owned by Alphabet. Thus, systems had to make decisions on whether to output any objects at all.
3. *Materialization.* Systems were required to output lists of objects for each subject-relation pair, hence had to make deliberate binary retain/discard decisions on candidates and could not hide behind ranking metrics.

We evaluated the resulting KBs using established precision, recall, and F_1 metrics.

Task Description Given an input tuple of a subject-entity s and a relation r, the task is to generate the correct object-entities $[o_1, o_2, \ldots, o_k]$, using language model probing.

For example, as shown in Table 1, for a given input consisting of a subject-entity and relation pair, when BERT is probed using the sample prompt, we obtain the following top predictions with likelihood in the placeholder position "`[MASK]`". The last column gives the correct ground-truth objects. The crux of the task is that across various subject-relation pairs, there is no optimal solution to make accept/reject decisions using a uniform threshold on the LM's likelihood. The problem lies even within a single relation: if we retain predictions up to 10.7% likelihood, Germany's neighbour Belgium would be dropped. Conversely, if the threshold is lowered to 2.2%, for Vietnam, wrongly, India would be asserted as its neighbour.

BERT-style models only annotate outputs with these problematic relative likelihoods over each other; nevertheless, participating systems need to make decisions on which and how many of the candidates to retain. Participants were allowed to paraphrase the input prompts manually or through existing prompt engineering techniques [23, 24], and could even form prompt ensembles [25] for final predictions.

Input	Sample Prompt	LM Prediction & Likelihood	Ground Truth
Vietnam, `shares-border`	Vietnam shares a land border with `[MASK]`.	Cambodia, 12.1% China, 10.7% India, 10.1%	China, Cambodia, Laos
Germany, `shares-border`	Germany shares a land border with `[MASK]`.	Austria, 17.7% ... Belgium, 2.2%	Austria, ... Belgium
Carbon dioxiode, `consists-of`	Carbon dioxiode consists of `[MASK]`.	Oxygen, 20.8% Water, 14% Nitrogen, 11.5%	Carbon, Oxygen
Angela Merkel, `speaks-language`	Angela Merkel can speak in `[MASK]`.	German, 89.1% English, 5.3% Italian, 0.5%	German, English, Russian
Elon Musk, `place-of-death`	Elon Musk died in `[MASK]`.	office, 4.8% prison, 3% Chicago, 2.8%	$\emptyset$

Table 1
Sample inputs and the corresponding top-3 predicted outputs by BERT.

The challenge had two tracks:

1. **BERT track**, where only computationally modest BERT-type models were allowed;
2. **Open track**, where any language model, also autoregressive or generative models, could be used.

Using a public training dataset, participants were allowed to prompt-engineer, retrain, fine-tune, use context examples (e.g., for GPT-3 [3]), or use additional textual data (e.g., Wikipedia snippets as prompt context), to optimize their output.

LM-KBC22 Dataset We curated a dataset comprising 12 relations, each comprising a set of subjects and a complete list of ground-truth objects per subject-relation-pair. For each relation, maximum of 100 subjects were provided for training, another 50 for validation and testing, while a third 50 were withheld (private test) for challenge evaluation. Table 2 gives more details on our released dataset. The relations were chosen so as to ensure diversity, and the subject-entities were of different types, e.g., person, country, organization. To further increase realism, 5 relations also contained subjects without any correct ground truth objects (e.g., Apple having no parent organization). We provided aliases for ground-truth objects that are known under multiple names, and outputting any one of them was sufficient. In particular, to facilitate usage of LMs like BERT (which are constrained by single-token predictions), we provided a valid single-token form for multi-token object-entities, wherever such a form was meaningful.

Evaluation For each test instance, predictions submitted by participating systems were evaluated by calculating precision, recall, and F_1 metrics against ground-truth values. Let $\mathcal{P}$

Relation	Description	Example	|Train|	|Dev|	|Test|	Range(Train)	Range(Dev)	Range(Test)
`shares-border`	country (s) shares a land border with another country (o)	⟨Argentina, `shares-border`, [Bolivia, Brazil, Paraguay, Chile, Uruguay]⟩	100	47	50	[0, 17]	[0, 14]	[0, 11]
`official-language`	country (s) has an official language (o)	⟨Belarus, `official-language`, [Belarusian, Russian]⟩	100	47	50	[1, 4]	[1, 15]	[1, 11]
`shares-border`	state (s) of a country shares a land border with another state (o)	⟨Oregon, `shares-border`, [California, Idaho, Washington, Nevada]⟩	100	50	50	[1, 14]	[1, 15]	[1, 14]
`basin-country`	river (s) basins in a country (o)	⟨Saar, `basin-country`, [Germany, France]⟩	100	50	50	[1, 6]	[1, 10]	[1, 9]
`consists-of`	chemical compound (s) consists of an element (o)	⟨Nitroglycerin, `consists-of`, [Hydrogen, Oxygen, Nitrogen, Carbon]⟩	100	50	50	[2, 6]	[2, 6]	[2, 6]
`speaks-language`	person (s) speaks in a language (o)	⟨Bruno Mars, `speaks-language`, [Spanish, English]⟩	100	50	50	[1, 6]	[1, 5]	[1, 7]
`plays-instrument`	person (s) plays an instrument (o)	⟨Chester Bennington, `plays-instrument`, [*None*]⟩	100	50	50	[0, 7]	[0, 14]	[0, 7]
`employer`	person (s) is employed by a company (o)	⟨Susan Wojcicki, `employer`, [Google]⟩	100	50	50	[1, 8]	[1, 8]	[1, 8]
`profession`	person (s) held a profession (o)	⟨Shakira, `profession`, [[Singer-Songwriter, Singer, Songwriter], Guitarist]⟩	100	50	50	[1, 23]	[1, 19]	[1, 20]
`place-of-death`	person (s) died at a location (o)	⟨Elvis Presley, `place-of-death`, [Graceland] ⟩	100	50	50	[0, 1]	[0, 1]	[0, 1]
`cause-of-death`	person (s) died due to a cause (o)	⟨John lewis, `cause-of-death`, [[Pancreatic Cancer, Cancer]]⟩	100	50	50	[0, 1]	[0, 1]	[0, 1]
`parent-org`	company (s) has another company (o) as its parent organization	⟨Apple Inc, `parent-org`, [*None*]⟩	100	50	50	[0, 5]	[0, 1]	[0, 3]

Table 2
Characteristics of the LM-KBC22 dataset. It contains 12 diverse relations, among which 5 relations can have subjects without any objects. Since there are less than 200 countries in the world, `shares-border` and `official-language` relations had less than 50 subjects in the dev. The no. of subjects in each data split is in col 3-5, and the cardinality of each relation—[min, max] no. of objects—is in col 6-9.

be the prediction list of object-entities for a test subject-entity and $\mathcal{GT}$ be it's corresponding ground-truth list of object-entities, then the metrics are calculated as follows:

$$\text{Precision} = \frac{\mathcal{P} \cap \mathcal{GT}}{|\mathcal{P}|} \qquad \text{Recall} = \frac{\mathcal{P} \cap \mathcal{GT}}{|\mathcal{GT}|} \qquad F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

When $\mathcal{P}$ is empty, and $\mathcal{GT}$ is not, precision $= 1$ and recall $= 0$, leading to $F_1 = 0$. On the other hand, when $\mathcal{GT}$ is empty, recall $= 1$ but precision $= 1$ only when $\mathcal{P}$ is empty, else precision $= 0$, leading to either 1 or 0 F_1-score. Scores were macro-averaged across subjects, and across relations, and systems were ranked by the final macro-F_1-score. Participants could submit their system predictions on CodaLab at https://codalab.lisn.upsaclay.fr/competitions/5815 to get scores on the private test dataset, and check their submission ranking on the leaderboard.

To ease participation, we released a baseline implementation that probed the BERT language model using one sample prompt per relation, like "China shares border with `[MASK]`", and selected the object-entities predicted in the `[MASK]` position with greater than or equal to 0.5 likelihood as outputs. This baseline achieved 31.08% F_1-score on the hidden test dataset. We also submitted a second baseline on CodaLab, where the predictions list $\mathcal{P}$ for all test instances was empty. This baseline achieve 18% F_1-score, highlighting that predicting nothing is also a plausible baseline, with non-zero F_1 scores, since in realistic KBC scenarios subjects without objects do occur. We also released a Juptyer Notebook for getting started at https://github.com/lm-kbc/dataset/blob/main/getting_started.ipynb, where the baseline is explained, and modularized.

2. System Submissions

The challenge received five submissions—four based on the BERT model (track 1) and one based on GPT-3 (track 2). Below we list the contributions and main insights of each participating system.

[Track 1 Winner]: Task-specific Pre-training and Prompt Decomposition for Knowledge Graph Population with Language Models

Tianyi Li, Wenyu Huang, Nikos Papasarantopoulos, Pavlos Vougiouklis, Jeff.Z Pan

The authors present a system that performed task-specific pre-training of BERT, employed prompt decomposition for progressive generation of candidate objects, and use adaptive thresholds for final candidate object selection. They collected additional knowledge triples from Wikidata KB and further pre-trained BERT on the masked token prediction objective. They formulated the input as a cloze-style prompt and masked the object-entity, ensuring that the model knows what to recover during prediction. In this modified pre-training step, they also experimented with additionally masking tokens (window size of 1 or 2) appearing in vicinity of the object-entity; however, this did not lead to a gain in the overall performance. They also showed that task-specific pre-training of BERT specific to a relation performed better than pre-training for all relations.

Following Jiang et al. [25], they mined prompts from Wikipedia, and used the top-20 retrieved sentences as potential prompts. These top-20 prompted where used in an ensemble fashion with averaged voting for the final object-entity prediction. For the `shares-border` relation with subject-entities as state-type, they proposed a pre-condition prompt, "`[SUBJECT]`, as a place, is a `[MASK]`", which generated the exact type (state, province , department, city or region) for the subject-entity leading to higher gain in performance.

Finally, for candidate selection, they proposed sticky thresholds, which essentially selected a candidate in the ranked list if its likelihood was at least 80% of the previous candidate's likelihood. This system scored 55.01% F_1-score on the private test dataset, and won track 1 of the challenge. The code for this system is available at github.com/Teddy-Li/LMKBC-Track1.

[Track 2 Winner]: Prompting as Probing: Using Language Models for Knowledge Base Construction

Dimitrios Alivanistos, Selene Baez Santamaria, Michael Cochez, Jan-Christoph Kalo, Thiviyan Thanapalasingam, Emile van Krieken

The authors present the Prompting as Probing (ProP) system, which probes the GPT-3 model under few-shot setting for KB construction. ProP combines various prompting techniques including careful manual prompt creation and question style prompts for checking the veracity of GPT-3 generated claims. Since the GPT-3 model performs well with in-context examples illustrating the task, ProP system uses four representative examples from the training set for each relation, and allows the model to generate after the subject entity of interest is mentioned in the end.

Their context examples had the following properties: 1) answer sets of varying length was used to force the model to generate multiple objects; 2) subjects with empty answer set was given whenever possible; 3) examples formulated as question-answer pairs, e.g., "Which countries neighbour Dominica? ['Venezuela']", to enforce learning the task style; 4) answer list formatted as a list to accurately post-process the generations. Following Jung et al. [26], ProP has a post-processing step called fact probing, which checks the veracity of GPT-3 generated answers. In fact probing, they probe GPT-3 by converting the previous generations into natural language fact prompt, and ask the model to further generate either *True* or *False*, leading to a high gain in performance.

Finally, they also experimented with GPT-3 models differing in size (Ada < Babbage < Curie < Da-vinci), and found an analogous increase in performance as the size increased. ProP won track 2 of the challenge, with an F_1-score of 67.56 % on the private test dataset. Their code is available at github.com/HEmile/iswc-challenge.

Knowledge Base Construction from Pre-trained Language Models by Prompt learning
Xiao Ning, Remzi Celebi
The authors used manual prompts, designed based on three automated sources, and also tried ensemble learning for generating the final predictions. The descriptive information from Wikidata is used in the following three ways for designing prompts: 1) "middle-word" strategy, which selects the words occurring between subject and object as a prompt, 2) "dependency-based", which uses the syntactic structure or dependency path of the description as the prompt, and 3) "paraphrasing-based", where the original prompts are paraphrased using semantically similar expressions. Each of these prompts are used to probe the BERT-large model, and for a given subject, the five most frequent and likely objects are selected from the ensemble. Before selecting the top-5 objects, the candidate list is post-processed by removing stopwords. They also treated the threshold for candidate selection as a hyper-parameter and tuned it on the train dataset for each relation separately. This system obtained 49.35 % F_1-score on the private test dataset. Their code is available at github.com/xiao-nx/LMKBC_2022.

Prompt Design and Answer Processing for Knowledge Base Construction from Pre-trained Language Models (KBC-LM)
Xiao Fang, Alex Kalinowski, Haoran Zhao, Ziao You, Huhao Zhang, Yuan An
The authors propose manual prompts for each relation and probe the BERT-large model. They used semantics and domain knowledge of each relation to craft the prompts carefully. Uniquely, they used the intuition behind word co-occurrences in a context to design the prompt for `place-of-death` and `cause-of-death` relations. The system first checked the relative likelihoods of *dead* or *alive* tokens using a question prompt, "`[SUBJECT]` (is|has) `[MASK]`", and then probed the model for original relation only when the *dead* token had a higher probability. This simple and intuitive idea led to an overall gain in performance. For `plays-instrument` relation, authors observed that changing the article from 'an' to 'a' in the prompt improved the performance, although 'an' was grammatically correct. Similarly, even for other relations,

UserID	Paper	Track	Precision	Recall	F$_1$-score
doctor_who	Alivanistos et al.	2	79.84 %	69.00 %	67.56 %
Teddy487	Li et al.	1	76.55 %	56.58 %	55.01 %
Xiao	Ning and Celebi	1	69.34 %	50.84 %	49.35 %
xf49	Fang et al.	1	73.43 %	53.29 %	49.27 %
anonuser123			66.44 %	47.42 %	45.63 %
SumitDalal	Dalal et al.	1	63.07 %	43.82 %	33.71 %
abhiseksharma			64.82 %	43.71 %	33.69 %
baseline-1			96.00 %	31.65 %	31.08 %
chitrank			35.75 %	53.82 %	27.96 %
baseline-2			100.00 %	18.00 %	18.00 %

Table 3
Challenge leaderboard showcasing the final scores on the test dataset.

authors tried to reason out the relationship between subject and objects in question for optimal prompt design. They achieved 49.27 % F_1-score on the private test dataset. Their code is available at github.com/anyuanay/KBC-LM-Drexel.

Manual Prompt Generation For Language Model Probing
Sumit Dalal, Abhisek Sharma, Sarika Jain, Mayank Dave
The authors experiment with various manual prompts and thresholds for candidate selection for each relation while probing the BERT model. Notably, they also check if selecting more candidate in the object list (100, 150, 180, or 200) has an effect on the overall performance. They also created an ensemble of their manually crafted prompts, finally achieving an F_1-score of 33.7 % on the private test dataset.

3. Discussion

The first edition of our *LM-KBC* challenge received encouraging uptake, with five teams going past the finish line and submitting both code and system descriptions. Table 3 presents the final leaderboard of our challenge.

3.1. Main Observations

The main findings across all the submissions towards KB construction using existing language models are:

1. **Designing optimal prompts is crucial for effective knowledge elicitation from LMs.** The majority of the submissions focused on manual prompt engineering, tuning them using domain knowledge and training data. Prompt choices, sometimes even just based on small syntactic variations, had a major impact on overall system performance, and all teams reported that variations there gave huge gains in evaluation metrics.
2. **Relation specific tuning of LMs leads to better performance** compared to iteratively tuning a single LM on all relations. This may appear surprising insofar as language models

are generally held to be multi-task learners. Still, it may be explained by the significant topical and distributional differences between relations, where transfer of learning results was not beneficial.

3. **Relation-specific thresholding is necessary.** Given that LMs heavily rely on word co-occurrences and patterns during training stage, LM's confidence score highly varied for object-entity prediction and a fixed threshold across all relations for candidate selection is inadequate.
4. **Subjects without objects are challenging**, and few systems identified them at high accuracy. For example, even the best-performing system incorrectly predicted some object for 10% of those subjects. Further research on how to identify whether objects exist for a given subject-relation pair at all appears necessary.

3.2. Challenge Extensions

Deciding on the challenge complexity required navigating a trade-off between ease of access, and realism. Several avenues for extension are:

1. **Including entity disambiguation:** We consciously decided not to require resolution to specific entity identifiers, but to match only on String labels, in order to keep the challenge pure (not require pipelined systems). Yet this also creates some challenges in evaluation, such as when lists of aliases are long, or labels are ambiguous (e.g., should *Korea* be accepted as correct as birth place for someone born in South Korea?). Evaluating systems on disambiguated identifiers is a possible extension, for example, by using an entity-aware LM as default [27].
2. **Expanding training data size:** The LM-KBC22 dataset contains 100 samples per relation, which is too little for most supervised approaches. Providing more training data could open the challenge to more machine-learning centric approaches.
3. **Other metrics:** Our evaluation focused on macro-averaged F_1-scores, which give equal weight to precision and recall. It might be interesting to explore other trade-offs, as for KBs, precision often is way more critical than recall. Also, as subjects with no objects dominate many domains (e.g., very few people hold political offices), a higher presence, or more weight on no-object-subjects, might be interesting.

4. Reviewing Process

All papers received 2-3 single-blind peer reviews. The following researchers contributed reviews:

1. Emile van Krieken, Vrije Universiteit Amsterdam, Netherlands
2. Dimitrios Alivanistos, Vrije Universiteit Amsterdam, Netherlands
3. Tianyi Li, University of Edinburgh, UK
4. Yuan An, Drexel University, USA
5. Xiao Ning, Southeast University, China
6. Abhisek Sharma, National Institute of Technology Kurukshetra, India
7. Sneha Singhania, Max Planck Institute for Informatics, Germany
8. Simon Razniewski, Max Planck Institute for Informatics, Germany

5. Acknowledgements

We thank the semantic web challenge chairs, Catia Pesquita and Daniele Dell'Aglio, for helping us host a successful first edition of our challenge. We very much appreciate all the effort by the PC members and thank them for their timely and detailed reviews. Finally, we thank the participating teams for their enthusiasm and contributions.

References

[1] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: EMNLP-IJCNLP, 2019, pp. 2463–2473.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019, pp. 4171–4186.

[3] T. Brown, et al., Language models are few-shot learners, in: neurIPS, 2020, pp. 1877–1901.

[4] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM (2014) 78–85.

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: ISWC, 2007, pp. 722–735.

[6] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: WWW, 2007, p. 697–706.

[7] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: AAAI, 2017, p. 4444–4451.

[8] N. Nakashole, M. Theobald, G. Weikum, Scalable knowledge harvesting with high precision and high recall, in: WSDM, 2011, p. 227–236.

[9] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: KDD, 2014, p. 601–610.

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: SIGMOD, 2008, p. 1247–1250.

[11] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, G. Weikum, Yago2: Exploring and querying world knowledge in time, space, context, and many languages, in: WWW, 2011, p. 229–232.

[12] K. Guu, K. Lee, Z. Tung, P. Pasupat, M. Chang, Retrieval augmented language model pre-training, in: ICML, 2020, pp. 3929–3938.

[13] A. Roberts, C. Raffel, N. Shazeer, How much knowledge can you pack into the parameters of a language model?, in: EMNLP, Online, 2020, pp. 5418–5426.

[14] S. Hao, B. Tan, K. Tang, H. Zhang, E. P. Xing, Z. Hu, Bertnet: Harvesting knowledge graphs from pretrained language models, 2022.

[15] H. Arnaout, T.-K. Tran, D. Stepanova, M. H. Gad-Elrab, S. Razniewski, G. Weikum, Utilizing language model probes for knowledge graph repair, in: Wiki Workshop 2022, 2022.

[16] T. McCoy, E. Pavlick, T. Linzen, Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference, in: ACL, 2019, pp. 3428–3448.

[17] N. Kassner, H. Schütze, Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly, in: ACL, 2020, pp. 7811–7818.

[18] S. Razniewski, A. Yates, N. Kassner, G. Weikum, Language models as or for knowledge bases, DL4KG (2021).

[19] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, J. Xu, Knowledgeable or educated guess? revisiting language models as knowledge bases, in: ACL, 2021, pp. 1860–1874.

[20] T.-P. Nguyen, S. Razniewski, Materialized knowledge bases from commonsense transformers, CSRR (2022).

[21] S. Razniewski, F. Suchanek, W. Nutt, But what do we actually know?, in: AKBC, 2016, pp. 40–44.

[22] S. Singhania, S. Razniewski, G. Weikum, Predicting document coverage for relation extraction, TACL (2022).

[23] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, AutoPrompt: Eliciting knowledge from language models with automatically generated prompts, in: EMNLP, 2020, pp. 4222–4235.

[24] Z. Zhong, D. Friedman, D. Chen, Factual probing is [MASK]: Learning vs. learning to recall, in: NAACL, 2021, pp. 5017–5033.

[25] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know?, TACL (2020) 423–438.

[26] J. Jung, L. Qin, S. Welleck, F. Brahman, C. Bhagavatula, R. L. Bras, Y. Choi, Maieutic prompting: Logically consistent reasoning with recursive explanations, CoRR (2022).

[27] N. De Cao, G. Izacard, S. Riedel, F. Petroni, Autoregressive entity retrieval, in: ICLR, 2020.