



Evaluating Language Models for Knowledge Base Completion

Blerta Veseli¹(✉), Sneha Singhania¹, Simon Razniewski²(✉),
and Gerhard Weikum¹

¹ Max Planck Institute for Informatics, Saarbrücken, Germany
`bveseli@mpi-inf.mpg.de`

² Bosch Center for AI, Renningen, Germany
`srazniew@mpi-inf.mpg.de`

Abstract. Structured knowledge bases (KBs) are a foundation of many intelligent applications, yet are notoriously incomplete. Language models (LMs) have recently been proposed for unsupervised knowledge base completion (KBC), yet, despite encouraging initial results, questions regarding their suitability remain open. Existing evaluations often fall short because they only evaluate on popular subjects, or sample already existing facts from KBs. In this work, we introduce a novel, more challenging benchmark dataset, and a methodology tailored for a realistic assessment of the KBC potential of LMs. For automated assessment, we curate a dataset called WD-KNOWN, which provides an unbiased random sample of Wikidata, containing over 3.9 million facts. In a second step, we perform a human evaluation on predictions that are not yet in the KB, as only this provides real insights into the added value over existing KBs. Our key finding is that biases in dataset conception of previous benchmarks lead to a systematic overestimate of LM performance for KBC. However, our results also reveal strong areas of LMs. We could, for example, perform a significant completion of Wikidata on the relations `nativeLanguage`, by a factor of ~ 21 (from 260k to 5.8M) at 82% precision, and `citizenOf` by a factor of ~ 0.3 (from 4.2M to 5.3M) at 90% precision. Moreover, we find that LMs possess surprisingly strong generalization capabilities: even on relations where most facts were not directly observed in LM training, prediction quality can be high. We open-source the benchmark dataset and code. (<https://github.com/bveseli/LMsForKBC>).

1 Introduction

Structured knowledge bases (KBs) like Wikidata [26], DBpedia [1], and Yago [25] are backbones of the semantic web and are employed in many knowledge-centric applications like search, question answering and dialogue. Constructing these KBs at high quality and scale is a long-standing research challenge and multiple knowledge base construction benchmarks exist, e.g., FB15k [2], CoDEX [21], and LM-KBC22 [24]. Text-extraction, knowledge graph embeddings, and LM-based

knowledge extraction have continuously moved scores upwards on these tasks, and leaderboard portals like Paperswithcode¹ provide evidence to that.

Recently, pre-trained language models have been purported as a promising source for structured knowledge. Starting from the seminal LAMA paper [17], a thrive of works have explored how to better probe, train, or fine-tune these LMs [12]. Nonetheless, we observe a certain divide between these late-breaking investigations, and practical knowledge base completion (KBC). While the recent LM-based approaches often focus on attractive and clean methodologies that produce fast results, practical KBC is a highly precision-oriented, extremely laborious process, involving a very high degree of manual labor, either for manually creating statements [26], or for building comprehensive scraping, cleaning, validation, and normalization pipelines [1, 25]. We believe previous works fall short in three aspects:

1. *Focus on high precision:* On the KB side, part of Yago’s success stems from its validated >95% accuracy, and the Google Knowledge Vault was not deployed into production, partly because it did not achieve 99% accuracy [27]. Yet, previous LM analyses balance precision and recall or report precision/hits@k values, implicitly tuning systems towards balanced recall scores resulting in impractical precision.
2. *Evaluation of completion potential:* Existing benchmarks often sample from popular subjects. This is useful for system comparison, but not for KBC. E.g., predicting capitals of countries with 99% accuracy does not imply practical value: they are already captured in established KBs like Wikidata.
3. *Prediction of missing facts:* Existing works test LMs on facts already stored in KBs, which does not provide us with a realistic assessment for completion. For KBC we need to predict objects for subject-relation pairs, previously not known to the KB.

It is also important to keep in mind the scale of KBs: Wikidata, for instance, currently contains around 100 Million entities, and 1.2B statements. The cost of producing such KBs is massive, one estimate from 2018 sets the cost per triple at 2 USD for manually curated triples [15], and 1 ct for automatically extracted ones.² Thus, even small additions in relative terms, might correspond to massive gains in absolute numbers. For example, even by the lower estimate of 1 ct/triple, adding one triple to just 1% of Wikidata humans, would come at a cost of 100k USD.

In this paper, we conduct a systematic analysis of LMs for KBC, where we focus on *high precision ranges* (90%). We evaluate by first using a new benchmark dataset WD-KNOWN, where we randomly sample facts from Wikidata (including many without any structured information, Wikipedia information, or English labels) and second by a manual evaluation on subject-relation pairs without object values, yet.

¹ <https://paperswithcode.com/task/knowledge-graph-completion>.

² Wikidata might broadly fall in between, as its aim is human-curated quality, but major portions are imported semi-automatically from other sources.

Technically, we focus on the BERT language model [7], and the Wikidata knowledge base. Although BERT has been superseded by newer LMs, its popularity is still matched only by the closed source GPT-3 and chatGPT models. Wikidata is by far the most prominent and comprehensive public KB, so evaluations against it provide the strongest yardstick.

Our main results are as follows:

1. In actual KBC, LMs perform considerably worse than benchmarks like LAMA indicated, but still achieve strong results for language-related, socio-demographic relations (e.g., *nativeLanguage*).
2. Simple changes on out-of-the-box LMs, in particular, vocabulary expansion and prompt formulation, can significantly improve their ability to output high-precision knowledge.
3. Using LMs, Wikidata could be significantly expanded for three relations, **nativeLanguage** by a factor of ~ 21 (from 260k to 5.8M) at precision 82%, **citizenOf** by a factor of ~ 0.3 (from 4.2M to 5.3M) at 90% precision and **usedLanguage** by a factor of ~ 2.08 (from 2.1M to 6.6M) at 82% precision.

2 Background and Related Work

KB Construction and Completion. Knowledge base construction is a field with considerable history. One prominent approach is by human curation, as done e.g., in the seminal CYC project [11], and this is also the backbone of today’s most prominent public KB, Wikidata [26]. Another popular paradigm is the extraction from semi-structured resources, as pursued in Yago and DBpedia [1, 25]. Extraction from free text has also been explored (e.g., NELL [4]). A popular paradigm has been embedding-based link prediction, e.g., via tensor factorization like Rescal [14], and KG embeddings like TransE [2].

An inherent design decision in KBC is the P/R trade-off – academic projects are often open to trade these freely (e.g., via F-1 scores), while production environments are often very critically concerned with precision, e.g., Wikidata generally discouraging statistical inferences³, and industrial players likely using to a considerable degree human editing and verification [27]. Although the P/R trade-off is in principle tunable via thresholds, the high-precision range is hardly investigated. For example in all of Rescal, TransE, and LAMA, the main results focus on metrics like hits@k, MRR, or AUC, which provide no bounds on precision.

LMs for KB Construction. Knowledge extraction from language models provides fresh hope for the synergy of automated approaches and high-precision curated KBs. Knowledge-extraction from LMs provides remarkably straightforward access to very large text corpora: The basic idea by [17] is to just define one template per relation, then query the LM with subject-instantiated versions, and

³ There is often a terminological confusion here: Automated editing is omnipresent on Wikidata, but the bots performing them typically execute meticulously pre-defined edit and insertion tasks (e.g., based on other structured sources), not based on statistical inference.

retain its top prediction(s). A range of follow-up works appeared, focusing, e.g., on investigating entities, improving updates, exploring storage limits, incorporating unique entity identifiers, and others [3, 5, 8, 10, 16, 18–20, 23]. Nonetheless, we observe the same gaps as above: The high-precision area, and real KBC, are underexplored.

Benchmarking KB Completion. KB completion (sometimes also referred to as link prediction) has a set of quasi-standard benchmarks. Here we review the important ones and outline why they do not help for the focus of our investigation.

Two of the most popular benchmarks, both introduced in [2], are **FB15k** and **WN18**. The former contains statements for 15k extremely popular entities from Freebase, entities that already in 2013, when KBs were small, had at least 100 statements. The latter contains 41k entities from WordNet, yet these are linguistic concepts where WordNet is already the authoritative reference, and the potential for completion is small. **DBP-5L** is a popular multilingual dataset of 14k entities, yet it is collected by iterative graph neighbourhood expansion, so by design is biased towards popular (well-connected) entities. **YAGO3-10** is another benchmark, that contains YAGO3 entities with at least 10 statements [6]. The recent **CoDEX** benchmark provides a much larger subset of Wikidata triples but again focuses on the more popular subjects, as even its hardest variant considers only entities with at least 5 statements [21]. The **LAMA** benchmark [16] is based on the T-REx dataset, which in turn restricts the scope of subjects to those having a Wikipedia page. **LAMA-UHN** [18] removes some surface correlations but does not remove the restriction to Wikipedia-known subjects. **LM-KBC22** [24] provides a curated mix of popular and long-tail subjects, but not a random sample, and only a small set of 12 relations. In summary, all these benchmarks provide non-random subject sets, and by taking ground truth from existing KBs, necessarily can not evaluate a method’s real KB completion potential. The **PKGC** work [13] uses human evaluation to account for KB incompleteness, but also uses subjects from previous benchmarks focused on popular entities.

3 Method

3.1 Knowledge Base Completion Tasks

Established KBs like Wikidata store a large number of facts of the form (*subject, relation, object*). However, such KBs still suffer from incompleteness, which limits their applicability. KBC tries to counteract this incompleteness and describes the task of predicting missing facts for a KB. KBC is often split into subtasks, such as predicting the relation, subject, or object slots in triples. In the following, we focus on the most prominent object slot filling task, i.e. predicting an object for a subject-relation pair without an object so far. Identifying plausible subject-relation pairs is another important task, as not every combination qualifies, e.g., (Albert Einstein, hasCapital, ·) has no object.

We will refer to facts that are present in a KB as *existing facts* and *existing-fact prediction* describes predicting the object for a subject-relation pair for which the object value already exists in the KB. Similarly, we refer to facts

that are missing in a KB as *missing facts* and *missing-fact prediction* describes predicting the object for a subject-relation pair with no object value yet.

3.2 Fact Prediction Using Pre-trained Language Models

The slot filling ability of an LM, i.e. predicting “Paris” for a pair (France, has-Capital), is essential for KBC. This is done by querying an LM using cloze-style statements, a.k.a. prompts, like “The capital of France is [MASK].” [17]. The LM’s goal is to predict the token at the masked position, i.e. the object (Fig. 1).

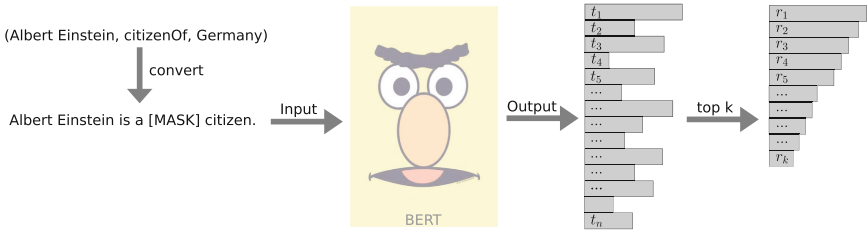


Fig. 1. To query the LM for an object slot, we convert triples into relation-specific prompts by masking the object, following [17]. The output is a probability vector over the model’s vocabulary. We retain the top $k = 10$ predictions.

We probe the masked language model BERT [7] and query for facts of different relations by using relation-specific predefined prompts. The prompts contain placeholders for subject and object, so that the input for the LM can be automatically created by replacing the subject placeholder with the actual subject and the object placeholder with [MASK]. For each cloze-style query like “The capital of France is [MASK].”, BERT returns a probability vector over its vocabulary $[t_1, \dots, t_n]$ ($\sim 29K$ tokens). From this vector, we select *top k* predictions $[r_1, \dots, r_k]$ with the highest probability as predictions for the object. We set $k = 10$.

3.3 Systematic Analysis Procedure

The goal of our analysis is to realistically assess the abilities of an LM for KBC. Therefore we perform a two-fold analysis by investigating 1) the existing-fact prediction ability of BERT in an automated evaluation and 2) the potential for KBC using LMs by predicting missing facts and evaluating the LM’s predictions using human annotations.

The first analysis part includes the automated evaluation of existing facts in WD-KNOWN compared to the LAMA-T-REx benchmark in order to get a realistic estimate of an LM’s prediction ability. Based on this evaluation we will extract relation-specific prediction thresholds considering precision and recall trade-offs to enable KBC on high precision and reasonable recall.

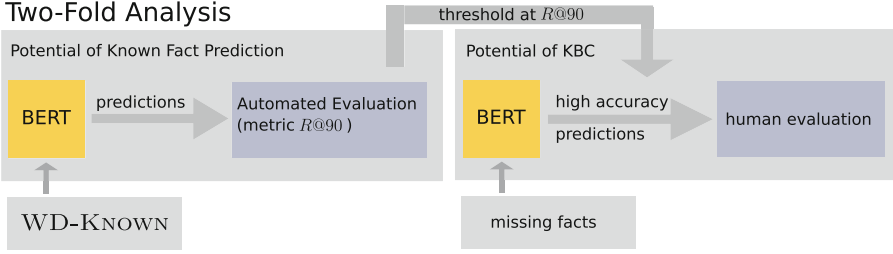


Fig. 2. Our systematic analysis is divided into two parts. First, we analyse existing-fact prediction by an automated evaluation computing the recall achieved at 90% precision ($R@P90$). The prediction at $R@P90$ is used as a threshold in the second part of the analysis. We analyse the potential of KBC for missing facts via a manual evaluation.

In the second analysis part, we produce high accuracy predictions for missing facts in Wikidata given the previously extracted threshold and evaluate the model’s predictions using human annotations. Given the human evaluation, we will provide estimations about the amount of addable new facts to Wikidata. In Fig. 2 we show an overview of our systematic analysis.

4 Datasets

4.1 LAMA-T-REx

The LAMA-T-REx dataset [17] is derived from T-REx dataset [9], which is a subset of Wikidata and provides alignment between Wikidata (WD) triples and Wikipedia texts. LAMA-T-REx consists of 41 relations with ~ 1000 facts per relation sampled from T-REx. Included with the dataset are relation-specific, manually defined prompts, which [17] also refers to as templates. We use these 41 relations and their corresponding templates throughout our work. The scope of subjects in LAMA-T-REx is restricted to having a Wikipedia page and contains little data as shown in Table 1. This makes it difficult to realistically assess LMs for KBs.

4.2 WD-KNOWN

To realistically assess the usability of LMs for KBC, we created a large-scale dataset with random facts from Wikidata, without subject restrictions to pages such as Wikipedia. One must observe that, while WD-KNOWN is an unbiased sample from Wikidata’s subject set per class, it is still biased towards reality, like Wikidata itself [22].

Creation. We extract facts from Wikidata for the same 41 relations as in LAMA-T-REx. We extract subject-relations pairs per relation and because these pairs may have multiple valid objects, e.g. N:M relations, we extract all associated valid objects along with the pairs. Otherwise, we would risk incomplete

ground truth data. This would falsify our evaluation as an LM’s prediction can not be recognized as correct when predicting another valid object than the extracted one. We sampled a maximum of 100,000 subject-relation pairs per relation along with all valid objects. If a relation contains fewer than 100,000 pairs, we extract all of them. The extracted facts consist of Wikidata-specific entity IDs, which we converted into natural language labels to allow probing an LM for facts. In contrast to [17], we do not remove multi-token objects like “Ball Aerospace & Technologies” because the inability to predict such objects is part of (some) LM’s limitations and at the time KBC is performed it is unknown which objects are multi-token objects and which are not as KBC includes predicting missing facts, i.e. facts without ground truth object yet. This dataset feature enables the testing of LMs with multi-token prediction capability in comparison to uni-token predictions for KBC. Additionally, a large dataset like WD-KNOWN enables fine-tuning for fact prediction. In Table 1 we report some dataset statistics in comparison to LAMA-T-REx showing the larger size of WD-KNOWN. The dataset is available at <https://github.com/bveseli/LMsForKBC>.

Table 1. Our WD-KNOWN dataset in comparison to LAMA-T-REx. We report the total number of distinct objects (#unique objects), distinct subjects (#unique subjects), and the number of triples (#triples) as well as the total number of objects consisting of more than one token (#multi-token objects), and the average object entropy.

Dataset		#unique subjects	#unique objects	#triple	#multi-token objects	object dist. entropy
LAMA-T-REx	total	31,479	61,85	34,039	0	–
	average	767	150	830	0	0.76
WD-KNOWN	total	2,956,666	709,327	3,992,386	1,892,495	–
	average	72,113	17,300	97,375	46,158	0.67

5 Potential for Existing-Fact Prediction

Existing-fact prediction describes the prediction of an object for a subject-relation pair for which the ground truth object already exists in a KB. We will analyse the prediction ability of BERT given existing facts from Wikidata in an automated evaluation, focusing on the recall achieved at 90% precision ($R@P90$).

5.1 Metric

We use a rank-based metric and calculate the recall that our method achieves at 90% precision ($R@P90$). To compute $R@P90$, we sort the predictions in descending order of their prediction probability, and threshold predictions when average precision drops below 90%. When determining which prediction is true or false, we have to consider that a subject-relation pair can be associated with multiple valid objects. Therefore, a prediction is true, when it is among the valid objects and false otherwise.

5.2 Baselines

We want to check if BERT’s fact prediction ability goes at least beyond just predicting statistically common objects, e.g. English for `spokenLanguage`. Therefore, we try two distribution-based baselines: random and majority vote. We compute relation-specific object distributions based on a relations ground truth data. In the case of a majority vote, we assign the most probable object to each fact and in the case of random, we assign a random object from the distribution. Additionally, we compare BERT to a relation extraction (RE) baseline and use the *Rosette Text Analytics*⁴ relation extractor. Given a text snippet, the relation extractor can extract up to 17 relations. For the intersection of these 17 and our 41 relations, we subsampled relation-wise 200 facts from WD-KNOWN and align each fact with text. For text alignments, we consider two different source types: web texts and Wikipedia pages to cover facts with Wikipedia-unknown and Wikipedia-known subjects. Per relation, we align 100 facts with web texts, i.e. top 10 Google search results after googling the subject of a fact and 100 facts with wikipages, i.e. the summary of a subject’s Wikipedia page. We get 200 facts with text alignments per relation `headquarteredIn` and `citizenOf`, 400 in total.

5.3 Evaluation and Results

Quantitative Analysis. In Table 2 we report $R@P90$ values achieved by BERT’s predictions on our dataset WD-KNOWN in comparison to *LAMA-T-REx*. On WD-KNOWN the LM achieves significantly lower values (marked in bold) suggesting a more realistic assessment of BERT’s fact prediction ability by WD-KNOWN. Only the relations `nativeLanguage`, `spokenLanguage`, `officialLanguage` show a smaller decrease and therefore stable results.

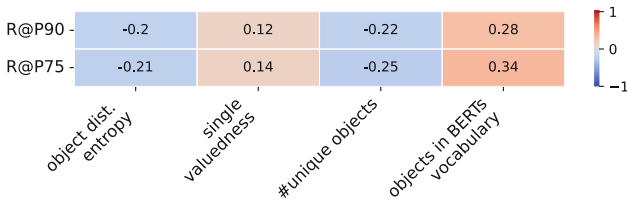


Fig. 3. Pearson Correlation Analysis

To investigate why BERT achieves low results, we perform a correlation analysis computing the Pearson correlation coefficient over $R@P90$ in Fig. 3. We notice a negative correlation between $R@P90$ and the object distribution entropy, meaning that a more uniform distribution makes the predictions harder. Furthermore, the number of unique objects is also negatively correlated with

⁴ <https://www.rosette.com/capability/relationship-extractor/#tech-specs>.

Table 2. BERT’s performance on data sets WD-KNOWN a LAMA-T-REx for the same 41 relations. Boldface marks significantly lower values, indicating an overestimation of BERT’s fact predicting ability on LAMA.

Relation	WD-KNOWN <i>R@P90</i>	LAMA-T-REx <i>R@P90</i>	Relation	WD-KNOWN <i>R@P90</i>	LAMA-T-REx <i>R@P90</i>
nativeLanguage	0.61	0.68	foundedIn	0.00009	0.001
spokenLanguage	0.26	0.33	deathPlace	0.00009	0
officialLanguage	0.25	0.37	namedAfter	0.00008	0
headquarteredIn	0.04	0.52	partOf	0.00006	0
developedBy	0.04	0.64	twinTown	0.00003	0.001
producedBy	0.03	0.86	sharesBorders	0.00001	0.001
countryOfJurisdiction	0.02	0.66	fieldOfWork	0	0
hasCapital	0.01	0.60	employedBy	0	0.002
locatedIn	0.008	0.20	hasReligion	0	0
bornIn	0.006	0.009	playerPosition	0	0
isCapital	0.006	0.81	subclassOf	0	0.01
CountryOfOrigin	0.005	0.08	holdsPosition	0	0
isA	0.004	0.06	diplomaticRelation	0	0
LanguageOfFilm	0.003	0	citizenOf	0	0
ownedBy	0.0008	0.16	consistsOf	0	0
hostCountry	0.0006	0.002	musicGenre	0	0
originalBroadcaster	0.0004	0.02	musicLabel	0	0.02
inTerritoryOf	0.0002	0.01	playsInstrument	0	0.003
writtenIn	0.0001	0.15	hasProfession	0	0
locationOfWork	0.0001	0	inContinent	0	0.004
memberOf	0.0001	0.52			

BERT’s performance, i.e. fewer possible objects benefit the performance. The single valuedness, i.e. the relation-wise proportion of subject-relation pairs with only one object, shows a low but positive correlation. This indicates the performance is better on N:1 or 1:1 relations confirming the observation in [17]. The performance is also positively correlated with the number of objects in BERT’s vocabulary, i.e. the more objects of a relation are covered by the LM’s vocabulary, the better the performance. The vocabulary acts as a bottleneck, preventing the model from predicting facts.

Looking at the baselines in Table 3, we see that the majority baseline is quite solid with an average precision of 0.18. Access to the underlying distribution of relation-specific ground truth data has a noticeable impact on assigning objects to a subject-relation pair. Still, BERT achieving an average precision in a higher range (>75%) shows that the prediction ability is based on more than predicting statistically common objects. The RE baseline is outperformed by BERT for two tested relations, while BERT and RE show lower results on facts aligned with webtexts than on facts aligned with Wikipedia pages.

Qualitative Analysis. Since we are interested in high precision for KB completion and quantitative analysis showed low R@P90 values for most of the relations, we need to increase BERT’s performance. We first perform a qualitative analysis of issues. Since analysing all 41 relations qualitatively is not feasible, we select a representative subset of relations. The subset is diverse regarding, e.g. semantics (e.g. language, demographic, goods), entity type (human, com-

Table 3. Random and Majority Baselines (left) and Relation Extraction (RE) Baseline for **citizenOf** and **headquarteredIn** (right). The RE Baseline is done on two datasets: 1) a (Wikipedia) dataset, where the triples are aligned with the Wikipedia summaries extracted from the subject’s Wikipedia pages, and 2) a (Google search) dataset, where the triples are aligned with texts from the top 10 Google search results after searching for the subject of the respective triple. Scores were computed on a subset of WD-KNOWN with text alignments as described in Sect. 5.2.

Distribution vs. BERT				RE vs. BERT						
	\bar{P}	\bar{R}	$F1$		Wikipedia			Google search		
					\bar{P}	\bar{R}	$F1$	\bar{P}	\bar{R}	$F1$
Random	0.09	0.05	0.06	Rosette	0.32	0.05	0.08	0.21	0.01	0.02
Majority	0.18	0.09	0.03	BERT @P75	0.87	0.31	0.45	0.37	0.28	0.31
BERT @P75	0.75	0.43	0.54	BERT @P90	0.95	0.22	0.17	0.45	0.23	0.30
BERT @P90	0.90	0.35	0.48							

pany, places), performance (lower vs. higher scores), and possible objects (all languages vs. cities worldwide). The chosen relations are shown in Table 4. We aim to identify and eliminate or at least reduce systematic prediction errors for those relations. Common error categories are: 1) predicting the wrong hierarchy, i.e. country instead of a city; 2) predicting the wrong category, i.e. country instead of nationality and 3) ambiguous prompts, i.e. predicting “local” for “Albert Einstein is a [MASK] citizen”. Such cases falsify the evaluation, since a distinction between actually true predictions (“German” or “Germany”) and actually false predictions (“local” or “German”) is not made. These errors are rooted in the conceptual discrepancy between LMs and KBs. While querying a KB is clearly defined and leads to unambiguous answers, an LM is queried by natural language prompts, which are as ambiguous as natural language itself. To use LMs for KBs we need to translate between them. Therefore we focus on three main model parts: input, model, and output: 1) input optimization by adjusting prompts; 2) model optimization by fine-tuning, and 3) output adjustment by converting ground truth and prediction into the same category. Input and model optimizations are done on relation-wise data splits for training and test (80 – 20), where we split based on subject-relation pairs avoiding having the same subject in training and test.

Input Optimization. *AutoPrompt* [23] introduced an automatic way of prompt generation given a specific task, such as fact retrieval. We generate our prompts as suggested by [23] on relation-specific training splits of WD-KNOWN. In Table 4 the input optimization with AutoPrompts achieves notable improvements for $R@P90$, e.g. **hasReligion** increased from 0 to 0.21, **citizenOf** from 0 to 0.15. There is no deterioration in any of the relations.

Model Optimization. Fine-tuning not only allows us to optimize an LM on the searched output but also enables adding new words to the model’s vocabulary as this has been shown to be a bottleneck for fact prediction. We fine-tune relation-specific LMs and for vocabulary extension, we add a *max.* of 2000 objects per relation. For fine-tuning the masked language model objective [7] is used as well

as our training splits. We show results on two setups: 1) fine-tuning $BERT_{large}$, denoted FT , and 2) fine-tuning $BERT_{large}$ and with extended vocabulary, denoted FT_{vocab} . In Table 4 the model optimization shows the biggest improvements. Only `developedBy`, `producedBy`, `headquarteredIn` deteriorate at first in the FT setup, but `producedBy` and `developedBy` improve significantly in the FT_{vocab} setup. We found that after fine-tuning the model predicts substrings that match the ground truth object, e.g. in relation `producedBy` “Len” is predicted if the ground truth object is “Lenovo” or “Xiao” for “Xiaomi”. The same happens in relation `developedBy`, e.g. “Core” for “Corel”, and in relation `headquarteredIn` e.g. “Wind” for the city “Windhoek”. After vocabulary expansion previously missing tokens like “Xiaomi” can now be fully predicted, so that $R@P90$ can increase in the FT_{vocab} setup. It is worth noting that `producedBy` and `developedBy` could only be improved significantly by expanding the vocabulary during fine-tuning. Only for `headquarteredIn` the precision does not improve. While `headquarteredIn` does degrade in precision, fine-tuning with an extended vocabulary increases the overall quality significantly (see Table 4, AVG $R@P90$).

Output Adjustment. To map prediction and ground truth, so a prediction “French” is correctly recognized for object “France”, we use manually crafted dictionaries. Methods like string matching can always lead to incorrect mappings and sometimes even do not work for examples like the one shown. Therefore, we used two dictionaries mapping nationalities and countries on the one hand and religions and religious affiliations on the other hand. A prediction will be true if it belongs to the same entry in a relation-specific dictionary as the ground truth object and false otherwise. The creation of such mapping dictionaries involves tremendous manual labor, contradicting our search for automated KBC. Therefore, we evaluate on only two `hasReligion` and `citizenOf` as these relations were also most affected by the second error category mentioned above so that the output adjustment here might have the greatest effect. We find, that automated fine-tuning significantly outperforms this approach.

Table 4. Table shows $R@P90$ values. Improvement approaches to maximize BERT’s performance on specific relations in comparison to pre-trained BERT-large (case-sensitive). Improvements were tested at three key points in the LM: *input*, *model*, *output*. Scores were computed on the test split of WD-KNOWN.

Relation	Base	Input Opt.	Model Opt.		Output Adjustment
	Pre-Trained	AutoPrompt	FT	FT_{vocab}	Manual Mapping
<code>nativeLanguage</code>	0.62	0.66	0.79	0.79	–
<code>hasReligion</code>	0	0.21	0.13	0.27	0.02
<code>citizenOf</code>	0	0.15	0.23	0.23	0.01
<code>producedBy</code>	0.03	0.03	0	0.15	–
<code>developedBy</code>	0.04	0.04	0.0004	0.11	–
<code>headquarteredIn</code>	0.04	0.04	0	–	–
<code>spokenLanguage</code>	0.26	0.42	0.51	0.5	–
<code>LanguageOfFilm</code>	0.003	0.04	0.29	0.27	–
AVG $R@P90$	0.12	0.20	0.24	0.29	0.015

Summary. We found that using biased datasets lead to an overestimation of an LM’s performance for fact prediction. To neither over- nor underestimate BERT’s performance, we tested it on the large dataset WD-KNOWN and implemented improvements to increase BERT’s performance and perform KBC with high precision later on. We have seen the model’s vocabulary to be a bottleneck for its performance. When fine-tuning with vocabulary extension, the model’s performance can be significantly improved for fact prediction.

6 Potential for Knowledge Base Completion

In the following, we will obtain plausible missing facts from Wikidata, produce high accuracy predictions respective to $R@P90$, and let annotators from Amazon Mechanical Turk (mturk) manually evaluate the predictions given a five-value scale *true*, *plausible*, *unknown*, *implausible*, *false*. The relations in focus are the same as in Table 4, except for **hasReligion**, which appeared with too sparse information on the web.

6.1 Human Evaluation

Obtaining Missing Facts. The key to KBC with LMs is the ability to predict *missing facts*. Directly extracting empty subject-relation pairs from Wikidata is not possible, since Wikidata consists of existing facts (s, r, o) . Also, randomly sampling an arbitrary subject to combine it with any relation would run the risk of violating our condition of plausible missing facts, where an object for a subject-relation pair exists (e.g. an implausible pair like (Albert Einstein, hasCapital, -) has no object). Therefore, we will only sample subject-relation pairs, where the subject has a relation-compatible entity type. Thus, we compute relation-wise the most frequent subject entity type within a relation. When sampling subject-relation pairs with missing objects, the subject is conditioned on having the most frequent entity type. This ensures extracting plausible missing facts. We randomly sample 10,000 missing facts per relation, 70,000 missing facts in total.

High Accuracy Predictions. To provide human annotators with reasonable predictions, we set a relation-specific prediction threshold to ensure the prediction quality and use the best possible model for predictions based on results in Table 4. Given these best models, the threshold is the prediction probability at $R@P90$ of each relation to respect the relation-specific precision and recall trade-offs. We keep only those missing facts, i.e. subject-relation pairs, whose predictions have a probability over the threshold. These are our *high accuracy predictions*.

Annotations. We filter the 70,000 missing facts relation-wise by keeping only the facts with high accuracy predictions and then sample 50 missing facts with high accuracy predictions. This results in one prediction per subject-relation

pair, i.e. 350 predicted missing facts in total. For these 350 missing facts, we use Amazon Mechanical Turk (mturk) for human annotations. An annotator is asked to evaluate a predicted missing fact. For readability reasons, each fact is formulated into relation-specific statements such as “The native language of Marcus Adams is English”, where “English” is the prediction and `nativeLanguage` the relation. The annotators are given a five-value scale: true, plausible, unknown, implausible, and false. Before voting, we ask them to look up the fact on the web. They are required to give an evidence link and copy a text snippet supporting their voting. In case they vote unknown, they have to explain their voting. This way we ensure that annotators leave reasonable votes. We can say that all annotators left understandable insights regarding their voting.

We also self-annotated the 350 missing facts with ground truth and evidence links. Along with ground truth annotations, we annotated if the subject was known to English (en) Wikipedia; if the ground truth consists of more than one word; if the ground truth is in BERT’s vocabulary, the position of the found evidence in Google search results and the search link used to find the ground truth. Furthermore, we rated each prediction using the same five-value scale as mturk annotators. We and mturk annotators reach an agreement of 69% given the five-value scale and an agreement of 94% given the upper categories true (true, plausible) and false (unknown, implausible, false).

Table 5. Overview of the results from the human evaluation. “True” denotes the summed up human ratings for “true with evidence” and “plausible” per relation. Similarly, “false” denotes the combined human ratings for “unknown”, “implausible”, and “false with evidence” per relation. The column “in Wikipedia (en)” describes the ratio of subjects with English Wikipedia articles to subjects without English Wikipedia articles per relation.

Relation	true	false	true with evidence	plausible	unknown	implausible	false with evidence	in Wikipedia (en)
<code>nativeLanguage</code>	82%	18%	48%	34%	6%	8%	4%	16%
<code>spokenLanguage</code>	82%	18%	46%	36%	6%	6%	6%	28%
<code>headquarteredIn</code>	82%	18%	34%	48%	6%	8%	4%	26%
<code>developedBy</code>	62%	38%	50%	12%	0%	0%	38%	74%
<code>producedBy</code>	22%	78%	20%	2%	6%	0%	72%	10%
<code>LanguageOfFilm</code>	76%	24%	56%	20%	0%	2%	22%	32%
<code>citizenOf</code>	90%	10%	52%	38%	4%	0%	6%	8%

6.2 Results

Quantitative Results. In Table 5, we see that the human annotators rate the model’s predictions as highly correct. Based on these values we have determined the potential for KBC in Table 6. Given the number of missing facts and the proportion of high accuracy predictions, we can estimate the amount of addable facts at a relations-specific accuracy. This accuracy was achieved through human evaluation as shown in Table 5. Given the relation `nativeLanguage` we could

add 5,550,689 new facts in a human-in-a-loop procedure or $7,871,085 \cdot 0.86 = 6,769,133$ with an accuracy of 82% automatically. In a human-in-a-loop procedure, the estimated costs of 2 USD (Sect. 1) for manually curated facts could drop to 40 Cents with approximately 2 min per annotation as we experienced with our mturk evaluation. Given our results in Table 6 we can perform significant KBC for relations `nativeLanguage` and `spokenLanguage` at precision 82% and `citizenOf` at precision 90%.

Qualitative Results. Looking into the annotations and predictions, we see that statements such as “Final Fantasy VII is developed by Square” are almost literally included in corresponding evidence links such as Wikipedia pages⁵. In contrast, relations like `nativeLanguage` include statements only implicitly, e.g. the native language of “Marcus Adams” is never mentioned explicitly in the corresponding Wikipedia page⁶. Yet, the LM achieves comparable results on most relations despite their more implicit or explicit factual basis.

Generalization or Retrieval? To investigate this further, we computed the proportion of subjects with English Wikipedia articles per relation. This enables us to estimate whether facts were mentioned in corresponding Wikipedia articles and thus, were present in BERT’s training set. It can be shown that the language-related and socio-demographic relations achieve high results despite their lower occurrence in Wikipedia. This means that BERT predicts never seen facts with a high accuracy for these relations, showing a high generalization capability. When given facts such as the headquarters of “Universal de Televisión Peru”, the model correctly predicts “Lima”. Or given a fact such as the original language of the movie “Il mio paese”, the model predicts “Italian” correctly. Socio-demographic relations such as `citizenOf`, `headquarteredIn` or language-related relations like `nativeLanguage`, `LanguageOfFilm` exhibit stronger correlations (e.g. person name and spoken language/origin country) than other relations (e.g. video game and developer, goods/products and manufacturer). These correlations are learned by the LM from the vast amounts of training data and used for the prediction. We recognize here that such learned correlations can be quite useful for fact prediction. Regarding the non-language or non-socio-demographic relations, we see that `producedBy` has the least Wikipedia-known subjects of 10% and shows also the lowest accuracy of 22%. In contrast, `developedBy` has the most Wikipedia-known subjects and still a high accuracy of 62% despite being a non-language or non-socio-demographic relation. In these relations, the model shows less generalization capability and is more in need of actual retrieval. As an example: the developer of “Pro Cycling Manager 2015” must be explicitly mentioned during training to know it, yet the model correctly predicts “Cyanide”.

Conclusion. Given these qualitative examples and quantified numbers the model is capable of generalization as well as retrieval. But it is still unclear in what mixture and to what extent for example fact retrieval is possible. Regarding

⁵ https://en.Wikipedia.org/wiki/Final_Fantasy_VII.

⁶ [https://en.Wikipedia.org/wiki/Marcus_Adams_\(Canadian_football\)](https://en.Wikipedia.org/wiki/Marcus_Adams_(Canadian_football)).

KBC both are beneficial. In case of precise retrieval, facts are addable automatically to an existing KB. In the case of generalization, which still achieves a high accuracy, human-in-a-loop procedures allow adding manually curated facts in a faster and cheaper way.

Table 6. The amount of missing facts and the percentage of high accuracy predictions denotes the number of new facts we could add at a relation-specific precision. The amount of addable facts indicates the number of potential new facts that could be added without error, e.g. in a human-in-a-loop procedure. The growth factor describes the potential growth of Wikidata given the current *cardinality*^{WD} in Wikidata and the amount of addable facts.

Relation	<i>cardinality</i> ^{WD}	#missing facts	high accuracy predictions(%)	accuracy	#addable facts	growth factor
nativeLanguage	264,778	7,871,085	86%	82%	5,550,689	20.96
spokenLanguage	2,148,775	7,090,119	77%	82%	4,476,701	2.08
headquarteredIn	409,309	55,186	8%	82%	3,443	0.008
developedBy	42,379	29,349	2%	62%	363	0.01
producedBy	123,036	31,239	0.8%	22%	55	0.0004
LanguageOfFilm	337,682	70,669	37%	76%	19,872	0.06
citizenOf	4,206,684	4,616,601	28%	90%	1,163,383	0.27

7 Discussion and Conclusion

In this paper, we investigated the potential of automated KB completion using LMs. We introduced a challenging benchmark dataset, WD-KNOWN, an unbiased random sample of Wikidata containing 3.9M existing facts. Using this dataset enabled a more realistic assessment of KB completion using LMs. This revealed that previous benchmarks lead to an overestimate of LM-based KB completion performance.

Our analysis showed that LMs are not able to obtain results at a high precision ($\sim 90\%$) for all relations equally, but LM-based knowledge covers language-related and socio-demographic relations particularly well. Furthermore, we discovered that an LM’s vocabulary can limit the capability of fact prediction and we achieved significant improvements with fine-tuning and vocabulary expansion.

Since the prediction of facts non-existent to the KB is crucial for KB completion, we extracted plausible subject-relation pairs with non-existent objects in the KB. By probing the LM for these facts, we received actual novel facts previously unknown to the KB. Since the ground truth for these facts is missing, we performed a human evaluation. Annotators rated the LM’s suggestions as highly correct. That showed a high potential for KB completion, either completely automated at a precision of up to 90% or as a strong recommender system for human-in-a-loop procedures. We demonstrated that in a human-in-a-loop procedure, LMs might reduce the costs for manually curated facts significantly, from approximately \$2 to \$0.4 per fact.

Moreover, we showed that LMs build surprisingly strong generalization capabilities for specific socio-demographic relations.

A promising direction for future work could be the construction of LMs specifically for KBs, which goes beyond fine-tuning. This could include defining specific vocabularies that are optimized for fact prediction.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS (2013)
3. Cao, N.D., Aziz, W., Titov, I.: Editing factual knowledge in language models. In: EMNLP (2021)
4. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI (2010)
5. Cohen, R., Geva, M., Berant, J., Globerson, A.: Crawling the internal knowledge-base of language models. In: Findings of EACL (2023)
6. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI (2018)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
8. Elazar, Y., et al.: Measuring and improving consistency in pretrained language models. *TACL* **9**, 1012–1031 (2021)
9. Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., Simperl, E.: T-REx: a large scale alignment of natural language with knowledge base triples. In: LREC (2018)
10. Heinzerling, B., Inui, K.: Language models as knowledge bases: on entity representations, storage capacity, and paraphrased queries. In: EACL (2021)
11. Lenat, D.B.: CYC: a large-scale investment in knowledge infrastructure. *CACM* **38**, 33–38 (1995)
12. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM CSUR* **55**, 1–35 (2022)
13. Lv, X., et al.: Do pre-trained models benefit knowledge graph completion? A reliable evaluation and a reasonable approach. In: Findings of ACL (2022)
14. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
15. Paulheim, H.: How much is a triple? Estimating the cost of knowledge graph creation. In: ISWC (2018)
16. Petroni, F., et al.: How context affects language models’ factual predictions. In: AKBC (2020)
17. Petroni, F., et al.: Language models as knowledge bases? In: EMNLP (2019)
18. Poerner, N., Waltinger, U., Schütze, H.: E-BERT: efficient-yet-effective entity embeddings for BERT. In: Findings of EMNLP (2020)
19. Razniewski, S., Yates, A., Kassner, N., Weikum, G.: Language models as or for knowledge bases. In: DL4KG (2021)

20. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: EMNLP (2020)
21. Safavi, T., Koutra, D.: CoDEX: a comprehensive knowledge graph completion benchmark. In: EMNLP (2020)
22. Shaik, Z., Ilievski, F., Morstatter, F.: Analyzing race and country of citizenship bias in wikidata (2021)
23. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: AutoPrompt: eliciting knowledge from language models with automatically generated prompts. In: EMNLP (2020)
24. Singhanian, S., Nguyen, T.P., Razniewski, S.: LM-KBC: Knowledge base construction from pre-trained language models. CEUR (2022)
25. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW (2007)
26. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledge base. CACM **57**, 78–85 (2014)
27. Weikum, G., Dong, L., Razniewski, S., Suchanek, F.M.: Machine knowledge: creation and curation of comprehensive knowledge bases. In: FnT (2021)