How to Query Language Models?

Anonymous ACL submission

Abstract

Large pre-trained language models (LMs) are capable of not only recovering linguistic but also factual and commonsense knowledge. To access the knowledge stored in mask-based LMs, we can use cloze-style questions and let the model fill in the blank. The flexibility advantage over structured knowledge bases comes with the drawback of finding the right query for a certain information need. Inspired by human behavior to disambiguate a question, we propose to query LMs by example. To clarify the ambivalent question *Who does Neuer play for?*, a successful strategy is to demonstrate the relation using another subject, e.g., *Ronaldo plays for Portugal. Who does Neuer play for?*. We apply this approach of querying by example to the LAMA probe and obtain substantial improvements of up to 37.8% for BERT-large on the T-REx data when providing only 10 demonstrations—even outperforming a baseline that queries the model with up to 40 paraphrases of the question. The examples are provided through the model's context and thus require neither fine-tuning nor an additional forward pass. This suggests that LMs contain more factual and commonsense knowledge than previously assumed—if we query the model in the right way.

1 Introduction

Language Models (LM) are omnipresent in modern NLP systems. In just a few years, they've been established as the standard *feature extractor* for many different language understanding tasks (Karpukhin et al., 2020; Zhang et al., 2020; Wang et al., 2019; He et al., 2020). Typically, they are used to create a latent representation of natural language input and then fine-tuned to the task at hand. However, recent work (Petroni et al., 2019; Jiang et al., 2020; Brown et al., 2020; Roberts et al., 2020) has shown that *off-the-shelve* language models capture not only linguistic features but also large amounts of relational knowledge, not requiring any form of re-training.

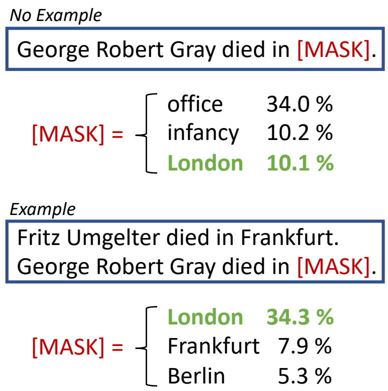

Figure 1: BERT's top-3 predictions with probabilites when prompted with the cloze-style question (top) versus when prompted with one additional example of the same relation (bottom).

The LAMA probe by Petroni et al. (2019) was designed to quantify the amount of relational knowledge present in (mask-based) language models. While the task of predicting the right object for a subject-relation tuple remains the same as for a standard knowledge base (KB) completion query, the input is structured in a cloze-style sentence. For example, a KB completion query of the form *(Dante, born-in, X)* becomes *"Dante was born in [MASK]."*. Petroni et al. (2019) show that BERT (Devlin et al., 2019) performs on par with competitive specialized models on factual and commonsense knowledge. The performance on this task can only be seen as a lower bound to the actual knowledge present in language models as the choice of natural language template for a given relation might be suboptimal (Petroni et al., 2019; Jiang et al., 2020). The more general question here is *"How to query an LM for a specific information need?"*. Jiang et al. (2020) propose to use multiple paraphrases of the probe and then aggregate the solutions. Petroni et al. (2020), on the other hand, add relevant context. Both approaches can be linked to common human behavior. In human dialog, a question can be made more precise both by paraphrasing or adding additional context information. Since language models are trained on large amounts of human-generated data, the intuition of phrasing the information need *most naturally* seems obvious. Humans excel at pattern recognition and pattern continuation for many different modes of representation (Shugen, 2002). Concepts embedded in language are no exception to this. Therefore, another common way to probe a human's knowledge is by providing examples and asking them to transfer the relation provided to a new object. For example, asking *Who plays Neuer for?* is ambiguous as both *Bayern Munich* and *Germany* would be correct answers. However, when contextualizing the question with an example, the answer is clear: *I know Ronaldo plays for Portugal. Who plays Neuer for?*.

In this work, we apply the concept of querying by example to probe language models. Additional to the cloze-style question, we provide other examples of the same relation to the model's input. The previous example's input then becomes *"Ronaldo plays for Portugal. Neuer plays for [MASK]."*. We show that by providing only a few demonstrations, standard language models' prediction performance improves drastically. So much so that for the TREx dataset, it becomes an even more powerful technique to retrieve knowledge than using an ensemble of up to 40 different paraphrases (Jiang et al., 2020), while requiring only a single forward pass instead of 40.

2 Related Work

Language Model Probes Petroni et al. (2019) started to investigate how much factual and commonsense knowledge LMs posses. They released the LAMA probe, which is a dataset consisting of T-REx (Elsahar et al., 2018), Google-RE, ConceptNet (Speer et al., 2018), and SQuAD (Rajpurkar et al., 2016). Each dataset is transformed to be a collection of ⟨subject, relation, object⟩-triplets and pruned to only contain single token objects present in BERT's vocabulary. Additionally, they provide templates in natural language for each relation. Their investigation reveals that BERT-large has remarkable capabilities in recalling factual knowledge, competitive to supervised baseline systems.
Since there is usually more than one way to express a relation, the LAMA probe score can only be regarded as a *lower bound* (Petroni et al., 2019; Jiang et al., 2020). To tighten this lower bound, Jiang et al. (2020) propose an automatic discovering mechanism for paraphrases together with an aggregation scheme. By querying the LM with a diverse set of prompts, they significantly improve the LAMA probe's baseline numbers for BERT models. However, this approach incurs the cost of additional queries to the LM, an optimization procedure to aggregate the results, and the extraction of paraphrases.
Machine reading comprehension (MRC) and open-domain question answering (QA) are fields in NLP dominated by large pre-trained LMs. Here, the premise typically is that the model is capable of extracting the answer from the provided context, rather than having it stored in its parameters[1]. Petroni et al. (2020) extend this line of thought to retrieve factual knowledge from LMs by providing relevant context but *without* fine-tuning the model. Their experiments show that providing relevant passages significantly improves the scores on the LAMA probe for BERT models.

Few-Shot Learning The term few-shot learning refers to the practice of only providing a few examples when training a model, compared to the typical approach of using large datasets (Wang et al., 2020). In the NLP domain, recent work by Brown et al. (2020) suggests to use these few examples only in the context, as opposed to actually *training* with it. Fittingly, they call this approach *in-context* learning. Here, they condition the model on a natural language description of the task together with a few demonstrations. Their experiments reveal that the larger the model, the better its in-context learning capabilities. Our approach is very similar to in-context learning, with the difference that we do not provide a description of the task and utilize natural language templates for the relations. The motivation is that this should closely resemble human behavior of providing examples of a relation: instead of providing a list of subject and objects and let the other person figure out the relation, a human typically provides the subject and objects embedded in the template relation. Moreover, we understand our approach not as a *learning* method, but rather as a querying technique that disambiguates the information need.
Schick and Schütze (2020b) argue that small LMs

[1]With the notable exception of the work of Roberts et al. (2020), which uses a T-5 model without any access to an additional knowledge base.

can be effective for few-shot learning too. However, they approach the problem of limited examples differently; instead of providing it as conditioning in the input, they actually train with it. By embedding the data into relation templates, they obtain training data that is closer in style to the pre-training data and, thus, can learn with fewer samples. Gao et al. (2020) take this concept even further and automate the template generation. Additionally, they also find that—when fine-tuning with few samples—providing good demonstrations in the context improves the model's performance.

3 Background

3.1 Language Models for cloze-style QA

In this work, we probe mask-based language models for their relational knowledge. The considered facts are triplets consisting of a subject, a relation, and an object $\langle s, r, o\rangle$. Language models are trained to predict the most probable word given the (surrounding) context. Hence, to test a model's factual knowledge, we feed it natural text with the object masked out. This requires a mapping from the relation r to a natural language prompt t_r with placeholders for subject and object, e.g., the relation r = *age* becomes t_r = *[s] is [o] years old*. When probing for a single $\langle s, r, o\rangle$-triplet, the input to the language model is the natural language prompt t_r of the relation r together with the subject s. It outputs a *likelihood* score $\mathrm{P_{LM}}$ for each token in its vocabulary $\mathcal{V}$ which we use to construct a top-k prediction subset $\mathcal{V}'$ for the object o:

$$\mathcal{V}' = \arg \max_{\mathcal{V}' \subset \mathcal{V}, |\mathcal{V}'|=k} \sum_{o' \in \mathcal{V}'} \mathrm{P_{LM}}(o'|s, t_r) \qquad (1)$$

The language model *succeeds* for the triplet @k if $o \in \mathcal{V}'$. For example, we say that it knows the fact $\langle s = \text{Tiger Woods}, r = \text{age}, o = 45\rangle$ @3, if for the query *"Tiger Woods is [MASK] years old"* it ranks the token *"45"* within the top-3 of the vocabulary.

3.2 Datasets

We use the LAMA probe in our experiments (Petroni et al., 2019). It's a collection of factual and commonsense examples provided as $\langle s, r, o\rangle$-triplets[2] with single token objects. Moreover, it provides human-generated templates t_r for each relation r. The statistics about the three considered corpora T-REx (Elsahar et al., 2018), Google-RE[3], and ConceptNet (Speer et al., 2018) are provided in Table 1.

Corpus	Relation	Statistics #Facts	#Relations
Google-RE	birth-place	2937	1
	birth-date	1825	1
	death-place	765	1
	Total	5527	3
T-REx	1-1	937	2
	N-1	20006	23
	N-M	13096	16
	Total	34039	41
ConceptNet	Total	11458	16

Table 1: Statistics for the corpora of the LAMA data.

3.3 Models

We investigate the usefulness of querying by example, for three individual language models: BERT-base, BERT-large (Devlin et al., 2019), and ALBERT-xxl (Lan et al., 2020). These models are among the most frequently used language models these days[4]. For both BERT models, we consider the cased variant, unless explicitly noted otherwise.

4 Method

Our proposed method for querying relational knowledge from LMs is simple yet effective. When we construct the query for the triplet $\langle s, r, o\rangle$, we provide the model with additional samples $\{\langle s', r, o'\rangle, \langle s'', r, o''\rangle, \dots\}$ of the same relation r. These additional examples are converted to their natural language equivalent using the template t_r and prepend to the cloze-style sentence representation of $\langle s, r, o\rangle$. The intuition is that the non-masked examples provide the model with an idea of filling in the gap for the relation at hand. As can be seen in Figure 1, providing a single example in the same structure clarifies the object requested for both humans and BERT. This is particularly useful when the template t_r does not capture the desired relation r between subject s and object o unambiguously, which in natural language is likely to be the case for many relations. In this sense, it tries to solve the same problem as paraphrasing. A

[2]We do not consider the SQuAD dataset of the probe as it has no clear notion of *relation*.

[3]https://github.com/google-research-datasets/relation-extraction-corpus

[4]According to the statistics from https://huggingface.co/models?filter=pytorch,masked-lm.

query is paraphrased multiple times to align the model's understanding of the query with the actual information need. When we provide additional examples, we do the same by showing the model how to apply the relation to other instances and ask it to generalize. Of course, the model does not reason in this exact way; rather, through its training data, it is biased towards *completing patterns* as this is a ubiquitous behavior in human writing.

Query	Predictions
No Example	
Rodmarton[5] is a ____.	farmer (3.9%) businessman (2.5%)
Random Example	
M.S.I. Airport is a airport. Rodmarton is a ____.	town (16.9%) **village** (14.7%)
Close Example	
Nantmor is a village. Rodmarton is a ____.	**village** (75.5%) hamlet (16.0%)
Arrow Operator	
Totopara → village The argument → album Tisza → river Rodmarton → ____	**village** (21.4%) town (8.7%)

Table 2: Example queries with predictions (from BERT-large) for the different querying methods. The correct answer is marked in bold.

Since we only adjust the context fed to the model, we do not incur the cost of additional forward passes. When paraphrasing, on the other hand, each individual template requires another query to the model. Moreover, our approach does *not* require any learning, i.e., backward passes, and hence is very different from the classic fine-tuning approach and pattern-exploiting training (Schick and Schütze, 2020a,b).

In Table 2, we compare different approaches of querying by example. The left column shows the input to the model, i.e., the query. The right column shows BERT-large's top-2 prediction, with its corresponding probabilities[6]. The first row of the table shows that completing the *is-a* relation for the village Rodmarton is tricky for the model. Its top predictions are not even close to the correct answer suggesting that BERT either does not know about this particular village or that the information need is not well enough specified. Interestingly, when prepending the query with another *random* example of the same relation (2nd row), the model's top predictions are *town* and the ground-truth *village*. This proves that BERT knows the type of instance Rodmarton is; only the extraction method (the cloze-style template) was not expressive enough.

Close Examples When humans use examples, they typically do not use a completely random subject but use one that is, by some measure, close to the subject at hand. In our introductory example, we used Ronaldo to exemplify an information need about Neuer. It would have been unnatural to use a musician here, even when describing a formally correct *plays-for* relation with them. We extend our approach by only using examples for which the subject is close in latent space to the subject querying for. We use the cosine similarity between the subject encodings using BERT-base. More formally, we encode a subject s using

$$f_\theta(s) = B_\theta([\text{CLS}] + s + [\text{SEP}])^{\text{CLS}}, \quad (2)$$

with $B(x)^{\text{CLS}}$ being the BERT encoding of the CLS-token for the input x, and θ being the BERT model's parameters. We then obtain the top-k most similar subjects to s in the dataset $\mathcal{D}$ through maximizing the cosine similarity, i.e.,

$$\mathcal{D}' = \arg \max_{\mathcal{D}' \subset \mathcal{D} \setminus \{s\}, |\mathcal{D}'| = k} \sum_{s' \in \mathcal{D}'} \frac{f_\theta(s)^\top f_\theta(s')}{\|f_\theta(s)\| \|f_\theta(s')\|} \quad (3)$$

From the top-k subset of most similar subjects $\mathcal{D}'$, we randomly sample to obtain our priming examples. Table 2 (3rd row) shows the chosen close example to Rodmarton, which is Nantmor, another small village in the UK. Provided with this particular example, BERT-large predicts the ground-truth label *village* with more than 75% probability.

Arrow Operator Brown et al. (2020) propose to use LMs as in-context learners. They suggest providing "training" examples in the model's context using the arrow operator, i.e., to express an $\langle s, r, o \rangle$ triplet they provide the model with $s \Rightarrow o$. We can apply this concept to the LAMA data by using the same template $t_r =$" $[s] \Rightarrow [o]$" $\forall r$. In Table 2 (last row), we see that by providing a few examples of

[5]A village in South West England.

[6]The probabilities are obtained by applying a softmax on the logit output over the token vocabulary.

the *is-a* relation, BERT-large can rank the ground-truth highest even though the relationship is never explicitly described in natural language. However, not using a natural language template makes the model less confident in its prediction, as can be seen by the lower probability mass it puts on the target.

5 Results

We focus the reporting of the results on the mean precision at k (P@k) metric. In line with previous work (Petroni et al., 2019, 2020; Jiang et al., 2020)[7], we compute the results per relation and then average across all relations of the dataset. More formally, for the dataset $\mathcal{D} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ that consists of n relations where each relation has multiple datapoints $\langle x, y \rangle$, we compute the P@k score as:

$$P@k = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{R}_i \in \mathcal{D}} \frac{1}{|\mathcal{R}_i|} \sum_{\langle x,y \rangle \in \mathcal{R}_i} 1_{\mathcal{V}'_x}(y), \qquad (4)$$

where 1 denotes the indicator function that is 1 if the ground truth y is in the top-k prediction set $\mathcal{V}'$ for the input x and 0 otherwise.

Table 3 shows the P@1 scores of different models and querying approaches across the LAMA probe's corpora. While for the Google-RE data, providing additional examples shows to be detrimental, we see massive prediction performance gains for T-REx and ConceptNet. Most notably, the P@1 score of BERT-large on T-REx increases by 37.8% to 44.8 when providing 10 close examples. Similarly, the lower bound on Albert's performance for T-REx (ConceptNet) can be improved by up to 72.3% (25.0%) with 10 close examples.

Google-RE For the Google-RE subset of the data, querying by example hurts the predictive capabilities of LMs. In the following, we provide an intuition of why we think this is the case. Looking at the baseline numbers of the individual relations for this data, we see that the performance is largely driven by predicting a person's birth and death place; the birth-date relation doesn't play a significant role because BERT is incapable of accurately predicting numbers (i.e., dates) (Lin et al., 2020; Wallace et al., 2019). The birth and death place of a person BERT-large predicts correctly 16.1% and 14.0% of the time, respectively; significantly lower than the 32.5% P@1 score among the relations of the T-REx data. Recent work describes that BERT has a bias to predict that a person with, e.g., an Italian sounding name is Italian (Rogers et al., 2020; Poerner et al., 2020). We suspect that this bias helps BERT predict birth and death places without knowing the actual person, and therefore it is not an adequate test of probing an LMs factual knowledge. As a consequence, the predictions it makes are more prone to errors when influenced by previous examples.

T-REx Figure 2 depicts the mean precision at 1 on the T-REx corpus for a varying number of examples provided. It shows that even a few additional examples can significantly improve the performance of the LMs. However, there is a saturation of usefulness for more examples that seems to be reached at around 10 examples already. Interestingly, with 10 examples, BERT-large even slightly improves upon the optimized paraphrase baseline from Jiang et al. (2020), while only requiring a single forward pass.
Table 4 shows the improvement in P@1 score for the individual relations that most (and least) benefit from additional examples for BERT-large. The relations for which demonstrations improve the performance the most typically have one thing in common: they are ambiguous. Prototypical ambiguous relations like *located-in* or *is-a* are among the top benefiting relations. One rather untypical improvement candidate is the top-scoring one of *religion-affiliation*. Suspiciously, this is also the most improved relation by the paraphrasing of Jiang et al. (2020). A closer look at the examples reveals the cause: the target object labels for the religions are provided as nouns (e.g., Christianity, Islam), while the template (*[s] is affiliated with the [o] religion*) indicates to use the religion as an adjective (e.g., Christian, Islamic). Hence, both paraphrasing the sentence such that it is clear to use a noun or providing example sentences that complete the template with nouns alleviate this problem. The relations that benefit the least from demonstrations are unambiguous, like *capital-of* or *developed-by*.

ConceptNet While T-REx probes for factual knowledge, the ConceptNet corpus is concerned with commonsense relations. The improvements of querying by example are significant with

[7] The P@1 score corresponds to Jiang et al. (2020)'s micro-averaged accuracy

Corpus	Relation	Baselines Bb	Bl	Al	Bb_{opt}	Bl_{opt}	LM Bb^3	Bb^{10}	Bb^{10}_{ce}	Bl^3	Bl^{10}	Bl^{10}_{ce}	Al^{10}_{ce}
Google-RE	birth-place	14.9	**16.1**	6.3	-	-	10.5 ±0.4	13.2 ±0.3	11.7 ±0.3	8.9 ±0.5	11.5 ±0.3	11.0 ±0.3	7.0 ±0.3
	birth-date	**1.6**	1.5	1.5	-	-	1.1 ±0.3	1.1 ±0.2	1.2 ±0.1	1.4 ±0.3	1.4 ±0.2	1.5 ±0.1	1.4 ±0.3
	death-place	13.1	**14.0**	2.0	-	-	9.2 ±0.5	11.8 ±0.7	10.4 ±1.0	7.2 ±0.7	9.1 ±0.5	8.5 ±1.1	5.0 ±0.6
	Total	9.9	10.5	3.3	10.4	**11.3**	6.9 ±0.1	8.7 ±0.2	7.8 ±0.4	5.8 ±0.4	7.4 ±0.1	7.0 ±0.4	4.5 ±0.3
T-REx	1-1	68.0	**74.5**	71.2	-	-	59.7 ±0.6	62.0 ±0.6	62.6 ±0.8	66.4 ±0.9	67.6 ±0.6	68.7 ±0.7	69.0 ±0.7
	N-1	32.4	34.2	24.9	-	-	32.3 ±0.1	37.9 ±0.2	41.7 ±0.4	38.8 ±0.2	44.8 ±0.2	**47.9** ±0.2	45.0 ±0.2
	N-M	24.7	24.8	17.2	-	-	27.9 ±0.4	31.3 ±0.2	34.8 ±0.1	31.4 ±0.4	35.0 ±0.1	**37.2** ±0.3	33.5 ±0.2
	Total	31.1	32.5	24.2	39.6	43.9	31.9 ±0.2	36.5 ±0.2	40.0 ±0.2	37.3 ±0.2	42.1 ±0.2	**44.8** ±0.1	41.7 ±0.1
ConceptNet	Total	15.9	19.5	21.2	-	-	15.2 ±0.2	16.2 ±0.2	17.1 ±0.2	19.6 ±0.3	21.2 ±0.2	22.0 ±0.3	**26.5** ±0.2

Table 3: Mean precision at one (P@1) in percent across the different corpora of the LAMA probe. The baseline models shown are BERT-base (Bb), BERT-large (Bl), Albert-xxlarge-v2 (Al), and the best versions of BERT-large and BERT-base by Jiang et al. (2020) that are optimized across multiple paraphrases[8](Bb_{opt} and Bl_{opt}). The LM section on the right shows the results for different querying by example approaches. Here, the superscript denotes the number of examples used and the subscript *ce* denotes that only close examples have been used. Since the choice of examples alters the predictions of the model and thus introduces randomness, we provide the standard deviation measured over 10 evaluations.

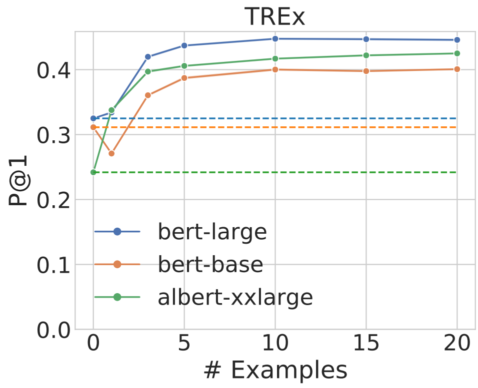

Figure 2: P@1 score for TREx over the number of examples provided. The dashed line shows the baseline value for when no additional example is given.

ID	Template	Δ P@1 n=1	n=3	n=5
P140	[s] is affiliated with the [o] religion .	51.0	67.4	70.0
P30	[s] is located in [o] .	47.8	55.3	55.8
P136	[s] plays [o] music .	12.8	44.0	54.5
P31	[s] is a [o] .	8.2	20.3	24.4
...				
P178	[s] is developed by [o] .	-8.3	-4.2	-6.8
P1376	[s] is the capital of [o] .	-16.3	-8.2	-8.6

Table 4: List of relations of T-REx that benefit the most (least) by additional examples. The right column provides the improvement in precision at 1 score when {1, 3, 5} examples are provided for BERT-large.

12%, 7.5%, and 25% relative improvement for BERT-base, BERT-large, and Albert-xxlarge.

More detailed plots for all the corpora and several metrics are provided in Appendix A.4.

5.1 The Change of Embedding

To further investigate the disambiguation effect of additional examples, we take a look at the latent space. In particular, we're interested in how the clusters of particular relations, formed by the queries' embeddings, change when providing the context with additional examples. Figure 3 visualizes BERT-large's [CLS]-token embedding for queries from the T-REx corpus, using t-SNE (van der Maaten and Hinton, 2008). The individual colors represent the relations of the queries. The first two images depict the clustering when using the natural language template without additional demonstrations (left) and ten demonstrations (right). The fact that the clusters become better separated is visual proof that providing examples disambiguates the information need expressed by the queries. The two plots on the right show the clustering when instead of a natural language template, the subject and object are only separated by the arrow operator "⇒". Here, we see an even more significant change in separability when providing additional demonstrations, as the actual information need is more ambiguous.

5.2 TextWorld Commonsense Evaluation

An emerging field of interest inside the NLP community is text-based games (TBG). An agent is placed inside an interactive text environment in these games and tries to complete specified goals–only using language commands. To succeed, it

[8]These models involve one query to the model per paraphrase.

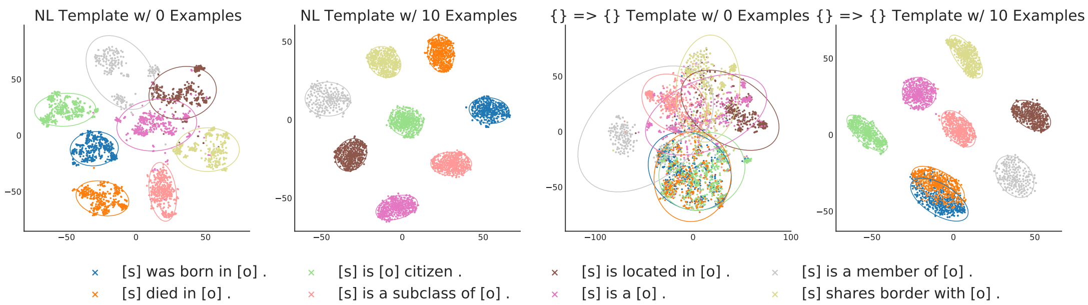

Figure 3: BERT-large's [CLS]-token embedding of a subset of T-REx queries visualized in two dimensions using t-SNE (van der Maaten and Hinton, 2008). Each point is a single query and the color represents the corresponding relation class. The ellipses depict the 2-std confidence intervals. The individual images show the clustering for both the natural language and the ([s]; [o]) template with either no examples or ten examples provided.

requires a deep language understanding to decide what are reasonable actions to take in the scene that move it closer to its final goal. These environments are often modeled on real-world scenes to foster the commonsense-learning capabilities of an agent. The TextWolrd Commonsense (`TWC`) game world by Murugesan et al. (2020) focus specifically on this aspect. There, the agent is placed in a typical modern-house environment to tidy up the room. This involves moving all the objects in the scene to their *commonsense* location, e.g., the dirty dishes belong in the dishwasher and not in the cupboard. Murugesan et al. (2020) approach this problem by equipping the agent with access to a commonsense knowledge base. Replacing a traditional KB with an LM for this task is very intriguing as the LM has relational knowledge stored implicitly and is capable of generalizing to similar objects. To test the feasibility of using LMs as commonsense knowledge source in the `TWC` environment, we design the following experiment[9]: We use a static agent that picks up any misplaced object o at random and puts it to one of the possible locations l in the scene according to a specific prior $p(l|o)$. This prior $p(l|o)$ is computed at the start of an episode for all object-location combinations in the scene, using an LM. We use the arrow operator as described in Table 2 and vary the number of examples provided. In Figure 4, we show the result for albert-xxlarge on the *hard* games of `TWC`, compared to a simple uniform prior (i.e., $p(l_i|o) = const. \forall i$), and Murugesan et al. (2020)'s RL agent with access to a commonsense KB. We see the same trend as in the LAMA experiments: providing additional examples of the same relation boosts performance significantly and saturates after 10-15 instances.

[9]Details and the pseudocode are provided in Apendix A.3

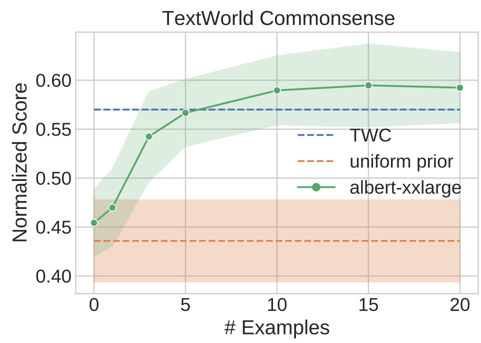

Figure 4: Normalized score for the *hard* games of the `TWC` environment over the number of examples provided for albert-xxlarge. The dashed baselines are the static agent with a uniform prior and the TWC commonsense agent by Murugesan et al. (2020). The shaded regions depict the standard deviation over 10 runs.

5.3 Word Analogy Evaluation

To evaluate the usefulness of querying pre-trained language models by examples for linguistic knowledge, we move to the word analogy task—a standard benchmark for non-contextual word embeddings. This evaluation is based on the premise that a good global word embedding defines a latent space in which basic arithmetic operations correspond to linguistic relations (Mikolov et al., 2013b). With the rise of contextual word embeddings and large pre-trained language models, this evaluation has lost significance. However, we consider approaching this task from the angle of querying linguistic knowledge from an LM instead of performing arithmetics in latent space. By providing examples of the linguistic relation with a regular pattern in the context of the LM, we prime it to apply the relation to the final word with its masked out correspondence.

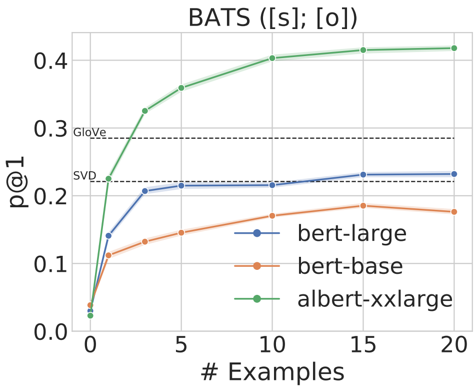

Figure 5: P@1 score on BATS over the number of examples provided. The performance of the GloVe and SVD benchmark models by Gladkova et al. is shown with the black, dashed lines.

We consider the *Bigger Analogy Test Set (BATS)* (Gladkova et al.) for our experiments. BATS consists of 40 different relations covering inflectional and derivational morphology, as well as lexicographic and encyclopedic semantics. Each relation consists of 50 unique word pairs. However, since most pre-trained LMs, including BERT and Albert, use subword-level tokens for their vocabulary, not all examples can be solved. In particular, 76.1% and 76.2% of the targets are contained in BERT's and Albert's vocabulary, respectively—upper bounding their P@1 performance.

Figure 5 depicts the P@1 score[10] for the individual LMs on BATS. Noticeably, also on this task, the LMs benefit from additional examples up to a certain threshold for which the usefulness stagnates. Both BERT models do not beat Gladkova et al.'s GloVe (Pennington et al., 2014) benchmark. This is in part because not all targets are present in the token vocabulary. Considering only the *solvable* word pairs, BERT-large achieves a P@1 score of 30.6% with 15 examples—beating the GloVe baseline achieving 28.5%. Interestingly, Albert-xxlarge outperforms all other models, including the baselines, by a large margin. Figure 7 in Appendix A.4 breaks down the LM's performance across the different relations of BATS and compares it against the GloVe baseline. Albert beats GloVe on almost all relations where its vocabulary does not limit it; the most significant improvements are in the derivational morphology and lexicographic semantics categories. It is outperformed by GloVe only on two relations: *country:capital* and *UK city:county*. Especially the former *country:capital* category is very prominent and constituted 56.7% of all semantic questions of the original Google test set (Mikolov et al., 2013a)—potentially influencing the design and tuning of non-contextual word embeddings.

[10]The P@1 score corresponds to Gladkova et al.'s reported accuracy score.

6 Discussion

Augmenting the context of LMs with demonstrations is a very successful strategy to disambiguate the query. Notably, it is as successful, on TRE-x, as using an ensemble of multiple paraphrases. The benefit of additional examples decreases when the information need is clear to the model; this is the case for unambiguous prompts or when enough (around 10) demonstrations are provided. Even in the extreme case of ambiguity, for example, when the arrow operator (*[s] => [o]*) is used to indicate a relation, providing only a handful of examples clarifies the relation sufficiently in many cases. We showed that the usefulness of providing additional demonstrations quickly vanishes. Hence, when having access to more labeled data and the option to re-train the model, a fine-tuning strategy is still better suited to maximize the performance on a given task. Moreover, casting NLP problems as language modeling tasks only works as long as the target is a single-token word of the LM's vocabulary. While technically large generation-based LMs as GPT (Brown et al., 2020; Radford et al., 2018) or T5 (Raffel et al., 2019) can generate longer sequences, it is not clear how to compare solutions of varying length.

7 Conclusion

In this work, we explored the effect of providing examples to probing LMs relational knowledge. We showed that already a few demonstrations—supplied in the context of the LM—disambiguate the query to the same extent as using an optimized ensemble of multiple paraphrases. We base our findings on experimental results of the LAMA probe, the BATS word analogy test, and a TBG commonsense evaluation. On the T-REx corpus' factual relations, providing 10 demonstrations improves BERT's P@1 performance by 37.8%. Similarly, on ConceptNet's commonsense relations, Albert's performance improves by 25% with access to 10 examples. We conclude that providing demonstrations is a simple yet effective strategy to clarify ambiguous prompts to a language model.

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L. Hamilton. 2021. Learning dynamic belief graphs to generalize on text-based games.

Leonard Adolphs and Thomas Hofmann. 2019. Ledeepchef: Deep reinforcement learning agent for families of text-based games. *CoRR*, abs/1909.01646.

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

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew J. Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2018. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW, address = San Diego, California, June 12-17, 2016, publisher = ACL, year = 2016, pages = 47-54 doi = 10.18653/v1/N16-2002, url = https://www.aclweb.org/anthology/N/N16/N16-2002.pdf.*

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Infocom. 1980. Zork i.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know?

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Pushkar Shukla, Sadhana Kumaravel, Gerald Tesauro, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. 2020. Text-based rl agents with commonsense knowledge: New challenges, environments and baselines.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions.

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

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-bert: Efficient-yet-effective entity embeddings for bert.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? *arXiv e-prints*, page arXiv:2002.08910.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few shot text classification and natural language inference.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners.

Wang Shugen. 2002. Framework of pattern recognition model based on the cognitive psychology. *Geospatial Information Science*, 5(2):74–78.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding.

Yaqing Wang, Quanming Yao, James Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension.

A Appendices

A.1 Implementation Details

The source code to reproduce all the experiments is provided in the supplementary material. All individual runs reported in the paper can be carried out on a single GPU (TESLA P100 16GB), though speedups can be realized when using multiple GPUs in parallel. The wall-clock runtime for the corpora of the LAMA probe is shown in Table 5.
All models used in this work are accessed from the Huggingface's list of pre-trained models for PyTorch (Wolf et al., 2019). Further details about these models are provided on the following webpage: https://huggingface.co/transformers/pretrained_models.html.

Corpus	Model	# Parameters	Avg. Input Length	Runtime [s]
Google-RE	bert-base-cased	109M	5.5	12.8
	bert-base-cased^{10}		60.3	36.1
	bert-base-cased$^{10}_{ce}$		60.1	39.6
	bert-large-cased	335M	5.5	20.5
	bert-large-cased^{10}		60.3	85.5
	bert-large-cased$^{10}_{ce}$		60.1	99.7
	albert-xxlarge-v2	223M	5.5	85.4
	albert-xxlarge-v2^{10}		60.3	466.0
	albert-xxlarge-v2$^{10}_{ce}$		60.1	544.9
T-REx	bert-base-cased	109M	7.6	72.6
	bert-base-cased^{10}		83.2	239.0
	bert-base-cased$^{10}_{ce}$		82.7	234.1
	bert-large-cased	335M	7.6	119.3
	bert-large-cased^{10}		83.2	747.5
	bert-large-cased$^{10}_{ce}$		82.7	596.5
	albert-xxlarge-v2	223M	7.6	504.1
	albert-xxlarge-v2^{10}		83.2	3227.4
	albert-xxlarge-v2$^{10}_{ce}$		82.7	3340.9
ConceptNet	bert-base-cased	109M	9.4	38.5
	bert-base-cased^{10}		102.8	121.9
	bert-base-cased$^{10}_{ce}$		104.5	124.6
	bert-large-cased	335M	9.4	80.4
	bert-large-cased^{10}		102.8	311.4
	bert-large-cased$^{10}_{ce}$		104.5	324.3
	albert-xxlarge-v2	223M	9.4	408.0
	albert-xxlarge-v2^{10}		102.8	1760.8
	albert-xxlarge-v2$^{10}_{ce}$		104.5	1853.6

Table 5: The runtime in seconds to go once through the full data from the LAMA probe on a single TESLA P100 GPU with a batch size of 32. The superscript of the model represents the number of examples used for querying and the subscript of *ce* indicates that close examples are used.

A.2 The Choice of Template

When providing examples, we give the model the chance to understand the relationship for which we query without providing additional instructions. This naturally raises the question of whether or not natural language templates are even necessary to query LMs. Most prominently, the in-context learning

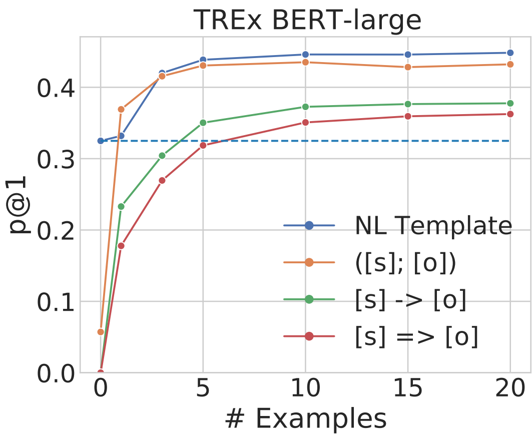

Figure 6: P@1 score for BERT-large on TREx over the number of examples provided. Each line corresponds to one *template* determining how the examples are provided: (i) with the natural language templates from the LAMA probe (NL Template), (ii) separated by a semicolon (([s]; [o])), (iii) separated by a one-lined arrow ([s] -> [o]), or (iv) separated by a double-lined arrow ([s] => [o]). The dashed line shows the baseline value for when no additional example is given.

of Brown et al. (2020) shows that large LMs can complete patterns even when not provided in natural language. In particular, they use the "=>"-operator to express the relation between input and output. In Figure 6, we compare the natural language cloze-style template against three different non-language templates: (i) *[s] => [o]*, (ii) *[s] -> [o]*, (iii) *([s]; [o])*. Surprisingly, Brown et al. (2020)'s "=>"-operator performs the worst for BERT-large on T-TREx, while separating the subject and objects by a semicolon works best—almost on par with the performance of the natural language template after providing just a single example. This result underlines BERT's remarkable pattern-matching capabilities and suggests that a natural language description of the relation is not always needed—even when querying relatively small LMs.

A.3 Details TextWorld Commonsense Evaluation

Text-based games (TBG) are computer games where the sole modality of interaction is text. Classic games like Zork (Infocom, 1980) used to be played by a large fan base worldwide. Today, they provide interesting challenges for the research field of interactive NLP. With the TextWorld framework by Côté et al. (2018), it is possible to design custom TBGs; allowing to adapt the objects, locations, and goals around the investigated research objectives. TBGs of this framework can vary from treasure hunting (Côté et al., 2018) to cooking recipes (Adhikari et al., 2021; Adolphs and Hofmann, 2019), or–as in the experiment at hand–tidying up a room (Murugesan et al., 2020). Murugesan et al. (2020) designed the TextWorld Commonsense environment `TWC` around the task of cleaning up a modern house environment to probe an agent about its commonsense abilities. For example, a successful agent should understand that dirty dishes belong in the dishwasher while clean dishes in the cupboard. Murugesan et al. (2020) approach this problem by developing an agent that, through a graph-based network, has access to relevant facts from the ConceptNet (Speer et al., 2018) commonsense knowledge base. Here, the obvious downside of static KBs for commonsense knowledge extraction becomes apparent: it does not generalize to not listed object-location pairs. Hence, slight deviations of typical entities require additional processing to be able to query the KB. A large pre-trained LM seems to be better suited for this task due to its querying flexibility and generalization capabilities. We test these abilities by designing a static agent as described in the following Algorithm 1, that has access to a large pre-trained LM.

Algorithm 1: LM-prior Agent

Input: `TWC` game `G`, pre-trained language model `LM`

$o_s \leftarrow$ objects in the scene
$l_s \leftarrow$ locations in the scene
$o \leftarrow$ large list of all possible objects across all games

Function `GetPrior`(o_s, l_s, o, LM) **:**
    /* Function to determine a probability distribution over the locations l_s for each object in o_s using the language model LM. */
    p $\leftarrow$ *empty array of size* $|o_s| \times |l_s|$
    forall *object* $o_i \in o_s$ **do**
        d $\leftarrow$ *Randomly sample demonstrations for objects* $\in o \setminus o_s$ *with locations* $\in l_s$
        /* Use demonstrations d to build context for LM, e.g.: */
        /* milk $\Rightarrow$ fridge */
        /* dirty dishes $\Rightarrow$ sink */
        /* o_i $\Rightarrow$ [MASK] */
        c $\leftarrow$ *build_context*(d)
        /* Compute MASK-token probabilities for the locations in l_s using LM */
        $p_{o_i} \leftarrow LM(c, l_s)$
        p.append(p_{o_i})
    end
    return p

prior $\leftarrow GetPrior(o_s, l_s, o, LM)$
while `G` *not finished & max steps not exhausted* **do**
    if *agent holds an object* o_i **then**
        $l_i \leftarrow$ *sample location according to* $prior[o_i]$
        if l_i *correct location for* o_i **then**
            remove o_i *from* o_s
        else
            prior$[o_i] \leftarrow 0$
        end
    else
        $o_i \leftarrow random\_choice(o_s)$
    end
end

A.4 Omitted Figures

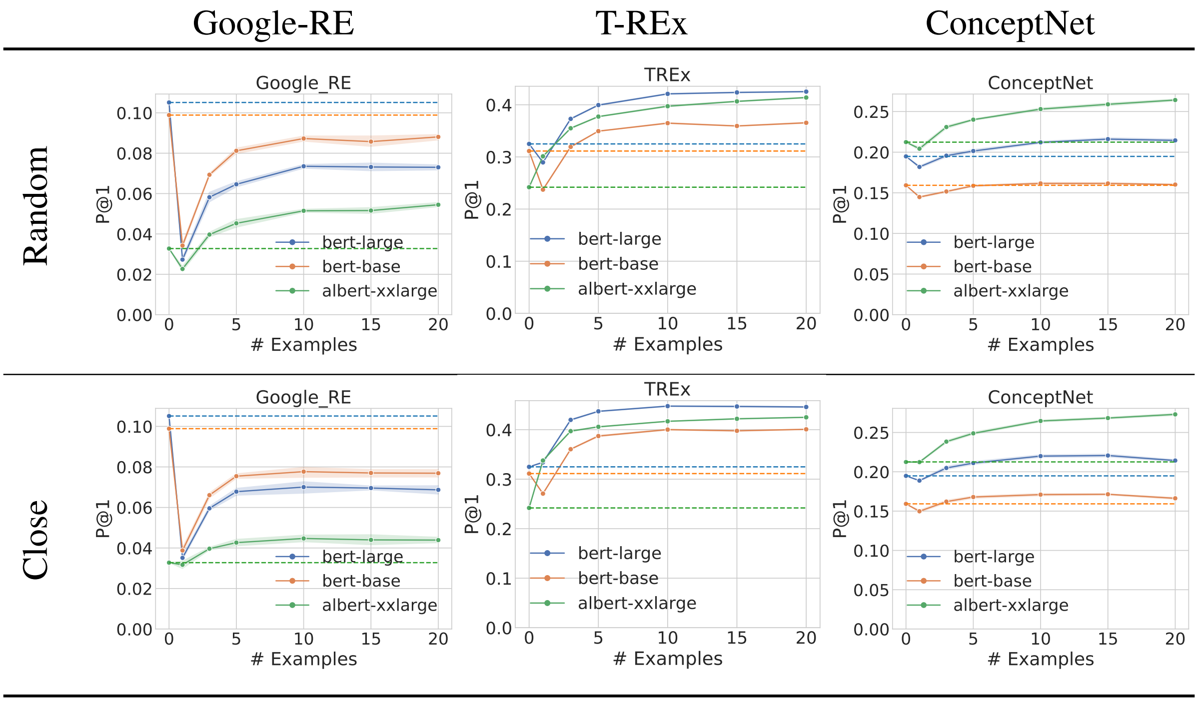

Table 6: P@1 score for the different corpora of the LAMA probe over the number of examples provided. The dashed line shows the baseline values for when no additional example is given. The upper row depicts the scores for when the examples are chosen randomly among the same relation, while the lower row only considers examples from *close* subjects as defined in Section 4.

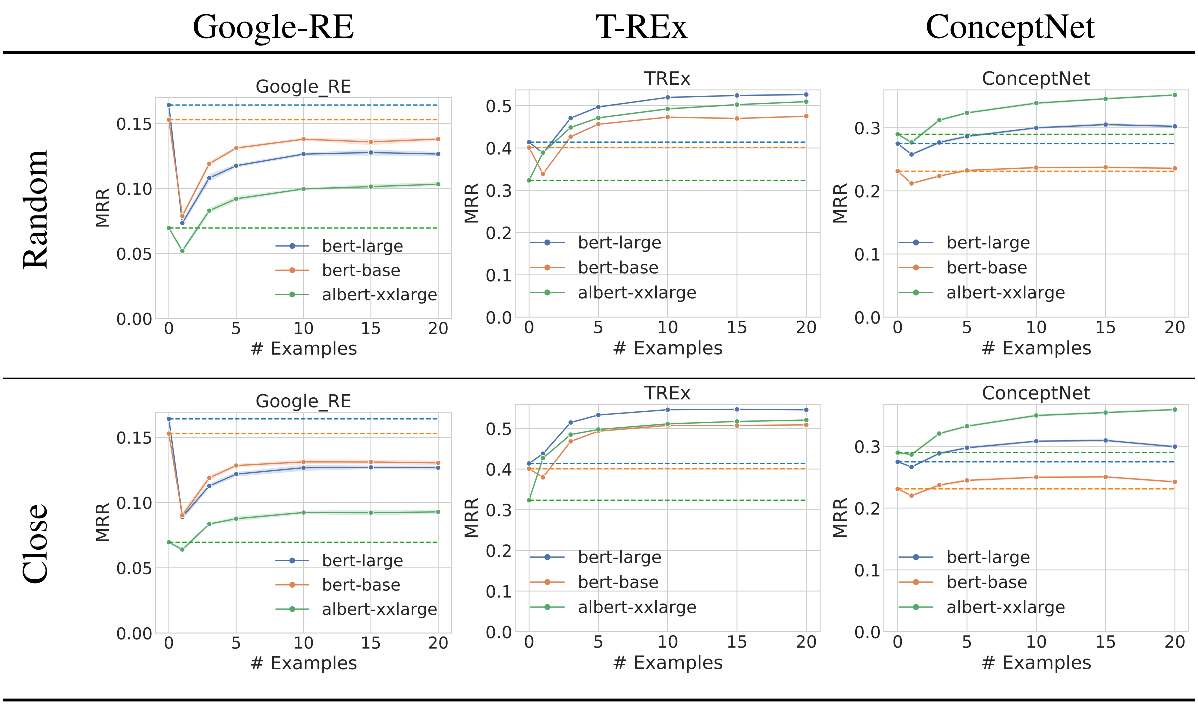

Table 7: Mean reciprocal rank (MRR) score for the different corpora of the LAMA probe over the number of examples provided. The dashed line shows the baseline values for when no additional example is given. The upper row depicts the scores for when the examples are chosen randomly among the same relation, while the lower row only considers examples from *close* subjects as defined in Section 4.

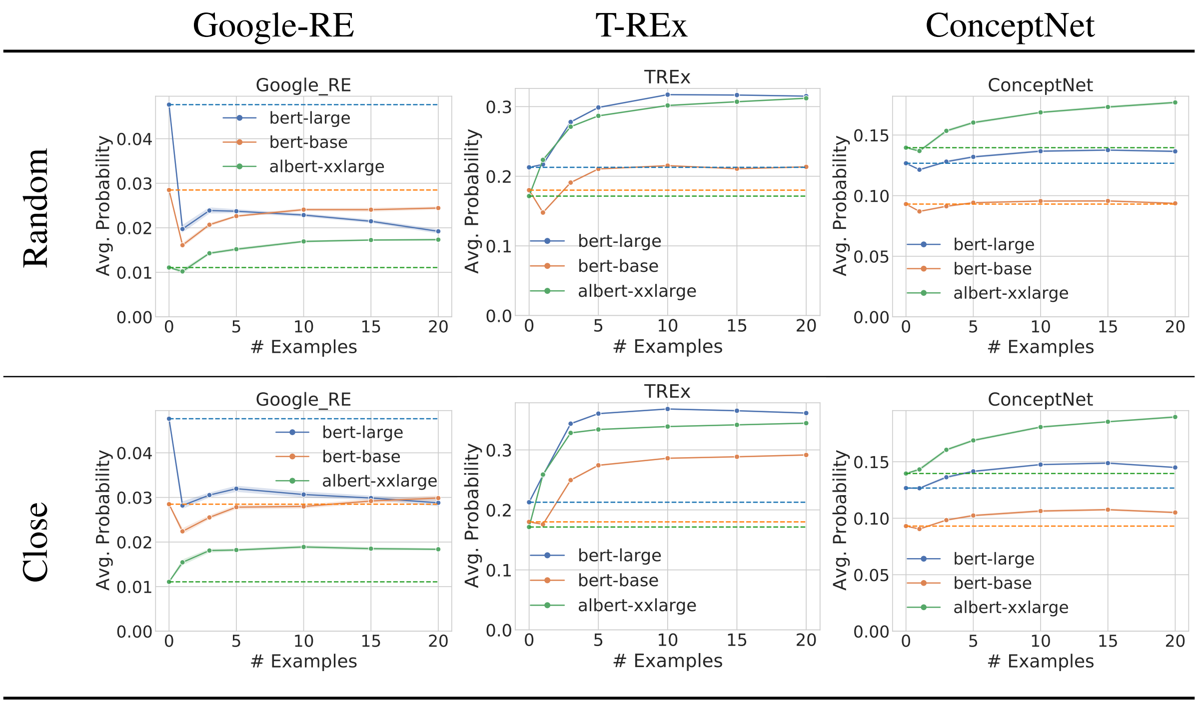

Table 8: Probability assigned to the ground-truth object for the different corpora of the LAMA probe over the number of examples provided. The dashed line shows the baseline values for when no additional example is given. The upper row depicts the scores for when the examples are chosen randomly among the same relation, while the lower row only considers examples from *close* subjects as defined in Section 4.

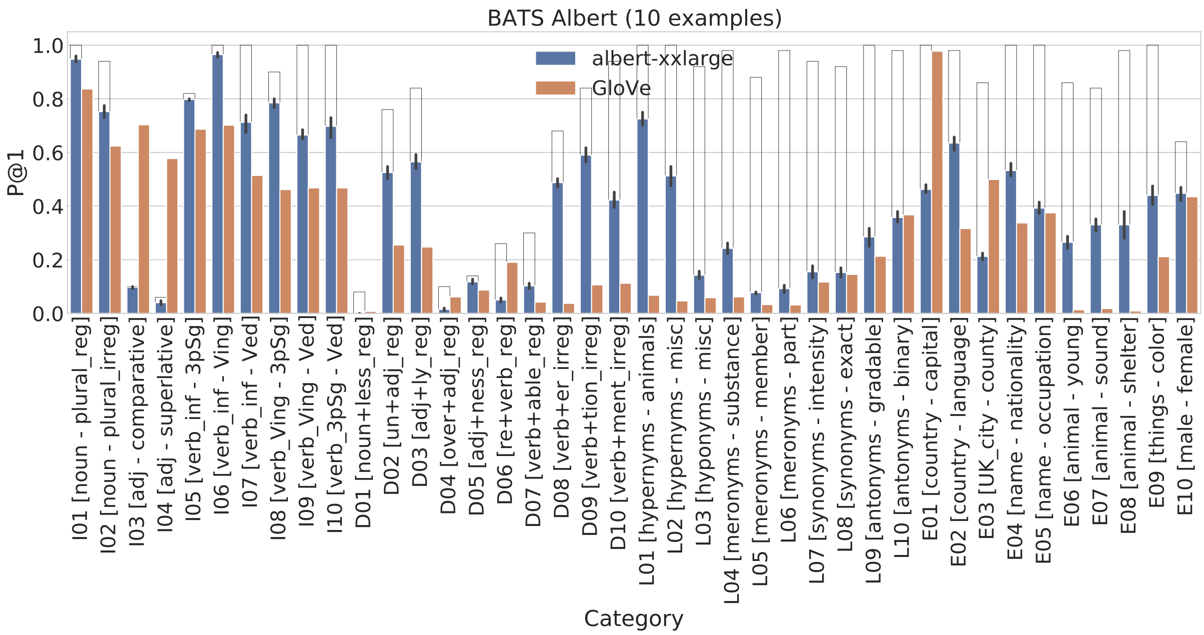

Figure 7: P@1 score on BATS for Albert-xxlarge with 10 examples that use the "([s]; [o])"-template. The x-axis breaks down the performance for the individual relations of the BATS dataset. As a benchmark, we use the GloVe model from Gladkova et al.. The frame around the bar indicates the maximum possible score that the Albert model could have scored because not all targets are tokens in its vocabulary.