Zero-shot Probing of Pretrained Language Models for Geography Knowledge

Nitin Ramrakhiyani[1,2] **Vasudeva Varma**[2] **Girish K. Palshikar**[1] **Sachin Pawar**[1]
[1]TCS Research, Tata Consultancy Services Ltd., India
[2]International Institute of Information Technology (IIIT) Hyderabad

Abstract

Gauging the knowledge of Pretrained Language Models (PLMs) about facts in niche domains is an important step towards making them better in those domains. In this paper, we aim at evaluating multiple PLMs for their knowledge about world Geography. We contribute (i) a sufficiently sized dataset of masked Geography sentences to probe PLMs on masked token prediction and generation tasks, (ii) benchmark the performance of multiple PLMs on the dataset. We also provide a detailed analysis of the performance of the PLMs on different Geography facts.

1 Introduction

Transformer based Pretrained Language Models (PLMs) have proven to be effective on multiple tasks in NLP ranging from the standard information extraction and text classification to more complex ones such as reading comprehension and text generation. Multiple such transformer based PLMs are available, either trained from scratch on large amounts of data or fine-tuned for specific tasks and domains. It is also being established (Liu et al., 2023) that on multiple NLP tasks, PLMs with billions of parameters (*LLMs*) such as GPT-4, Bloom and OPT, perform better than PLMs with significantly lesser number of parameters ('small PLMs') such as BERT and RoBERTa.

As PLMs are being widely used in multiple applications, their performance needs to be improved either by rigorous methods such as full scale fine-tuning or through efficient methods such as prompt based few-shot fine-tuning (Gao et al., 2020), adapters (Houlsby et al., 2019; Pfeiffer et al., 2020) and Low Rank Adaptation (LoRA) (Hu et al., 2021). In this paper we attempt to gauge the performance of multiple PLMs on facts pertaining to Geography. The facts we check include information pertaining to three types of Geographical entities - *Natural* (rivers, mountain ranges, natural reserves, etc.), *Geo-political* (countries, cities, etc.) and *Public/Industrial facilities* (dams, power plants, amusement parks, etc.). We hypothesize that PLMs may not be trained well on such niche Geography knowledge and efforts must be invested to enrich this aspect of their learning. This evaluation exercise is the first step towards such an enrichment effort.

To build a corpus of such Geography facts from text, we obtain geography facts in the form of triples from Wikidata and use templates to arrange them as masked sentences (prompts) to probe the PLMs (Section 2). To gauge the quality of the developed prompts, we carry out a manual examination of randomly sampled sets of prompts and check for triviality, grammatical incorrectness and noise (Section 2.2).

As the first contribution, we release this benchmark dataset[1] of 5268 masked sentences pertaining to various aspects of world geography, which can be used for probing and fine-tuning exercises. We consider a host of PLMs and probe them on the created masked sentences and report the comparative performance. We present an analysis of the behaviour of different PLMs on the different kinds of geography facts we probe. We also present which of the considered fact types are easiest or hardest for the PLMs to answer. This analysis forms our second contribution (Section 4) of benchmarking the performance of multiple small PLMs on this task, thereby suggesting application designers of knowledge systems to consider the reported analysis.

2 Dataset Creation

We create a dataset of sentences which discuss spatial information about various geographical entities,

[1]The dataset of prompts and resources such as prompt templates will be made available publicly on paper acceptance.

Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems, pages 49–61

Fact Type	Example	Wikidata property ($prop_p$)	Example Triple ($entity_S$, $prop_p$, $entity_O$)
Geo-political			
Country in which a certain city is located.	London is located in England	*country* (P17)	(London, P17, England)
Continent in which a certain country is located	Japan is located in Asia.	*continent* (P30)	(Japan, P30, Asia)
Capital of a certain country	Tokyo is the capital of Japan.	*capital* (P36)	(Japan, P36, Tokyo)
Natural			
Countries which are basin countries to a certain sea	Rivers from Greece and Turkey flow into the Aegean Sea.	*basin country* (P205)	(Aegean Sea, P205, Greece), (Aegean Sea, P205, Turkey)
Highest point of a mountain range	Mount Everest is the highest point of the Himalayan mountain range.	*highest point* (P610)	(Mount Everest, P610, the Himalayas)
Waterbody which has created a canyon	The Grand Canyon is created by the Colorado River.	*located in or next to a body of water* (P206)	(Grand Canyon, P206, Colorado River)
Public and Industrial Facilities			
Waterbody on which a certain dam is located	The Aswan dam is located on the Nile river.	*located in or next to a body of water* (P206)	(Aswan dam, P206, the Nile)
Country in which a certain power station is located	The Turbigo Power Station is located in Italy.	*country* (P17)	(Turbigo Power Station, P17, Italy)
Country in which a certain amusement park is located	The Wonderland Amusement Park is located in China.	*country* (P17)	(Wonderland Amusement Park, P17, China)

Table 1: Example Fact Types with Examples, Corresponding Wikidata properties and Triples (Full list in Appendix B)

having tokens masked at appropriate position depending on the information to be probed in the PLM. For example, in a sentence presenting the capital of a certain country, the token denoting the capital city is masked (replaced with a special token such as [MASK]) leaving the rest of the sentence as is. Though all these sentences are suitable for probing encoder models, a subset of sentences which has the masked token at the end, allow us to probe Generative LMs (decoder or encoder-decoder) by asking them to generate text at the [MASK] token and later positions.

We collect instances of 23 different types of geographical facts which we would like to test the PLMs for and categorize them under three heads. As part of the head - *Natural*, facts pertaining to natural entities namely `sea`, `mountain range`, `forest`, `river`, `desert`, `waterfall`, `canyon` and `natural reserve` are considered. As part of the head - *Geo-political*, facts pertaining to geo-politically relevant entities namely `continent`, `country`, `city`, `air base` and `naval base` are considered. As part of the head - *Public/Industrial Facilities*, facts pertaining to entities relevant to public life (work and leisure), namely `dam`, `power station`, `mine`, `amusement park` and `stadium` are considered. The different fact types considered with their examples are shown in Table 1.

2.1 Collecting Probing Sentences using Wikidata

For each of the 18 entities highlighted above, we query Wikidata for a list of most hyperlinked (number of wiki sitelinks) instances of the entity and consider top k (100 to 200) instances from the query result. E.g. For the entity type *city*, we query wikidata to obtain a list of cities ordered descending by number of sitelinks. The sitelinks count is a crude estimate of the popularity of the entity mention which implicitly benefits a PLM while probing, as it would have observed that entity more frequently than other less frequently referenced entities.

Wikidata captures spatial information about various entities through properties/relations such as *located in or next to a body of water* (P206), *shares land borders with* (P47), *continent* (P30), and *country* (P17). This can be used to obtain triples of the form ($entity_S$, $prop_p$, $entity_O$), where $entity_S$ is the instance of the subject entity and its property $prop_p$ has the value $entity_O$ (instance of the object entity). In Table 1, along with each fact type, we show the corresponding Wikidata property which forms the

Example Triple	Templates
Geo-political	
(London, P17, England)	entity$_S$ is a city located in entity$_O$.
Realization: London is a city located in [MASK]. Answer: England	
(Japan, P30, Asia)	entity$_S$ is part of the entity$_O$ continent.
(Japan, P36, Tokyo)	(i) entity$_O$ is the capital of entity$_S$. (ii) entity$_S$ has its capital city as entity$_O$.
Natural	
(Aegean Sea, P205, Greece), (Aegean Sea, P205, Turkey)	(i) Rivers from countries such as entity$_{O1}$ and entity$_{O2}$ flow into the entity$_S$. (ii) The entity$_S$ is bound by countries such as entity$_{O1}$ and entity$_{O2}$.
Realization (i): Rivers from countries such as [MASK] and Turkey flow into the Aegean Sea. Answer: Greece Realization (ii): Rivers from countries such as Greece and [MASK] flow into the Aegean Sea. Answer: Turkey	
(Mount Everest, P610, the Himalayas) (Grand Canyon, P206, Colorado River)	(i) entity$_O$ is the highest point of the entity$_S$ mountain range (ii) The highest point of the entity$_S$ mountain range is entity$_O$ The entity$_S$ canyon is created by water bodies namely entity$_{O1}$ and entity$_{O2}$
Public and Industrial Facilities	
(Aswan dam, P206, the Nile)	(i) The entity$_S$ dam is located on the entity$_O$ river. (ii) The entity$_S$ dam bounds the flow of the entity$_O$ river.
Realization (i): The Aswan dam is located on the [MASK] river. Realization (ii): The Aswan dam bounds the flow of the [MASK] river.	
(Turbigo Power Station, P17, Italy)	(i) The entity$_S$ supplies electricity to states in entity$_O$. (ii) The entity$_S$ is located in entity$_O$.
(Wonderland Amusement Park, P17, China)	The entity$_S$ amusement park is located in entity$_O$.

Table 2: Example Templates to convert wikidata triples to masked sentences (Full list in Appendix C)

triple alongwith the subject entity and the resulting object entity/entities. For each fact-type, we take the instances of the subject entities (based on the sitelink rank as explained earlier), query Wikidata for the corresponding property and obtain the value of the object entity to obtain triples of the form (entity$_S$, prop$_p$, entity$_O$).

To convert the collected triples (entity$_S$, prop$_p$, entity$_O$) into masked sentences, we devise a number of templates to arrange the triple elements into a sentence with a suitable token masked. It is important to note two important nuances at this step of the conversion. Firstly, the choice of the mask token location is not a straightfoward decision. In the current scope, we only mask the object entity (entity$_O$) during the conversion. We follow this convention for all fact types, except the fact type of country capitals wherein we take the additional option of masking the subject entity (entity$_S$) i.e. the country. Moreover, for multi-word object entities, we mask the first token (for e.g., Arabian Sea $\longrightarrow$ [MASK] sea) or the token after the preposition "of" if it is present (for e.g., Forest of Dean $\longrightarrow$ Forest of [MASK]). We also take care of specific cases where the second word should be masked (for e.g., Mount Everest $\longrightarrow$ Mount [MASK]). Secondly, there can be multiple possible entity$_O$ values for a combination of entity$_S$ and property p such as rivers having multiple basin countries and deserts spanning multiple countries. To handle such conversions, we devise multi-value templates where any two of the multiple answers can be placed in the sentence. During masking, one of the values can be masked while keeping the other as-is and vice-versa for another realization of the masked sentence. In Table 2, we show the list of the different templates for each fact type and representative realizations of how the masked sentences are formed from a specific triple. In this manner, based on 32 templates, we create about 5268 masked sentences.

Out of these 5268 masked sentences, a total 3650 are structured such that the [MASK] token occurs at the end of the sentence, thereby making them suitable for probing generative models. Specifically for evaluating the generative models, we use this subset of 3650 sentences and remove the [MASK] token at the end before providing the sentence for further text generation. Irrespective, we employ the entire dataset for evaluating encoder PLMs under consideration.

A straightforward placement of the subject and object entities in a lexical template is not sufficient to arrive at clean and noise-free prompts. This is because of repetitions of words that can happen because of their presence both in the tokens of the entity (obtained as-is from Wikidata) and in the

template. We apply a cleaning procedure explained in detail in Appendix A

2.2 Evaluating the quality of the generated prompts

A benefit we get from this automatic process of developing probing sentences is the scalability. Given any such triples and appropriate templates, a set of masked sentences can be created. In spite of this automation, it is important to check the quality of the generated prompts to correct any inconsistencies that may have co-developed. For this quality evaluation, we sample two sets of 225 prompts (approx. 5% of total number of prompts) in such a manner that prompts from each of the 23 types are selected. We then ask two non-author annotators to manually check these sets respectively. The annotators were asked to check each prompt on three important aspects, inspired from the manual evaluation criteria of "Acceptability" and "Grammaticality" in (Cheng et al., 2022):

- Leaky Prompts: If the prompt has the MASK token at a position where the context is a give away for the answer. For example, `[MASK] D.C. is the capital of USA.`; `The Yarlung Tsangpo Grand Canyon is created by the [MASK] Tsangpo River.`; `The Northeast Greenland National Park is located in [MASK].`
- Repetition: If the prompt has repetition (discussed previously) due to the presence of a word both in the entity value and the template. For example, `Disneyland Park amusement Park is located in [MASK].`; `The Mangla Dam is located on the [MASK] River river.`
- Grammatically incorrect: A prompt which is not grammatically correct such as `USA is located in the [MASK] America continent` instead of `USA is located in the [MASK] American continent.`. Similarly, `The Atlantic South-East Reserves is located in [MASK].` instead of `The Atlantic South-East reserves are located in [MASK].`

Both annotators reported that no repetitions were observed. This validates that the approach of muting repeat tokens in the templates (Appendix A) worked effectively. Secondly, the amount of leaky prompts was 5.78% and 6.67% for the two sets respectively. We currently allow these to be part of the dataset and keep their handling as part of future work. Thirdly, grammatically incorrect prompts were limited to around 1% for the two sets. Further, a third annotator was employed to check both sets and to compute inter-annotator agreement. An agreement of 96% and 98% was seen between the third annotator and the two primary annotators respectively, confirming the manual quality check to be worthy.

Apart from the manual quality check, an automatic check particularly focused on grammatical correctness of the prompts was also performed. Observation of the kind of grammatical issues that were pointed out by the annotators in the previous manual checking exercise, motivated this automatic check. To enable this, the T5 language model's capability of checking the linguistic acceptability of an input text was used. As the focus is on ensuring whether a prompt is grammatically correct, the prompt was converted to a regular sentence by inserting the gold answer in place of the [MASK] token and the regular sentence was then checked using T5-base's linguistic acceptability prompt ("cola sentence:"). If the output is "unacceptable", the prompt is kept aside for further investigation. A total of 557 sentences were flagged as unacceptable out of the total 5268. The third annotator was tasked with checking all the 557 and only 46 of those were found to be really problematic grammar wise. 42 of the 46 actually belonged to a class of issues spawning from plural noun-verb disagreement ("... mountains runs in", "... sanctuaries is located in"). This pattern was fixed through a simple regular expression leading to 103 corrections. The rest 4 in the 46 sentences were manually corrected, leading to overall 107 corrections from this T5 based automatic quality check.

3 Probing Pre-trained Language Models

We aim to evaluate PLMs for their geography knowledge. Given our constraints of using license friendly and less resource consuming models, we consider the following set of language models - Encoders: BERT (Large-cased), RoBERTa (Large-cased), ALBERT (Large-uncased) and DistilBERT (uncased); Decoders: GPT-Neo (2.7B), Falcon (7B), Falcon-instruct (7B) and MPT (7B), lead-

ing us to a total of 8 different PLMs to probe and evaluate. We currently do not report on the encoder-decoder models such as Flan-T5 and BART as their preliminary results are poor and hence require more investigation. We describe in brief (i) the PLMs considered in the exercise and (ii) our preliminary experiments with encoder-decoder models, in Appendix D.

3.1 The Probing process

Training on the MLM task allows an encoder PLM to predict a token at a masked location in a given sentence. In case of decoder models, we ensure to use their generation capability. Given a sentence from the probing dataset created earlier, we query the PLMs to predict the correct token at the location of the MASK token in the sentence. An encoder PLM returns a list of probabilities/logits corresponding to all tokens in the vocabulary to fill the MASK token and we order it in descending order and consider the top ones as answers for evaluation.

Similarly for generative PLMs, and to reiterate, we consider sentences where the MASK is at the final position in the sentence and check the generated text for tokens which can fill the MASK position. Also in case of generative PLMs we avoid sampling the generations and keep the temperature as 0.1, for ensuring a more factual and less creative generation. This tighter setting is in line to what (Sun et al., 2023) have employed in their work on evaluating LLMs for knowledge. Additionally, we observed that the when we prompt the raw sentence to generative LMs, the performance was quite low, however on prefixing the sentence with a suitable instruction, we got reasonable results. We experiment with 5 different instructions and report the results when using the best one for these LMs; (we detail a comparative study of the different instructions in Appendix E). We use the huggingface transformers package[2] as part of the implementation.

4 Evaluation and Analysis

It is desired that the token predicted with highest probability for the MASK token's place should be correct, indicating the learning of the PLM to be complete for that fact. Similarly for generative LMs, the token predicted right after the input text completes should be correct to consider it a valid answer. However, evaluating using only the highest probability prediction (in case of encoder models) and the first generated token (in case of generative models) would be too strict as the PLM may predict some token based on other lexical contexts in the input sentences, while still bringing the correct answer later down (or further ahead). This prompts us to consider a lenient accuracy based metric for evaluation:

in-top-k / in-near-k tokens: This evaluation metric, in case of encoder LMs, gives a score of 1 to the PLM if the correct answer comes in the top-k places of the prediction probability based rank list of tokens. Similarly in case of generative LMs, it awards a score of 1 to the PLM if the correct answer is spotted in the k nearest tokens generated after the input text. This metric assuages the concern of checking only the top most (or nearest) predicted token and gives the due benefit to the PLM. We try with k = 5 and 10 for encoder models and k = 5 and 15 for decoder models.

We can also consider a softer representation based similarity between the predicted tokens and the desired ones instead of exact match to handle variations such as *US*, *USA* and *America*. However, that would inevitably bring into play, some form of thresholds on the similarity score, which would be difficult to guess without training. We believe that the top-k/near-k evaluation metric also helps handle this aspect.

PLM	top-5	top-10
BERT$_{large}$ (c)	**0.506**	**0.558**
RoBERTa$_{large}$ (c)	0.485	0.530
ALBERT$_{large}$ (u)	0.396	0.465
DistilBERT (u)	0.465	0.541
	near-5	**near-15**
GPT-Neo (2.7B)	0.181	0.272
Falcon (7B)	0.194	**0.343**
Falcon-instruct (7B)	**0.220**	0.328
MPT (7B)	0.208	0.314

Table 3: Comparative Evaluation over the Datasets (Macro-Averaged over individual fact-types)

4.1 Overall Analysis

As can be observed from Table 3, the BERT encoder model works well and mostly outperforms all other encoder and generative models on different evaluation metrics. The RoBERTa model is close second and also performs relatively well. Distilled encoders ALBERT and DistilBERT demonstrate comparable performance in the top-10 met-

[2] https://pypi.org/project/transformers/

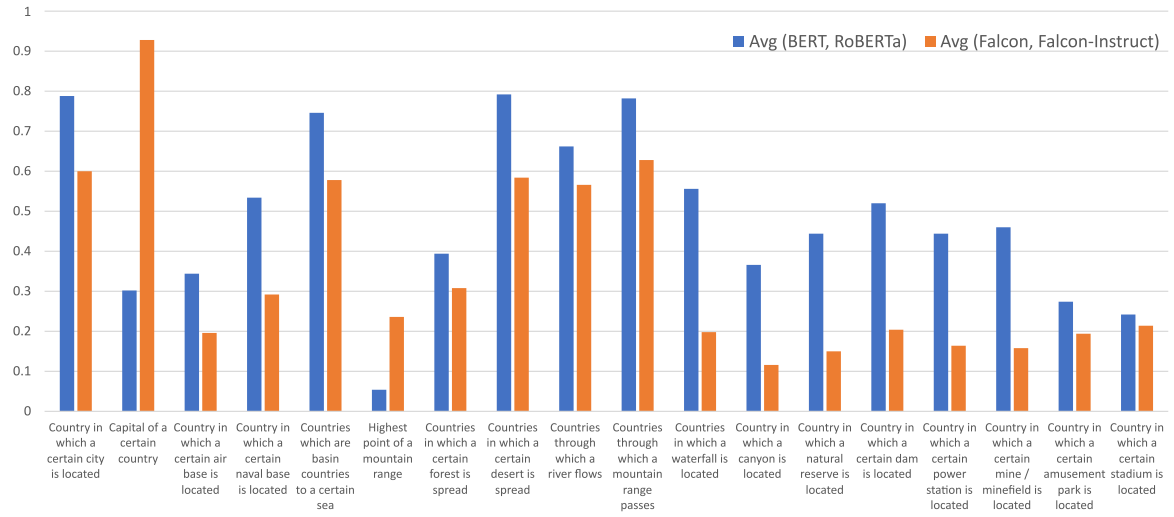

Figure 1: Performance (Averaged top-5/near-5 across major Fact-Types

ric. Specifically in case of generative models, the Falcon and Falcon-instruct models show better performance over the smaller GPT-Neo and the equivalently sized MPT. The MPT model is however better than the Falcon one as per the near-5 metric.

We define major fact-types as the subset of all considered types which consists of 20 or more masked sentences. To analyze the comparative difficulty of the fact types, we plot in Figure 1 the average of top-5 scores for 2 encoder models (BERT and RoBERTa) and near-5 values for 2 generative models (Falcon and Falcon-instruct) for each of the major fact-types. In most fact-types we observe that the encoder models perform better than the generative models. We can also observe that the *Public and Industrial Facilities* related facts are the most difficult category with both kinds of models finding it difficult to answer the prompts. This is probably because of low discourse on these entities in the LM's pre-training data. The most difficult fact type overall is - *Highest Point of a mountain range* under *Natural*, though on this fact type the generative models perform better than the encoder ones. Other difficult fact types are - *Country in which a certain stadium is located* and *Country in which a certain amusement park is located.*

Overall, the location of cities in countries and multiple others under the head `Natural` such as location of mountain ranges, rivers and deserts in countries are the major fact types, of which both encoder and generative models are aware of. It is not difficult to realize that information regarding these fact types is most frequently seen in the text on the web in multiple contexts such as news, Wikipedia and blogs, allowing the PLMs to learn this information multiple times and in turn helping them answer these prompts with high accuracy. Particularly for the fact-type `Capital of a certain country`, the generative models beat the encoder models by a significant margin. This is an interesting finding and may be attributed to better learning of contextual attention between the country name, the word *capital* and capital names by the larger models. On the other hand, fact-types regarding locations of natural entities such as reserves and waterfalls and locations of industrial sites such as dams and power stations, are better answered by the encoder models. We plan to investigate this performance gaps in further detail as part of future work.

The results indicate a good scope for efforts required for tuning the models for better performance on this knowledge. The top-5/near-5 scores for all the PLMs considered are either around or less than 0.6, which means that the answer arrives late in the rank list or further away and hence, improvements to push the answer up the rank list are possible.

4.2 Detailed Analysis

In Table 4, we present the top-5 values for the BERT encoder model and near-5 values for Falcon-Instruct generative model, for the major fact types. In most cases, the top-5 values of the BERT model are better than the near-5 values of the Falcon-Instruct model. We try explaining the gap by examining some example prompts and their completions where there is significant difference between the

Fact Type	BERT Large (cased)	Falcon Instruct (7B)
Geo-political		
Country in which a certain city is located	0.819	0.453
Capital of a certain country	0.418	0.922
Country in which a certain air base is located	0.314	0.230
Country in which a certain naval base is located	0.494	0.315
Natural		
Countries which are basin countries to a certain sea	0.766	0.59
Highest point of a mountain range	0.054	0.156
Countries in which a certain forest is spread	0.294	0.303
Countries in which a certain desert is spread	0.632	0.5
Countries through which a river flows	0.547	0.541
Countries through which a mountain range passes	0.597	0.599
Countries in which a water-fall is located	0.362	0.171
Country in which a canyon is located	0.238	0.087
Country in which a natural reserve is located	0.293	0.144
Public and Industrial Facilities		
Country in which a certain dam is located	0.410	0.123
Country in which a certain power station is located	0.299	0.192
Country in which a certain mine / minefield is located	0.299	0.124
Country in which a certain amusement park is located	0.166	0.282
Country in which a certain stadium is located	0.075	0.279

Table 4: top-5 comparison for major fact-types

values.

In the fact type on *Capital of a certain country*, the Falcon-Instruct model outperforms the BERT model by a very large margin. On close observation of the answers, we observe a peculiar behavior of the BERT model. In multiple instances it predicts other larger and famous cities of the country instead of the capital. For e.g., it predicts `Saigon` in case of `Vietnam has its capital city as [MASK].` which is another name for Ho-Chi-Minh city, the largest city in Vietnam located south of the actual capital `Hanoi`. Similarly it predicts `Karachi`, `Lahore` and `Sindh` instead of `Islamabad` as `Pakistan`'s capital. Another kind of inaccuracy we observed was that it was predicting, higher up the list, capitals of related countries which are more famous instead of the country under consideration. For example, for the sentence `Kazakhstan has its capital city as [MASK].`, it predicted cities such as `Baku` (Azerbaijan's capital), `Beijing` (China's capital) and `Minsk` (Belarus' capital). Similarly it predicted `Dhaka`, `Bangkok` and `Kolkata` as `Myanmar`'s capital (in place of `Naypyidaw`).

We now investigate an example under the head Public/Industrial sites, where the BERT model outperforms the Falcon-Instruct model. For instance in the fact-type `Country where a certain power station is located`, we observe this performance gap. On examination of the answers, we find that for the template `The entity`$_S$ `Power Station supplies electricity to states in [MASK]`, the generative model prefers to generate `the midwest` or `the midwestern`. Probably this is because that it gets biased by the phrase `states in` and completes it not with a specific country/location but more general text. In some instances, it generates an entire region/area as the answer instead of a specific country. For example, for the sentence `The Gobo Thermal Power Plant supplies electricity to states in`, it generates: `the Gobo basin region.`, instead of `Japan`. Mapping this generic answer to a specific country would require non-trivial reasoning and hence it is difficult to give it a benefit of doubt even during evaluation. Similar observations were marked from other fact-types such as `Country where a certain mine/ mine-field is located`. The generative model either referred to larger regions or entities for e.g. (`The Drmno mine is located in`, *the Dnepropetrovsk region*) & (`The Yanacocha mine is located in`, *the Andes, Mountains*) or entirely incorrect predictions (`The Ombilin coal mine is located in`, *the province of South Africa*) & (`The Gargamel mine is located in`, *the fictional town of Garg*).

5 Note on other PLM Probing benchmarks

An interesting research direction is gauging whether LLMs can replace Knowledge Graphs and latest work such as (Sun et al., 2023) conclude that such replacement is far from reality. This calls for increasing research focus towards making

LLMs more knowledgeable both generally and domain-wise. In the current context, it hence becomes important to highlight the need for a specific geography focused PLM probing dataset when there are several PLM probing benchmarks available in the literature (Petroni et al., 2019; Lin et al., 2020; Aroca-Ouellette et al., 2021). Firstly, to the best of our knowledge this is the first only geography focused PLM probing benchmark. Secondly, we believe that the existing ones cover a wide variety of general facts and information and hence for this focused domain, they would cover only a limited set of facts. Specifically, we discuss one of the foremost ones - the LAMA benchmark (Petroni et al., 2019). The LAMA benchmark considers four sources to build their probing benchmark out of which only the ConceptNet and TRex sources consist of concepts related to Geography or Spatial. A closer exploration of the ConceptNet source reveals that the LAMA authors include a "AtLocation" relation but the facts checked are too general, such as "Something you find at the [MASK=library] is reference materials.". This is different from the current goal of discovering geography knowledge. The TRex source comes closer to our method and considers a set of Wikidata based relations including a few geography ones. However, the TRex's procedure maps a given Wikidata triple to multiple sentences from Wikipedia text (Wiki text) sentences. LAMA's procedure selects one of the multiple candidates randomly for probing, which may include other facts about the entities in the triples not necessarily relevant to geography. E.g., Entities Egypt and Africa occurring in non-geographical context as in the TRex sentence - `The song's lyrics of unity mention a number of countries, including England, Russia, China, Egypt and Israel, as well as the continent of [MASK=Africa].` Moreover, probing PLMs, which have seen Wikipedia text as part of their training data, on masked sentences made form Wikipedia text itself might give them an advantage as compared to our template-based generation of masked sentences which would be different structure wise from the training data, leading to a more effective probing exercise.

Another closely related benchmark dataset is the GeoGLUE (Li et al., 2023), which also presents a set of evaluation tasks to gauge geographic language understanding, but is primarily in Chinese. Other relevant literature focused on spatial and geography knowledge exploration in text though focused largely towards Question Answering is discussed in Mirzae et al. (2021), Li et al. (2021) and Contractor et al. (2019).

6 Limitations (and Future Work)

We are aware about the facets and avenues that the current exercise doesn't consider and they remain to be explored in detail. A few important ones form part of the Future Work are listed as follows:

Penchant for Hardware Poor: Currently we do not include very large models such as the 13/40 billion or even larger models due to our goal of exploring resource poor and license friendly PLMs. This allows us better reach, deployment and use-case wise. However, we do plan to include larger models in the benchmarking exercise.

Fine-tuning: The focus on using smaller models also prompts us to improve the performance through different PLM fine-tuning techniques. A comprehensive Fine-Tuning exercise is underway and will be separately discussed.

Deeper Understanding: Investigation into the attention patterns of the LMs' transformer blocks might be necessary to gain deeper insight into what conspires when geography prompts are seen by the LM. We plan to employ AttentionFlows (DeRose et al., 2020) and AttentionViz (Yeh et al., 2023) in this regard.

Better Templates: Currently the templates are encoder model friendly as we started with these models and are catching up with the more recent LLMs. This urges us to design better templates which can work seamlessly for both encoder and generative models.

7 Conclusion and Future Work

We aimed at evaluating the learning of pre-trained language models in the space of geography knowledge. To carry out the evaluation we created a probing dataset of 5268 masked sentences based on Wikidata triples. Using the masked token prediction and text generation tasks, we probe 8 different PLMs (4 encoders and 4 decoders) and report the results. We observe that encoder models such as BERT showcase relatively better knowledge of Geography facts than the generative models considered. We elaborate the results through various analyses and examples of fact-types and prompts where the PLMs perform well and otherwise.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. Prost: Physical reasoning of objects through space and time. *arXiv preprint arXiv:2106.03634*.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Sijie Cheng, Zhiyong Wu, Jiangjie Chen, Zhixing Li, Yang Liu, and Lingpeng Kong. 2022. Unsupervised explanation generation via correct instantiations. *arXiv preprint arXiv:2211.11160*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling Instruction-finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Danish Contractor, Krunal Shah, Aditi Partap, Parag Singla, et al. 2019. Large scale question answering using tourism data. *arXiv preprint arXiv:1909.03527*.

Joseph F DeRose, Jiayao Wang, and Matthew Berger. 2020. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1160–1170.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making Pre-trained Language Models better Few-shot Learners. *arXiv preprint arXiv:2012.15723*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942*.

Dongyang Li, Ruixue Ding, Qiang Zhang, Zheng Li, Boli Chen, Pengjun Xie, Yao Xu, Xin Li, Ning Guo, Fei Huang, et al. 2023. Geoglue: A geographic language understanding evaluation benchmark. *arXiv preprint arXiv:2305.06545*.

Haonan Li, Ehsan Hamzei, Ivan Majic, Hua Hua, Jochen Renz, Martin Tomko, Maria Vasardani, Stephan Winter, and Timothy Baldwin. 2021. Neural factoid geospatial question answering. *Journal of Spatial Information Science*, 23(23):65–90.

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. *arXiv preprint arXiv:2005.00683*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. Spartqa:: A textual question answering benchmark for spatial reasoning. *arXiv preprint arXiv:2104.05832*.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A Framework for Adapting Transformers. *arXiv preprint arXiv:2007.07779*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*.

Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. 2023. Attentionviz: A global view of transformer attention. *arXiv preprint arXiv:2305.03210*.

A Post-processing of template based prompts

A straightforward placement of the subject and object entities in a lexical template is not sufficient to arrive at clean and noise-free prompts. This is because of repetitions of words that can happen because of their presence both in the tokens of the entity (obtained as-is from Wikidata) and in the template. For example, in case of rivers, some values in Wikidata explicitly have the mention of the word "river" at the end, for e.g. `Jhelum River` and some values simply mention the name of the river without the qualifier noun, for e.g. `Nile`. Now, in the relevant template - The entity$_S$ river flows through entity$_O$, the former case would lead to creation of the prompt as `The Jhelum River river flows through [MASK].` Such repetition is undesirable and needs to be handled before the prompt can be tried on a PLM. Such repetition can happen not only with same words but also with words which are different but contextually similar. For e.g., `The Everland Resort amusement park is located in [MASK].`, where not having the qualifier "amusement park" after resort would have made a better prompt - `The Everland Resort is located in [MASK].` To handle such repetition, we first manually observe all entity names and identify all possible instances where such repetitions can occur. In Table 5, we report all such suffix tokens which if present in the entity value, we mute the tokens in the corresponding template which would cause repetition. We post-process the generated template based prompts for handling all these different repetitions to arrive at the final prompts.

Entity (Tokens muted in the templates)	Suffix tokens that are observed in entity values
river	river
dam	dam, station, plant, barrage, reservoir
mountain range	mountains, ghats, range, ranges, highlands, hills, escarpment
forest	forest, forests, park, forest complex, plateau, woodlands, woodland, wilderness, recreation area
desert	desert, dunes, sand sea, scablands, scabland
waterfall	falls, fall, waterfall, waterfalls
canyon	valley, canyon, gorge, valleys, canyons, gorges, dells, ravine, ravines
amusement park	park, resort, resorts
stadium	stadium, arena, ground, sports complex, convention center
mine	mine, mines, quarry

Table 5: Template token muting for Preventing Repetition

B Complete list of fact types captured in the dataset (Table 6)

C Example Triples and masked sentence instantiation (Table 7)

D Brief Description of the PLMs considered

BERT: Bidirectional Encoder Representatations for Transformers (Devlin et al., 2018) model is a transformer which is trained on 16 GB of Books and Wiki data using the Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) tasks. The large version has 340M parameters and the base one has 110M parameters. We probe all the four versions of BERT namely BERT$_{base}$ uncased, BERT$_{base}$ cased, BERT$_{large}$ uncased and BERT$_{large}$ cased.

RoBERTa: Robust BERT or RoBERTa (Liu et al., 2019), use the similar architecture but significantly higher amount of training data (160 GB) also training the model on better compute resources for a longer period of time. The training is only based on the MLM task with dynamic masking. As RoBERTa models are cased, we experiment with the RoBERTa$_{base}$ cased and RoBERTa$_{large}$ cased models.

ALBERT (Lan et al., 2019): This model changes the original BERT architecture by introduction of shared parameters and low dimension projections of the high dimensional embedding space, thereby leading to a reduction of nearly 90 million parameters. The model training also involves a sentence order prediction task with about 10 times more data than on which BERT is trained. As AL-

Fact Type	Example	Wikidata property ($prop_p$)	Example Triple ($entity_S$, $prop_p$, $entity_O$)
Geo-political			
Country in which a certain city is located.	London is located in England	*country* (P17)	(London, P17, England)
Continent in which a certain country is located	Japan is located in Asia.	*continent* (P30)	(Japan, P30, Asia)
Capital of a certain country	Tokyo is the capital of Japan.	*capital* (P36)	(Japan, P36, Tokyo)
Country in which a certain air base is located	The Edwards Air Force Base is located in USA.	*country* (P17)	(Edwards Air Force Base, P17, USA)
Country in which a certain naval base is located	The Erdek Naval Base is located in Turkey.	*country* (P17)	(Erdek Naval Base, P17, Turkey)
Waterbody in which a certain naval base operates	The Bandar Abbas Naval Base operates in the waters of the Persian Gulf.	*located in or next to a body of water* (P206)	(Bandar Abbas Naval Base, P206, Persian Gulf)
Natural			
Countries which are basin countries to a certain sea	Rivers from Greece and Turkey flow into the Aegean Sea.	*basin country* (P205)	(Aegean Sea, P205, Greece), (Aegean Sea, P205, Turkey)
Highest point of a mountain range	Mount Everest is the highest point of the Himalayan mountain range.	*highest point* (P610)	(Mount Everest, P610, the Himalayas)
Countries in which a certain forest is spread	The Sundarban forest is spread over India and Bangladesh.	*country* (P17)	(Sundarbans, P17, India), (Sundarbans, P17, Bangladesh)
Continent in which a certain desert is located	The Sahara desert is located on the African continent.	*continent* (P30)	(Sahara desert, P30, Africa)
Countries in which a certain desert is spread	The Gobi Desert is spread over China and Mongolia.	*country* (P17)	(Gobi Desert, P17, China), (Gobi Desert, P17, Mongolia)
Countries through which a river flows	The Danube flows through Germany.	*basin country* (P205)	(Danube, P205, Germany)
Countries through which a mountain range passes	The Atlas mountain range passes through Algeria, Morocco and Tunisia.	*country* (P17)	(Atlas mountain range, P17, Algeria), (Atlas mountain range, P17, Morocco)
Countries in which a waterfall is located	The Rhine Falls is located in Switzerland.	*country* (P17)	(Rhine Falls, P17, Switzerland)
Country in which a canyon is located	The Kings Canyon is located in Australia.	*country* (P17)	(Kings Canyon, P17, Australia)
Waterbody which has created a canyon	The Grand Canyon is created by the Colorado River.	*located in or next to a body of water* (P206)	(Grand Canyon, P206, Colorado River)
Country in which a natural reserve is located	The Rila National Park is located in Bulgaria.	*country* (P17)	(Rila National Park, P17, Bulgaria)
Public and Industrial Facilities			
Waterbody on which a certain dam is located	The Aswan dam is located on the Nile river.	*located in or next to a body of water* (P206)	(Aswan dam, P206, the Nile)
Country in which a certain dam is located	The Aswan dam is located in Egypt.	*country* (P17)	(Aswan dam, P17, Egypt)
Country in which a certain power station is located	The Turbigo Power Station is located in Italy.	*country* (P17)	(Turbigo Power Station, P17, Italy)
Country in which a certain mine / minefield is located	The Grasberg Mine is located in Indonesia.	*country* (P17)	(Grasberg Mine, P17, Indonesia)
Country in which a certain amusement park is located	The Wonderland Amusement Park is located in China.	*country* (P17)	(Wonderland Amusement Park, P17, China)
Country in which a certain stadium is located	The Stadium of Light is located in England.	*country* (P17)	(Stadium of Light, P17, England)

Table 6: Fact Types with Examples, Corresponding Wikidata properties and Example Triples

BERT models are uncased, we experiment with the ALBERT_{base} uncased and ALBERT_{large} uncased models.

DistilBERT (Sanh et al., 2019): This is a model learnt on the same amount of data as BERT, but the learning is through distillation wherein the posterior probabilities in the prediction tasks learnt by BERT are approximated by a smaller network

Example Triple	Templates
Geo-political	
(London, P17, England)	entity$_S$ is a city located in entity$_O$.
Realization: London is a city located in [MASK]. Answer: England	
(Japan, P30, Asia)	entity$_S$ is part of the entity$_O$ continent.
(Japan, P36, Tokyo)	(i) entity$_O$ is the capital of entity$_S$. (ii) entity$_S$ has its capital city as entity$_O$.
(Edwards Air Force Base, P17, USA)	(i) The entity$_S$ serves the Air Force of entity$_O$. (ii) The entity$_S$ is located in entity$_O$.
(Erdek Naval Base, P17, Turkey)	(i) The entity$_S$ serves the Navy of entity$_O$. (ii) The entity$_S$ is located in entity$_O$.
(Bandar Abbas Naval Base, P206, Persian Gulf)	The entity$_S$ operates in the waters of the entity$_O$.
Natural	
(Aegean Sea, P205, Greece), (Aegean Sea, P205, Turkey)	(i) Rivers from countries such as entity$_{O1}$ and entity$_{O2}$ flow into the entity$_S$. (ii) The entity$_S$ is bound by countries such as entity$_{O1}$ and entity$_{O2}$.
Realization (i): Rivers from countries such as [MASK] and Turkey flow into the Aegean Sea. Answer: Greece Realization (ii): Rivers from countries such as Greece and [MASK] flow into the Aegean Sea. Answer: Turkey	
(Mount Everest, P610, the Himalayas)	(i) entity$_O$ is the highest point of the entity$_S$ mountain range (ii) The highest point of the entity$_S$ mountain range is entity$_O$
(Sundarbans, P17, India), (Sundarbans, P17, Bangladesh)	The entity$_S$ forest is spread over countries such as entity$_{O1}$ and entity$_{O2}$.
(Sahara desert, P30, Africa)	The entity$_S$ desert is part of the entity$_O$ continent.
(Gobi Desert, P17, China), (Gobi Desert, P17, Mongolia)	The entity$_S$ desert is spread over countries such as entity$_{O1}$ and entity$_{O2}$.
(Danube, P205, Germany)	(i) The entity$_S$ river flows through entity$_O$. (ii) entity$_O$ has the entity$_S$ river as one of its rivers.
(Atlas mountain range, P17, Algeria), (Atlas mountain range, P17, Morocco)	(i) The entity$_S$ mountain range passes through countries such as entity$_{O1}$ and entity$_{O2}$. (ii) The entity$_S$ mountain range runs through various countries such as entity$_{O1}$ and entity$_{O2}$.
(Rhine Falls, P17, Switzerland)	The entity$_S$ waterfall is located in countries namely entity$_{O1}$ and entity$_{O2}$.
Kings Canyon, P17, Australia)	The entity$_S$ canyon is spread over countries namely entity$_{O1}$ and entity$_{O2}$.
(Grand Canyon, P206, Colorado River)	The entity$_S$ canyon is created by water bodies namely entity$_{O1}$ and entity$_{O2}$
(Rila National Park, P17, Bulgaria)	The entity$_S$ is spread over countries such as entity$_{O1}$ and entity$_{O2}$.
Public and Industrial Facilities	
(Aswan dam, P206, the Nile)	(i) The entity$_S$ dam is located on the entity$_O$ river. (ii) The entity$_S$ dam bounds the flow of the entity$_O$ river.
Realization (i): The Aswan dam is located on the [MASK] river. Realization (ii): The Aswan dam bounds the flow of the [MASK] river.	
(Aswan dam, P17, Egypt)	The entity$_S$ dam is located in entity$_O$.
(Turbigo Power Station, P17, Italy)	(i) The entity$_S$ supplies electricity to states in entity$_O$. (ii) The entity$_S$ is located in entity$_O$.
(Grasberg Mine, P17, Indonesia)	(i) The entity$_S$ mines are spread over countries such as entity$_{O1}$ and entity$_{O2}$. (ii) The entity$_S$ mine is located in entity$_O$.
(Wonderland Amusement Park, P17, China)	The entity$_S$ amusement park is located in entity$_O$.
(Stadium of Light, P17, England)	The entity$_S$ stadium is located in entity$_O$.

Table 7: Templates to convert wikidata triples to masked sentences

(with half the number of parameters). DistilBERT achieves about 97% of BERT's performance on benchmark tasks. As a single base version is available, we experiment with the DistilBERT$_{base}$ uncased and DistilBERT$_{base}$ cased variants.

GPT-Neo-2.7B (Black et al., 2021): GPT-Neo 2.7B is a transformer model designed using EleutherAI's replication of the GPT-3 architecture and has 2.7 billion parameters. It was trained on the Pile, a large scale curated dataset created by EleutherAI. This model was trained for 420 billion tokens over 400,000 steps and was trained as a masked autoregressive language model, using cross-entropy loss.

Falcon and Falcon-Instruct (Almazrouei et al., 2023): Falcon-7B is a 7 billion parameters causal decoder-only model built by TII and trained on 1,500B tokens of RefinedWeb enhanced with curated corpora. The Falcon-Instruct version is fine-tuned on a mixture of chat/instruct datasets and hence the name 'instruct'. The Falcon family of

models also has a larger 40 billion model.

MPT-7B (MosaicML, 2023): MPT-7B is a 7 billion parameter decoder-style transformer pretrained from scratch on 1T tokens of English text and code by MosaicML. The MPT models use a modified transformer architecture optimized for efficient training and inference. These architectural changes include performance-optimized layer implementations and the elimination of context length limits by replacing positional embeddings with Attention with Linear Biases (ALiBi).

Experimenting with Encoder-Decoder models As can be observed, we have excluded PLMs which are encoder-decoder models. In our initial set of experiments we did include LMs namely BART and Flan-T5 (XL version) and used their text generation capability for the probing exercise (as their encoder layers are primarily used for representations and not MLM like tasks). We encountered some specific issues. For example, in case of Flan-T5, on providing it all possible instructions for answer generation mentioned either as part of the Flan-T5 paper (Chung et al., 2022) or as examples in their HuggingFace webpage, it was unable to generate proper answers. We tried multiple different instructions which Flan-T5 is already made aware duruing training such as (i) `Please answer the following question. The Turbigo Power station supplies power to states in?` (ii) `Q: The Turbigo Power station supplies power to states in? A:` and (iii) `Please answer the following question. What token best fills the [MASK] token in the sentence:The Turbigo Power station supplies power to states in [MASK].` But for none of these variations was an answer found leading to zero hits in all of near-1, near-5 and near-10 metrics. We believe this calls for developing special templates which would cast the Wikidata triple as a Wh-question, but we keep this investigation as part of Future work.

E Comparing different Prompt Instructions

We observed a peculiar behavior in all generative models that providing them with the geography sentence for completion as the prompt itself without any instruction leads to very low performance. Hence, it became imperative to prepend them an instruction to form the prompt and extract an appropriate answer. To decide on a suitable prompt, we carry out a small exercise. We evaluate the GPT-Neo 2.7B model on the dataset with 5 different prompts. We then select the one that works the best and use it for all models to keep the results comparable. The different prompt instructions we tried and the corresponding results of the GPT-Neo-2.7B model on the set of 3650 generative sentences are reported in Table 8. We observe that both instruction type 2 (second row in Table 8) and type 4 prompts worked the best and the type 2 one was employed in the experiments.

Instruction	on-top-15
Complete the following sentence:	0.236
For the following sentence about geography, generate the most probable text to complete it.	0.300
Generate the most probable text to complete the following sentence.	0.277
Complete the following geography fact.	0.300
Answer the question (with a '?' appended to the sentence)	0.167

Table 8: Different Instructions and Corresponding GPT-Neo-2.7B results