Extracting Geographic Knowledge from Large Language Models: An Experiment

Konstantinos Salmas[1], Despina-Athanasia Pantazi[1] and Manolis Koubarakis[1]

[1]*Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens*

Abstract

We perform an experimental analysis of how the inner architecture of large language models behaves whilst extracting geographic knowledge. Our aim is to conclude on whether models actually incorporate geospatial information or simply follow statistical patterns in the data; hence to contribute to the research area of creating knowledge graphs from large language models. To achieve this, we study one specific geospatial relation and explore different techniques that leverage the masked language modeling abilities of BERT and RoBERTa. Our study should be construed as a stepping stone to the general study of the ways large language models encapsulate geospatial knowledge. In addition, it has allowed us to observe important points one should focus on when querying language models, which we discuss in detail.

Keywords

large language models, geospatial data, geospatial knowledge, knowledge graphs

1. Introduction

There has been a lot of interest recently in the development of large language models (LLMs) and their relationship to knowledge graphs (KGs) [1]. LLMs such as BERT [2] and RoBERTa [3] were pre-trained in a self-supervised manner using a great amount of textual corpora. In the probing setting [4], LLMs can fill a masked word included in a text sequence to extract a relational assertion for a given subject [5]. For example, BERT successfully fills the sentence *"Athens is located in"* with *"Greece"*, which can be preceived as a KG subject-predicate-object triple `<Athens, located in, Greece>`.

Witnessing that LLMs are constantly evolving and that they have been exposed on a vast quantity of data, the question of whether we can extract their knowledge and automatically create KGs arises. An analysis of the factual and commonsense knowledge in publicly available pretrained language models was published in [1], where the authors concluded that it is not trivial to extract a KG from them. Another work proposed the injection of factual knowledge into BERT [6], [7] discussed how we can create structured KBs from LLMs, and [8] explored how robustly world knowledge is stored in LLMs.

In this paper, we are interested in *geospatial* KGs [9] and their relationship to LLMs. Geospatial KGs include YAGO2 [10], YAGO2geo [11], WorldKG [12], KnowWhereGraph [13], the geospatial extension of YAGO4 of [14] and others. Geospatial KGs contain information about *geographic entities* (e.g., the city of Athens) and their *geospatial* (e.g., Athens' geometry in the WGS84

KBC-LM'23: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2023

sdi1700133@di.uoa.gr (K. Salmas); dpantazi@di.uoa.gr (D. Pantazi); koubarak@di.uoa.gr (M. Koubarakis)

CEUR Workshop Proceedings (CEUR-WS.org)

coordinate reference system) and *thematic* characteristics (e.g., Athens' population or the name of its mayor). Geospatial KGs can store both *qualitative* (e.g., Thessaloniki being north of Athens), and *quantitative* (e.g., their distance being 504km) knowledge about geographic entities.

LLMs like BERT and RoBERTa have been trained on Wikipedia, book corpora, web text etc. As these sources (especially Wikipedia) contain a lot of geospatial knowledge like the one stored in geospatial KGs, it is very interesting to explore the LLMs' capabilities in this regard. For example, there is a lot of qualitative knowledge about geographic entities in Wikipedia, such as cardinal direction knowledge to describe the relative position of various entities on the world map (e.g., "Bulgaria borders Greece to the north"). Additionally, many geospatial properties included in Wikipedia can be implicitly reasoned through different texts. For instance, Athens is in Greece (Europe) and Accra is in Ghana (Africa). The conclusion that Athens is north of Accra can be inferred by the additional fact that Europe is north of Africa. Understanding such relations and being able to easily answer qualitative geospatial questions that depend on them can be an important capability of LLMs.

Motivated by the above discussion, we seek to broaden our perspective regarding the ways LLMs can answer geospatial questions. The contributions of the paper are the following:

- We carry out an experiment regarding the ways LLMs can answer qualitative geospatial questions of the form *X is a city in Y*. To achieve this, we exploit the fill-the-mask pipeline on the pre-trained models BERT and RoBERTa without fine-tuning them. In this way, we aim to understand if LLMs are actually able to answer simple qualitative geospatial questions correctly.
- We present our findings on answering geospatial questions using LLMs, and we experiment with the effect different layers and their attention heads have on the final results. We also explore different techniques on mask filling while retrieving the answers from the LLMs. To support our findings, we include detailed results for each experiment. In addition, in Appendix A we conduct a complementary study to further analyse the results of our experimental evaluation concerning different variations of LLMs.

The rest of the paper is structured as follows: Section 2 discusses related work, while Section 3 gives a detailed overview of the methodology we followed to show if the LLMs studied can answer geospatial questions. Section 4 presents the experimental evaluation we conducted. Section 5 concludes the paper and makes recommendations for extending this work. In Appendix A we present our complementary experiments using different variations of the LLMs. As a contribution to the research community we release our source code[1].

2. Related Work

Recent works have studied the possibility that LLMs could be used for KG construction and augmentation. Generally the LM-as-KG paradigm embodies three different approaches: prompt-base retrieval, case-based analogy, and context-based inference [15]. In prompt-base retrieval, one masks the desired answer and deems the returned token to be the answer; e.g., Paris is the capital of [MASK]. In case-based analogy, the prompt includes an example prior to the mask one

[1] https://github.com/AI-team-UoA/gsp-knowledge-extraction-llms

wants filled; e.g., Athens is the capital of Greece. Paris is the capital of [MASK]. In context-based inference, the prompt is enriched with relating information; e.g., Athens is in Greece.

Petroni et al. [1] introduced the LAMA probe, which is a prompt-based retrieval approach. Using the LAMA probe, they explored the factual knowledge an LLM encapsulates, simply from its pre-training procedure. Their contribution consists of a systematic analysis which reaches the conclusion that BERT-large is better at knowledge extraction compared to its competitors, that relation extraction performance is not easily improved simply by increasing the data volume, and finally, that we need better understanding regarding aspects of knowledge that LLMs capture. Wang et al. [7] moved one step further and proposed a framework that can construct KGs from LLMs. Their approach suggests a single forward pass of the LLMs without fine-tuning through textual corpora. The proposed MaMa (Match and Map) framework consists of two stages that result in an open KG with mapped facts being in a fixed schema, while unmapped ones being in an open schema. They argue that the resulting KGs (with a measured precision of more than 60%) indicate that their approach is reliable. Hao et al. [16] presented a more advanced prompt-based retrieval approach by introducing a framework that can construct a relational KG via an LLM without textual corpora parsing, simply by the utilisation of some examples and a prompt. They paraphrase the initial prompt and use the alternatives to find out which of them can help the LLM to effectively produce valid answers through the use of the said examples and the score they achieve. Their approach leverages the masked language modeling (MLM) abilities of LLMs and retrieves knowledge via fill-the-mask tasks.

Razniewski et al. [5] argue that LLMs should be a means to curate and augment KBs and not simply replace them. They propose some pragmatic and intrinsic considerations such as a common bias of the aforementioned techniques, namely the lack of disambiguation between statistical correlation and explicit knowledge. Considerable attention has also been paid to the inner workings of the LLMs: their layers and the corresponding attention heads. Clark et al. [17] hypothesise that some attention heads of BERT-base appear to behave in specific patterns that could indicate that BERT learns syntactic dependencies of the English language. A similar study has been conducted by Kovaleva et al. [18], also focusing on BERT's self attention mechanisms, suggesting that BERT can benefit from attention heads disabling in some tasks.

In the area of KGs, there is research on the enhancement of KGs along temporal and spatial dimensions, the latter being the topic of this paper. YAGO2 [10] is such an example that extends the classic Subject-Property-Object (SPO) triples adding Time and Location. It was further extended with richer geospatial knowledge (not just coordinates) in YAGO2geo [11] and YAGO4 [14]. Other recent approaches to geospatial knowledge graphs are WorldKG [12] and KnowWhereGraph [13].

As per the geospatial abilities of LLMs, Roberts et al. [19] probed GPT4 which performed generally well but remains unclear if it did so by reasoning or simple memorization. Cohn et al. [20] conducted dialectal evaluations on state-of-the-art models and suggest the models do not always succeed in spatial reasoning. Faisal et al. [21] proposed a framework to examine LLMs' geographic knowledge and biases. They seemed to understand geographic proximity but serious limitations exist. Hoffman et al. [22] propose *geoadaptation*, a task-agnostic training step performed on pretrained LLMs which they argue allows models to learn geographic and dialectal knowledge. Finally, Mai et al. [23] discuss the development of a foundation model for geospatial AI and introduce a framework to achieve such goal.

3. Methodology

We focus on Transformer-based language models that have been pre-trained through the masked language modeling (MLM) paradigm. BERT [2] was trained on BookCorpus (800M words) [24] and the English version of Wikipedia (2,500M words), excluding lists, headers and tables. As for the MLM tasks, 15% of the tokens were masked (i.e., replaced with the special [MASK] token). More specifically, in 10% of that 80%, the masked token was replaced with another random token and in 10% it was left unchanged. BERT was also pre-trained on Next Sequence Prediction (NSP). In this task, the model should predict if a sentence A was following sentence B or not. RoBERTa [3] follows similar training techniques (MLM, NSP) and almost identical architecture to BERT. However, the authors have changed some major points; the MLM is performed via dynamic masking and tokenization is replaced with byte-pair encoding (BPE). Finally, the data upon which it was trained (apart from that BERT used) include CC-News[2], OpenWebText[3] and Stories [25]. We use the *pretrained* versions of BERT and RoBERTa without fine-tuning them.

In the rest of the paper we try to answer the question: Can BERT and RoBERTa answer the very simple geospatial question *"Is X a city in Y?"*. The question is posed as a *geospatial phrase (GSP)* of the form *X is a city in Y*. For example, if an LLM has learned that "Athens is a city in Greece" it should be able to answer the GSP *Athens is a city in Y*.

We chose to carry out this very simple experiment since *"in"* is an important topological relation in all qualitative spatial reasoning models [26] and, as such, it deserves to be studied first. Also, *"city"* is an important class of geographic features and there is plenty of knowledge about cities and the administrative divisions they belong to (e.g., states of countries) in the data BERT and RoBERTa have been trained on (e.g., Wikipedia or BookCorpus).

3.1. Knowledge extraction settings

Layers. BERT-like models take a sequence of tokens as an input and pass it through their inner layers. When they are used for MLM, a specific head is added on top of the models that takes the contextual embeddings as an input, passes them through a feed-forward neural network (FNN) and returns a sequence of predicted tokens. We aim to explore how the answers are constructed at each layer of the model. In order to record that, we changed the default forward function of these models to have the answer from a layer of our choice. When asking a model with K layers to fill the masked tokens from the layer N, we actually allow the model to use all layers $1 \leq i \leq N$ and then bridge the gap between the remaining layers and the MLM head (i.e., layers $N < j \leq K$ were not used at all). If one requests to get an answer from the K^{th} layer, the process is identical to a simple *fill-the-mask* task in which the model would use all of its layers to produce the outcome.

Top-K answers. A softmax function is applied to the embeddings every BERT-like model returns. These embeddings are then sorted and the *top-k* of them (along with the confidence of the model) are kept as the most probable answers. We tampered with a few different *top-k* values, but we settled to a *top-k* value of 10 and 100. Note that a large model (24 layers) with *top-k=100* would return a total of 2400 answers.

[2] https://commoncrawl.org/2016/10/news-dataset-available/

[3] https://skylion007.github.io/OpenWebTextCorpus/

Multi-mask filling methods. Some geospatial relations like the relation *"in"* we are working with, require more than one mask to be filled, e.g., *[MASK] is a city in [MASK].* We test two different approaches as to how the full answer would be constructed, as described below:

- **Left-To-Right (LTR)**: Firstly, the model fills the left most mask and proceeds to the remaining ones on the right. For example:
  [MASK] is a city in [MASK] $\longrightarrow$ ***Athens** is a city in [MASK]* $\longrightarrow$ ***Athens** is a city in **Greece**.*

- **Right-To-Left (RTL)**: This is the exact opposite of LTR and starts the process of filling from the right most [MASK] token. The above example would be reformulated as:
  [MASK] is a city in [MASK] $\longrightarrow$ *[MASK] is a city in **Greece*** $\longrightarrow$ ***Athens** is a city in **Greece**.*

The reason both these different approaches were tested lies mainly in the fact that inserting biases while attempting to extract geospatial knowledge is fairly easy. According to the *GSP*, the choice of the method can affect the outcome greatly. For instance, using *LTR* in the relation *"X is a city in Y"* creates the following problem; for the answer A_i of the *top-k* answers, the model would attempt to produce *top-k* tokens for the other mask. However, even if the model was an oracle and could safely predict A_j as the correct answer (A_i *is a city in the country of* A_j), it would continue to produce *top(k-1)* more answers which would be wrong. As a result, the percentage of correct answers is severely limited by a human induced bias. For this reason, we introduce one more parameter; the *cutoff*.

Cutoff. When *cutoff* is enabled, the model constructs *top-k* answers for the first mask to be filled (according to the opted method), and then returns one token for each of the *top-k* answers. Alternatively, when *cutoff* is disabled, the total answers produced are $L \cdot K^M$, where L is the number of layers, *K* is the *top-k* value, and *M* is the number of masks.

Layer Drop. In some experiments we explore if some specific layers affect the final results to a great extent. That is why we drop some of them from the model. Simply freezing a layer would still allow the tokens to flow through it and be susceptible to its normalization mechanism. We aim however to completely remove a layer and disallow it from influencing the data. In this regard, when removing the i^{th} layer, we copy the internal encoder structure except for the i^{th} layer, and assign the new layer list to the model. As a result, we are able to keep all the other layers unaffected by the removal and examine the influence such tweaks have on the results.

Attention Heads Drop. In a similar mindset, we also examine the extent to which specific attention heads (from specific layers) affect a model's answers. We utilize the internal mechanisms of a model that allow us to easily prune said heads; when pruned they serve as a no-op.

3.2. Compatibility Matrices

We are not only interested in the correct percentage of the answers, we also want to examine the consistency of the models' results. This is the reason why we construct compatibility matrices with which we are able to compare the percentage of compatibility between the layers. They are 2D heat maps that visually demonstrate how similar the answers yielded at each state of the model while answering a geospatial question. We constructed the following two types of compatibility matrices:

- **Self-Compatibility**: These matrices are symmetrical and compare a model to itself. By examining them, we are able to see how much the model changes its answers throughout the layers. Note that the main diagonal is not always 100% because we count the compatibility discarding the duplicate answers.
- **Cross-Model Compatibility**: Similar to the self-compatibility matrices, these compare different models (with the same architecture or not). We note some interesting points utilizing these graphs as to how different models behave whilst constructing their answers.

3.3. Validation

We utilize the GeoPy python API[4] to validate the answers to a geospatial question provided by an LLM. It is a module for geocoding that uses the OpenStreetMap (OSM) Nominatim service[5]. It takes a name as an input and returns the location matching that name along with its characteristics such as the feature type (e.g., city). The validation of an answer (X_i, Y_j) to "X is a city in Y." would proceed as follows. Running $geopy(name = X_i, feature\_type = city, country = Y_j)$ we would get the *city* that belongs in the country of Y_j and matches the name X_i. Note that the feature type of *city* may also allow towns, villages and communes. Unfortunately, sometimes GeoPy might confuse a clearly wrong answer for an existing location. For example $geopy(name = 1982)$ returns *Elewijt, Belgium*[6]. That is why we further restrict GeoPy's answer and model's answer to have an edit distance (i.e., changes needed to match the two strings) of 0 or 1[7].

4. Experiments

Settings and evaluation metrics. In our experimental evaluation, we use the two major variations of the BERT and RoBERTa models' sizes - base and large. As far as BERT is concerned, we also experiment upon the different casing versions. The uncased version (as opposed to the cased one) was trained with all textual data being lowered during the pre-processing procedure. For the *top-k* variable, we selected the values 10 and 100. More values were tested (e.g., 300, 1000), but ultimately we settled down on these for the following reasons. Higher *top-k* values dramatically reduced *P@C* (percentage of correct answers) on each layer, whilst simultaneously blunting the fluctuations at lower layers. As the answers on the results in the early stages are not yet sufficiently evaluated by the model, a lot of correct answers lie lower than they should be; hence a high *top-k* reveals those answers and tends to yield somewhat linear graphs. Such results would not allow us to focus on different layers' performance and their deeper analysis. Moreover, we needed a sufficient, yet small, *top-k* to be used as a reference for future *GSPs*' experimentation that have a finite number of correct answers. For example there exist approximately 190 countries, hence "[MASK] is a country." cannot have 300 correct responses. To evaluate our experiments, for each layer of a model, we compute the correct percentage of the answers produced by this layer (*P@C*) and we depict it in graphs. When a model returns a

[4] https://pypi.org/project/geopy/

[5] https://nominatim.org/

[6] *1982* is the zip code of *Elewijt, Belgium*

[7] $edit\_distance(1982, elewijt) = 7$

specific token as an answer, it also assigns a score to it, corresponding to the confidence it has for the token to be the actual answer. For each layer we compute the mean score of the mentioned confidence for the returned tokens, we normalize it and then feed it to a *MinMaxScaler* so that we can depict the different levels of confidence in the graphs. The closer the color of a scatter point is to black, the more confident the model was on that specific layer.

Task 1 - Results. We attempt to construct answers for the *GSP* of *"_ is a city in Europe."*. In Figure 1, we present the percentages of the correct answers each layer produces for the said phrase. As we can observe, the BERT models perform adequately in this specific task, while the RoBERTa models struggle with higher *top-k* values. This holds true probably because of the datasets that were used during pre-training; RoBERTa processed a great volume of data irrelevant to the Wikipedia textual corpus.
Intuitively, we can assume that the *uncased* versions of the models would perform worse, for the selected *GSP*, as we are searching for answers that are cities and almost always appear capitalized. However in some examples *uncased* versions achieve better scores. We can also observe that almost all the models appear to be strongly confident on lower levels with moderate *P@C*; this could indicate a high level of randomness to the answers. Another difference between the BERT and RoBERTa models appears on the final layers. RoBERTa seems to rearrange its answers and performs worse even though it had previously reached a higher score.

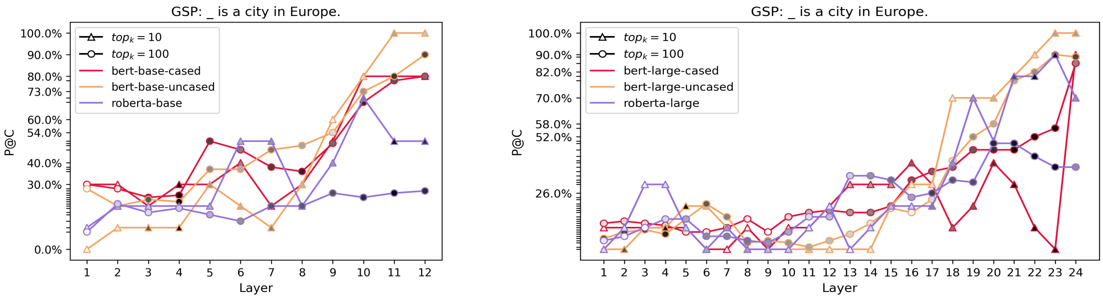

Figure 1: Correct percentage of answers per layer

Task 2 - Layer Pruning. In 1 we can see that sometimes local minima appear on the graphs, indicating layers that affect the general model performance. We therefore remove the said layers and observe the resulting graphs. Our results are included in Figure 2. What we can see is that such pruning allows the models to reach similar maximum scores with fewer layers. Even though a slight drop appears on the maximum *P@C*, in some cases we have pruned enough layers to decrease the total model size (trade-off); this could indicate that not all layers are necessary for such tasks and should we attempt to fine-tune them for better results, the process would be less computationally heavy and expensive.

Task 3 - Attention Head Pruning. As shown in Figure 3, we experimented with different combinations of what heads to keep and what to prune. It has been argued that specific heads

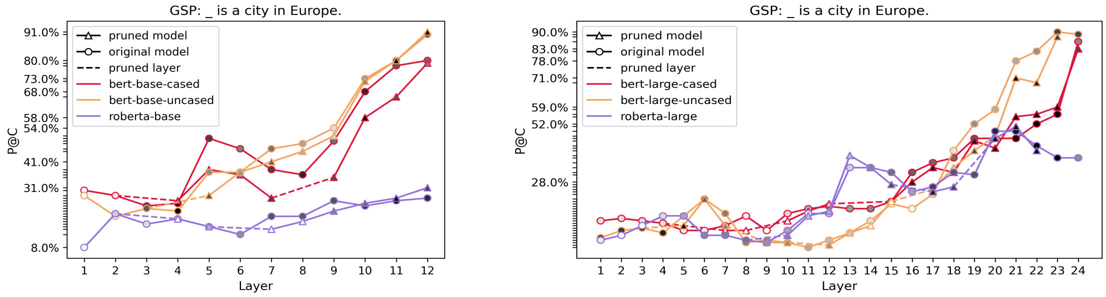

Figure 2: Pruned Models vs Original Models, *top-k* $= 100$

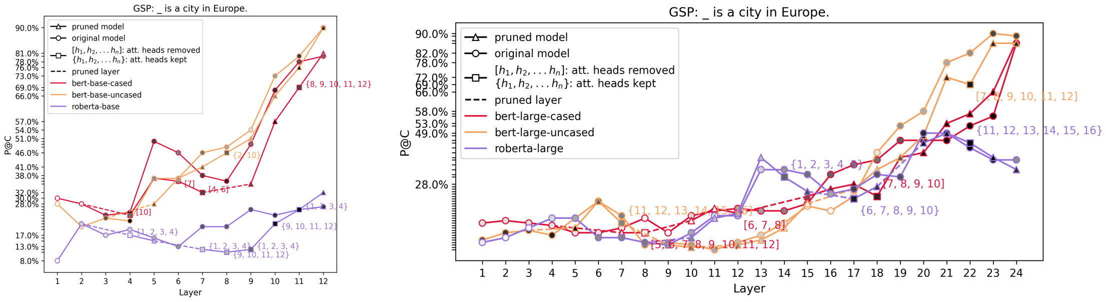

Figure 3: Attention Heads Pruning, *top-k* $= 100$

are able to perform better at certain tasks as classifiers [18, 17]. We could not however specify a general rule of thumb in our experiments. Through trial and error we were able to sometimes rectify the final scores or moderately affect them while having pruned a significant amount of heads. It seems that attention heads are crucial, but sometimes the models come equipped with more that the necessary amount [27]. What is also believed to be true, is that some heads on a layer often exhibit similar behaviours [17]. Hence, in some cases we were able to remove approximately half of a layer's heads without significant performance reduction.

Task 4 - Multiple Masks. As discussed in Section 3, when more than one mask appears, we need to specify the order we will fill them. It is understandable from the results shown

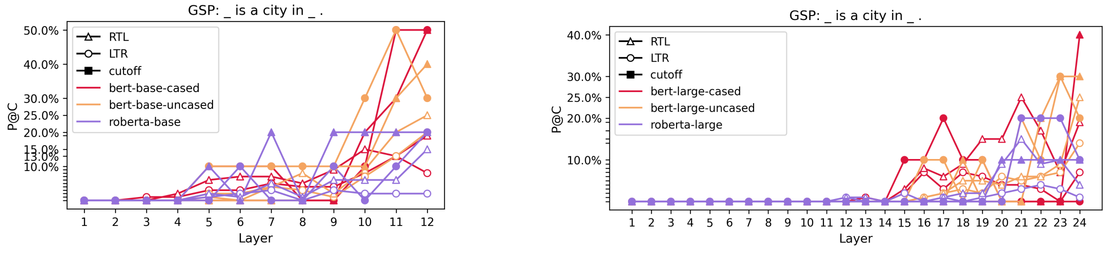

Figure 4: LTR vs RTL, *top-k* $= 10$

in Figure 4 that *cutoff* affects the total number of the produced answers. This is the reason cutoff-enabled curves perform better. Moreover, we observe that *RTL* achieves higher scores since multiple cities belong to a country while the reverse is non true. This fact indicates again the lack of a general rule, as the optimal method has to be chosen in regards to the *GSP*.

Task 5 - Compatibility Matrices. In some cases, the models seem to return similar answers; both between their layers and generally between different models. That is why we constructed compatibility matrices that count the number of common answers per layer, which helps us determine some insights on the similarities between them. For this experimental setting, we present in Figure 5 the compatibility matrices for the layers of the uncased BERT-base model ans its corresponding matrix for the negation of the same phrase, with *top-k* equal to 100. As we can observe, close layers in a model seem to yield similar answers. This compatibility span may appear to be slightly higher on upper layers; answers are more stable with fewer changes. Regarding the negation matrix, many scores are higher than what we expected. This is an indication that the models almost discard the negation. An interesting point lies also to the comparison of *base* and *large* models where the *base*'s 12 layers are more compatible to the last 24 layers of the *large* versions. The additional matrices that reflect different model cases can be found in Appendix A.

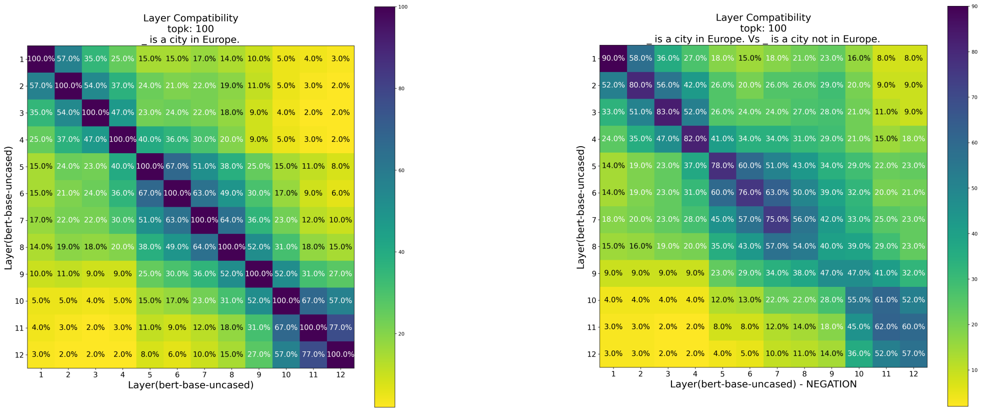

Figure 5: On the left lies the bert-base-uncased self compatibility matrix on the *GSP: "[MASK] is a city in Europe."* that portrays the similarities between each layers' answers. On the right, a compatibility matrix of the same model comparing the aforementioned phrase with negation. We observe that many answers are identical for both *GSPs* despite the negation.

5. Summary and Future Work

In this work, we carried out an experiment to investigate how LLMs behave whilst constructing simple geographic knowledge of the form *X is a city in Y* via the MLM paradigm. We experimented with the effect different layers and their attention heads have on the final output of the model and explored different techniques for mask filling. We found out that the results of geographic knowledge extraction from LLMs can vary highly; different methods and/or models greatly fluctuate the metric *P@C*.

In future work, we would like to study whether other kinds of geospatial knowledge can be extracted from LLMs using appropriate templates; in the introduction we have discussed various kinds of such knowledge. We also want to study how much geospatial answering and geospatial reasoning can be done by more recent language models such as ChatGPT, Bard, Claude and LLaMA. Through these models we would like to extend our research to more relations that are also important in the geospatial dimension such as cardinal relations. We aim to validate our future work using geospatial KGs and evaluate the aforementioned models on their geographic knowledge.

Acknowledgments

This work was supported by the first call for H.F.R.I. Research Projects to support faculty members and researchers and the procurement of high-cost research equipment grant (HFRI-FM17-2351).

References

[1] F. Petroni, T. Rocktäschel, S. Riedel, P. S. H. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, Language models as knowledge bases?, in: EMNLP-IJCNLP, Hong Kong, China, 2019.

[2] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, Minneapolis, MN, USA, 2019.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, CoRR (2019).

[4] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, CoRR (2021).

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

[6] N. Pörner, U. Waltinger, H. Schütze, E-BERT: efficient-yet-effective entity embeddings for BERT, in: EMNLP, Online Event, 2020.

[7] C. Wang, X. Liu, D. Song, Language models are open knowledge graphs, CoRR (2020).

[8] B. Heinzerling, K. Inui, Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries, in: EACL, Online, 2021.

[9] M. Koubarakis (Ed.), Geospatial Data Science: A Hands-on Approach for Building Geospatial Applications Using Linked Data Technologies, volume 51, 1 ed., Association for Computing Machinery, New York, NY, USA, 2023.

[10] J. Hoffart, F. M. Suchanek, K. Berberich, G. Weikum, YAGO2: A spatially and temporally enhanced knowledge base from wikipedia, Artificial Intelligence (2013).

[11] N. Karalis, G. M. Mandilaras, M. Koubarakis, Extending the YAGO2 knowledge graph with precise geospatial knowledge, in: ISWC, Auckland, New Zealand„ 2019.

[12] A. Dsouza, N. Tempelmeier, R. Yu, S. Gottschalk, E. Demidova, WorldKG: A world-scale geographic knowledge graph, in: CIKM, 2021.

[13] K. Janowicz et al., Know, know where, knowwheregraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence, AI Mag. 43 (2022).

[14] M. D. Siampou, N. Karalis, M. Koubarakis, Extending YAGO4 knowledge graph with geospatial knowledge, in: The 5th International Workshop on Geospatial Linked Data at ESWC, Hersonissos, Greece, 2022.

[15] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, J. Xu, Knowledgeable or educated guess? revisiting language models as knowledge bases, in: ACL/IJCNLP, Online, 2021.

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

[17] K. Clark, U. Khandelwal, O. Levy, C. D. Manning, What does BERT look at? an analysis of bert's attention, in: ACL Workshop, Florence, 2019.

[18] O. Kovaleva, A. Romanov, A. Rogers, A. Rumshisky, Revealing the dark secrets of BERT, in: EMNLP-IJCNLP, Hong Kong, China, 2019.

[19] J. Roberts, T. Lüddecke, S. Das, K. Han, S. Albanie, GPT4GEO: how a language model sees the world's geography (2023).

[20] A. G. Cohn, J. Hernández-Orallo, Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms (2023).

[21] F. Faisal, A. Anastasopoulos, Geographic and geopolitical biases of language models (2022).

[22] V. Hofmann, G. Glavas, N. Ljubesic, J. B. Pierrehumbert, H. Schütze, Geographic adaptation of pretrained language models (2022).

[23] G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu, C. Cundy, Z. Li, R. Zhu, N. Lao, On the opportunities and challenges of foundation models for geospatial artificial intelligence (2023).

[24] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler., Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: ICCV, Santiago, Chile, 2015.

[25] T. H. Trinh, Q. V. Le, A simple method for commonsense reasoning, CoRR (2018).

[26] A. G. Cohn, J. Renz, Qualitative spatial representation and reasoning, in: Handbook of Knowledge Representation, 2008.

[27] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, ACL, Florence, Italy (2019).

A. Appendix

A.1. Additional Compatibility Matrices

In Figure 6, we can see the self compatibility matrices for the large versions of the models BERT-uncased and RoBERTa, which are symmetrical, and the diagonals correspond to a layer being compared to itself. Moreover, on a comparison between BERT's base (discussed in Section 4) and the large versions shown in Figure 7, we can easily observe that the answers the base version yields shifted to the last 12 layers of the large versions. The early levels of BERT-large achieve very low similarity scores compared to the BERT-base layers.

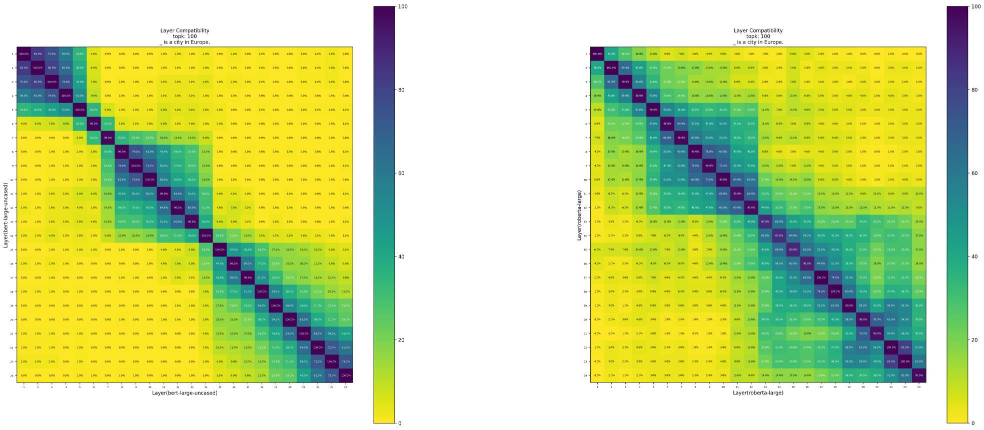

Figure 6: Self compatibility between the large models BERT-uncased and RoBERTa

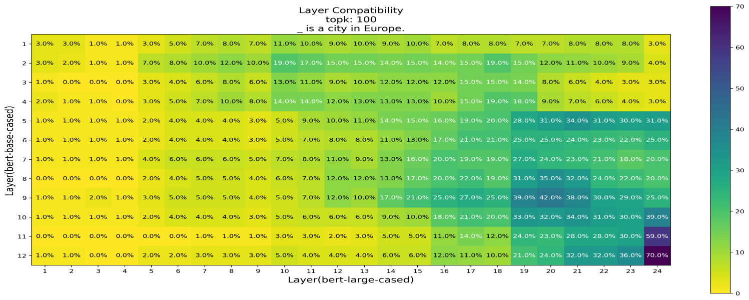

Figure 7: Layer compatibility between cased BERT-base and BERT-large models

Table 1
Highest scores for *top-k* $\in \{10, 100\}$

GSP	Model	Layer	Top-k	Method	Cutoff	Score (%)
_ is a city in Europe	bbu	12	10	-	-	100
_ is a city in Europe	bbu	12	100	-	-	90
_ is a city in Europe	blu	23	100	-	-	90
_ is a city in _	bbc	12	10	LTR	true	50
_ is a city in _	bbc	12	100	LTR	true	52

A.2. Highest Scores

In Table 1, we present the highest scores for *top-k* $\in \{10, 100\}$ and how it was achieved. The model column consists of abbreviations of the models (bert-base-cased: bbc, bert-base-uncased: bbu, bert-large-uncased:blu). Unsurprisingly, a *top-k* value equal to 10 is able to yield better scores than the higher value. When we include multiple masks, scores decline dramatically and only with the *cutoff* variable enabled are the models able to perform better. Moreover, the LTR method seems to outperform the RTL one. What is more, bert-large-uncased achieved a higher score on the 23^{rd} layer.