Extracting geographic knowledge from large language models

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 a thorough analysis on how the inner architecture of large language models behaves whilst extracting geographic knowledge. Our aim is to conclude on weather models actually incorporate geospatial information or simply follow statistical relevance of data; hence we hope to contribute to the public discuss of creating Knowledge Graphs from LLMs. In order to do that, we probe specific geospatial relations and explore different techniques that leverage the masked language modeling abilities of transformers. Our study should be construed as a stepping stone to the general probing of the ways LLMs encapsulate knowledge. 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, knowledge bases

1. Introduction

The field of Artificial Intelligence (AI) and Deep Learning (DL) is blossoming and continuously offers us a great multitude of intelligent applications. One prime example is the Large Language Models (LLMs) which seem to gain more and more abilities. These models were pre-trained using a huge amount of textual corpora and are generally believed to encapsulate explicit and implicit factual knowledge. The AI community however has yet to fully understand the internal mechanisms of such models. More importantly, even though extensive research on this matter has been conducted, there is no unequivocal conclusion of whether in fact transformers acquisite knowledge or simply follow the statistical relevance of data.

On top of that, a major percentage of intelligent applications leverage Knowledge Graphs for their operation. YAGO was one of the first studies that worked on the automation of Knowledge Graphs creation. Seeing that LLMs (and their variations) are constantly evolving and that they have been exposed on a vast quantity of data, naturally the question of whether we could extract their knowledge and automatically create KGs occurs.

Parts of KGs contain information concerning geospatial entities and their properties. Obviously, such knowledge is extremely useful in many environments. As Wikipedia entails a lot of corpora about said entities, it would seem natural for one to explore LLMs capabilities in this regard. Building specifically Geospatial Knowledge Graphs has already been put to the public

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)

https://cgi.di.uoa.gr/~dpantazi/ (D. Pantazi); https://cgi.di.uoa.gr/~koubarak/ (M. Koubarakis)

discussion. A geospatial entity usually incorporates both qualitative and quantitative attributes; Thessaloniki being north of Athens, their distance being 504km are two examples respectively [1]. In this study we are more interested in qualitative attributes mainly because LLMs are not able to *understand* numbers and their meaning. For humans, parsing a simple geographic textual corpus (even if fictional) allows them to extract meaningful qualitative knowledge (e.g., X belong to the continent of Y), without the need of visual stimuli such as maps. On this scope, it seems natural to entertain the idea that LLMs should ultimately be able to perform similarly. Qualitative characteristics of a geospatial entity are existent in many Wikipedia texts. Cardinal directions relations for example are extensively used in order to roughly describe an entity's position in the world. Additionally, many geospatial properties 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 simply be drawn by the fact that Africa is north of Europe. Understanding such relations and being able to easily answer qualitative geospatial queries appears to be an important aspect of natural language processing. What is more, models could be influenced by the statistical frequency and linkage of some data. For example even though Constantinople does not lie in Greece it frequently comes up as a Greek city; probably because of the historic entanglement of Greece and Asia Minor.

We believe that the main goal should not be the rectification of the extracted (from LLMs) KGs per se. Au contraire, the main focus should be on perfecting the techniques themselves used for such extraction with the final goal being that KGs and KBs we get from these models should represent their actual inner knowledge of the world (if any). Thereby, we would be able to probe them (ndlr. the KGs) and finally understand whether or not they really learn. In the name of aforementioned rectification, researchers may accidentally introduce biases that artfully achieve state-of-the-art results [2].

Motivated by the previously mentioned facts, we conduct an analysis about the ways LLMs attempt to answer geospatial related queries. We explore the fill-the-mask pipeline on pre-trained models without fine tuning; attempting to understand whether LLMs are actually able to answer such questions reliably. Through this study, we propose that deeper probing on transformers has to be conducted. In order for LMs-As-KB paradigm to be full-proof the golden ratio between most models has to be found. Are all their components needed for such tasks? Do their pre-training techniques dramatically affect the result? How similar and to what extend factually true the answers yielded are? And ultimately can we use LLMs so as to extract geographic knowledge?

2. Related Work

Recent works have studied the possibility that Large Language Models (LLMs) could be used as a means to Knowledge Graphs (KGs) and/or Knowledge Bases (KBs) construction and augmentation. Petroni et al. [3] introduced the LAMA probe that can explore the factual knowledge an LLM encapsulates, simply from its pre-training. Their contribution consists of a systematic analysis which reaches the conclusion that BERT-large is better at knowledge extraction compared to its adversaries, that relation extraction performance is not easily improved simply by increasing the data volume and finally they support that we need better understanding

regarding aspects of data that LLMs capture.

Wang et al. [4] 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 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 support that the resulting KGs (with a measured precision of more than 60%) indicate their approach being reliable.

Hao et al.[5] presented another framework that can construct a relational KG via an LLM without textual corpora parsing, simply by the utilisation of an initial prompt and some shots of examples. 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 said examples and the score they perform. Their approach leverages the masked language modeling (MLM) abilities of LLMs and retrieves knowledge via fill-the-mask tasks.

Razniewski et al.[6] on the other hand 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. Generally the LM-as-KB paradigm embodies three different approaches. Prompt-base retrieval in which one masks the desired answer and deems the returned token to be it; Paris is the capital of [MASK]. Case-based analogy, in which the prompt is enriched with an example prior to the mask one wants filled; Athens is the capital of Greece. Paris is the capital of [MASK]. Context-based inference, in which the prompt is enriched with relating information; Athens is in Greece. Athens is the capital of [MASK]. In this context, Cao et al.[2] analysed these three different methods and state for each one of them the biases that they consider to be the actual reason for previous approaches performing good as factual knowledge extraction techniques.

Concurrently, there is extensive research on the enhancement of knowledge bases along temporal and spatial dimensions. YAGO2 [7] is such an example that extends the classic Subject-Property-Object (SPO) triples adding Time and Location; which was further extended with precise geospatial knowledge [8].

Considerable attention is also being paid to the inner workings of LLMs; their layers and the corresponding attention heads. Clark et al.[9] hypothesise that some attention heads of BERT-base appear to behave in specific patterns that could indicate BERT learning syntactic dependencies of the English language. A similar study has been conducted by Kovaleva et al.[10], also focusing on BERT's self attention mechanisms suggesting that BERT can benefit from attention heads disabling in some tasks.

The aforesaid works, motivated us to perform a thorough analysis of the way LLMs' inner mechanisms behave whilst extracting geospatial knowledge.

3. Methodology

3.1. Models

We focus on Transformer-based language models that have been pre-trained through the masked language modeling (MLM) paradigm. BERT (Devlin et al.[11]) was trained on the BookCorpus

(800M words) (Zhu et al.[12]) 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 (ie. replaced with the special [MASK] token). More specifically, 10% of that 80%, the masked token was replaced with another random token and 10% was left unchanged. Another useful task BERT was pre-trained on, was Next Sequence Prediction (NSP). In this method, the model should predict whether a sentence A was following sentence B or not.

RoBERTa (Liu et al.[13]) follows similar training techniques (MLM, NSP) and almost identical architecture to BERT. They have however changed some major points. The MLM is performed via dynamic masking and tokenizing is replaced with byte-pair encoding (BPE). Finally, the data upon which it was trained (appart from those BERT used) include CC-News, OpenWebText and Stories.

We use the two major variations in models' sizes - base and large - for both BERT and RoBERTa. 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 pre-processing.

3.2. GeoSpatialPhrases

As a GeoSpatialPhrase (GSP) we define any clause that contains geospatial information for one entity or more (e.g., *Athens* is north of *Chania*.) In the previous example we deem *Athens* and *Chania* as instances that populate the more generic GSP format of "**X** is north of **Y**". We focus mainly on the ability LLMs have to produce viable and valid answers for such GSPs. In the following experiments we target IS-A relations for two main reasons. Firstly, we are able to validate them in a more full-proof way; "Paris is a city in Germany." is undeniably wrong, while the cardinal direction of Greenland compared to Iceland is arguably north, south, west and east simultaneously. Secondly, IS-A relations are very common in the Wikipedia corpus, on all the pages concerning geospatial entities. As a result, it is more than interesting to explore how much LLMs have incorporated that knowledge (if any).

3.3. Knowledge extraction hyperparameters

3.3.1. Layers

BERT like models take a sequence of tokens as an input and pass them 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 FeedForwardNeuralNetwork (FFNN) and returns a sequence of predicted tokens. We wanted to explore how the answers are constructed at each point of the model. In order to record that, we changed the classic forward function of these models so that we could have the answer from an arbitrary layer. When asking a model (with **K** layers) to fill the masked tokens from the layer **N** we actually allowed the model to use all layers $1 \leq i \leq N$ and then bridged the gap between the remaining layers and the MLM head (ie. layers $N < j \leq K$ were not used at all). If one requests they get an answer from the **Kth** 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.

3.3.2. Top-K answers

A softmax function is applied to the embeddings any BERT like model returns. They 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 of 10 and 100. Note that a large model (24 layers) with $topk = 100$ would return a total of 2400 answers.

3.3.3. Multi-mask filling methods

Some relations 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.

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 reason both these different approaches were tested lies mainly to 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 the country of **Y**" creates the following problem. For one of the topk answers ($A1_i$) the model would attempt to produce topk tokens for the other mask. However, even if the model was an oracle and could safely predict $A2_j$ as the correct answer ($A1_i$ is a city in the country of $A2_j$) it would continue to produce topk-1 more answers which would definetely be wrong. As a result, the percentage of correct answers is severely limited by a human induced bias. For that reason, we introduce one more parameter; cutoff.

Cutoff When cutoff is enabled, the model constructs topk answers for the first mask to fill (according to the opted method) and then returns 1 token for each of the topk answers. Alternatively, when cutoff is disabled, the total answers produced are

$$L \cdot K^M$$

Where L =number of layers, $K = topk$ and M =number-of-masks

3.3.4. Layer Drop

In some experiments we want to explore whether some specific layers affect the final results in a great extent. That is why, we may drop some of them from the model. Simply freezing the layer would still allow the tokens to flow through it and be susceptible to its Normalization mechanism. We wanted however to completely remove a layer and disallow it from influencing the data. In this regard when removing the i^{th} layer we simply copy the internal encoder

structure except the i^{th} and assign this 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 final results.

3.3.5. 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.

3.4. 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. That is why we construct compatibility matrices with which we are able to compare the percentage of compatibility between the layers. They are 2D heatmaps that visually demonstrate how similar the answers yielded at each point, are.

Self-Compatibility These matrices are symmetrical and compare a model to itself. We are able, examining them, 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 (either with the same architecture or not). We are able to note some interesting points utilizing these graphs as to how different models behave whilst constructing their answers.

3.5. Validation

We utilize the GeoPy python API for geocoding. Mainly, an answer is fed to the geocoder and the returned value (which is a location) is examined. According to the GSP format we want validated, we define different ways of answers examination. In "**X** is a city in **Y**" relation for example we restrict that the returned location be of a city feature type and in the country of **Y**. Note that GeoPy does not think of 'city' in a strict manner; towns, villages, even communes are allowed to be returned as cities. Some times however we may need to validate an answer from a model that was wrongly produced, eg. **1982** is a city in Europe. GeoPy - probably considering **1982** to be a location code - returns an existing city whose name however is not **1982**. For that reason, we further restrict that the returned location of GeoPy and the initial answer of the model, have a Levenshtein Distance of maximum 1.

3.6. Graph Construction

In any experiment, we compute for each layer the correct percentage of the answers (P@C) and depict it in graphs. When a model returns a 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. Per layer we compute the mean score of the tokens. We normalize them and then feed them 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. The actual mean scores may not be as different as the colors; that is why we normalized and scaled them so that the changes would be clearly visible.

4. Experiments

Task #1 - Results We attempt to construct answers for the GSP of "_ is a city in Europe.". In fig.1 we present the percentages of correct answers each layer produces for said phrase. A few points are more than apparent. BERT models perform adequately in this specific task while RoBERTa models struggle with bigger topks. 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.
What is more, intuitively, one would assume that *uncased* versions of the models would perform worse (for this GSP) as we are actually looking for answers that are cities and almost always appear capitalized. We suggest better scores appear on the *uncased* versions because of their vocabulary. Many of the extra results (that are indeed factually true) belong to the *uncased* vocabulary and not to the *cased* one. 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 BERT and RoBERTa models, appear on the final layers. RoBERTa seems to rearrange its answers and perform worse even though it had previously reached a higher score. Finally, we were not surprised to see the models performing better for lower topks.

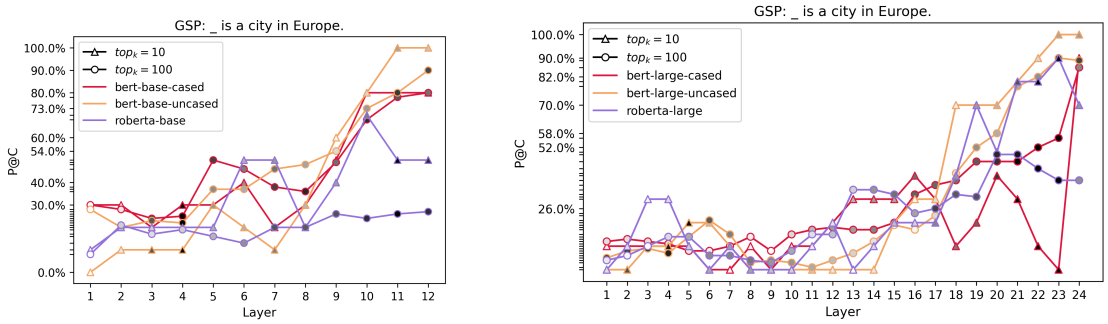


Figure 1: Correct percentage of answers per layer

Task #2 - Layer Pruning It it more than apparent that sometimes local minima appear on the graphs, indicating layers that affect the general model performance. We were intrigued to remove said layers and observe the resulting graphs. What we discern is that such pruning allows the models to reach similar maximum scores with fewer layers. Even if a slight drop appears on the maximum P@C, at 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

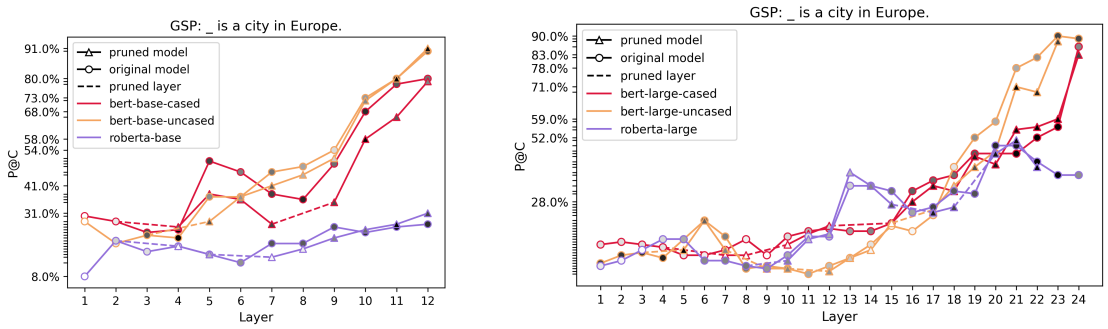


Figure 2: Pruned Models Vs Original Models, $top_k = 100$

should we attempt to fine-tune them for better results the process would be less computationally heavy and expensive.

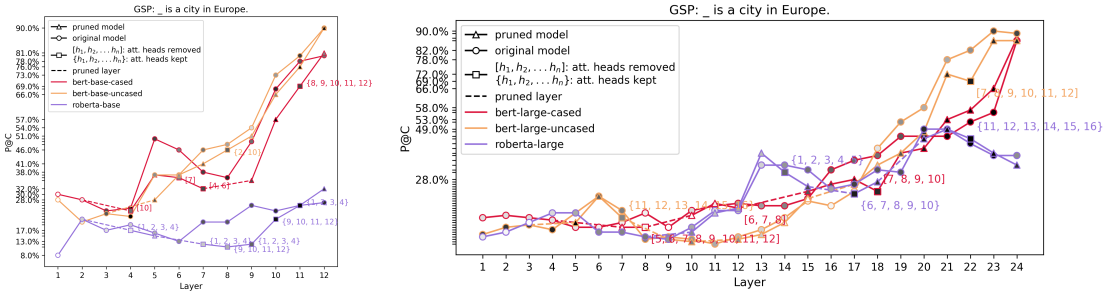


Figure 3: Att. Heads Pruning, $top_k = 100$

Task #3 - Att. Head Pruning As it is shown on fig.3 we experimented with different combinations of what heads to keep and what to prune. It is generally speculated that specific heads are able to perform better at certain tasks as classifiers [10, 9]. We could not however specify a general rule of thumb. Through trial and error we were able to some times rectify the final scores or moderately affect it while having pruned a significant amount of heads. It seems that attention heads are crucial but sometimes the sheer number models comes equipped with, is not necessary [14]. What is also believed to be true, is that heads on a layer often exhibit similar behaviours [9]. 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 Methodology, when more than one masks appear, an order of filling them has to be opted. It is easily understandable that *cutoff* affects the total number of produced answers. That is mainly the reason one can see cutoff-enabled curves performing better. Moreover, we observe that *RTL* method (here) achieves higher scores; multiple cities belong to a country, the reverse does not hold true. This fact indicates again the lack of a general rule, the optimal method has to be chosen in regards to the GSP.

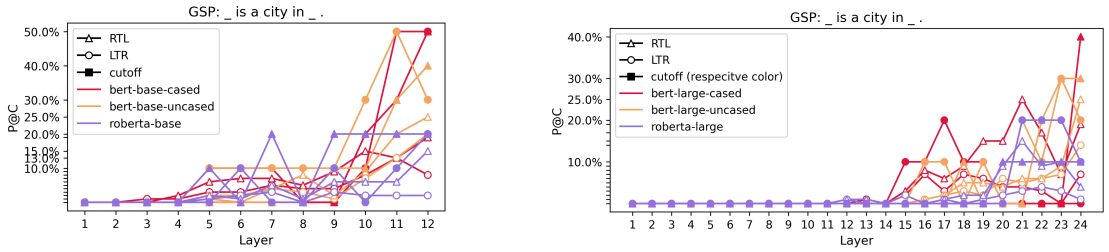


Figure 4: LTR Vs RTL, $top_k = 10$

Task #5 - Compatibility Matrices Studying the matrices, a few points are more than clear. Close layers (in a model) seem to yield similar answers. This compatibility span may appear to be slightly bigger on upper layers; answers are more stable with fewer changes. 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 12 layers of the *large* versions. Matrices can be found on the Appendix

5. Summary and Future Work

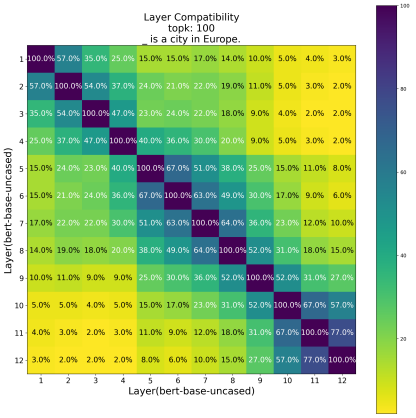
Conclusions We presented a thorough analysis of how large language models behave whilst constructing factual geographic knowledge via the MLM paradigm. We experimented with the effect different layers and their attention heads have on the final results. We also explored different techniques on mask filling. Geographic knowledge extraction from LLMs is highly entropical; different methods and/or models greatly fluctuate the P@C. The answers not being stable enough, allows one to seriously doubt the reliability of knowledge extraction from LLMs. It appears vital the models' knowledge incorporation be further researched to discern whether the results are such because of data statistical correlation or not; negation for example yields greatly homogeneous results to the equivalent affirmative clause (see Appendix).

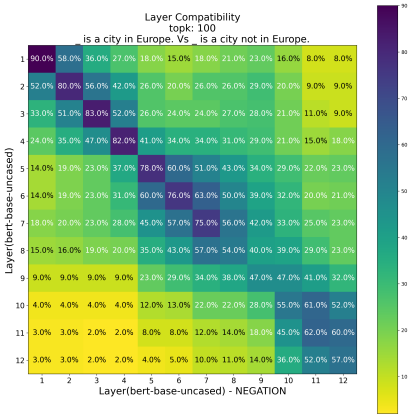
Future Work In due course we would also like to explore how the models behave with whole-word-masking; taking half-tokens into consideration and/or answers that consist of multiple words. What is more, paraphrasing prompts in specific ways (e.g., prepositions, negations) and measuring the different performances would be highly interesting. One could also examine a temporal dimension of such analyses; namely, how the scores change when querying for temporally wrong clauses. Most models have been trained for NSP tasks too. These abilities could be utilized to examine similar queries that can be constructed via NSP. Lastly, we can attempt to gradually construct geographic factual knowledge using the MLM abilities of the models by repetitively deepening a phrase with more masked queries to be filled and find the maximum depth a model can reach.

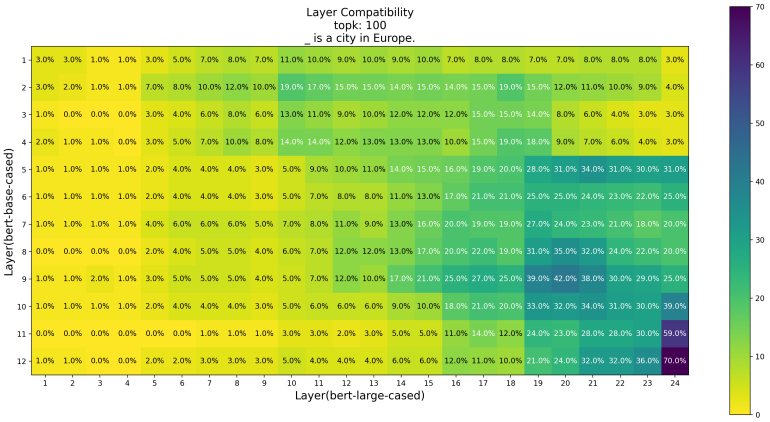
6. Acknowledgments

7. Appendices

7.1. Compatibility Matrices







7.2. Highest Scores

Highest scores that were achieved from unpruned models.

GSP	Model	Layer	TopK	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
The _ is a river in Europe	blu	24	100	-	-	31
The _ is a river in Europe	bbu	12	10	-	-	80
The _ is a river in _	TODO	TODO	TODO	TODO	TODO	TODO

References

[1] M. Koubarakis, Geospatial data modeling, in: Geospatial Data Science: A Hands-on Approach for Building Geospatial Applications Using Linked Data Technologies, 2023, pp. 9–30.

[2] 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, 2021.

[3] F. Petroni, T. Rocktäschel, S. Riedel, P. S. H. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, Language models as knowledge bases?, 2019.

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

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

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

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

[8] N. Karalis, G. M. Mandilaras, M. Koubarakis, Extending the YAGO2 knowledge graph with precise geospatial knowledge, 2019.

[9] K. Clark, U. Khandelwal, O. Levy, C. D. Manning, What does BERT look at? an analysis of bert's attention, 2019.

[10] O. Kovaleva, A. Romanov, A. Rogers, A. Rumshisky, Revealing the dark secrets of BERT, 2019.

[11] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, 2019.

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

[13] 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 (2019).

[14] 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, 2019.