Semesterarbeit I - Prompt Engineering / Einfaches RAG

Task 1 (Notebook: Llama)

- siehe Code: Python4RAG_pdf.ipynb

Task 2 (Notebook: Prompt Principles)

Folgende Prompt Patterns habe ich getestet:

Persona Pattern (Act as Oriental Story Teller)
USER_PROMPT_3 = """ \
What is Matryoshka Representation Learning? Act as a oriental story teller.
"""

Audience Pattern (Explain to Child)
USER_PROMPT_2 = """ \
What is Matryoshka Representation Learning? Explain to me like I'm 11 years old.
"""

Response Template Pattern (Use Sections)
USER_PROMPT_4 = """ \
What is Matryoshka Representation Learning?
Format your answer as follows:
#Question: [The question being asked by User]
Summary
[a one paragraph summary of the answer]
Details
[a detailed answer]
"""

Question Refinement Pattern
Is Matryoshka Representation useful at all?
If question is not clearly formulated, suggest a better version of the question and ask user if he would like to use it instead.
"""

Observations

- Prompt Patterns scheinen mit sehr machtvoll um das verhalten der LLM zu steuern. Sie hängen jedoch auch stark von der Qualität der Formulierung und den Fähigkeiten der LLM ab.
- Schwierig scheint mir auch, die Qualität des Prompts und seine Wirkung messbar zu beurteilen/ zu testen.
- Die Entwicklung, Wartung und Test komplexer Prompt basierender Anwendungen scheint mir herausfordernd. Eine kleine Veränderung im Prompt hat Einfluss auf das ganze System. Schon mit einfachen Prompts kann man viel Zeit mit Trial and Error verbringen.
- Eine weitere Herausforderung sehe ich darin, dass nicht nur der aktuelle Prompt, sonder auch die Message History bzw. Verwend und vom User/System Message ein einflussnehmender Faktor ist.
- Die Frage ‚Funktioniert es?' kann ich allenfalls mit einem ‚meistens' oder ‚wahrscheinlich' beantworten. Ob das in Real Life Anwendungen genügt?

Task 3 (Notebook: Prompt, Retrieval und Resultat)

- Der Satz „*When training such representations, it is often the case that computational and statistical constraints for each down- stream task are unknown.*" wird von keiner der getesteten lokalen LLM berücksichtigt. Ein test mit gpt-4o, gpt-4o-mini und gpt-3.5-turbo war erfolgreich. Der Satz wird erwähnt und zitiert. (Screenshots am Ende des Dokuments)
- Es scheint eine Herausforderung, ein Modell zu finden, welches klein genug ist, aber dennoch Leistungsfähig genug für den jeweiligen Anwendungsfall.

- Des Weiteren ist es für mich, als klassischer Softwareentwickler, sehr ungewohnt, mit solchen ‚Unberechenbarkeiten' umzugehen. Wie kann ich beispielsweise beweisen/testen, ob eine Software funktioniert, Resultate einer Komponente (LLM) nicht 100% vorhergesagt werden können?

Task 4 (Simple Chat Bot RAG)

- siehe Video / Code

Task 5 (Wozu kann MRL eingesetzt werden)

Mein Verständnis ist folgendermassen:

MRL kann genutzt werden, um ein Embeddings-Modell zu trainieren.

In einem nach dem MRL Prinzip trainierten Embeddings-Modell wird die Semantik eines Textabschnitts als eine ineinander verschachtelte Struktur von grob zu fein abgebildet. (Matrjoschka-Analogie). Jede tiefere Ebene repräsentiert dabei die Kern-Semantik des Textes in einer höheren/genaueren Präzision. Der generierte Vektor ist dann ebenfalls so gestaltet, dass jede Dimension die Kern-Semantik in einer feineren Weise repräsentiert, von grob zu fein.

Bei der Verwendung von Vektor-Datenbanken zum Speichern und Abfragen von Embeddings sind die Anzahl der Dimensionen ein wichtiger Kostenfaktor. Ziel ist daher, nur so viele Dimensionen zu verwenden, wie unbedingt erforderlich.

Hier bietet der Einsatz von MRL-Embedding Modellen einen flexiblen Ansatz, da die Dimensionen der Vektoren reduziert werden können (abgeschnitten), ohne dass dabei die Kern-Semantik wesentlich an Qualität einbüsst.

Das im OpenAI-Artikel vorgestellte Modell „text-embedding-3-large" ist ein Beispiel für ein Modell, bei dem Embedding Dimensionen gekürzt werden können, ohne wesentlichen Qualitätsverlust (z. B. von ursprünglich 1536 auf 256 Dimensionen). Zwar wird es nicht explizit erwähnt, aber es scheint mir naheliegend, dass „text-embedding-3-large" ein Modell ist, das mit dem MRL-Ansatz trainiert wurde.

Abgaben

Video (ChatBot Demo, Code)
Code/GIT

- Python4RAG_pdf.ipynb (llama)
- Bot4RAG_pdf.py (ein wenig erweitert)

Abstract

Learned representations are a central component in modern ML systems, serving a multitude of downstream tasks. When training such representations, it is often the case that computational and statistical constraints for each downstream task are unknown. In this context, rigid fixed-capacity representations can be either over or under-accommodating to the task at hand. This leads us to ask: *can we design a flexible representation that can adapt to multiple down-*

Are there computational and statistical constraints?

[MODEL: llama3.1:8b, TEMP: 0.2, CONTEXT: 56]

Yes, there are computational and statistical constraints mentioned in the context.

The context mentions that running things multiple times with models like ResNet and ViT is extremely expensive (point 3 under "Computational Complexity"). This suggests a constraint on the number of computations or runs that can be performed due to resource limitations.

Additionally, the context also mentions that the authors created a new subset of ImageNet-21K for downstream evaluation of retrieval performance at scale (point C under "4. If you are using existing assets..."), which implies that there may be statistical constraints related to data availability or quality.

Therefore, the answer is: Yes, there are computational and statistical constraints mentioned in the context.Yes, there are computational and statistical constraints mentioned in the context.

The context mentions that:

- The authors used a large-scale dataset (ImageNet-21K) which is expensive to run multiple times with models like ResNet and ViT. (Section 4)
- They created a new subset of ImageNet-21K for downstream evaluation of retrieval performance at scale, indicating the computational cost of working with such large datasets. (Appendix J)

Additionally, the context mentions that:

- The authors used a combination of GPUs and internal cluster resources to run their experiments, but did not provide detailed information on the specific resources used. (Section 4)
- They mentioned that they used public datasets and code under MIT, BSD or CC licenses, but did not mention any new assets created for this work. (Section 4)

These points suggest that there are indeed computational and statistical constraints to working with large-scale vision tasks like ImageNet-21K.

Are there computational and statistical constraints?

[MODEL: mistral:latest, TEMP: 0.2, CONTEXT: 56]

Yes, there are computational constraints mentioned in the context. The authors state that using their proposed method (MRL) with models like ResNet and ViT on large-scale datasets like ImageNet-21K can be expensive due to the need for multiple runs, which may require significant compute resources (see Appendix C and Appendix I). However, no specific details about the type of GPUs or other resources used are provided. Additionally, there is a statistical constraint in terms of the information bottleneck size, where a smaller bottleneck leads to a decrease in accuracy on fine-grained classes but not as significantly when evaluated at a superclass level (Figure 3 and Table 24 in Appendix J). Yes, there are computational constraints mentioned in the context. The authors state that using their proposed method (MRL) with models like ResNet and ViT on large-scale datasets like ImageNet-21K can be expensive due to the need for multiple runs and the use of resources such as GPUs and internal clusters or cloud providers. However, no specific details about the total amount of compute or the type of resources used are provided in the context.

Additionally, there is a statistical constraint mentioned in the context. The authors show that as the information bottleneck becomes smaller, the overall accuracy on fine-grained classes decreases rapidly (Figure 3). However, the drop-off is not as significant when evaluated at a superclass level (Table 24 in Appendix J). This suggests that there may be a trade-off between the amount of information retained and the accuracy achieved on different levels of classification.

Are there computational and statistical constraints?

[MODEL: gpt-4o-mini, TEMP: 0.2, CONTEXT: 56]

Yes, there are computational and statistical constraints. The paper discusses this context by stating, "When training such representations, it is often the case that computational and statistical constraints for each downstream task are unknown." This highlights the challenges posed by these constraints in designing flexible representations that can adapt to various tasks.

Are there computational and statistical constraints?

[MODEL: gpt-4o, TEMP: 0.2, CONTEXT: 56]

Yes, there are computational and statistical constraints mentioned in the context. Specifically, the context states:

"When training such representations, it is often the case that computational and statistical constraints for each downstream task are unknown. In this context, rigid fixed-capacity representations can be either over or under-accommodating to the task at hand."

Additionally, it mentions:

"The rigidity in these representations forces the use of high-dimensional embedding vectors across multiple tasks despite the varying resource and accuracy constraints that require flexibility."

And:

"Compute costs for the latter part of the pipeline scale with the embedding dimensionality as well as the data size (N) and label space (L). At web-scale this utilization cost overshadows the feature computation cost."

These quotations highlight the presence of computational and statistical constraints in the context of representation learning.