Szenario Ideen

Inhalt

Harry Potter Experte

Harry Potter hat einen Hype erfahren, du kannst aber nicht mitreden, weil du die Bücher nie gelesenen hast. Zeit hast du dafür auch nicht.

Damit du mitreden kannst, willst du dir in diesem Workshop einen Experten für das Harry Potter Universum zu schaffen mithilfe von KI. Als Wissensgrundlage werden dir von uns vorbereitete Dateien dienen.

Wenn das nächste Mal Harry Potter thematisiert wird, bist du der Profi. Die KI soll dir ermöglichen, dass dir das Wichtigste zusammengefasst wird.

Aufgabe 1 – Erstellung eines Harry-Potter-Experten durch Prompt Engineering

Dein Ziel ist es, einen Chatbot zu entwickeln, der sich ausschließlich durch die Gestaltung des System Prompts als umfassender Experte für das Harry-Potter Universum auskennt. Es dürfen keine zusätzlichen Dokumente oder externen Informationen verwendet werden.

Schaue dir den initialen System Prompt an und baue ihn so um, dass der Chatbot in der Lage ist, detaillierte, präzise und informative Antworten auf alle Fragen zu den sieben Büchern der Hauptreihe von J.K. Rowling zu geben.

Der Chatbot soll sich wie ein enthusiastischer und allwissender Harry-Potter-Fan verhalten und in der Lage sein, auf spezifische Ereignisse, Charaktere, Orte, Zaubersprüche, magische Gegenstände und historische Hintergründe einzugehen. Achte darauf, dass der Prompt den Chatbot anweist, Spekulationen zu vermeiden und sich strikt an den Kanon der Bücher zu halten.

Nutze dazu die Techniken Role Prompting und Shot Prompting. Nachdem du den System Prompt angepasst hast, stelle dem Chatbot Fragen und prüfe das Ergebnis.

Anleitung:
1. Optimiere den System Prompt gem. Szenario
2. Erstelle eine Anweisung im System Prompt
3. Ergänze den System Prompt um eine Rolle (Role Prompting)
4. Ergänze Informationen, wie der Chatbot antworten soll (z.B. enthusiastisch)
5. Ergänze den System Prompt um Beispiele (Few Shot Prompting)
6. Stelle dem Chatbot Fragen

Ziel: Code und den Chatbot Client kennenlernen

Aufgabe 2 - Dokument laden

Der Chatbot kennt zwar den Inhalt der Hauptreihe, aber nun möchtest du dem Chatbot neues Wissen hinzugeben, welches er noch nicht kennt. Dazu sind in der Datei HP_erfundene_aussagen.md zwei frei erfundene Aussagen hinterlegt, die etwas über Harry Potter erzählen. Nachdem du den Code so angepasst hast, dass das Dokument geladen und auch übergeben wird, schaue dir den Inhalt der Datei an und frage den Chatbot etwas ganz Spezielles, was er nur wissen kann, wenn er den Inhalt von HP_Kindheit.md kennt. So kannst du überprüfen, ob der Chatbot das Dokument auch wirklich kennt.

P.S.: Denk daran, damit der Chatbot die neuen Informationen auch hat, musst du die Anwendung einmal neustarten

Anleitung:

1. Lade das Dokument „HP_erfundene_aussagen.md“ mit dem Document Loader von Langchain
2. Erweitere den System Prompt um Context und passe es so an, dass der Context aus dem bereitgestellten Dokument priorisiert wird.
3. Stelle sicher durch Anpassung des System Prompt: Wenn eine Frage nicht beantwortet werden kann, soll der Chatbot das offen zu geben und nichts erfinden.
4. Füge in der "predict function" das Dokument als Input hinzu
5. Frage etwas ganz Spezielles, was er nur wissen kann

(Optional) Challenge: Weitere Document Loader inkl. zusätzlichem Dokument ausprobieren, z.B. PyPDF

Aufgabe 3 – Embeddings

1. **Erstelle aus den Dokumenten einen Vector Store:**
   Wandele alle Dokumente in Vektor-Embeddings um und lege sie in einem Vector Store ab. Dadurch kannst du später gezielt nach ähnlichen Dokumenten suchen.

2. **Erstelle und optimiere eine Query für die Similarity Search:**
   Formuliere eine Suchanfrage (Query) so, dass nur Dokumente zurückgegeben werden, die tatsächlich etwas mit *Harry Potter* zu tun haben. Achte darauf, dass beispielsweise das Dokument „HdR_erfundene_Aussagen“ (Herr der Ringe) bei der Suche und Auswahl der Embeddings **nicht** berücksichtigt wird.

   - Teste das, indem du den Chatbot z.B. nach Informationen aus „HdR_erfundene_Aussagen“ fragst.
   - Falls der Chatbot Informationen dazu liefern kann, solltest du deine Query anpassen, sodass dieses Dokument zukünftig ausgeschlossen wird.

3. **Nutze die Similarity Search, um passende Dokumente als Kontext zu holen:**
   Suche mit deiner optimierten Query die kontext-relevanten Dokumente im Vector Store heraus – es sollten nur passende Dokumente mit Harry-Potter-Inhalten zurückgegeben werden.

4. **Binde die relevanten Dokumente als Kontext in den Chatbot ein:**
   Füttere den Chatbot mit den ausgewählten Dokumenten (bzw. deren Embeddings), sodass er die Informationen daraus verwenden kann.

5. **Teste den Chatbot zu verschiedenen Themen:**

   - Stelle gezielte Fragen zu den Harry-Potter-Dokumenten und prüfe, ob die Antworten korrekt sind.
   - Frage bezüglich des „HdR_erfundene_Aussagen“-Dokuments (Herr der Ringe), um sicherzustellen, dass der Chatbot keine Infos zu diesem Thema kennt.

6. Schaue dir das Dokument „HP_der_verschwundene_kuchen.md“ an, welches auch Inhalte enthält, die nichts mit Harry Potter zu tun haben. Teste dazu Fragen wie:
   - „Kennst du Simon?“
   - „Weißt du, was Glück ist?“
     Damit überprüfst du, ob der Chatbot nur relevante Harry-Potter-Informationen nutzt.

Aufgabe 4 – Text-Splitter

1. **Verwende die gleichen Dokumente wie in Aufgabe 3:**
   Nutze dieselben Dateien, um die Auswirkungen des Text-Splittings auf die Qualität der Chatbot-Antworten zu untersuchen.

2. **Implementiere einen eigenen Text-Splitter:**
   Erstelle einen Text-Splitter, der die Dokumente in überschaubare Abschnitte („Chunks“) unterteilt.

   - **Parameter:**
     - **chunk_size = 1000** (ca. 1000 Zeichen pro Chunk)
     - **chunk_overlap = 100** (Überlappung von 100 Zeichen zwischen den Chunks, um Kontextverluste zu vermeiden)

3. **Zerlege die Dokumente in Chunks:**
   Wende den Text-Splitter auf jedes Dokument an und erzeuge die einzelnen Chunks.

   - Gib die Anzahl der generierten Chunks aus, um einen Überblick zu bekommen, wie viele Teile jedes Dokument umfasst.

4. **Erzeuge einen Vector Store aus den Chunks:**
   Verwandle die einzelnen Text-Chunks in Embeddings und speichere sie im Vector Store ab. Dies ermöglicht eine präzisere Similarity Search und ggf. bessere Antworten.

5. **Teste die Chatbot-Antwortqualität:**
   Prüfe, ob sich die Qualität der Antworten verändert hat, nachdem die Dokumente in Chunks zerlegt wurden.

   - Nutze wieder Fragen rund um das Dokument „HP_der_verschwundene_kuchen.md“, speziell auch zu den Sätzen, die **nichts mit Harry Potter zu tun haben**, z.B.:
     - „Kennst du Simon?“
     - „Weißt du, was Glück ist?“

   So kannst du feststellen, ob der Chatbot die Informationen aus den nicht-Harry-Potter-Sätzen weiterhin findet und verarbeitet.

6. **Variante der Parameterauswertung:**
   Experimentiere mit den Parametern **chunk_size** und **chunk_overlap**, um zu sehen, wie sich unterschiedliche Einstellungen auf die Antwortqualität und Anzahl der Chunks auswirken. Dokumentiere deine Beobachtungen.

7. **(Optional) Challenge – Persistente Speicherung:**
   Baue eine persistent gespeicherte ChromaDB ein, indem du einen **persist_directory**

einrichtest. Dadurch bleiben die Indexdaten auch nach einem Neustart erhalten, was Zeit spart und den Workflow verbessert.

Aufgabe 5 - Retriever

1. **Hintergrund:**
   Falls du den Text-Splitter bereits implementiert hast, ist dir möglicherweise aufgefallen, dass der Chatbot Fragen zu den zufälligen, nicht-Harry-Potter-Sätzen im Dokument „HP_der_verschwundene_kuchen.md“ nicht mehr beantworten konnte. Dies lag daran, dass die bisherige Query und die Similarity Search diese Sätze aufgrund der Filterung ausgeschlossen haben.

2. **Langchain Retriever erstellen:**
   Implementiere einen Langchain Retriever. Dieser hilft, relevante Dokumente bzw. Chunks zu finden und direkt im Kontext verfügbar zu machen, ohne dass du eine explizite Query mit Filterkriterien formulieren musst.

3. **Query und Similarity Search anpassen:**
   Entferne oder deaktiviere die bisher verwendete manuelle Query und Similarity Search, da der Retriever diese Funktion übernimmt und dadurch flexibler und umfassender sucht.

4. **Retriever in der predict-Funktion nutzen:**
   Rufe im Vorhersage- oder Antwort-Handler die Retriever-Funktion mit dem vom Nutzer eingegebenen Text (User Input) auf. Der Retriever liefert passende Dokumentenabschnitte zurück, die du dann als Kontext für die Antwortgenerierung nutzt.

5. **Dokumente als Kontext hinzufügen:**
   Füge die vom Retriever zurückgegebenen Dokument- oder Textabschnitte als Kontext in deinen Prompt an den Chatbot ein, damit dieser fundierte und relevante Antworten geben kann.

6. **Teste den Chatbot erneut:**
   Stelle spezifische Fragen zu den nicht-Harry-Potter-Inhalten im „HP_der_verschwundene_kuchen.md“ (bzw. zu den zufälligen Sätzen), zum Beispiel:

   - „Kennst du Simon?“
   - „Weißt du, was Glück ist?“
   - „Was ist mein Lieblingsgericht?“
     Prüfe, ob der Chatbot diese Fragen jetzt beantworten kann.