NLP_final

June 3, 2024

1 Installation and importing

[1]:
```
%%capture
!pip install tika
!pip install matplotlib
!pip install langchain
!pip install -qU llama-index-llms-openai
!pip install langchain_community
!pip install langchain_core
!pip install langchain_openai
!pip install langchain_experimental
!pip install faiss-cpu
!pip install datasets
!pip install ragas
```

[2]:
```
import os
import re
import getpass
import pandas as pd
from tika import parser
from datasets import load_dataset, Dataset
from ragas import evaluate
import matplotlib.pyplot as plt
from ragas.metrics import (answer_relevancy, faithfulness, context_recall,␣
 ↪context_precision)
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_community.vectorstores import FAISS
from langchain_openai.embeddings import OpenAIEmbeddings
from langchain_openai import ChatOpenAI
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.runnables import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
from langchain_experimental.text_splitter import SemanticChunker
```

[3]:
```
os.environ["OPENAI_API_KEY"]=␣
 ↪"sk-proj-EpcuRNoHC6vZvieDWSR2T3BlbkFJVgs9EQBu6tx0CO9VchbF"
```

1.1 Introduction

Our project aims to create a Question Answering (QA) model that delves into the foundation of human rights, exploring seminal legal documents like Justinian's Code and Magna Carta. To achieve this, we're developing a RAG (Retrieval-Augmented Generation) model that leverages the capabilities of Large Language Models (LLMs). Our RAG model will retrieve relevant passages from these historical texts and generate accurate answers to users' questions, providing insights into the evolution of human rights and their significance in modern times.

Our project will:

- Develop a RAG model that focuses on legal-human rights documents
- Train the model on a dataset comprising Justinian's Code, Magna Carta, and other relevant texts
- Evaluate the model's performance on a test set of questions
- Fine-tune the model for improved accuracy and relevance

Our RAG model builds upon the advancements in LLMs, which have demonstrated exceptional language understanding and generation capabilities. By integrating retrieval capabilities into our RAG model, we can ensure that the generated answers are grounded in the original texts, providing a more reliable and informative QA experience.

1.2 Universal Declaration of the Human Rights

[125]:
```
text_file = parser.from_file("datasets/Universal Declaration of the Human␣
↪Rights.pdf")
print(text_file['content'].replace('\n', '')[:100])
```

```
Universal Declaration of Human Rights Preamble Whereas recognition of the
inherent dignity and of th
```

1.2.1 Chunking methods

Chunking is a crucial step in preprocessing text data for our RAG model. We explore three approaches: naive chunking, semantic chunking and manual chunking.

Recursive Character Text Splitter (also called naive chunking) - it is a fundamental tool in the LangChain suite for breaking down large texts into manageable, semantically coherent chunks. This method is particularly recommended for initial text processing due to its ability to maintain the contextual integrity of the text. It operates by recursively splitting text based on a list of user-defined characters, ensuring that related pieces of text remain adjacent to each other, thus preserving their semantic relationship.

[42]:
```
naive_chunker = RecursiveCharacterTextSplitter(
    chunk_size=256,
    chunk_overlap=0,
    length_function=len,
    is_separator_regex=False
)

naive_chunks = naive_chunker.split_text(text_file['content'])
```

```
naive_chunks_df = pd.DataFrame(naive_chunks)
naive_chunks_df.head()
```

[42]:

	0
0	Universal Declaration of Human Rights \n\nPrea…
1	and peace in the world, \n\nWhereas disregard…
2	in which human beings shall enjoy freedom of s…
3	resort, to rebellion against tyranny and oppre…
4	Whereas the peoples of the United Nations have…

Semantic Chunking - it considers the relationships within the text. It divides the text into meaningful, semantically complete chunks. This approach ensures the information's integrity during retrieval, leading to a more accurate and contextually appropriate outcome. It is slower compared to the previous chunking strategy.

Interquartile chunking - in this method, the interquartile distance is used to split chunks.

[43]:
```
interquartile_chunker = SemanticChunker(
    OpenAIEmbeddings(), breakpoint_threshold_type="interquartile"
)

interquartile_chunks = interquartile_chunker.
 ↪create_documents([text_file['content']])
interquartile_chunks_df = pd.DataFrame(interquartile_chunks)
interquartile_chunks_df.head()
```

[43]:

	0	1	\
0	(page_content, \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\…	(metadata, {})	
1	(page_content, This right may not be invoked i…	(metadata, {})	
2	(page_content, Everyone has the right to own p…	(metadata, {})	
3	(page_content, No one shall be arbitrarily dep…	(metadata, {})	
4	(page_content, Parents have a prior right to c…	(metadata, {})	

	2
0	(type, Document)
1	(type, Document)
2	(type, Document)
3	(type, Document)
4	(type, Document)

Percentile chunking - in this method, all differences between sentences are calculated, and then any difference greater than the X percentile is split.

[44]:
```
percentile_chunker = SemanticChunker(
    OpenAIEmbeddings(), breakpoint_threshold_type="percentile"
)

percentile_chunks = percentile_chunker.create_documents([text_file['content']])
percentile_chunks_df = pd.DataFrame(percentile_chunks)
```

```
percentile_chunks_df.head()
```

[44]:
```
                                                   0               1  \
0  (page_content, \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\…  (metadata, {})
1  (page_content, This right may not be invoked i…  (metadata, {})
2  (page_content, Everyone has the right to own p…  (metadata, {})
3  (page_content, No one shall be arbitrarily dep…  (metadata, {})
4  (page_content, Parents have a prior right to c…  (metadata, {})

                   2
0  (type, Document)
1  (type, Document)
2  (type, Document)
3  (type, Document)
4  (type, Document)
```

Manual chunking - this chunking method was manually created by us in order to capture all of the articles in the document in different chunks. It uses basic regular expressions.

[65]:
```
manual_chunks = re.split(r'Article \d+\s*|Preamble', text_file['content'])[1:] ␣
 ↪# Split on "ART. X " and keep the chunks
manual_chunks_df = pd.DataFrame(manual_chunks)
manual_chunks_df.head()
```

[65]:
```
                                                   0
0   \n\nWhereas recognition of the inherent digni…
1  All human beings are born free and equal in di…
2  Everyone is entitled to all the rights and fre…
3  Everyone has the right to life, liberty and th…
4  No one shall be held in slavery or servitude; …
```

1.2.2 Model definition

In this part, we will define our RAG model. Firstly, we will define the template for our model. By template, we mean:

- If the model can't find a related context to our question in the text, it will output "I don't know"
- Otherwise, it will take the related context and augment it with the user defined question

[66]:
```
rag_template = """\
Use the following context to answer the user's query. If you cannot answer,␣
 ↪please respond with 'I don't know'.

User's Query:
{question}

Context:
{context}
```

```
"""

rag_prompt = ChatPromptTemplate.from_template(rag_template)
```

RAG models retrieve the context from vector databases. In this part, we will define 4 vector databases for each of our chunks. And to transfer our text file into numbers in the vector databases, we will use the "text-embedding-3-large" embedding model from OpenAI.

[67]:
```
naive_vectorstore = FAISS.from_texts(naive_chunks,␣
 ↪embedding=OpenAIEmbeddings(model="text-embedding-3-large"))
naive_retriever = naive_vectorstore.as_retriever(search_kwargs={"k" : 4})
```

[68]:
```
interquartile_vectorstore = FAISS.from_documents(interquartile_chunks,␣
 ↪embedding=OpenAIEmbeddings(model="text-embedding-3-large"))
interquartile_retriever = interquartile_vectorstore.
 ↪as_retriever(search_kwargs={"k" : 4})
```

[69]:
```
percentile_vectorstore = FAISS.from_documents(percentile_chunks,␣
 ↪embedding=OpenAIEmbeddings(model="text-embedding-3-large"))
percentile_retriever = percentile_vectorstore.as_retriever(search_kwargs={"k" :␣
 ↪4})
```

[70]:
```
manual_vectorstore = FAISS.from_texts(manual_chunks,␣
 ↪embedding=OpenAIEmbeddings(model="text-embedding-3-large"))
manual_retriever = manual_vectorstore.as_retriever(search_kwargs={"k" : 4})
```

Our base model for question answering system will be ChatOpenAI.

[71]:
```
base_model = ChatOpenAI()
```

These are the main bodies of ouor models. We will first define the context by setting it to relevant retriever from our vectorbase and in the same way, we will define the question by setting it to the user defined question. Then these two will go through our pre-defined prompt. As we know from earlier, the prompt will retrieve the context and augment it with the question before feeding them to our base model. When the base model receives these two, it will generate an answer. The answer will solely be related to our database.

[72]:
```
naive_rag_chain = (
    {"context" : naive_retriever, "question" : RunnablePassthrough()}
    | rag_prompt
    | base_model
    | StrOutputParser()
)
```

[73]:
```
interquartile_rag_chain = (
    {"context" : interquartile_retriever, "question" : RunnablePassthrough()}
    | rag_prompt
    | base_model
```

```
    | StrOutputParser()
)
```

[74]:
```
percentile_rag_chain = (
    {"context" : percentile_retriever, "question" : RunnablePassthrough()}
    | rag_prompt
    | base_model
    | StrOutputParser()
)
```

[75]:
```
manual_rag_chain = (
    {"context" : manual_retriever, "question" : RunnablePassthrough()}
    | rag_prompt
    | base_model
    | StrOutputParser()
)
```

1.2.3 Testing with random questions

For comparing the different chunking methods and in general, for comparing the models, we will create some random questions based on our text and get answers from these models. We will also create some questions that are not in the document to see if it outputs "I don't know".

[76]:
```
# Test 1
question = "What is the natural and fundamental group in society?"

naive_answer_1 = naive_rag_chain.invoke(question)
interquartile_answer_1 = interquartile_rag_chain.invoke(question)
percentile_answer_1 = percentile_rag_chain.invoke(question)
manual_answer_1 = manual_rag_chain.invoke(question)

print(f"The question is: {question}")
print(f"\nThe answer of naive chunking is: {naive_answer_1}")
print(f"\nThe answer of interquartile chunking is: {interquartile_answer_1}")
print(f"\nThe answer of percentile chunking is: {percentile_answer_1}")
print(f"\nThe answer of manual chunking is: {manual_answer_1}")
```

```
The question is: What is the natural and fundamental group in society?

The answer of naive chunking is: The natural and fundamental group in society is
the family.

The answer of interquartile chunking is: The natural and fundamental group in
society is the family, according to the context provided.

The answer of percentile chunking is: The natural and fundamental group in
society is the family, according to the context provided.
```

```
The answer of manual chunking is: The natural and fundamental group in society
is considered to be the family, according to the context provided.
```

[78]:
```python
# Test 2
question = "Who has the right to freedom of thought?"

naive_answer_2 = naive_rag_chain.invoke(question)
interquartile_answer_2 = interquartile_rag_chain.invoke(question)
percentile_answer_2 = percentile_rag_chain.invoke(question)
manual_answer_2 = manual_rag_chain.invoke(question)

print(f"The question is: {question}")
print(f"\nThe answer of naive chunking is: {naive_answer_2}")
print(f"\nThe answer of interquartile chunking is: {interquartile_answer_2}")
print(f"\nThe answer of percentile chunking is: {percentile_answer_2}")
print(f"\nThe answer of manual chunking is: {manual_answer_2}")
```

```
The question is: Who has the right to freedom of thought?

The answer of naive chunking is: Everyone has the right to freedom of thought
according to the context provided.

The answer of interquartile chunking is: Everyone has the right to freedom of
thought.

The answer of percentile chunking is: Everyone has the right to freedom of
thought, conscience, and religion, as stated in the provided document.

The answer of manual chunking is: Everyone has the right to freedom of thought,
as stated in the context provided.
```

1.2.4 Model Evaluation

In the final part, we will ask the OpenAI model to generate questions (to behave like a teacher preparing questions for the students) from the document in order to assess our model's ability to answer questions.

In order to achieve this, we'll first create synthetic documents from our original data by chunking them "naively". Next, we'll create our question prompt for the OpenAI model, which will take this prompt and generate questions based on the synthetic documents (it will behave like a teacher trying to prepare questions for an exam). Then in order to be able to compare the generated answers with the real answers, we'll create also a ground truth prompt for the OpenAI model. This ground truth will always give the right answers because we extract the context from which the question was generated from and feed it to the ground truth prompt.

[79]:
```python
synthetic_data_splitter = RecursiveCharacterTextSplitter(
    chunk_size=256,
    chunk_overlap=0,
    length_function=len,
```

```
    is_separator_regex=False
)

synthetic_data_chunks = synthetic_data_splitter.
 ↪create_documents([text_file['content']])
```

[80]:
```
questions = []
ground_truths_semantic = []
contexts = []
answers = []
```

We'll start with percentile chunking.

[81]:
```
question_prompt = """\
You are a teacher preparing a test. Please create a question that can be␣
 ↪answered by referencing the following context.

Context:
{context}
"""

ground_truth_prompt = """\
Use the following context and question to answer this question using *only* the␣
 ↪provided context.

Question:
{question}

Context:
{context}
"""

question_prompt = ChatPromptTemplate.from_template(question_prompt)
ground_truth_prompt = ChatPromptTemplate.from_template(ground_truth_prompt)

question_chain = question_prompt | ChatOpenAI(model="gpt-3.5-turbo") |␣
 ↪StrOutputParser()
ground_truth_chain = ground_truth_prompt |␣
 ↪ChatOpenAI(model="gpt-4-turbo-preview") | StrOutputParser()

for chunk in synthetic_data_chunks[10:20]:
  questions.append(question_chain.invoke({"context" : chunk.page_content}))
  contexts.append([chunk.page_content])
  ground_truths_semantic.append(ground_truth_chain.invoke({"question" :␣
 ↪questions[-1], "context" : contexts[-1]}))
  answers.append(percentile_rag_chain.invoke(questions[-1]))
```

[82]:
```
qagc_list = []

for question, answer, context, ground_truth in zip(questions, answers,␣
 ↪contexts, ground_truths_semantic):
  qagc_list.append({
      "question" : question,
      "answer" : answer,
      "contexts" : context,
      "ground_truth" : ground_truth
  })

eval_dataset = Dataset.from_list(qagc_list)
```

[83]:
```
result = evaluate(eval_dataset, metrics=[context_precision,
                                         faithfulness,
                                         answer_relevancy,
                                         context_recall]);
```

```
Evaluating:   0%|          | 0/40 [00:00<?, ?it/s]
```

[84]:
```
result_df = result.to_pandas()
result_df
```

[84]:

	question	answer	contexts
0	Question: According to the context provided, w…	The principle emphasized in Article 2 of the D…	
1	Question: According to the provided context, w…	Answer: Factors such as race, nationality, rel…	
2	Question: According to the context provided, w…	According to the context provided, the specifi…	
3	Question: According to the context provided, w…	I don't know.	
4	Question: According to the context provided, w…	According to the context provided, individuals…	
5	Question: According to Article 8 of the Declar…	Answer: Everyone has the right to equal protec…	
6	Question: How does the constitution protect in…	The constitution protects individuals from arb…	
7	Question: According to the context provided, w…	According to the context provided, Article 11 …	
8	Question:\nAccording to the context provided, …	Everyone charged with a penal offence has the …	
9	Question:\nAccording to the context provided, …	The principle being described in relation to c…	

```
0  [spirit of brotherhood.  \n\nArticle 2  \n\nEv…
1  [political or other opinion, national or socia…
2  [belongs, whether it be independent, trust, no…
3  [No one shall be held in slavery or servitude;…
4  [Everyone has the right to recognition everywh…
5  [discrimination in violation of this Declarati…
6  [for acts violating the fundamental rights gra…
7  [Everyone is entitled in full equality to a fa…
8  [1. Everyone charged with a penal offence has …
9  [2. No one shall be held guilty of any penal o…

                                        ground_truth  context_precision  \
0  The principle emphasized in Article 2 of the D…                   0.0
1  According to the provided context, factors tha…                   1.0
2  According to the context provided, under Artic…                   1.0
3  According to the context provided, under Artic…                   1.0
4  According to the context provided, individuals…                   1.0
5  According to Article 8 of the Declaration, eve…                   1.0
6  The constitution protects individuals from arb…                   1.0
7  According to the context provided, Article 11 …                   1.0
8  According to the context provided, everyone ch…                   1.0
9  The principle being described in relation to c…                   1.0

   faithfulness  answer_relevancy  context_recall
0           0.5          0.957460             0.0
1           1.0          0.954312             1.0
2           1.0          0.975967             1.0
3           1.0          0.000000             1.0
4           1.0          0.916578             1.0
5           0.5          0.898954             1.0
6           1.0          0.862841             1.0
7           1.0          0.906340             1.0
8           1.0          0.928532             1.0
9           1.0          0.970546             1.0
```

[86]:
```python
for metric, score in result.items():
    print(f"Metric: {metric}, score: {score}", end='\n')
```

```
Metric: context_precision, score: 0.89999999991
Metric: faithfulness, score: 0.9
Metric: answer_relevancy, score: 0.8371529146590362
Metric: context_recall, score: 0.9
```

[111]:
```python
for _, row in result_df.query('faithfulness < 1').iterrows():
    print(row.question, end='\n')
    print("Answer: ", row.answer, end='\n')
    print("Ground truth: ", row.ground_truth, end='\n')
    print("Faithfulness: ", row.faithfulness, end='\n')
```

```
    print('----------------------')
```

```
Question: According to the context provided, what principle is emphasized in
Article 2 of the Declaration being referenced?
Answer:  The principle emphasized in Article 2 of the Declaration is non-
discrimination and equality, stating that everyone is entitled to all the rights
and freedoms without any distinction based on various factors.
Ground truth:  The principle emphasized in Article 2 of the Declaration being
referenced is the principle of non-discrimination.
Faithfulness:  0.5
----------------------
Question: According to Article 8 of the Declaration, what right does everyone
have in regards to discrimination and incitement to discrimination?
Answer:  Answer: Everyone has the right to equal protection against any
discrimination and incitement to such discrimination according to Article 8 of
the Declaration.
Ground truth:  According to Article 8 of the Declaration, everyone has the right
to an effective remedy by the competent national tribunals for acts of
discrimination in violation of this Declaration and against any incitement to
such discrimination.
Faithfulness:  0.5
----------------------
```

Let's do the same things for manual chunking.

[87]:
```
answers_manual = []

for question in questions:
  answers_manual.append(manual_rag_chain.invoke(question))
```

[88]:
```
qagc_manual = []

for question, answer, context, ground_truth in zip(questions, answers_manual,␣
 ↪contexts, ground_truths_semantic):
  qagc_manual.append({
      "question" : question,
      "answer" : answer,
      "contexts" : context,
      "ground_truth" : ground_truth
  })

eval_dataset_manual = Dataset.from_list(qagc_manual)
```

[89]:
```
result_manual = evaluate(eval_dataset_manual, metrics=[context_precision,
                                        faithfulness,
                                        answer_relevancy,
                                        context_recall])
```

```
Evaluating:   0%|          | 0/40 [00:00<?, ?it/s]
```

[93]:
```
result_manual_df = result_manual.to_pandas()
result_manual_df
```

[93]:

	question	answer	contexts	ground_truth	context_precision
0	Question: According to the context provided, w…	The principle of non-discrimination is emphasi…	[spirit of brotherhood. \n\nArticle 2 \n\nEv…	The principle emphasized in Article 2 of the D…	0.0
1	Question: According to the provided context, w…	According to the provided context, factors suc…	[political or other opinion, national or socia…	According to the provided context, factors tha…	1.0
2	Question: According to the context provided, w…	The specific rights guaranteed to everyone und…	[belongs, whether it be independent, trust, no…	According to the context provided, under Artic…	1.0
3	Question: According to the context provided, w…	I don't know.	[No one shall be held in slavery or servitude;…	According to the context provided, under Artic…	1.0
4	Question: According to the context provided, w…	Answer: The fundamental right that individuals…	[Everyone has the right to recognition everywh…	According to the context provided, individuals…	1.0
5	Question: According to Article 8 of the Declar…	Answer: According to Article 8 of the Declarat…	[discrimination in violation of this Declarati…	According to Article 8 of the Declaration, eve…	1.0
6	Question: How does the constitution protect in…	The constitution protects individuals from arb…	[for acts violating the fundamental rights gra…	The constitution protects individuals from arb…	1.0
7	Question: According to the context provided, w…	According to the context provided, Article 11 …	[Everyone is entitled in full equality to a fa…		
8	Question:\nAccording to the context provided, …	According to the context provided, everyone ch…	[1. Everyone charged with a penal offence has …		
9	Question:\nAccording to the context provided, …	The principle being described in relation to c…	[2. No one shall be held guilty of any penal o…		

```
7  According to the context provided, Article 11 …                1.0
8  According to the context provided, everyone ch…                1.0
9  The principle being described in relation to c…                1.0

   faithfulness  answer_relevancy  context_recall
0      1.000000          0.871105             0.0
1      0.700000          0.971180             1.0
2      1.000000          0.975967             1.0
3      1.000000          0.000000             1.0
4      1.000000          0.967665             1.0
5      1.000000          0.901003             1.0
6      0.333333          0.964884             1.0
7      1.000000          0.943339             1.0
8      1.000000          0.943714             1.0
9      0.666667          0.967770             1.0
```

[94]:
```
for metric, score in result_manual.items():
    print(f"Metric: {metric}, score: {score}", end='\n')
```

```
Metric: context_precision, score: 0.89999999991
Metric: faithfulness, score: 0.8699999999999999
Metric: answer_relevancy, score: 0.8506626506289852
Metric: context_recall, score: 0.9
```

[95]:
```
result_manual_df.query('faithfulness < 1')
```

[95]:
```
                                          question  \
1  Question: According to the provided context, w…
6  Question: How does the constitution protect in…
9  Question:\nAccording to the context provided, …

                                            answer  \
1  According to the provided context, factors suc…
6  The constitution protects individuals from arb…
9  The principle being described in relation to c…

                                          contexts  \
1  [political or other opinion, national or socia…
6  [for acts violating the fundamental rights gra…
9  [2. No one shall be held guilty of any penal o…

                                        ground_truth  context_precision  \
1  According to the provided context, factors tha…                1.0
6  The constitution protects individuals from arb…                1.0
9  The principle being described in relation to c…                1.0

   faithfulness  answer_relevancy  context_recall
1      0.700000          0.971180             1.0
```

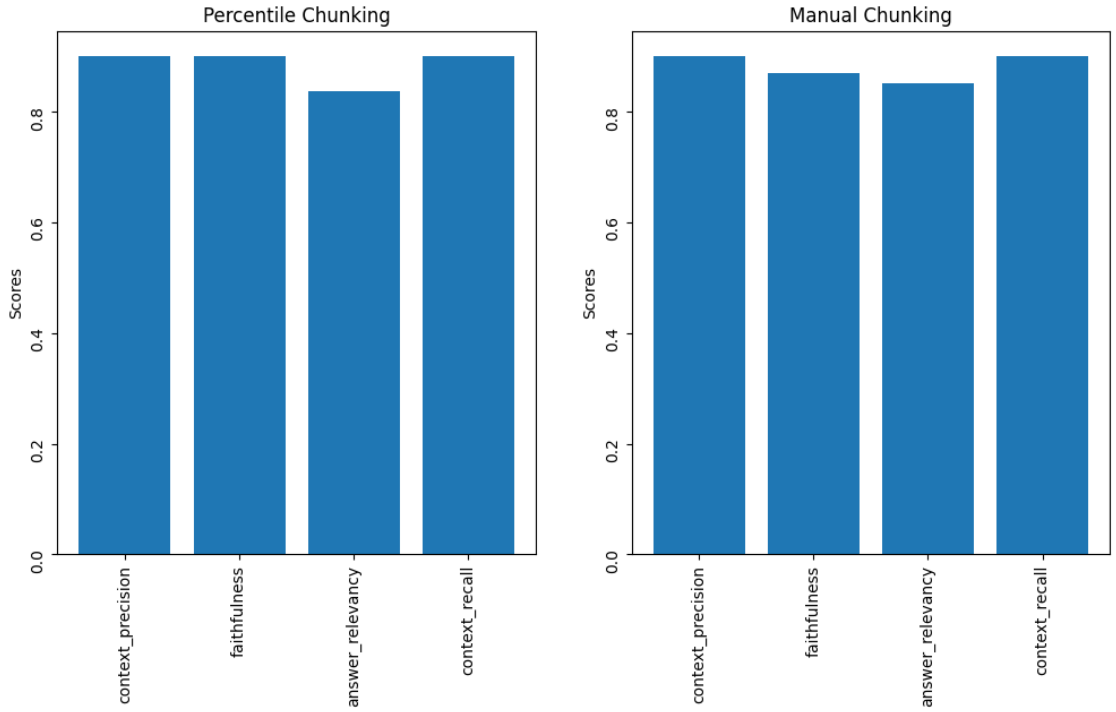
```
6       0.333333          0.964884             1.0
9       0.666667          0.967770             1.0
```

[96]:
```
for _, row in result_manual_df.query('faithfulness < 1').iterrows():
    print(row.question, end='\n')
    print("Answer: ", row.answer, end='\n')
    print("Ground truth: ", row.ground_truth, end='\n')
    print("Faithfulness: ", row.faithfulness, end='\n')
    print('----------------------')
```

```
Question: According to the provided context, what factors should not be used to
discriminate against individuals?
Answer:  According to the provided context, factors such as race, color, sex,
language, religion, political or other opinion, national or social origin,
property, birth, or other status should not be used to discriminate against
individuals.
Ground truth:  According to the provided context, factors that should not be
used to discriminate against individuals include political or other opinion,
national or social origin, property, birth, or other status. Additionally, no
distinction shall be made based on the political, jurisdictional, or
international status of the country or territory to which a person belongs.
Faithfulness:  0.7
----------------------
Question: How does the constitution protect individuals from arbitrary actions
such as arrest, detention, or exile?
Answer:  The constitution protects individuals from arbitrary actions such as
arrest, detention, or exile by stating that no one shall be subjected to
arbitrary arrest, detention, or exile. It also guarantees the right to life,
liberty, and security of person, as well as protection against arbitrary
interference with privacy, family, home, or correspondence. Additionally,
individuals have the right to own property and cannot be arbitrarily deprived of
it.
Ground truth:  The constitution protects individuals from arbitrary actions such
as arrest, detention, or exile through Article 9, which explicitly states that
"No one shall be subjected to arbitrary arrest, detention or exile."
Faithfulness:  0.3333333333333333
----------------------
Question:
According to the context provided, what principle is being described in relation
to criminal offenses committed in the past?
Answer:  The principle being described in relation to criminal offenses
committed in the past is the principle of non-retroactivity of criminal law.
This means that individuals cannot be held guilty of a penal offense for an act
that was not considered a crime at the time it was committed, and they cannot be
subjected to a heavier penalty than what was applicable at the time of the
offense.
Ground truth:  The principle being described in relation to criminal offenses
committed in the past is the principle of legality, specifically that no one
```

```
shall be found guilty of a crime for actions that were not considered criminal
under national or international law at the time they were committed. This
includes the prohibition against retroactive application of criminal law to
impose a heavier penalty than what was applicable at the time the act was
committed.
Faithfulness:  0.6666666666666666
----------------------
```

[112]:
```
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
axes[0].bar(list(result.keys()), list(result.values()))
axes[1].bar(list(result_manual.keys()), list(result_manual.values()))
axes[0].tick_params(labelrotation=90)
axes[1].tick_params(labelrotation=90)
axes[0].set_title("Percentile Chunking")
axes[1].set_title("Manual Chunking")
axes[0].set_ylabel("Scores")
axes[1].set_ylabel("Scores")
plt.show()
```



Let's now compare all of the chunking methods in order to choose the best among them for our next dataset.

[113]:
```
answers_quartile = []

for question in questions:
```

```
  answers_quartile.append(interquartile_rag_chain.invoke(question))
```

[114]:
```
qagc_quartile = []

for question, answer, context, ground_truth in zip(questions, answers_quartile,␣
 ↪contexts, ground_truths_semantic):
  qagc_quartile.append({
      "question" : question,
      "answer" : answer,
      "contexts" : context,
      "ground_truth" : ground_truth
  })

eval_dataset_quartile = Dataset.from_list(qagc_quartile)
```

[115]:
```
result_quartile = evaluate(eval_dataset_quartile, metrics=[context_precision,
                                          faithfulness,
                                          answer_relevancy,
                                          context_recall])
```

```
Evaluating:   0%|          | 0/40 [00:00<?, ?it/s]
```

[116]:
```
answers_naive = []

for question in questions:
  answers_naive.append(naive_rag_chain.invoke(question))
```

[117]:
```
qagc_naive = []

for question, answer, context, ground_truth in zip(questions, answers_naive,␣
 ↪contexts, ground_truths_semantic):
  qagc_naive.append({
      "question" : question,
      "answer" : answer,
      "contexts" : context,
      "ground_truth" : ground_truth
  })

eval_dataset_naive = Dataset.from_list(qagc_naive)
```

[118]:
```
result_naive = evaluate(eval_dataset_naive, metrics=[context_precision,
                                          faithfulness,
                                          answer_relevancy,
                                          context_recall])
```

```
Evaluating:   0%|          | 0/40 [00:00<?, ?it/s]
```

[119]:
```
for metric, score in result_quartile.items():
    print(f"Metric: {metric}, score: {score}", end='\n')
```

```
Metric: context_precision, score: 0.89999999991
Metric: faithfulness, score: 0.8557142857142856
Metric: answer_relevancy, score: 0.9263559279295326
Metric: context_recall, score: 0.9
```

[120]:
```
for metric, score in result_naive.items():
    print(f"Metric: {metric}, score: {score}", end='\n')
```

```
Metric: context_precision, score: 0.89999999991
Metric: faithfulness, score: 0.9400000000000001
Metric: answer_relevancy, score: 0.8506374851518224
Metric: context_recall, score: 0.9
```

Seems like naive chunking performs the best. Let's choose it for further analysing other datasets.

1.3 Magna Carta

The text was preprocessed; the header and page numbers were removed.

[126]:
```
magna_carta = parser.from_file("datasets/Magna Carta.pdf")
print(magna_carta['content'].replace('\n', '')[:100])
```

```
Preamble: John, by the grace of God, king of England, lord of Ireland, duke of
Normandy and Aquitain
```

1.3.1 Chunking

In the first part, we saw that naive chunking results in a better performance compared to others (though the differences are too small).

[127]:
```
naive_chunker = RecursiveCharacterTextSplitter(
    chunk_size=256,
    chunk_overlap=0,
    length_function=len,
    is_separator_regex=False
)

naive_chunks = naive_chunker.split_text(magna_carta['content'])
naive_chunks_df = pd.DataFrame(naive_chunks)
naive_chunks_df.head()
```

[127]:

	0
0	Preamble:
1	John, by the grace of God, king of England, lo…
2	stewards, servants, and to all his bailiffs an…
3	God and the advancement of his holy Church and…
4	England and cardinal of the holy Roman Church,…

1.3.2 Model Definition

We will apply the same model defined in the first part.

[130]:
```
rag_template = """\
Use the following context to answer the user's query. If you cannot answer,␣
 ↪please respond with 'I don't know'.

User's Query:
{question}

Context:
{context}
"""

rag_prompt = ChatPromptTemplate.from_template(rag_template)
```

[131]:
```
naive_vectorstore = FAISS.from_texts(naive_chunks,␣
 ↪embedding=OpenAIEmbeddings(model="text-embedding-3-large"))
naive_retriever = naive_vectorstore.as_retriever(search_kwargs={"k" : 4})
```

[132]:
```
base_model = ChatOpenAI()
```

[133]:
```
naive_rag_chain = (
    {"context" : naive_retriever, "question" : RunnablePassthrough()}
    | rag_prompt
    | base_model
    | StrOutputParser()
)
```

1.3.3 Testing with random questions

[135]:
```
# Test 1
question = "Where should common pleas be held?"

naive_answer_1 = naive_rag_chain.invoke(question)

print(f"The question is: {question}")
print(f"\nThe answer of naive chunking is: {naive_answer_1}")
```

```
The question is: Where should common pleas be held?

The answer of naive chunking is: Common pleas should be held in some fixed
place, not following the court.
```

[137]:
```
# Test 2
question = "When are the foreign born knights banished?"
```

```
naive_answer_2 = naive_rag_chain.invoke(question)

print(f"The question is: {question}")
print(f"\nThe answer of naive chunking is: {naive_answer_2}")
```

```
The question is: When are the foreign born knights banished?

The answer of naive chunking is: When peace is restored, the foreign born
knights are banished from the kingdom.
```

Our model answers correctly all the questions. Let's evaluate it on sets of questions generated by OpenAI.

1.3.4 Model evaluation

[155]:
```
synthetic_data_splitter = RecursiveCharacterTextSplitter(
    chunk_size=256,
    chunk_overlap=0,
    length_function=len,
    is_separator_regex=False
)

synthetic_data_chunks = synthetic_data_splitter.
 ↪create_documents([magna_carta['content']])
```

[156]:
```
questions = []
ground_truths_semantic = []
contexts = []
answers = []
```

[157]:
```
question_prompt = """\
You are a teacher preparing a test. Please create a question that can be␣
 ↪answered by referencing the following context.

Context:
{context}
"""

ground_truth_prompt = """\
Use the following context and question to answer this question using *only* the␣
 ↪provided context.

Question:
{question}

Context:
{context}
"""
```

```
question_prompt = ChatPromptTemplate.from_template(question_prompt)
ground_truth_prompt = ChatPromptTemplate.from_template(ground_truth_prompt)

question_chain = question_prompt | ChatOpenAI(model="gpt-3.5-turbo") |␣
 ↪StrOutputParser()
ground_truth_chain = ground_truth_prompt |␣
 ↪ChatOpenAI(model="gpt-4-turbo-preview") | StrOutputParser()

for chunk in synthetic_data_chunks[:15]:
  questions.append(question_chain.invoke({"context" : chunk.page_content}))
  contexts.append([chunk.page_content])
  ground_truths_semantic.append(ground_truth_chain.invoke({"question" :␣
 ↪questions[-1], "context" : contexts[-1]}))
  answers.append(percentile_rag_chain.invoke(questions[-1]))
```

[158]:
```
qagc_list = []

for question, answer, context, ground_truth in zip(questions, answers,␣
 ↪contexts, ground_truths_semantic):
  qagc_list.append({
      "question" : question,
      "answer" : answer,
      "contexts" : context,
      "ground_truth" : ground_truth
  })

eval_dataset = Dataset.from_list(qagc_list)
```

[159]:
```
result = evaluate(eval_dataset, metrics=[context_precision,
                                         faithfulness,
                                         answer_relevancy,
                                         context_recall]);
```

```
Evaluating:   0%|          | 0/60 [00:00<?, ?it/s]

No statements were generated from the answer.
No statements were generated from the answer.
```

[160]:
```
result_df = result.to_pandas()
result_df
```

[160]:

	question	\
0	The Preamble to the United States Constitution…	
1	Question: According to the context provided, w…	
2	Question: To whom is the letter addressing in …	
3	What is the primary purpose of the actions bei…	
4	Question: Who were some of the prominent churc…	

```
5   Question: Who are some of the notable individu…
6   Question: Who were some of the illustrious men…
7   Question:\nWho were some of the liegemen menti…
8   Question: What did the Magna Carta grant to th…
9   In the context provided, what specific aspect …
10  Question: Who granted, confirmed, and obtained…
11  Question: According to the context, who has be…
12  Question:\nAccording to the context provided, …
13  Question:\nAccording to the context provided, …
14  Question: According to the context, how much w…

                                              answer  \
0                                        I don't know.
1                                        I don't know.
2      The letter is addressing the General Assembly.
3   The primary purpose of the actions being descr…
4                                        I don't know.
5                                        I don't know.
6                                        I don't know.
7                                        I don't know.
8                                        I don't know.
9   The specific aspect of the English Church high…
10                                       I don't know.
11  According to the context, everyone is entitled…
12  According to the context provided, everyone is…
13                                       I don't know.
14                                       I don't know.

                                                contexts  \
0                                             [Preamble:]
1   [John, by the grace of God, king of England, l…
2   [stewards, servants, and to all his bailiffs a…
3   [God and the advancement of his holy Church an…
4   [England and cardinal of the holy Roman Church…
5   [of Coventry, Benedict of Rochester, bishops; …
6   [and of the illustrious men William Marshal, e…
7   [Peter Fitz Herbert, Hubert De Burgh (senescha…
8   [1. In the first place we have granted to God,…
9   [liberties inviolate; and we will that it be t…
10  [pure and unconstrained will, did grant, and d…
11  [and this we will observe, and our will is tha…
12  [liberties, to be had and held by them and the…
13  [2. If any of our earls or barons, or others h…
14  [by the old relief, to wit, the heir or heirs …

                                         ground_truth  context_precision  \
0   According to the Preamble to the United States…                   1.0
```

```
1   Based on the provided context, the author of t…              1.0
2   The letter is addressing stewards, servants, b…              1.0
3   The primary purpose of the actions being descr…              1.0
4   Some of the prominent church figures in Englan…              1.0
5   The notable individuals mentioned in the conte…              1.0
6   The illustrious men mentioned in the context p…              1.0
7   Some of the liegemen mentioned in the context …              1.0
8   The Magna Carta granted the English Church fre…              1.0
9   The specific aspect of the English Church high…              1.0
10  The speaker granted, confirmed, and obtained r…              1.0
11  According to the context, the rights outlined …              1.0
12  According to the context provided, the liberti…              1.0
13  According to the context provided, the conditi…              1.0
14  According to the context, the heir or heirs of…              1.0

    faithfulness  answer_relevancy  context_recall
0       0.000000          0.000000             1.0
1       0.000000          0.000000             1.0
2       0.000000          0.944142             1.0
3       0.000000          1.000000             1.0
4       0.000000          0.000000             1.0
5       0.000000          0.000000             1.0
6       1.000000          0.000000             1.0
7       1.000000          0.000000             1.0
8       1.000000          0.000000             1.0
9       0.333333          0.989799             1.0
10           NaN          0.000000             1.0
11      0.000000          0.920955             1.0
12      0.000000          0.965144             1.0
13           NaN          0.000000             1.0
14      0.000000          0.000000             1.0
```

[161]:
```python
for metric, score in result.items():
    print(f"Metric: {metric}, score: {score}", end='\n')
```

```
Metric: context_precision, score: 0.9999999999
Metric: faithfulness, score: 0.25641025641025644
Metric: answer_relevancy, score: 0.3213360102965338
Metric: context_recall, score: 1.0
```

[162]:
```python
for _, row in result_df.query('faithfulness < 1').iterrows():
    print(row.question, end='\n')
    print("Answer: ", row.answer, end='\n')
    print("Ground truth: ", row.ground_truth, end='\n')
    print("Faithfulness: ", row.faithfulness, end='\n')
    print('----------------------')
```

```
The Preamble to the United States Constitution begins with the famous words, "We
```

the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defence, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America."

Question:
According to the Preamble to the United States Constitution, what are the six purposes for which the Constitution was established?
Answer: I don't know.
Ground truth: According to the Preamble to the United States Constitution, the six purposes for which the Constitution was established are:

1. To form a more perfect Union
2. Establish Justice
3. Insure domestic Tranquility
4. Provide for the common defence
5. Promote the general Welfare
6. Secure the Blessings of Liberty to ourselves and our Posterity
Faithfulness: 0.0

Question: According to the context provided, who is the author of the document?
Answer: I don't know.
Ground truth: Based on the provided context, the author of the document is John, by the grace of God, king of England, lord of Ireland, duke of Normandy and Aquitaine, and count of Anjou.
Faithfulness: 0.0

Question: To whom is the letter addressing in the given context?
Answer: The letter is addressing the General Assembly.
Ground truth: The letter is addressing stewards, servants, bailiffs, and liege subjects.
Faithfulness: 0.0

What is the primary purpose of the actions being described in the given context?
Answer: The primary purpose of the actions being described in the given context is to promote respect for human rights and freedoms and to secure their universal and effective recognition and observance.
Ground truth: The primary purpose of the actions being described in the given context is to promote the interests of God and the holy Church, as well as to bring about improvements or reforms within the realm. This is being done with the counsel of religious leaders, indicating a collaboration between the church and the ruling authority for these aims.
Faithfulness: 0.0

Question: Who were some of the prominent church figures in England during the time period mentioned in the context?
Answer: I don't know.

Ground truth: Some of the prominent church figures in England during the time period mentioned in the context include:

- Henry, archbishop of Dublin
- William of London
- Peter of Winchester
- Jocelyn of Bath and Glastonbury
- Hugh of Lincoln
- Walter of Worcester
Faithfulness: 0.0

Question: Who are some of the notable individuals mentioned in the context provided?
Answer: I don't know.
Ground truth: The notable individuals mentioned in the context provided are Benedict of Rochester, Master Pandulf, and brother Aymeric (master of the Knights of the Temple in England).
Faithfulness: 0.0

In the context provided, what specific aspect of the English Church is highlighted as being of utmost importance and very essential?
Answer: The specific aspect of the English Church highlighted as being of utmost importance and very essential is the right to freedom of thought, conscience, and religion.
Ground truth: The specific aspect of the English Church highlighted as being of utmost importance and very essential is the freedom of elections.
Faithfulness: 0.3333333333333333

Question: According to the context, who has been granted the rights outlined in the document?
Answer: According to the context, everyone is entitled to the rights outlined in the document.
Ground truth: According to the context, the rights outlined in the document have been granted to all freemen of the kingdom, for the grantor and their heirs forever.
Faithfulness: 0.0

Question:
According to the context provided, who is entitled to the liberties mentioned in the text?
Answer: According to the context provided, everyone is entitled to the liberties mentioned in the text.
Ground truth: According to the context provided, the liberties mentioned in the text are entitled to "them and their heirs" of "us and our heirs forever."
Faithfulness: 0.0

Question: According to the context, how much would the heir or heirs of a knight be required to pay as relief?

```
Answer:  I don't know.
Ground truth:  According to the context, the heir or heirs of a knight would be
required to pay 100 shillings at most as relief.
Faithfulness:  0.0
----------------------
```

[163]:
```
plt.bar(list(result.keys()), list(result.values()))
plt.ylabel("Scores")
plt.show()
```