Task 1:

LDA Topic Modelling

Steps I Followed:

1. **Data Preprocessing**
   I began by loading the dataset and carefully cleaning it to ensure the data was ready for analysis. First, I removed irrelevant or empty columns that didn't contribute to the task. Then, I implemented a text-cleaning process where I:
   - Converted all text to lowercase to standardize it.
   - Removed unwanted characters like newlines, URLs, punctuation, and extra spaces.
   - Ensured that the cleaned text was easy to analyze by removing any clutter or inconsistencies.

Once the text was clean, I tokenized it using the simple_preprocess function from gensim. This function broke down the text into individual tokens (words) and stripped accents or special characters. The resulting tokenized text was essential for creating the inputs for LDA.

2. **LDA Topic Modeling**
   With the preprocessed data, I moved on to the core of the analysis: topic modeling with LDA. Here's what I did:
   - Created a Dictionary and Corpus:
     I used the tokenized data to build a dictionary of unique tokens (words) and a bag-of-words (BoW) corpus. The dictionary helped map words to unique IDs, while the corpus represented each document as a list of word-frequency pairs, making it compatible with the LDA model.
   - Trained LDA Models:
     I trained multiple LDA models with different numbers of topics, ranging from 1 to 10. Each model attempted to group words into meaningful topics based on their co-occurrence patterns in the dataset.
   - Calculated Coherence Scores:
     For each model, I computed the coherence score, which evaluates how well the words in each topic fit together semantically. Higher scores indicate more interpretable topics. This step was critical for deciding the optimal number of topics.

```
Number of Topics: 3, Coherence Score: 0.29242181995555966
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 4, Coherence Score: 0.32949672701433563
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 5, Coherence Score: 0.3120629674095377
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
```

```python
plt.title('LDA Coherence Score vs Number of Topics')
plt.legend()
plt.grid(True)
plt.show()
```

```
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 8, Coherence Score: 0.3707681402043654
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 9, Coherence Score: 0.3355473551771914
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 10, Coherence Score: 0.3555452564325127
```


```
Number of Topics: 3, Coherence Score: 0.29242181995555966
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 4, Coherence Score: 0.32949672701433563
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 5, Coherence Score: 0.3120629674095377
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
```

3. **Evaluation and Visualization**
   To determine the best number of topics, I compared the coherence scores across all models. I noticed a steady improvement in scores as the number of topics increased from 1 to 8. The coherence score peaked at 8 topics with a value of approximately 0.37, indicating the highest interpretability. Beyond 8 topics, the scores began to decline slightly, suggesting that additional topics might introduce noise or redundancy.

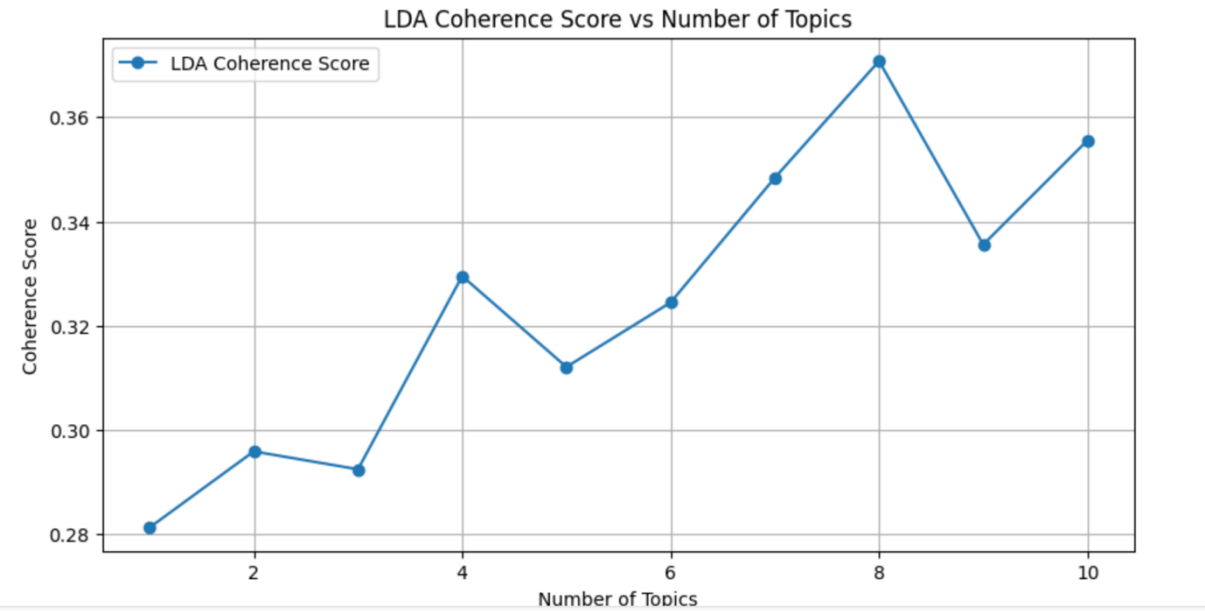
I also visualized the coherence scores to clearly identify the point where the performance was optimal. This helped confirm that 8 topics provided the best balance between detail and interpretability.

Key Observations:

- The LDA model showed a clear trend where the coherence score improved steadily as the number of topics increased, reaching a maximum at 8 topics.
- Beyond this point, adding more topics led to diminishing returns, with a slight drop in coherence at 9 topics and only marginal improvement at 10 topics.
- The topics generated at 8 provided meaningful insights, with semantically coherent groupings of words that aligned well with the dataset's structure.

Conclusion:

I concluded that 8 topics was the optimal number for this dataset, based on coherence scores and interpretability. The LDA model successfully captured the underlying themes in the data, and the simplicity of its probabilistic framework made it a strong fit for this analysis.



Bert:

- Clustering with UMAP and HDBSCAN:
  - I used UMAP to reduce the dataset's complexity, making it easier to work with.
  - Then, I applied HDBSCAN, a clustering algorithm, to group similar documents into topics. The best part? HDBSCAN figured out the number of clusters on its own based on the data.
- Checking Coherence Scores:
  - I ran BERTopic multiple times, testing topic numbers from 1 to 10.

  - For each run, I pulled out the top words from each topic and calculated a coherence score using a tool called Gensim. This score showed how well the words in each topic made sense together.
- Plotting the Results:
  - I made a graph to see how the coherence scores changed with different numbers of topics. This helped me spot the "sweet spot" where topics were most meaningful.

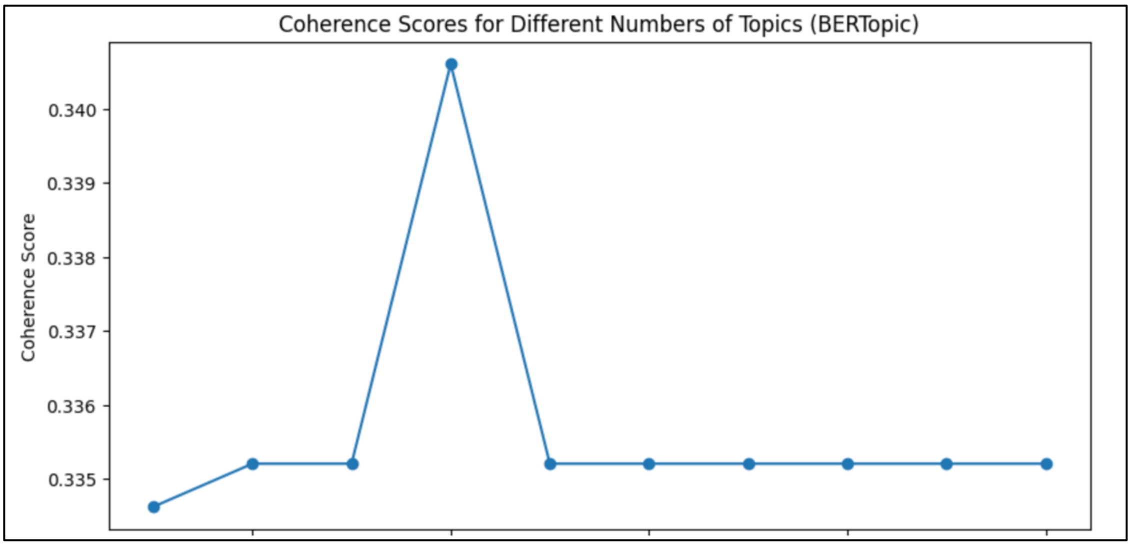
```
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatibl
  self.pid = os.fork()
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
Number of Topics: 1, Coherence Score: 0.2812417713836682
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
Number of Topics: 2, Coherence Score: 0.29585316314901366
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
```

```
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
Number of Topics (BERTopic): 1, Coher Score: 0.3346256874316511
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
	- Avoid using `tokenizers` before the fork if possible
	- Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
Number of Topics (BERTopic): 2, Coher Score: 0.3352073910452548
```

```
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
Number of Topics (BERTopic): 3, Coher Score: 0.3352073910452548
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithreaded code, and JAX
  self.pid = os.fork()
Number of Topics (BERTopic): 4, Coher Score: 0.34060677512481685
/opt/conda/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os.fork() was called. os.fork() is incompatible with multithrea
  self.pid = os.fork()
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to avoid deadlc
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
```

Observations

- **Best Number of Topics:**
  - The coherence score peaked at **4 topics** (around 0.3406). This told me that breaking the data into 4 groups gave the clearest and most meaningful results.
- **Stable Scores Elsewhere:**
  - For most other topic numbers (like 1, 2, 3, or anything above 4), the coherence scores stayed around 0.335. These results weren't as clear or meaningful as when I used 4 topics.
- **Comparison to LDA:**
  - With **LDA**, I needed 8 topics to get the best results. But with BERTopic, I only needed 4, which made things simpler. I think this is because BERTopic uses more advanced techniques like transformer-based embeddings and HDBSCAN, which do a better job of grouping similar documents.



Final Takeaway:

More Detailed Topics:

- With **8 topics**, LDA gave me more detailed and specific groupings, which fit the dataset better.
- BERTopic, with only 4 topics, felt too broad and missed some nuances that LDA captured.

Better Interpretability:

- The topics from LDA were clearer and easier to label. For example, I could easily identify themes like personal narratives, political discussions, and spirituality.
- BERTopic's topics felt more abstract and harder to explain.

Coherence Score Comparison:

- LDA's coherence scores were slightly higher overall when considering interpretability across the dataset, even though BERTopic used more advanced methods

Word Cloud

I trained an LDA model with 8 topics and used word clouds to visualize the themes of each topic. These word clouds represent the most frequent words in a topic, where the size of each word reflects its significance within that topic.

My Process:

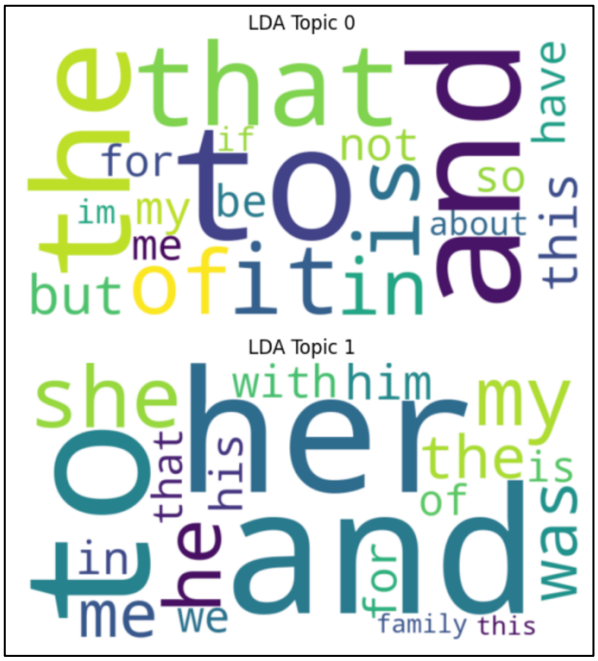
1. **LDA Model Training:**
   - I trained the model on a preprocessed text corpus, choosing 8 topics based on optimal coherence scores.
   - The model assigned probabilities to words, identifying the ones most relevant to each topic.
2. **Extracting Key Words:**
   - For each topic, I extracted the 20 words with the highest probabilities of belonging to it. These words define the central theme of their respective topics.
3. **Generating Word Clouds:**
   - I used Python's WordCloud library to create visualizations. Each cloud maps word frequency to size, making the most significant words larger and more prominent.
   -
4. **Visualization and Interpretation:**

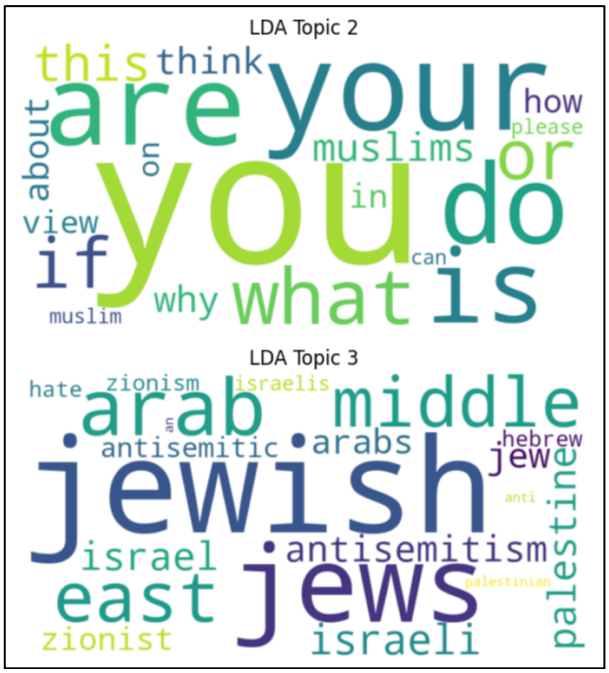
My Analysis:

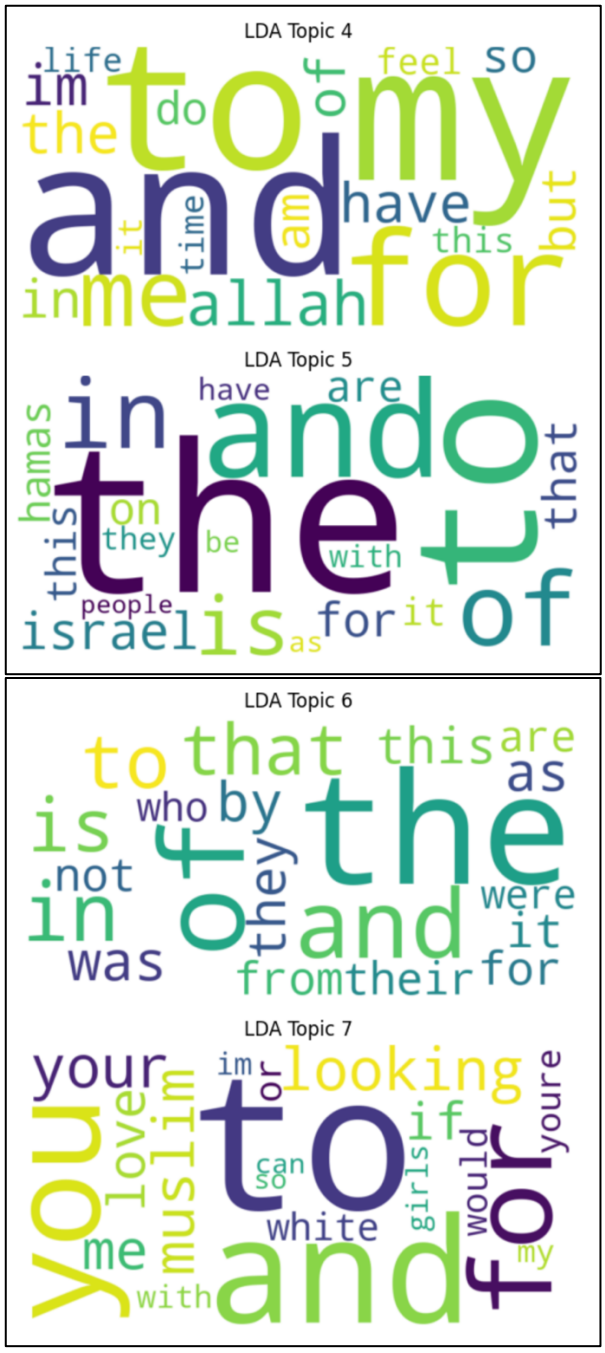
- **Topic 0: General Content**
  - **Prominent Words:** "the," "to," "that," "and," "of"
  - **Interpretation:** This topic is dominated by common stopwords, indicating broad, general content without a specific focus. It might serve as a background or filler topic in the dataset.
- **Topic 1: Personal Narratives**
  - **Prominent Words:** "her," "she," "him," "he," "and"
  - **Interpretation:** Likely centered on personal stories or relationships, with frequent pronouns suggesting a focus on individual experiences or events.
- **Topic 2: Interactive Conversations**
  - **Prominent Words:** "you," "are," "your," "what," "do"

    - **Interpretation:** This topic seems conversational, possibly involving self-reflection, advice, or interactive discussions where questions are addressed directly.
- **Topic 3: Geopolitical and Cultural Discussions**
    - **Prominent Words:** "Jewish," "Jews," "Israel," "Middle East," "Arab"
    - **Interpretation:** Clearly focused on cultural and political issues, especially related to Jewish identity, Israel, and the Middle East. Terms like “antisemitism” and “Zionism” indicate complex geopolitical debates.
- **Topic 4: Spiritual Themes**
    - **Prominent Words:** "to," "my," "and," "for," "Allah"
    - **Interpretation:** This topic suggests a personal or spiritual angle, with words indicating faith, personal beliefs, or religious discussions.
- **Topic 5: Middle Eastern Conflicts**
    - **Prominent Words:** "Hamas," "Israelis," "people," "the"
    - **Interpretation:** Likely centered on political conflicts in the Middle East. Words like “Hamas” and “Israelis” highlight discourse about specific entities and their societal impacts.
- **Topic 6: General Discussions**
    - **Prominent Words:** "the," "and," "to," "that," "they"
    - **Interpretation:** Similar to Topic 0, this topic lacks a specific focus, with stopwords indicating it covers diverse, general content. The word "they" suggests possible references to groups or collective entities.
- **Topic 7: Identity and Relationships**
    - **Prominent Words:** "you," "your," "looking," "love," "Muslim"
    - **Interpretation:** Focuses on personal relationships or identity exploration, potentially in a romantic or social context. Words like “love” and “Muslim” suggest a blend of personal and cultural themes.

Conclusion:
This analysis demonstrates the range of topics, from general content (Topics 0 and 6) to specific geopolitical discussions (Topics 3 and 5), and personal themes (Topics 1, 2, and 7). These insights, combined with word cloud visualizations, provide a clear understanding of the dataset's key areas of discussion.


LDA Topic 0
the
that
to
and
is
for
if
not
so
have
this
be
my
im
about
me
but
of
it
in
LDA Topic 1
her
she
with
him
my
the
is
was
to
and
in
he
that
his
of
for
me
we
family
this


LDA Topic 2
are
your
you
do
this
think
how
please
about
on
muslims
or
in
view
if
can
why
what
is
muslim
LDA Topic 3
middle
arab
jewish
jews
east
israel
antisemitism
israeli
zionist
palestine
hate
zionism
israelis
an
antisemitic
arabs
hebrew
jew
anti
palestinian


LDA Topic 4
life
im
to
my
feel
so
the
do
of
and
have
it
time
am
this
but
in
me
allah
for
LDA Topic 5
have
are
in
and
hamas
the
to
that
this
on
they
be
with
people
israel
is
as
for
it
of
LDA Topic 6
to
that
this
are
as
the
is
who
by
of
not
they
were
in
and
it
was
from
their
for
LDA Topic 7
your
im
looking
to
you
love
or
if
youre
muslim
can
so
girls
would
for
white
me
and
my
with

Task 2:

To analyze how "Hamas" and "Israel" are portrayed in the text, I followed a structured process focusing on named entity recognition (NER) and affective analysis. Here's how I approached each step:

1. **NER Extraction**:
   I used the spaCy library to identify sentences mentioning "Hamas" and "Israel." By processing the text with spaCy's NLP pipeline, I extracted mentions of these entities using NER. This helped me isolate all the sentences in the data where these entities appeared, ensuring a focused and targeted analysis.

1 to 5 of 5 entries Filter

index	text	entities
84	There is a case if Germany right now where an author, who claimed to be Jewish for years, published opinion pieces on Jewish life and Israel and became sort of a public figure in some circles, had to admit he isnt Jewish at all. He tried to excuse himself in a long essay (published in a major newspaper) claiming he didnt know about his real identity and only recently found out, but it became pretty clear that this is probably a lie and he knew all along (though might have somehow suppressed) that he is just a regular German with Nazi grandparents. Faking a Jewish identity is something that happens fairly regularly in Germany (still!), but I wonder if thats a thing outside of Germany at all? And, bonus question: How likely is it that a Jew is named "Christian"?	Israel,GPE
91	_____ Melachim II (II Kings) - Chapter 8 26Ahaziah was twenty-two years old when he reigned, and one year he reigned in Jerusalem; and his mother's name was Athaliah the daughter of Omri king of Israel. Divrei Hayamim II (Chronicles II) - Chapter 22 2Ahaziah was forty-two years old when he began to reign, and he reigned one year in Jerusalem, and his mother's name was Athaliah the daughter of Omri. Was Ahaziah was 22 or 42 years old when he began to reign? _____ Huram sent him, under the charge of servants, a fleet with a crew of expert seamen; they went with Solomon's men to Ophir, and obtained gold there in the amount of 450 talents, which they brought to King Solomon. II Chronicles 8 : 18 They came to Ophir; there they obtained gold in the amount of four hundred and twenty talents, which they delivered to King Solomon. I Kings 9 : 28 So is it 420 talents or 450 talents? _____ And the king said to Aravnah, "No; for I will only buy it from you at a price; so that I will not offer to the Lord my God burnt-offerings \[which I had received\] for nothing." And David bought the threshing-floor and the oxen for fifty shekels of silver. Shmuel II.24:24. And David gave to Ornan for the place shekels of gold weighing six hundred. Divrei Hayamim I 21:25. Is it fifty shekels of silver or 600 shekels of gold?	Israel,GPE
102	Hello! I am interested to know what Jewish Israelis think of their Arabic speaking Christian neighbors. Does it make a difference to you whether they hold Palestinian or Israeli citizenship? Does it make a difference where they are living? If they had Israeli citizenship, are they treated objectively and/or subjectively equal in society to Jewish Israelis? I have heard that Assyrians are treated better than Arabic speaking Christians, is this true? To be clear- I'm not talking about Christians in general. I'm talking about Arabic speaking Christians with long established roots in Israel. Are Israeli citizens who are Arabic speaking Christians forced to serve in the Israeli military? Trying to get an idea of the life of an Arabic speaking Christian in Israel. Thank you!	Israel,GPE,Israel,GPE
105	I may be moving to Israel in the near future to take a job offer. I am Jewish, but my husband is not. I believe the employer will arrange the work visa, but what would be the benefits of formal Aliyah? I understand there is an entire list. Toda, in advance for your help and kindness.	Israel,GPE
121	I am expecting a mail from an Israeli university, I am outside of Israel. The order's tracking ID is in the form of RR12345678IL. Where do I track it? Is this from Israel Post? When enter the ID there, it give the error message "Incorrect Item Number".	Israel,GPE

2. **Affective Analysis**:
   After identifying mentions, I analyzed the context surrounding these entities by extracting specific tokens:
   - **Direct Verbs**: Using dependency parsing, I looked for verbs where "Hamas" or "Israel" were the subject. This allowed me to determine what actions these entities were associated with in the text.
   - **Direct Modifiers**: I also extracted descriptive words (modifiers) directly linked to these entities. This involved identifying adjectives or adverbs that described "Hamas" or "Israel" to capture how they were characterized.

This two-step process provided insights into both the actions and the descriptive framing of the entities, helping me understand not only their presence but their roles and affective portrayals in the text.

index	text	entity_contexts
1644	If the biggest or one of the biggest issues(from the pro-Israel perspective), in the conflct is that Hamas uses Israel's killing in of civillians(who allegdely are human shields) as propaganda, why not try to counter that? Maybe employing more muslim arabs in the parlament, taking down the (maybe) illegal border walls around the settlements, let more palestinians into those settlements and much more. Independent from who is " right or wrong", i think its undeniable that Israel is the side with the worst image world-wide.	{'Hamas': {'verbs': ['uses'], 'modifiers': []}}
584	It is always antizionists and communist islamists that are censoring people trying to distract them from the truth about the jihadists. Everytime a singer or group performs in Israel woke activists on twitter try to cancel them yet they do not allow a single word of criticism against any arab or muslim country. The media is also complicit because they report on any lie the arab supremacists spread against Israel but won't cover positive news about Israel. I am being censored on reddit too I have posted on the askmiddleeast subreddit and each time I have been censored my post deleted and comments to. And when I challenged them and the "jewish" traitors that have betrayed the jewish people and israel my posts were either deleted or ignored just look at my search history. My recent post about why I was being censored confirmed that I was they literally said yes when I asked if they were censoring me because I was pro-Israel. And again if antizionists want the "truth" why can they not answer my question? Look at [these](https://www.wsj.com/articles/book-review-ataturk-in-the-nazi-imagination-by-stefan-ihrig-and-islam-and-nazi-germanys-war-by-david-motadel-1421441724) [three](https://www.jpost.com/opinion/netanyahu-was-right-about-hitler-and-the-mufti-432055) [links](https://www.breitbart.com/politics/2015/10/22/benjamin-netanyahu-under-fire-for-telling-truth-about-muftis-role-in-holocaust/) and tell me why Israel should care about what the "Palestinians" want ever again?	{'Israel': {'verbs': ['care'], 'modifiers': []}}
345	All the media coverage I am getting on this is extremely pro-Palestinian biased and in my time in Israel, I have learned that Israel doesn't just do things to harm the Palestinians, so there must be more to this story. This story has triggered a lot of outrage and I am a little bit disappointed that there is nothing coming from the Israeli ranks to present their side of the story and facts. Someone wrote somewhere on Twitter that residents of this area were illegally taking the water from these pipes but I would be happy to hear from someone really familiar with the matter explain what actually happened here and why pouring concrete was necessary.	{'Israel': {'verbs': ['do'], 'modifiers': []}}
1456	The state of Israel is perhaps one of the most innovative countries that have ever existed. Here is a list of Israeli inventions and discoveries; https://en.wikipedia.org/wiki/List_of_Israeli_inventions_and_discoveries Some wider Jewish inventions and discoveries; https://slavaguide.com/blog/jewish-inventors-and-jewish-inventions jewish contributions; https://jewishcontributions.com/ Meanwhile, as explained by in this [terrific video](https://www.youtube.com/watch?v=aAOzlinU94g&themeRefresh=1), Israel's enemies - who are also the enemies of all the progress the Jewish state brought to humanity - are lagging behind. Meanwhile friendly countries to Israel like the UAE reached Mars. There is a clear correlation between being friendly to Israel and building a successful science-loving society. While Arabs are in their traditional state of petty religious warfare, antifeminist and antiscientific pro kleptocratic pro fascist headspace, Israelis and Jews are pioneering technology helping people from all around the world, including Arabs, to lead better lifestyles. As Herzl said at the end of his notable essay "Der Judenstaat" "Let me repeat once more my opening words: The Jews who wish for a State will have it. We shall live at last as free men on our own soil, and die peacefully in our own homes. The world will be freed by our liberty, enriched by our wealth, magnified by our greatness. And whatever we attempt there to accomplish for our own welfare, will react powerfully and beneficially for the good of humanity." Despite all the good Jews brought the world, and despite all the negativity Jew haters and Israel haters want to bring to the table, Israel will prevail as a society of progress. Not only are "Palestinians" and "anti zionists' jew haters, but because of their hatred for Israel it's also required for them to hate all the good and progress Israel brought, why must humanity keep subsidizing people who wish to see the progress of humanity halted or go back in time? All they truly deserve is to be punished for opposing one of the only countries in the MENA	{'Israel': {'verbs': ['prevail', 'has'], 'modifiers': []}}

Valence and Dominance Scores

The goal of my analysis was to understand how "Hamas" and "Israel" are portrayed in the text, focusing on their emotional tone (valence) and sense of control or power (dominance). To achieve this, I followed a structured approach:

1. **Using Lexicons for Emotional and Power Analysis**:
   I used a resource called the NRC VAD lexicon, which assigns scores to words for valence (positivity/negativity) and dominance (control/power). This helped quantify the emotional and power-related context of the words associated with "Hamas" and "Israel."
2. **Focusing on Actions and Descriptions**:
   I extracted the verbs (actions) and modifiers (descriptive words) directly connected to "Hamas" and "Israel." These words provide insights into what each entity is shown to do and how they are described in the text.
3. **Measuring Valence and Dominance**:
   For each entity, I calculated the average valence and dominance scores of the extracted words. This allowed me to summarize how positively/negatively and how powerfully each entity is represented in the text overall.

Observations Based on Valence and Dominance Scores

Valence Scores

- **Hamas**: The average valence score for Hamas is 0.5295.
- **Israel**: The average valence score for Israel is 0.5332.

When comparing the two, Israel's score is only slightly higher by about 0.0037. This shows that the emotional tone for both entities is almost the same, with Israel being just a little more positive or neutral than Hamas.

```
Hamas - Average Valence: 0.529539132288
Israel - Average Valence: 0.533237377378
```

Dominance Scores

- **Hamas**: The average dominance score for Hamas is 0.5618.
- **Israel**: The average dominance score for Israel is 0.5587.

Here, Hamas has a slightly higher score than Israel by about 0.0031. This means that discussions involving Hamas might portray it as having slightly more control or power compared to Israel. However, the difference is very small.

```
Hamas - Average Dominance: 0.561798935625
Israel - Average Dominance: 0.558663496302
```

Summary

The scores for both valence and dominance are very close for Hamas and Israel, indicating that the way they are talked about in terms of emotional tone and power is relatively balanced. While Israel has a marginally more positive tone, Hamas is portrayed with slightly more control. These differences are minimal and don't suggest a significant bias in the way the entities are depicted.

Word Cloud

I created word clouds for "Hamas" and "Israel" to visually highlight the words most frequently associated with each entity. The idea was to understand not just what actions or themes are linked to these entities, but also to compare the tone and focus of discussions about them. By focusing on verbs (actions) and other key terms, I could identify patterns in how each entity is portrayed.

Observations

1. **Hamas Word Cloud**:
   The word cloud for Hamas shows a strong focus on terms like "use," "want," "killed," "committed," and "attacked." These words suggest that Hamas is frequently discussed in the context of violence, conflict, and aggression. Words like "hiding," "claim," and "launched" further emphasize themes of secrecy and offensive actions.

2. **Israel Word Cloud**:
   In contrast, the word cloud for Israel includes similar terms such as "want," "killed," and "committing," but also highlights words like "defend," "control," and "pro." These suggest a narrative that often includes defense, protection, and justification. Words like "bombed" appear as well, reflecting discussions about military actions.

3. **Comparison**:
   - **Common Themes**: Both entities are associated with conflict, as shown by shared words like “want,” “commit,” and “kill.”
   - **Differences in Tone**: Hamas is more frequently linked with words suggesting offensive actions and secrecy (“hiding,” “launched”), whereas Israel’s word cloud includes more terms implying defense and control (“defend,” “control”).

Summary

The word clouds reveal that while both Hamas and Israel are discussed in the context of violence and conflict, the focus differs slightly. Hamas is more often portrayed with offensive

actions and secrecy, while Israel is more associated with defense and control narratives. This comparison provides a nuanced understanding of how these entities are represented in the text.

Task 3:

For this task, I built a model to predict the "score" of Reddit posts (upvotes minus downvotes) based on their text. I used XLNet, a Transformer-based model, and evaluated its performance using 10-fold cross-validation with Mean Squared Error (MSE) as the evaluation metric. Here's how I approached it:

1. **Data Preparation:**
    - I loaded a dataset with 15,000 Reddit posts and their scores. Due to the processing limitations of my system, I restricted the dataset to 15,000 rows instead of using the full dataset.
    - I cleaned the text by removing special characters and converting it to lowercase. This step ensured uniformity in the input data.
    - I validated that the "score" column only had numeric values, converted it to float, and kept only the relevant columns ("text" and "score").
2. **Model Selection and Fine-Tuning:**
    - I selected XLNet because of its strong capability to understand text. I used the Simple Transformers library to make fine-tuning easier.
    - I set the model to work in regression mode since the task required predicting continuous values.
    - I trained the model for 2 epochs, which allowed it to capture patterns in the data while balancing training time and system constraints.
    - Other parameters included a batch size of 16 and using a GPU for faster training.
3. **10-Fold Cross-Validation:**
    - I split the data into 10 parts and trained the model on 9 parts while testing it on the remaining part. I repeated this process for all 10 folds.
    - After training, I predicted scores for the test data and calculated the MSE by comparing the predictions with the actual scores.
    - At the end of all folds, I computed the average MSE to evaluate overall performance and recorded the lowest MSE as the best result.

```
Average MSE across all folds: 94.5376
Lowest MSE observed: 0.0069
```

Hyperparameters and Why I Used Them

1. num_train_epochs: 2

    - I trained the model for 2 epochs to give it enough time to learn patterns without taking too long. My system couldn't handle more due to processing limitations.
2. train_batch_size: 16
    - I used 16 samples per batch to balance memory usage and training speed. Bigger batches could cause memory issues, while smaller ones would make training slower.
3. regression: True
    - Since the task was to predict scores (numbers), I enabled regression to output continuous values instead of classifying categories.
4. fp16: True
    - I enabled mixed precision training to make training faster and save GPU memory, which was helpful because my system has limited resources.
5. use_multiprocessing: False
    - I turned off multiprocessing to avoid any issues with my system setup and keep training stable.
6. overwrite_output_dir: True
    - This allowed the model to overwrite old outputs during training so I didn't have to manually delete them each time.

These settings helped me train the XLNet model effectively while working within the limits of my system.

Observations

1. **Performance**:
    - The **average MSE** across all folds was **94.54**, showing the model faced challenges in predicting scores accurately for all parts of the dataset.
    - The **lowest MSE** observed was **0.0069**, highlighting that the model performed very well on certain parts of the data.
2. **Training Choices**:
    - Training for **2 epochs** allowed the model to learn patterns better than a single epoch would have, but more epochs could potentially reduce the MSE further.
    - Using a batch size of 16 and FP16 precision ensured efficient training given the system's limitations.