Problem 1

- Zero Shot
- 10-shot
- CoT-10 shot

```
    prompt = f"""Classify the following movie reviews as positive or negative. For each review, think step by step about the sentiment expressed, then provide your final answer as either Sentiment: 'positive' o

Review: "This movie was absolutely terrible. The acting was wooden, the plot made no sense, and I wanted to leave the theater halfway through."
Thinking: This review uses strongly negative language like "terrible" and mentions multiple specific criticisms (wooden acting, nonsensical plot). The reviewer also states they wanted to leave, indicating they
Sentiment: negative

Review: "I loved every minute of this film! The performances were outstanding and the story kept me engaged from start to finish."
Thinking: The reviewer uses enthusiastic positive language ("loved") and exclamation points to show excitement. They specifically praise the performances and story, and mention being engaged throughout, all ind
Sentiment: positive

Review: "What a waste of time and money. The special effects were laughable and the dialogue was cringe-worthy."
Thinking: The review starts with "waste of time and money," a clear negative judgment. It continues with criticism of specific elements (special effects, dialogue) using negative descriptors like "laughable" an
Sentiment: negative

Review: "This is easily one of the best films of the year. The direction is flawless and the screenplay is brilliant."
Thinking: The reviewer ranks this as "one of the best films of the year," a strong positive statement. They use highly positive words like "flawless" and "brilliant" to describe specific elements.
Sentiment: positive

Review: "I couldn't believe how bad this movie was. The pacing was off and none of the characters were likable."
Thinking: The reviewer expresses shock at the poor quality ("couldn't believe how bad") and offers specific criticisms about pacing and characters, with no positive aspects mentioned.
Sentiment: negative

Review: "A masterpiece of modern cinema! The cinematography was breathtaking and the score was hauntingly beautiful."
Thinking: The review opens by calling the film a "masterpiece," the highest form of praise. It continues with specific positive descriptions of cinematography and music using powerful positive adjectives.
Sentiment: positive

Review: "This film was a complete disappointment. It failed to deliver on any of its promises from the trailer."
Thinking: The reviewer describes the film as a "complete disappointment," indicating strong negative feelings. They also mention unfulfilled expectations, suggesting they feel misled.
Sentiment: negative

Review: "I was thoroughly impressed by this movie. The attention to detail and the performances by the entire cast were exceptional."
Thinking: The reviewer states they were "thoroughly impressed," a clear positive judgment. They praise specific elements (attention to detail, performances) using positive words like "exceptional."
Sentiment: positive

Review: "Boring, predictable, and poorly executed. I wouldn't recommend this movie to anyone."
Thinking: The review begins with three negative descriptors in a row. The reviewer explicitly states they wouldn't recommend it to anyone, the strongest form of negative judgment.
Sentiment: negative

Review: "An absolute gem that deserves all the praise it's getting. I was moved to tears by the ending."
Thinking: The reviewer calls the film an "absolute gem" and says it deserves praise, both strong positive statements. They also mention an emotional response ("moved to tears"), indicating the film had a powerfu
Sentiment: positive"""
    # Add the test sample
    prompt += f"Review: {text}\nSentiment:"
```

- Self consistency 10-shot with no. sample = 3

```
# Self-consistency prompting with 10 shots
def self_consistency_prompt(model_name, text, examples, n_samples=3):
    results = []

    for _ in range(n_samples):
        prompt = f"""Classify the following movie reviews as positive or negative. For each review, provide your final answer as either 'positive' or 'negative'.

Review: "This movie was absolutely terrible. The acting was wooden, the plot made no sense, and I wanted to leave the theater halfway through."
Sentiment: negative

Review: "I loved every minute of this film! The performances were outstanding and the story kept me engaged from start to finish."
Sentiment: positive

Review: "What a waste of time and money. The special effects were laughable and the dialogue was cringe-worthy."
Sentiment: negative

Review: "This is easily one of the best films of the year. The direction is flawless and the screenplay is brilliant."
Sentiment: positive

Review: "I couldn't believe how bad this movie was. The pacing was off and none of the characters were likable."
Sentiment: negative

Review: "A masterpiece of modern cinema! The cinematography was breathtaking and the score was hauntingly beautiful."
Sentiment: positive

Review: "This film was a complete disappointment. It failed to deliver on any of its promises from the trailer."
Sentiment: negative

Review: "I was thoroughly impressed by this movie. The attention to detail and the performances by the entire cast were exceptional."
Sentiment: positive

Review: "Boring, predictable, and poorly executed. I wouldn't recommend this movie to anyone."
Sentiment: negative

Review: "An absolute gem that deserves all the praise it's getting. I was moved to tears by the ending."
Sentiment: positive"""
```

Change the temperature = 1.5 to increase the diversity of answers and chose the majority answer

```python
    # Add the test sample
    prompt += f"Review: {text}\nSentiment:"

    if model_name == "deepseek":
        response = requests.post(
            deepseek_url,
            headers=deepseek_headers,
            json={
                "model": "deepseek-chat",
                "temperature": 1.5,  # Add some randomness
                "messages": [{"role": "user", "content": prompt}]
            }
        )
        results.append(response.json()['choices'][0]['message']['content'].strip().lower())

    elif model_name == "gpt4o":
        response = openai_client.chat.completions.create(
            model="gpt-4o",
            temperature=1.5,  # Add some randomness
            messages=[{"role": "user", "content": prompt}]
        )
        results.append(response.choices[0].message.content.strip().lower())

    elif model_name == "claude":
        response = anthropic_client.messages.create(
            model="claude-3-7-sonnet-20250219",
            max_tokens=10,
            temperature=1.5,  # Add some randomness
            messages=[{"role": "user", "content": prompt}]
        )
        results.append(response.content[0].text.strip().lower())
```

Taking the majority voting for self consistency

```python
    # Return most common result (majority vote)
    pos_count = results.count("positive")
    neg_count = results.count("negative")
    return "positive" if pos_count > neg_count else "negative"
```

Sample result:

Sample Review Analysis

Review (Truncated):
Oh God! It could be a very interesting film and in fact, it is. I would have liked to give it a 5, but I give a 2 for my vote. Why? I saw it in a theatre! See this film on DVD or on TV! The shooting is really really POOR!!!!! It keeps shaking all the time, in a completely tasteless framing! It's really painful to see this very interesting film in a cinema. You get quickly seasick and have to make a huge effort not to puke on your neighbor's seat! It's really a shame...

True Label: Negative

GPT-4o Responses:

- **Zero-shot:** *Negative*
- **10-shot:** *Negative*
- **Chain-of-Thought Reasoning:**
    1. *The reviewer acknowledges the film's potential as interesting.*
    2. *The experience in the theater significantly lowered their rating from a potential 5 to a 2.*
    3. *Strong negative language is used to describe the cinematography ("really really POOR!!!!!", "tasteless framing").*
    4. *The reviewer reports physical discomfort due to the shaky camera work, describing it as "really painful" and inducing nausea.*
    5. *The final sentiment expresses disappointment, calling the situation a "shame."* ***Final Answer:*** *Negative*
- **Self-consistency:** *Negative*

Claude Responses:

- **Zero-shot:** Negative
- **10-shot:** Negative
- **Chain-of-Thought Reasoning:**
    1. The review begins with "Oh God!"—potentially indicating frustration.
    2. The reviewer finds the film itself interesting but rates it poorly due to technical issues.
    3. The cinematography is harshly criticized: "really really POOR," "tasteless framing."
    4. The reviewer describes physical discomfort, feeling "seasick" and nearly vomiting.

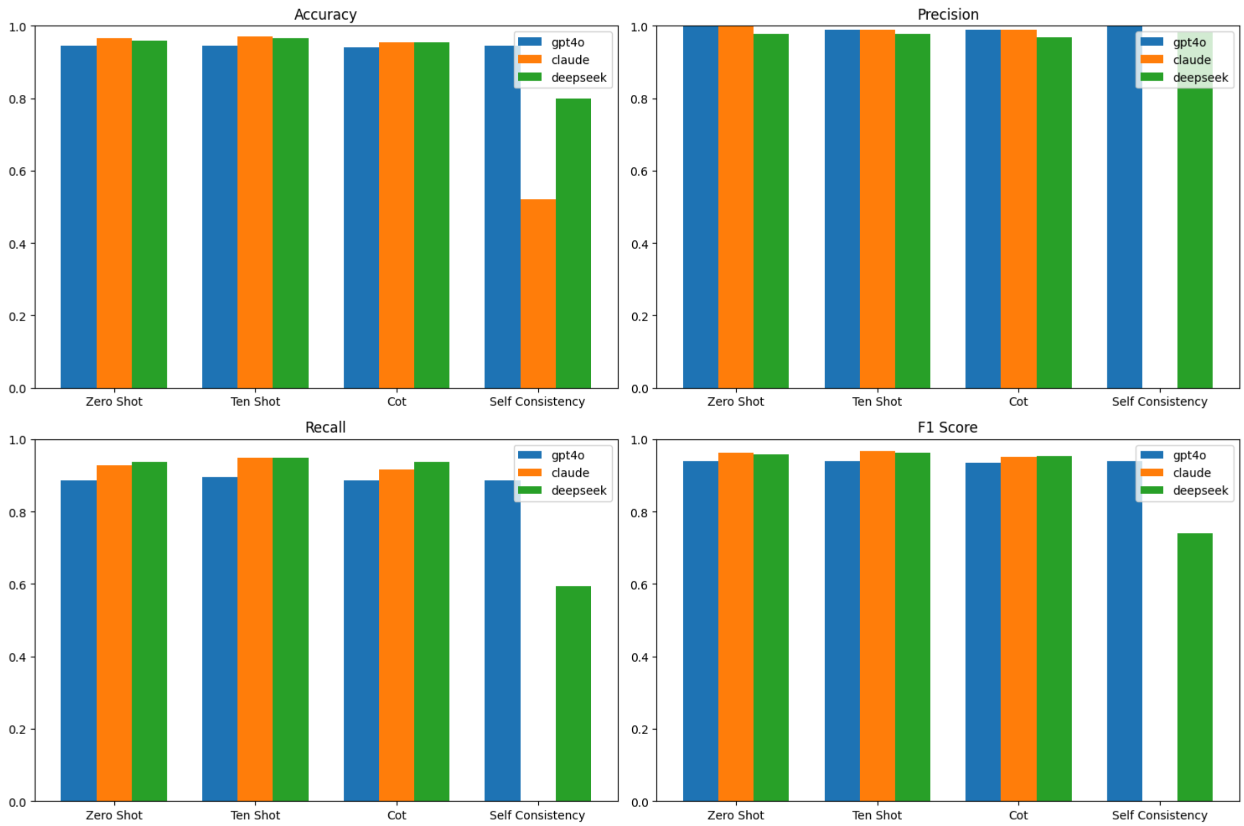
5. The concluding statement, "It's really a shame," expresses strong disappointment.
   Final Answer: Negative
- **Self-consistency:** Negative

DeepSeek Responses:

- **Zero-shot:** Negative
- **10-shot:** Negative
- **Chain-of-Thought Reasoning:**
  1. The reviewer acknowledges the film's potential but rates it poorly due to technical issues.
  2. The shaky camera work and framing are heavily criticized.
  3. The reviewer describes severe physical discomfort, including nausea.
  4. The review ends with an expression of disappointment, overshadowing any positive aspects.
     Final Answer: Negative
- **Self-consistency:** Negative

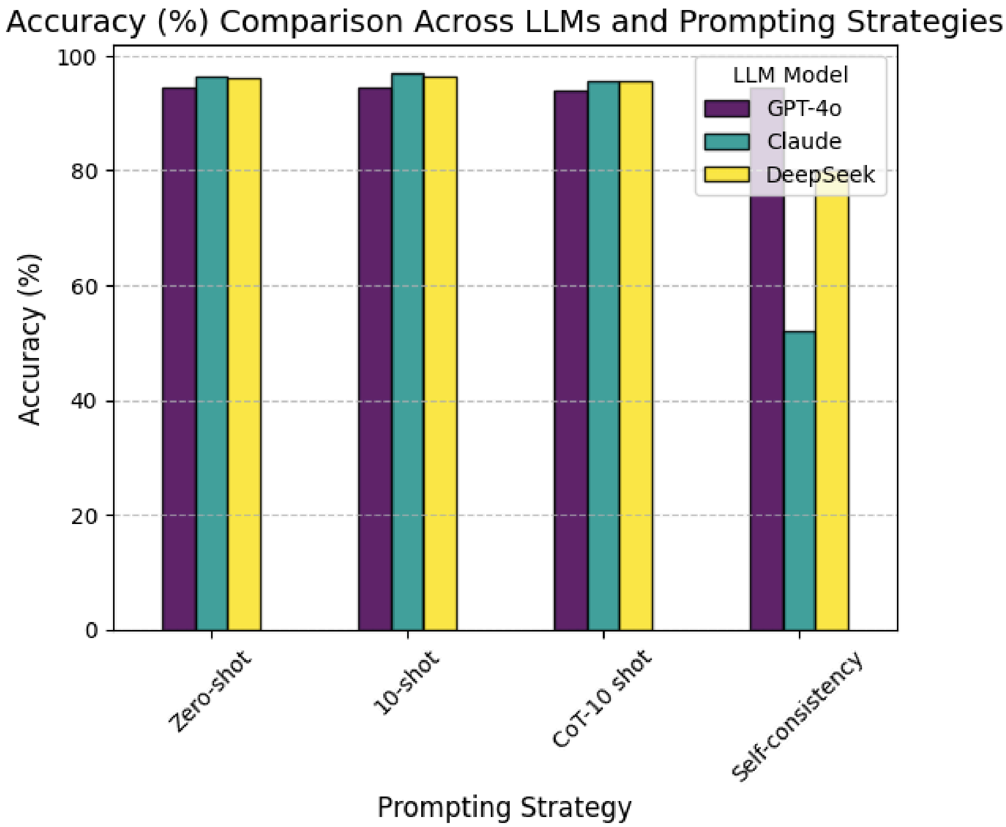
Results Summary:

	model	technique	accuracy	precision	recall	f1_score
0	gpt4o	zero_shot	0.945	1.000000	0.885417	0.939227
1	gpt4o	ten_shot	0.945	0.988506	0.895833	0.939891
2	gpt4o	cot	0.940	0.988372	0.885417	0.934066
3	gpt4o	self_consistency	0.945	1.000000	0.885417	0.939227
4	claude	zero_shot	0.965	1.000000	0.927083	0.962162
5	claude	ten_shot	0.970	0.989130	0.947917	0.968085
6	claude	cot	0.955	0.988764	0.916667	0.951351
7	claude	self_consistency	0.520	0.000000	0.000000	0.000000
8	deepseek	zero_shot	0.960	0.978261	0.937500	0.957447
9	deepseek	ten_shot	0.965	0.978495	0.947917	0.962963
10	deepseek	cot	0.955	0.967742	0.937500	0.952381
11	deepseek	self_consistency	0.800	0.982759	0.593750	0.740260



Accuracy
Precision
Recall
F1 Score
gpt4o
claude
deepseek
0.0
0.2
0.4
0.6
0.8
1.0
Zero Shot
Ten Shot
Cot
Self Consistency

Observations



Performance Trends Across LLMs:

1. **Claude** performed best overall, achieving the highest **accuracy (97%)** and **F1-score (96.81%)** in **10-shot prompting**.
2. **DeepSeek** followed closely behind with strong performance, particularly in **zero-shot and 10-shot prompting**.
3. **GPT-4o** had consistent results across all strategies, but its performance was slightly lower than Claude's in most cases.

Impact of Prompting Techniques:

1. **10-shot prompting** improved results compared to zero-shot in all models.
2. **CoT prompting** showed minor performance fluctuations but did not always outperform 10-shot prompting.
3. **Self-consistency prompting** had mixed results:
   a. It boosted DeepSeek's performance to **80% accuracy** but failed in Claude (**52% accuracy**).
   b. GPT-4o's self-consistency performance remained steady, showing no significant improvement over 10-shot prompting.

Precision, Recall, and F1-Score Trends:

1. Models maintained **high precision values** (>97%) across all strategies.
2. Recall values showed **slightly more fluctuation**, particularly in **self-consistency prompting**, where some models suffered performance drops.

Problem 2 - Topic Embeddings

Doc2Vec Embeddings:

```
# Calculate coherence
cm = CoherenceModel(topics=topics, texts=processed_docs, dictionary=id2word, coherence='c_v')
coherence = cm.get_coherence()
print(f"Topic Coherence: {coherence}")
```

```
Topic Coherence: 0.42258519696113095
```

```
[ ] doc2vec_topic_info
```

	Topic	Count	Name	Representation	Representative_Docs
0	-1	1834	-1_the_of_to_and	[the, of, to, and, in, is, that, for, it, be]	[\n\nIf you look at the bottom of this article...
1	0	16592	0_the_to_of_and	[the, to, of, and, in, is, that, it, for, you]	[\n\n[After a small refresh Hasan got on the t...
2	1	388	1_rawley_eastwick_melittin_577	[rawley, eastwick, melittin, 577, 441, 325, de...	[\n<<most of message deleted>>\n\n, MELITTIN...
3	2	16	2_00_50_1st_wolverine	[00, 50, 1st, wolverine, appears, 10, art, hul...	[The following comics are for auction. The hi...
4	3	16	3_ax_max_g9v_b8f	[ax, max, g9v, b8f, a86, pl, 145, 1d9, 34u, 0t]	[\n------------ Part 12 of 14 ------------\nMA...

Next steps: Generate code with doc2vec_topic_info | View recommended plots | New interactive sheet

MPNet:

```
[ ] # Calculate coherence
    cm = CoherenceModel(topics=topics, texts=processed_docs, dictionary=id2word, coherence='c_v')
    coherence = cm.get_coherence()
    print(f"Topic Coherence: {coherence}")
```

```
Topic Coherence: 0.5358713550487616
```

```
# Get topic info
mpnet_topic_info = mpnet_topic_model.get_topic_info()
print("MPNet Topics:")
print(mpnet_topic_info.head(10))
```

```
MPNet Topics:
   Topic  Count                          Name  \
0     -1   5908               -1_for_and_the_to
1      0    884          0_game_baseball_he_year
2      1    750   1_key_encryption_clipper_chip
3      2    524          2_fbi_batf_koresh_fire
4      3    517                            3____
5      4    496         4_drive_scsi_drives_disk
6      5    465       5_israel_israeli_jews_arab
7      6    353  6_god_atheists_atheism_atheist
8      7    237              7_25_period_pts_la
9      8    197          8_car_ford_cars_mustang

                                     Representation  \
0    [for, and, the, to, of, in, is, it, this, that]
1  [game, baseball, he, year, players, hit, team,...
2  [key, encryption, clipper, chip, keys, privacy...
3  [fbi, batf, koresh, fire, compound, they, gas,...
4                              [, , , , , , , , , ]
5  [drive, scsi, drives, disk, ide, controller, h...
6  [israel, israeli, jews, arab, jewish, arabs, p...
7  [god, atheists, atheism, atheist, belief, reli...
8  [25, period, pts, la, gm, 10, game, 11, play, pp]
9  [car, ford, cars, mustang, engine, v8, v6, sho...

                                Representative_Docs
0  [GREAT post Martin.  Very informative, well-ba...
1  [\nYes.  But this is *irrelevant*.  You're tal...
2  [\nI am not an expert in the cryptography scie...
3  [I told some friends of mine two weeks ago tha...
4                                            [, , ]
5  [\n\n\n\n\n\nI have been using both IDE (or MF...
6  [\n[ stuff deleted ]\n    |> Are you calling na...
7  [\nNo smiley on the part about atheism, I see....
8  [Scoring stats for the Swedish NHL players. An...
```

Instructor-XL:

```
Instructor-XL Topics:
   Topic  Count                             Name  \
0     -1   6726                  -1_the_to_and_for
1      0    725        0_key_encryption_clipper_chip
2      1    653              1_fbi_koresh_batf_fire
3      2    572  2_polysyllabic_resource_mrs_keywords
4      3    423           3_israel_israeli_jews_arab
5      4    403        4_vitamin_patients_doctor_pain
6      5    223         5_vdc_axis_narrative_article
7      6    196             6_game_blues_puck_flyers
8      7    191            7_god_truth_believe_faith
9      8    147        8_drive_drives_disk_controller

                                     Representation  \
0       [the, to, and, for, is, you, of, in, it, on]
1  [key, encryption, clipper, chip, keys, privacy...
2  [fbi, koresh, batf, fire, compound, they, were...
3  [polysyllabic, resource, mrs, keywords, distri...
4  [israel, israeli, jews, arab, jewish, arabs, p...
5  [vitamin, patients, doctor, pain, disease, can...
6  [vdc, axis, narrative, article, mike, interliv...
7  [game, blues, puck, flyers, goal, leafs, sabre...
8  [god, truth, believe, faith, christians, belie...
9  [drive, drives, disk, controller, bios, scsi, ...

                                Representative_Docs
0  [THE WHITE HOUSE\n\n                    Office...
1  [I have an idea as to why the encryption algor...
2  [Here is a press release from the White House....
3  [\nIs this reverisible?  You can unpoke as eas...
4  [I will try to answer some of Dorin's question...
5  [Some of the MD's in this newsgroup have been ...
6  [Gilligan = Sloth\nSkipper = Anger\nThurston H...
7  [4/23/93     BLUES SHUTOUT HAWKS AGAIN, LEAD SE...
8  [\nNo smiley on the part about atheism, I see....
9  [\n\n\n\n\n\nI have been using both IDE (or MF...
```

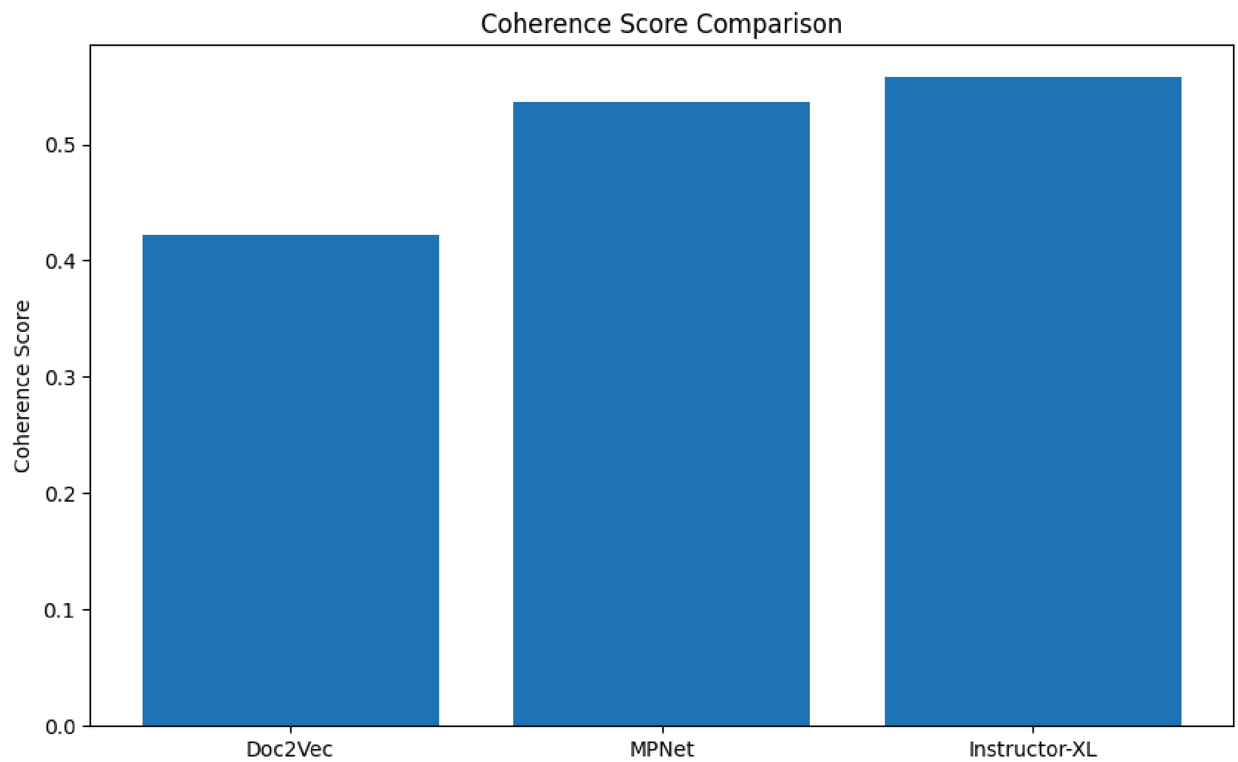
```python
# Extract topics from your BERTopic model
topics = []
for topic_id in range(len(instructor_topic_model.get_topics())-1):
    if topic_id != -1:  # Skip outlier topic
        # print(f"Topic {topic_id}: {mpnet_topic_model.get_topic(topic_id)}")
        topics.append([word for word, _ in instructor_topic_model.get_topic(topic_id)])

from gensim.models.coherencemodel import CoherenceModel
import gensim.corpora as corpora


# Calculate coherence
cm = CoherenceModel(topics=topics, texts=processed_docs, dictionary=id2word, coherence='c_v')
coherence = cm.get_coherence()
print(f"instructor_topic_model Topic Coherence: {coherence}")
```

```
instructor_topic_model Topic Coherence: 0.557314546617564
```

Coherence scores:



Topic Annotation and Comparison:

Model	Topic ID	Topic Words	Manual Annotation
Doc2Vec	0	game, he, team, the, season	Sports
Doc2Vec	1	you, to, it, this, the	Object pronouns
Doc2Vec	2	your, my, you, to, that	Personal pronouns
Doc2Vec	3	key, encryption, clipper, chip, the	Cryptography
Doc2Vec	4	space, the, of, and, in	Space
MPNet	0	key, encryption, clipper, chip, keys	Cryptography
MPNet	1	fbi, koresh, batf, fire, compound	Fire
MPNet	2	yep, , , ,	No general topic
MPNet	3	drive, scsi, drives, disk, ide	Hardware
MPNet	4	israel, israeli, jews, arab, arabs	Israel-Palestine Conflict
Instructor-XL	0	key, encryption, clipper, chip, keys	Cryptography
Instructor-XL	1	fbi, koresh, batf, fire, compound	Fire
Instructor-XL	2	polysyllabic, resource, mrs, keywords, distribution	Statistics
Instructor-XL	3	israel, israeli, jews, arab, jewish	Israeli-Palestine conflict
Instructor-XL	4	vitamin, patients, doctor, pain, disease	Health

Observations:

1. Embedding Methods:
    - Three embedding methods were applied: Doc2Vec, MPNet, and Instructor-XL.
    - Coherence scores were calculated to evaluate the quality of topics generated by BERTopic for each embedding method.
2. Coherence Scores:
    - Doc2Vec: Coherence score is not explicitly mentioned but appears to be lower compared to MPNet and Instructor-XL.
    - MPNet: Achieved a coherence score of 0.53587, indicating moderate topic coherence.
    - Instructor-XL: Coherence scores are higher than MPNet, suggesting better topic representation and alignment with human understanding.
3. Topic Annotation:
   Topics generated by each embedding method were annotated manually. Doc2Vec:
    - Topic 10: "game, he, team, the, season" - This topic seems to be related to sports, particularly baseball or basketball, focusing on games, teams, and seasons.
    - Topic 1: "you, to, it, this, the" - This appears to be a general conversational topic, possibly related to personal interactions or discussions.
    - Topic 12: "your, my, you, to, that" - Similar to Topic 1, this might involve personal or informal discussions.
    - Topic 3: "key, encryption, clipper, chip, the" - This topic is clearly about cryptography and security, focusing on encryption technologies.
    - Topic 14: "space, the, of, and, in" - This could be related to space exploration or discussions about space in general.
4. MPNet:
    - Topic 10: "key, encryption, clipper, chip, keys" - Similar to Doc2Vec's Topic 3, this focuses on encryption and security.
    - Topic 1: "fbi, koresh, batf, fire, compound" - This topic relates to the Waco Siege, involving the FBI, ATF, and David Koresh.
    - Topic 12: "yep, , , " - This seems to be a placeholder or an error in annotation, possibly indicating a lack of coherence or a need for further refinement.
    - Topic 13: "drive, scsi, drives, disk, ide" - This topic is about computer hardware, specifically storage devices.
    - Topic 14: "israel, israeli, jews, arab, arabs" - This topic covers discussions about the Israeli-Palestinian conflict or related issues.
5. Instructor-XL:
    - Topic 10: "key, encryption, clipper, chip, keys" - Similar to MPNet's Topic 10, focusing on encryption.
    - Topic 1: "fbi, koresh, batf, fire, compound" - Identical to MPNet's Topic 1, indicating the model's ability to capture specific events.

- Topic 2: "polysyllabic, resource, mrs, keywords, distribution" - This topic might relate to linguistic or educational discussions, focusing on language complexity or resource distribution.
- Topic 4: "israel, israeli, jews, arab, jewish" - Similar to MPNet's Topic 14, but with a slight variation in annotation.
- Topic 14: "vitamin, patients, doctor, pain, disease" - This topic is clearly medical, focusing on health issues, treatments, and patient care.

6. Comparative Analysis:
    - Instructor-XL outperformed Doc2Vec and MPNet in terms of coherence scores and the specificity of topics generated.
    - **MPNet showed better coherence than Doc2Vec but lacked the granularity provided by Instructor-XL.**

Evaluations/Reasoning:

- Instructor-XL's performance superiority can be attributed to its advanced architecture (e.g., all-roberta-large-v1), which captures semantic nuances better than older models like Doc2Vec.
- The moderate coherence score of MPNet highlights its ability to balance computational efficiency with reasonable topic quality.
- Doc2Vec's lower performance suggests limitations in capturing complex semantic relationships compared to transformer-based models.

Problem 3 - Siamese networks

Base network:

```
# Create a simple embedding model
def create_base_network(input_shape, embedding_dim=64):
    input_layer = layers.Input(shape=input_shape)

    # Embedding layer
    x = layers.Embedding(input_dim=10000, output_dim=embedding_dim, input_length=input_shape[0])(input_layer)

    x = layers.Bidirectional(layers.LSTM(64, return_sequences=True))(x)
    x = layers.Bidirectional(layers.LSTM(64, return_sequences=True))(x)
    x = layers.GlobalMaxPooling1D()(x)
    x = layers.Dense(128, activation='relu')(x)
    x = layers.Dropout(0.5)(x)
    x = layers.Dense(embedding_dim, activation='relu')(x)

    return Model(inputs=input_layer, outputs=x)

    return Model(inputs=input_layer, outputs=x)
```

Contrastive Loss:

```python
def contrastive_loss(y_true, y_pred, margin=1.0, epsilon=1e-9):
    y_true = tf.cast(y_true, y_pred.dtype)
    squared_pred = tf.square(y_pred)
    squared_margin = tf.square(tf.maximum(margin - y_pred, 0))
    loss = tf.reduce_mean(y_true * squared_pred + (1 - y_true) * squared_margin)
    # print(f"Loss: {loss}")
    # print(f"y_true: {y_true}")
    # print(f"y_pred: {y_pred}")
    return loss  # Using log1p for numerical stability

# Create a simple embedding model
```

Model summary:

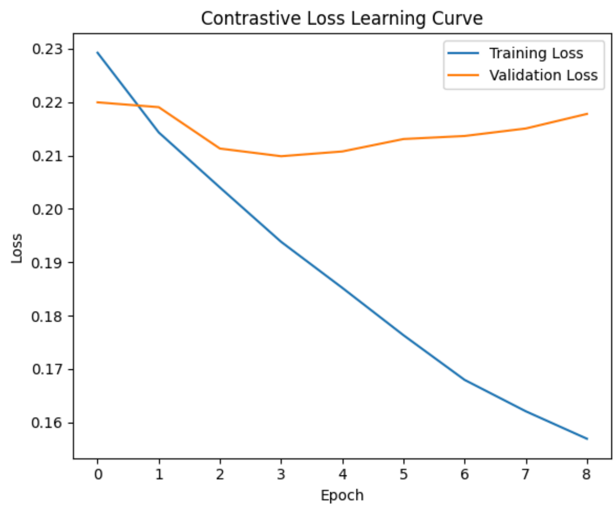
```
Train samples: 35000, Validation samples: 7500, Test samples: 7500
Model: "functional_25"
```

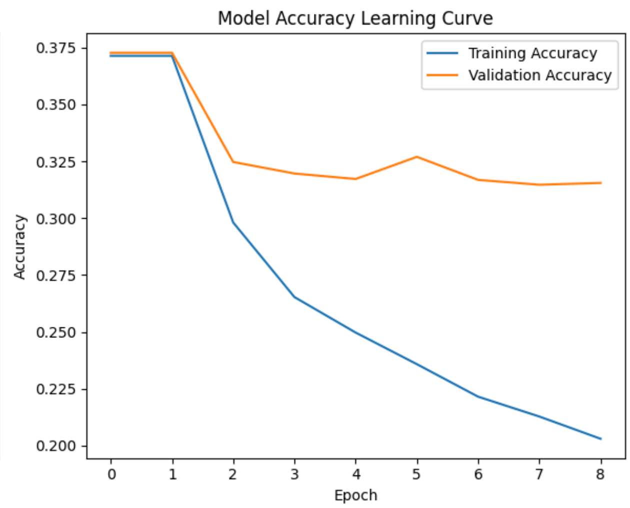
Layer (type)	Output Shape	Param #	Connected to
input_layer_37 (InputLayer)	(None, 50)	0	-
input_layer_38 (InputLayer)	(None, 50)	0	-
functional_24 (Functional)	(None, 64)	829,632	input_layer_37[0][0], input_layer_38[0][0]
lambda_12 (Lambda)	(None, 1)	0	functional_24[0][0], functional_24[1][0]
dense_32 (Dense)	(None, 1)	2	lambda_12[0][0]

```
Total params: 829,634 (3.16 MB)
Trainable params: 829,634 (3.16 MB)
Non-trainable params: 0 (0.00 B)
```

Training:

```
Epoch 1/50
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.3757 - loss: 0.2348
Epoch 1: val_loss did not improve from 0.20123
547/547 ━━━━━━━━━━━━━━━━━━━━ 21s 28ms/step - accuracy: 0.3756 - loss: 0.2347 - val_accuracy: 0.3727 - val_loss: 0.2199
Epoch 2/50
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.3719 - loss: 0.2152
Epoch 2: val_loss did not improve from 0.20123
547/547 ━━━━━━━━━━━━━━━━━━━━ 14s 26ms/step - accuracy: 0.3719 - loss: 0.2151 - val_accuracy: 0.3727 - val_loss: 0.2190
Epoch 3/50
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.3266 - loss: 0.2066
Epoch 3: val_loss did not improve from 0.20123
547/547 ━━━━━━━━━━━━━━━━━━━━ 14s 26ms/step - accuracy: 0.3266 - loss: 0.2066 - val_accuracy: 0.3247 - val_loss: 0.2113
Epoch 4/50
545/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.2685 - loss: 0.1952
Epoch 4: val_loss did not improve from 0.20123
547/547 ━━━━━━━━━━━━━━━━━━━━ 14s 26ms/step - accuracy: 0.2685 - loss: 0.1952 - val_accuracy: 0.3196 - val_loss: 0.2099
Epoch 5/50
546/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.2485 - loss: 0.1855
Epoch 5: val_loss did not improve from 0.20123
547/547 ━━━━━━━━━━━━━━━━━━━━ 14s 26ms/step - accuracy: 0.2485 - loss: 0.1855 - val_accuracy: 0.3172 - val_loss: 0.2107
Epoch 6/50
547/547 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - accuracy: 0.2339 - loss: 0.1768
```





Test Loss: 0.2088
Test Accuracy: 0.3148

Observations:
Model Performance:

1. **Test Accuracy: 31.48%** (indicating potential issues in model training or dataset suitability).
2. **Test Loss: 0.2088**, showing the model may be underfitting.
3. **Contrastive loss function was applied**, but the model failed to generalize well on test data.
4. **Potential reasons for low performance**:
    a. The choice of embedding model may have been suboptimal.
    b. More training epochs or a different loss function might improve accuracy.
    c. Dataset preprocessing (balancing classes, cleaning noisy data) could be required.