

Câu hỏi so sánh mẫu & lời giải

Câu hỏi	Trả lời mẫu
Supervised vs Unsupervised	Supervised: cần dữ liệu gán nhãn, độ chính xác cao; Unsupervised: không cần nhãn, linh hoạt, nhưng độ chính xác phụ thuộc lexicon.
Lexicon-based vs ML-based	Lexicon: dựa từ điển, dễ hiểu, khó mở rộng; ML: học tự động, cần dữ liệu, có thể tổng quát.
Explicit vs Implicit Aspect	Explicit: xuất hiện rõ, dễ trích xuất; Implicit: ngụ ý, khó nhận diện, cần phân tích ngữ cảnh.

1.1 Thực thể (Entity)

- **Định nghĩa:** Đối tượng chính được đánh giá (sản phẩm, dịch vụ, sự kiện, tổ chức...).
- **Ví dụ:** iPhone, máy ảnh, dịch vụ giao hàng.
- **Cặp biểu diễn:** $e = (T, W)$
 - T (Type Hierarchy): Thực thể cấp cao và các thành phần con (ví dụ: Điện thoại → Màn hình, Pin).
 - W (Attributes): Tập thuộc tính của thực thể hoặc thành phần (ví dụ: độ phân giải, dung lượng pin).

1.2 Khía cạnh (Aspect)

- **Định nghĩa:** Khía cạnh là một thành phần hoặc thuộc tính cụ thể của thực thể mà ý kiến được gán.
- **Phân loại:**
 - **Explicit Aspect:** Xuất hiện rõ ràng bằng danh từ/danh từ kép ("camera", "thời lượng pin").
 - **Implicit Aspect:** Được ngụ ý qua tính từ/động từ ("lâu" ⇒ pin, "nhanh" ⇒ hiệu năng).

- **Ví dụ:**
 - Câu: "Camera của điện thoại rất nét." ⇒ Aspect expression: "Camera".
 - Câu: "Máy chạy mượt." ⇒ Implicit aspect: "hiệu năng".

1.3 Người giữ ý kiến (Opinion Holder)

- **Định nghĩa:** Cá nhân hoặc tổ chức phát biểu ý kiến.
- **Ví dụ:** Người dùng Amazon, blogger, reviewer chuyên nghiệp.

1.4 Ý kiến (Opinion) & Cực tính (Polarity)

- **Opinion:** Thái độ, cảm xúc hoặc đánh giá của holder về entity/aspect.
- **Orientation/Polarity:**
 - **Discrete:** {positive, negative, neutral}.
 - **Continuous:** Thang điểm (1–5 sao, [-1,1]).
- **Loại ý kiến:**
 - **Regular Opinion:** Đánh giá đơn lẻ ("Sản phẩm tốt").
 - **Comparative Opinion:** So sánh hai thực thể/aspect ("iPhone 13 nhanh hơn Galaxy S21").
 - **Direct vs Indirect:** Trực tiếp ("Camera đẹp"), gián tiếp ("Chụp ảnh đẹp ⇒ camera tốt").

1.5 Ngũ túr Ý kiến (Opinion Quintuple)

- Cấu trúc 5-tuple: $(e_i, a_{ij}, o_{ijkl}, h_k, t_l)$
 - e_i : Thực thể; a_{ij} : Khía cạnh; o_{ijkl} : Cực tính; h_k : Holder; t_l : Timestamp.
- **Ví dụ:** (iPhone13, battery_life, positive, user123, 2025-04-18).

1.6 Quy trình Khai thác Ý kiến

1. **Entity Extraction:** Nhận diện tên thực thể qua NER hoặc từ vựng.
2. **Aspect Extraction:** Tìm explicit/implicit aspects qua POS-tagging, dependency.

3. **Holder & Time Extraction:** Trích tên người dùng, thời gian từ metadata.

4. **Aspect Sentiment Classification:** Gán polarity cho từng aspect.

5. **Quintuple Generation:** Kết hợp thông tin thành ngũ tú.

6. 1.7 So sánh các khái niệm chính

Khái niệm	Entity	Aspect	Opinion Holder	Polarity
Mô tả	Đối tượng đánh giá	Thành phần/thuộc tính	Nguồn ý kiến	Hướng đánh giá
Trích xuất NER, dictionary		POS, dependency	Metadata, pronouns	Lexicon, classifier
Ví dụ	iPhone	camera, pin	user123	positive, negative

2.2 Câu hỏi lý thuyết (Exam-style)

1. Định nghĩa: Sự khác nhau giữa Explicit và Implicit Aspect

- **Explicit Aspect:** Khía cạnh được đề cập trực tiếp bằng danh từ/noun phrases rõ ràng trong văn bản, ví dụ: "camera", "thời lượng pin". Đề phát hiện qua POS-tagging và từ điển.
- **Implicit Aspect:** Khía cạnh không được nhắc tên mà chỉ ngụ ý qua tính từ/động từ/adjectives, ví dụ: "máy chạy mượt" (ngụ ý về hiệu năng), "pin nhanh hết" (ngụ ý về thời lượng pin). Cần phân tích ngữ cảnh sâu để nhận diện.
- **Khác biệt chính:** Explicit aspects rõ ràng, ít nhầm lẫn, còn implicit aspects linh hoạt hơn nhưng khó xác định, phụ thuộc ngữ nghĩa.

So sánh: Supervised vs Unsupervised Sentiment Classification

Tiêu chí	Supervised	Unsupervised
Dữ liệu	Cần tập dữ liệu gán nhãn (stars, positive/negative).	Không cần dữ liệu nhãn, dùng lexicon hoặc PMI/SO stats.
Độ chính xác	Thường cao nếu đủ dữ liệu chất lượng.	Thấp hơn, phụ thuộc lexicon và thống kê từ corpus.

Chi phí & Công sức	Tốn thời gian gán nhãn, huấn luyện mô hình.	Nhanh, dễ thực hiện với lexicon có sẵn.
Khả năng tổng quát	Tốt trên miền huấn luyện, có thể kém trên miền khác.	Thích ứng linh hoạt với ngôn ngữ miền, nhưng khó đạt độ phủ lexicon.
Phụ thuộc Miền	Cần gán nhãn lại cho mỗi miền/ngữ cảnh mới.	Có thể điều chỉnh lexicon cho từng miền.

3. Phân biệt: Lexicon-based vs Machine Learning-based Approaches

- **Lexicon-based Approaches:**

- Dựa trên từ điển sentiment lexicon (list of positive/negative words).
- Tính polarity bằng cách đếm hoặc gán trọng số các từ tích cực/tiêu cực.
- **Ưu điểm:** Minh bạch, không cần huấn luyện, dễ triển khai nhanh.
- **Nhược điểm:** Khó mở rộng, không xử lý tốt ngữ cảnh, negation, sarcasm kém.

- **Machine Learning-based Approaches:**

- Sử dụng mô hình ML (Naïve Bayes, SVM, Neural Networks) với features (TF-IDF, embeddings).
- Huấn luyện trên dữ liệu gán nhãn để học quy tắc phân loại.
- **Ưu điểm:** Có khả năng học ngữ cảnh phức tạp, độ chính xác cao với đủ dữ liệu.
- **Nhược điểm:** Cần lượng dữ liệu lớn, mất thời gian huấn luyện, thiếu tính giải thích (black-box).

2.2 Câu hỏi lý thuyết (Exam-style)

1. Định nghĩa: Sự khác nhau giữa Explicit và Implicit Aspect

- **Explicit Aspect:** Khía cạnh được đề cập trực tiếp bằng danh từ/noun phrases rõ ràng trong văn bản, ví dụ: "camera", "thời lượng pin". Đề phát hiện qua POS-tagging và từ điển.
- **Implicit Aspect:** Khía cạnh không được nhắc tên mà chỉ ngụ ý qua tính từ/động từ/adjectives, ví dụ: "máy chạy mượt" (ngụ ý về hiệu năng), "pin nhanh hết" (ngụ ý về thời lượng pin). Cần phân tích ngữ cảnh sâu để nhận diện.
- **Khác biệt chính:** Explicit aspects rõ ràng, ít nhầm lẫn, còn implicit aspects linh hoạt hơn nhưng khó xác định, phụ thuộc ngữ nghĩa.

2. So sánh: Supervised vs Unsupervised Sentiment Classification

Tiêu chí	Supervised	Unsupervised
Dữ liệu	Cần tập dữ liệu gán nhãn (stars, positive/negative).	Không cần dữ liệu nhãn, dùng lexicon hoặc PMI/SO stats.
Độ chính xác	Thường cao nếu đủ dữ liệu chất lượng.	Thấp hơn, phụ thuộc lexicon và thống kê từ corpus.
Chi phí & Công sức	Tốn thời gian gán nhãn, huấn luyện mô hình.	Nhanh, dễ thực hiện với lexicon có sẵn.
Khả năng tổng quát	Tốt trên miền huấn luyện, có thể kém trên miền khác.	Thích ứng linh hoạt với ngôn ngữ miền, nhưng khó đạt độ phủ lexicon.
Phụ thuộc Miền	Cần gán nhãn lại cho mỗi miền/ngữ cảnh mới.	Có thể điều chỉnh lexicon cho từng miền.

3. Phân biệt: Lexicon-based vs Machine Learning-based Approaches

• Lexicon-based Approaches:

- Dựa trên từ điển sentiment lexicon (list of positive/negative words).
- Tính polarity bằng cách đếm hoặc gán trọng số các từ tích cực/tiêu cực.
- **Ưu điểm:** Minh bạch, không cần huấn luyện, dễ triển khai nhanh.

- **Nhược điểm:** Khó mở rộng, không xử lý tốt ngữ cảnh, negation, sarcasm kém.
- **Machine Learning-based Approaches:**
 - Sử dụng mô hình ML (Naïve Bayes, SVM, Neural Networks) với features (TF-IDF, embeddings).
 - Huấn luyện trên dữ liệu gán nhãn để học quy tắc phân loại.
 - **Ưu điểm:** Có khả năng học ngữ cảnh phức tạp, độ chính xác cao với đủ dữ liệu.
 - **Nhược điểm:** Cần lượng dữ liệu lớn, mất thời gian huấn luyện, thiếu tính giải thích (black-box).

2.3 Câu hỏi so sánh mẫu & lời giải Câu hỏi so sánh mẫu & lời giải

Câu hỏi	Trả lời mẫu
Supervised vs Unsupervised	Supervised: cần dữ liệu gán nhãn, độ chính xác cao; Unsupervised: không cần nhãn, linh hoạt, nhưng độ chính xác phụ thuộc lexicon.
Lexicon-based vs ML-based	Lexicon: dựa từ điển, dễ hiểu, khó mở rộng; ML: học tự động, cần dữ liệu, có thể tổng quát.
Explicit vs Implicit Aspect	Explicit: xuất hiện rõ, dễ trích xuất; Implicit: ngụ ý, khó nhận diện, cần phân tích ngữ cảnh.

2.4 Bài tập Ví dụ / Example Exercises & Solutions

Bài tập 1: Xây dựng mô hình SVM phân loại tin tức tích cực/tiêu cực

- **Dữ liệu:** Sử dụng Reuters News hoặc IMDB (news vs review).
- **Các bước:**
 1. **Tiền xử lý:** Làm sạch text, tokenize, loại bỏ stopwords, stemming/lemmatization.
 2. **Feature extraction:** Áp dụng TF-IDF (unigrams + bigrams).

3. **Chia tập:** Train/test (80/20).
4. **Huấn luyện SVM:** Sử dụng `sklearn.svm.SVC(kernel='linear')`.
5. **Đánh giá:** Accuracy, Precision, Recall, F1-score.

Bài tập 2: Triển khai Word2Vec trên tập tweet, thực nghiệm dimensionality

- **Dữ liệu:** Tập tweet (có thể thu thập qua Twitter API hoặc sử dụng tập có sẵn).
- **Các bước:**
 1. **Tiền xử lý:** Lowercase, remove mentions/hashtags/URLs, tokenize.
 2. **Huấn luyện Word2Vec:** Dùng `gensim.models.Word2Vec` với các giá trị `vector_size = [50,100,200]`.
 3. **So sánh chất lượng:** Lấy một tập câu so sánh (ví dụ king - man + woman) và tính cosine similarity.

Bài tập 3: Fine-tune BERT cho Sentiment Classification trên IMDB

- **Dữ liệu:** IMDB reviews (25k train, 25k test).
- **Các bước:**
 1. **Chuẩn bị:** Cài `transformers`, tải `bert-base-uncased`.
 2. **Tokenization:** Dùng `BertTokenizer` với `max_length=256`, padding.
 3. **Tạo Dataset & DataLoader (PyTorch):**
 4. **Xây dựng mô hình:** `BertForSequenceClassification.from_pretrained(...)`.
 5. **Huấn luyện:** AdamW optimizer, lr=2e-5, epochs=3.
 6. **Đánh giá:** đo accuracy, F1.

1. Định nghĩa: Sự khác nhau giữa Explicit và Implicit Aspect

- *Define: What is the difference between explicit and implicit aspect?*
- Trả lời (Tiếng Việt):** Explicit aspect là khía cạnh được nhắc trực tiếp bằng danh từ hoặc cụm danh từ, ví dụ "camera", "thời lượng pin". Implicit aspect

được ngũ ý qua tính từ, động từ hoặc trạng từ, ví dụ "máy chạy mượt" hay "pin nhanh hết". Explicit dễ trích xuất, implicit cần phân tích ngữ cảnh.

2. So sánh: Supervised vs Unsupervised Sentiment Classification

- *Compare: What are the advantages and disadvantages of supervised versus unsupervised sentiment classification?*

Trả lời (Tiếng Việt): Supervised cần dữ liệu gán nhãn, độ chính xác cao với đủ dữ liệu nhưng tốn thời gian; unsupervised không cần nhãn, triển khai nhanh nhưng độ chính xác thấp và phụ thuộc lexicon.

3. Phân biệt: Lexicon-based vs Machine Learning-based Approaches

- *Distinguish: How do lexicon-based approaches differ from machine learning-based approaches in sentiment analysis?*

Trả lời (Tiếng Việt): Lexicon-based dựa vào từ điển từ cảm xúc, dễ hiểu nhưng kém ngữ cảnh; ML-based học từ dữ liệu gán nhãn, có thể học ngữ cảnh phức tạp nhưng cần nhiều dữ liệu.

4. Phân biệt: Document-level vs Sentence-level vs Aspect-level

- *Distinguish: What are the differences between document-level, sentence-level, and aspect-level sentiment classification?*

Trả lời (Tiếng Việt): Document-level gán nhãn toàn văn bản, đơn giản; sentence-level gán từng câu, chi tiết hơn nhưng lẩn khía cạnh; aspect-level gán từng khía cạnh, chính xác nhất nhưng phức tạp.

5. Giải thích: Handle Negation trong Lexicon-based Approach

- *Explain: How is negation handled in lexicon-based sentiment analysis?*

Trả lời (Tiếng Việt): Các từ phủ định như "không", "never" đảo ngược polarity của từ gần nhất ("không tốt" từ +1 thành -1), cần thêm quy tắc cho cấu trúc đặc biệt.

6. Giải thích: PMI trong Unsupervised Sentiment Classification

- *Explain: What role does PMI play in unsupervised sentiment analysis?*

Trả lời (Tiếng Việt): PMI đo xác suất đồng xuất hiện giữa hai từ so với độc lập. Sentiment Orientation = $\text{PMI}(\text{phrase}, \text{"positive"}) - \text{PMI}(\text{phrase}, \text{"negative"})$, giúp xác định polarity.

7. So sánh: CBOW vs Skip-gram (word2vec)

- *Compare: What are the differences between CBOW and Skip-gram architectures in word2vec?*

Trả lời (Tiếng Việt): CBOW dự đoán từ từ ngữ cảnh, nhanh và hiệu quả với dữ liệu lớn nhưng kém với từ hiếm; Skip-gram dự đoán ngữ cảnh từ từ, chậm hơn nhưng tốt với từ hiếm.

8. Định nghĩa: Sentiment Orientation vs Sentiment Polarity

- *Define: What is the difference between sentiment orientation and sentiment polarity?*

Trả lời (Tiếng Việt): Sentiment orientation là hướng chung (tích cực/tiêu cực), sentiment polarity là giá trị số biểu diễn độ mạnh yếu (ví dụ +1, -1 hoặc thang sao).

9. Phân biệt: Direct vs Indirect Opinions

- *Distinguish: How do direct opinions differ from indirect opinions?*

Trả lời (Tiếng Việt): Direct opinions đánh giá trực tiếp entity/aspect; indirect thể hiện gián tiếp qua kết quả hoặc ảnh hưởng ("thuốc làm tay tôi đỡ đau" ⇒ positive opinion về thuốc).

10. Giải thích: Phương pháp Trích xuất Opinion Holder

- *Explain: What methods are used to extract opinion holders?*

Trả lời (Tiếng Việt): Dùng Named Entity Recognition (NER), khai phá pronouns, trích metadata (user ID, author), rule-based dựa trên cấu trúc câu.

11. So sánh: PMI vs Word Embedding Similarity for Sentiment

- *Compare: How does PMI-based sentiment scoring differ from using word embedding similarity?*

Trả lời (Tiếng Việt): PMI dựa trên thống kê co-occurrence và lexicon tĩnh; embedding similarity dùng khoảng cách vector, linh hoạt ngữ cảnh nhưng cần training embedding.

12. Giải thích: TF-IDF và vai trò trong feature extraction

- *Explain: What is TF-IDF and why is it useful for feature extraction?*

Trả lời (Tiếng Việt): $\text{TF-IDF} = \text{tf}(t,d) \times \text{idf}(t,D)$, cân bằng tần suất term trong document và tần suất across corpus, giúp giảm trọng số từ phổ biến không phân biệt.

13. Giải thích: Vai trò của N-grams trong Sentiment Analysis

- Explain: What is the role of n-grams in sentiment analysis?

Trả lời (Tiếng Việt): N-grams (bi-grams, tri-grams) giúp nắm bắt cụm từ, idioms, phủ định và phrase-level features tốt hơn so với unigram.

14. Phân biệt: CNN vs RNN trong Text Classification

- Distinguish: How do CNN and RNN architectures differ for text classification?

Trả lời (Tiếng Việt): CNN nắm local patterns qua convolution, xử lý nhanh, RNN (LSTM) model sequence, nắm ngữ cảnh dài nhưng chậm và khó train sâu.

15. Giải thích: Attention Mechanism và Transformer

- Explain: What is the attention mechanism and how does the Transformer architecture use it?

Trả lời (Tiếng Việt): Attention tính trọng số giữa các token, cho phép model tập trung vào vị trí quan trọng. Transformer sử dụng multi-head self-attention để parallel và capture long-range dependencies.

16. So sánh: ELMo vs BERT Embeddings

- Compare: How do ELMo and BERT embeddings differ?

Trả lời (Tiếng Việt): ELMo dùng bi-directional LSTM, contextualized nhưng không truly bidirectional; BERT dùng bidirectional Transformer, pre-trained với Masked LM & NSP, mạnh hơn.

17. Giải thích: Masked Language Modeling (MLM)

- Explain: What is masked language modeling and why is it used in BERT?

Trả lời (Tiếng Việt): MLM randomly masks tokens and trains model dự đoán token bị mask, giúp học representation bidirectional và ngữ cảnh đầy đủ.

18. Phân biệt: Fine-tuning vs Feature-based Use of Pre-trained Models

- Distinguish: What is the difference between fine-tuning and feature-based use of pre-trained language models?

Trả lời (Tiếng Việt): Fine-tuning cập nhật toàn bộ tham số model trên

task; feature-based chỉ giữ embedding output làm feature, không cập nhật weights core.

19. Giải thích: ROC-AUC Metric

- Explain: What does ROC-AUC measure in classification?

Trả lời (Tiếng Việt): ROC-AUC đo trade-off giữa TPR và FPR qua threshold, giá trị gần 1 cho hiệu quả phân loại tốt.

20. So sánh: Precision vs Recall

- Compare: What is the difference between precision and recall?

Trả lời (Tiếng Việt): Precision = TP/(TP+FP) đo độ chính xác dự đoán positive; recall = TP/(TP+FN) đo khả năng tìm ra positive thực.

21. Định nghĩa: What is aspect-based opinion mining?

Trả lời (Tiếng Việt): Aspect-based opinion mining là quá trình trích xuất và phân loại ý kiến dựa trên từng khía cạnh của thực thể trong văn bản.

22. Phân biệt: On what basis do you choose supervised learning algorithms in sentiment classification?

Trả lời (Tiếng Việt): Chọn dựa trên độ phức tạp của dữ liệu, kích thước tập nhãn, tốc độ huấn luyện và khả năng mở rộng.

23. Explain: What is the purpose of feature selection in sentiment analysis?

Trả lời (Tiếng Việt): Giảm chiều dữ liệu, loại bỏ nhiễu, cải thiện hiệu suất và tốc độ huấn luyện mô hình.

24. Compare: Bigrams vs Trigrams in capturing context.

Trả lời (Tiếng Việt): Bigrams bắt cặp hai từ, trigrams bắt ba từ, trigrams nắm ngữ cảnh rộng hơn nhưng tăng kích thước không gian feature.

25. Define: Word Sense Disambiguation in sentiment analysis.

Trả lời (Tiếng Việt): WSD là quá trình xác định nghĩa đúng của từ đa nghĩa trong ngữ cảnh, giúp phân tích polarity chính xác.

26. Distinguish: Polarity Shift vs Intensity Modulation.

Trả lời (Tiếng Việt): Polarity shift là đảo chiều tính cực/tiêu cực; intensity modulation thay đổi độ mạnh của polarité (ví dụ rất tốt vs tốt).

27. Explain: What is the role of part-of-speech tagging in aspect extraction?

Trả lời (Tiếng Việt): POS-tagging giúp xác định danh từ, danh từ kép, tính từ, từ đó trích xuất explicit aspects và opinion words.

28. Compare: Rule-based vs Statistical approaches in opinion holder extraction.

Trả lời (Tiếng Việt): Rule-based dùng các quy tắc ngôn ngữ, chính xác với mẫu cố định; statistical dựa mô hình ML, linh hoạt nhưng cần dữ liệu huấn luyện.

29. Define: What is sentiment lexicon?

Trả lời (Tiếng Việt): Sentiment lexicon là tập các từ/ngữ mang tính tích cực hoặc tiêu cực kèm trọng số hoặc nhãn.

30. Explain: What is the difference between macro-average and micro-average F1-score?

Trả lời (Tiếng Việt): Macro-average tính trung bình F1 qua các lớp; micro-average tổng hợp TP, FP, FN rồi tính F1, phù hợp với data imbalance.

31. Define: What is topic modeling and its relation to sentiment analysis?

Trả lời (Tiếng Việt): Topic modeling trích xuất chủ đề chính từ văn bản; tích hợp với phân tích sentiment để hiểu sentiment theo chủ đề.

32. Distinguish: Named Entity Recognition vs Entity Extraction in opinion mining.

Trả lời (Tiếng Việt): NER xác định tên riêng (người, tổ chức), entity extraction có thể bao gồm tên sản phẩm, khái niệm, rộng hơn NER.

33. Explain: What is the impact of imbalanced classes in sentiment datasets?

Trả lời (Tiếng Việt): Gây bias mô hình về lớp chiếm đa số, làm giảm recall lớp thiểu số và F1-score chung.

34. Compare: Oversampling vs Undersampling for handling class imbalance.

Trả lời (Tiếng Việt): Oversampling nhân bản dữ liệu lớp thiểu số, có thể gây overfitting; undersampling giảm dữ liệu lớp đa số, mất thông tin.

35. Define: What is a confusion matrix?

Trả lời (Tiếng Việt): Ma trận biểu diễn TP, TN, FP, FN, dùng để đánh giá chi tiết hiệu quả phân loại.

36. Explain: What are stopwords and why remove them?

Trả lời (Tiếng Việt): Stopwords là từ thông dụng không mang nghĩa phân biệt (và, là, nhưng), loại bỏ để giảm nhiễu và kích thước dữ liệu.

37. Define: What is stemming vs lemmatization?

Trả lời (Tiếng Việt): Stemming cắt gốc từ cơ bản bằng rule, nhanh nhưng không chính xác; lemmatization dùng từ điển, ra lemma đúng ngữ nghĩa nhưng tốn công sức.

38. Compare: Batch vs Stochastic Gradient Descent in training sentiment models.

Trả lời (Tiếng Việt): Batch cập nhật gradient trên toàn bộ data, ổn định nhưng chậm; SGD cập nhật trên mỗi sample, nhanh nhưng nhiều cao.

39. Explain: What is overfitting and how to prevent it in sentiment models?

Trả lời (Tiếng Việt): Overfitting là mô hình học quá kỹ noise, kém khái quát; dùng regularization, dropout, early stopping để ngăn.

40. Define: What is cross-validation and its purpose?

Trả lời (Tiếng Việt): Cross-validation chia data thành nhiều folds, train/test nhiều lần để đánh giá mô hình ổn định và tránh bias dữ liệu.

41. Analyze: How can sarcasm detection be integrated into sentiment analysis?

Trả lời (Tiếng Việt): Có thể tích hợp bằng cách sử dụng mô hình học sâu huấn luyện trên tập dữ liệu gán nhãn sarcasm, kết hợp các đặc trưng như ngữ cảnh trái nghĩa, dấu hiệu từ vựng và biểu thức cảm xúc mâu thuẫn.

42. Explain: What are the challenges in multilingual sentiment analysis?

Trả lời (Tiếng Việt): Khó khăn gồm từ vựng đặc trưng ngôn ngữ, cấu trúc cú pháp khác nhau, thiếu lexicon và dữ liệu gán nhãn, cũng như sự khác biệt văn hóa trong diễn đạt cảm xúc.

43. Compare: Zero-shot vs Few-shot learning in cross-domain sentiment tasks.

Trả lời (Tiếng Việt): Zero-shot không cần dữ liệu mới ở miền đích, khó đạt chính xác cao; few-shot sử dụng một số ít ví dụ để fine-tune, cải thiện độ phù hợp nhưng vẫn đòi hỏi khả năng tổng quát tốt.

44. Evaluate: Why is BERT considered better than LSTM in many NLP tasks?

Trả lời (Tiếng Việt): BERT học ngữ cảnh song song theo cả hai chiều nhờ kiến trúc transformer, trong khi LSTM tuần tự, khó bắt dependencies dài và chậm hơn.

45. Discuss: What are hybrid models in sentiment analysis and give an example.

Trả lời (Tiếng Việt): Hybrid models kết hợp phương pháp từ điển và học máy để tận dụng ưu điểm cả hai. Ví dụ: dùng lexicon để gán nhãn ban đầu rồi huấn luyện SVM.

46. Justify: When would you use rule-based sentiment systems over ML-based?

Trả lời (Tiếng Việt): Khi dữ liệu gán nhãn hạn chế, yêu cầu giải thích rõ ràng, áp dụng trong môi trường có cấu trúc ngôn ngữ ổn định và nhỏ gọn như chatbot đơn giản.

47. Define: What is domain adaptation in sentiment analysis?

Trả lời (Tiếng Việt): Domain adaptation là điều chỉnh mô hình huấn luyện trên miền nguồn sao cho hoạt động tốt ở miền đích có khác biệt từ vựng, phong cách hoặc cấu trúc.

48. Analyze: How do attention scores help interpret sentiment models?

Trả lời (Tiếng Việt): Attention score cho biết mô hình tập trung vào từ/cụm từ nào khi đưa ra dự đoán, giúp lý giải quyết định mô hình một cách trực quan và minh bạch.

49. Compare: Latent Dirichlet Allocation (LDA) vs Non-negative Matrix Factorization (NMF) in topic extraction.

Trả lời (Tiếng Việt): LDA dựa trên phân phối xác suất, tốt cho dữ liệu lớn, nhưng chậm và khó hội tụ; NMF nhanh hơn, đơn giản hơn nhưng yêu cầu chuẩn hóa dữ liệu tốt.

50. Design: How would you build a real-time sentiment dashboard for Twitter data?

Trả lời (Tiếng Việt): Kết hợp Twitter API để thu thập dữ liệu, xử lý stream bằng Spark hoặc Kafka, phân loại với mô hình ML (hoặc BERT), lưu kết quả vào database, hiển thị bằng dashboard như Grafana hoặc Dash.

Tính TF-IDF và Bài tập nâng cao

1. Công thức TF-IDF và Giải thích thông số (Chi tiết)

TF-IDF (Term Frequency - Inverse Document Frequency) là một phương pháp trọng số trong khai phá văn bản. Nó đánh giá mức độ quan trọng của một từ trong một văn bản so với toàn bộ tập tài liệu.

- **Công thức tổng quát:** $TF-IDF(t,d,D) = tf(t,d) \times idf(t,D)$
- **Giải thích các thành phần:**
 - **t:** Từ cần tính trọng số.
 - **d:** Văn bản hiện tại.
 - **D:** Tập hợp toàn bộ các văn bản.

- **tf(t, d)**: Tần suất của từ t trong văn bản d.
 - $tf(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$: Số lần từ t xuất hiện chia cho tổng số từ trong d.
 - Hoặc dùng logarit: $tf(t,d) = \log(1 + f_{t,d})$
 - **idf(t, D)**: Độ quan trọng của từ t trong toàn bộ tập D.
 - $idf(t,D) = \log(\frac{N}{df(t)})$, trong đó:
 - N: Tổng số tài liệu.
 - df(t): Số tài liệu có chứa từ t.
 - Có thể thêm smoothing: $idf(t,D) = \log(\frac{N+1}{df(t)+1})$
-

2. Ví dụ cơ bản: "excellent"

Câu hỏi: Tính TF-IDF của từ "excellent" trong tài liệu d1, biết:

- $tf = 3$ ("excellent" xuất hiện 3 lần trong d1)
- $df = 5$ (xuất hiện trong 5 tài liệu)
- $N = 100$ (tổng số tài liệu)

Lời giải:

$$TF-IDF = \frac{tf}{N} \times \log\left(\frac{N}{df}\right) = \frac{3}{100} \times \log\left(\frac{100}{5}\right) = 0.03 \times \log(20) \approx 0.03 \times 3.903 = 0.117$$

3. Ví dụ tổng hợp: "apple"

Tập văn bản:

- d1: "apple orange apple fruit"
- d2: "orange fruit banana"

- d3: "banana fruit orange"

Yêu cầu: Tính TF-IDF cho từ "apple" trong d1.

Giải:

- $f("apple", d1) = 2$
 - Tổng số từ d1 = 4 $\Rightarrow tf = \frac{2}{4} = 0.5$
 - $df("apple") = 1$ (chỉ có trong d1)
 - $N = 3 \Rightarrow idf = \log_{10}(3) \approx 0.4771$
 - $TF-IDF = 0.5 \times 0.4771 = \mathbf{0.2386}$
-

4. Bài tập nâng cao có giải thích

Bài 1: Từ "data" trong d1

Tập văn bản:

- d1: "data science is awesome"
- d2: "big data is powerful"
- d3: "data and science are related"
- d4: "statistics is important in science"
- d5: "data is everywhere"

Giải:

- $tf("data", d1) = 1 / 4 = 0.25$
- $df("data") = 4$
- $N = 5 \Rightarrow idf = \log(5/4) \approx 0.0969$
- $TF-IDF = 0.25 \times 0.0969 \approx \mathbf{0.0242}$

Bài 2: So sánh từ "learning" và "the"

- $\text{tf}(\text{"learning"}) = 2$, $\text{df} = 10$
- $\text{tf}(\text{"the"}) = 20$, $\text{df} = 10000$
- $N = 10000$

Giải:

- $\text{idf}(\text{"learning"}) = \log(10000 / 10) = \log(1000) \approx 3.0000 \Rightarrow \text{TF-IDF} = 2 \times 3 = 6$
 - $\text{idf}(\text{"the"}) = \log(10000 / 10000) = 0 \Rightarrow \text{TF-IDF} = 20 \times 0 = 0 \Rightarrow \text{"learning" đặc trưng hơn "the"}$
-

5. Bài tập tự luyện có gợi ý giải

Câu 1: "machine" trong d1

Tập văn bản:

- d1: "deep learning is part of machine learning"
- d2: "machine learning is used in AI"
- d3: "AI is transforming the world"

Gợi ý:

- $\text{tf}(\text{"machine"}, \text{d1}) = 1 / 6$
- $\text{df} = 2$ (xuất hiện trong d1 và d2)
- $N = 3 \Rightarrow \text{idf} = \log(3 / 2) \approx 0.1761$
- $\text{TF-IDF} = \text{tf} \times \text{idf} \approx (1/6) \times 0.1761 \approx 0.02935$

Câu 2: So sánh các cách tính tf và idf

- $\text{tf} = 4$
- $\text{df} = 2$
- $N = 50$

(a) TF nguyên bản:

- $tf = 4$, $idf = \log(50 / 2) \approx \log(25) \approx 1.3979$
- $TF-IDF = 4 \times 1.3979 \approx \mathbf{5.5916}$

(b) TF logarit:

- $tf = \log(1 + 4) = \log(5) \approx 0.699$
- $TF-IDF = 0.699 \times 1.3979 \approx \mathbf{0.9778}$

(c) IDF smoothing:

- $idf = \log(50 / (2 + 1)) = \log(50 / 3) \approx 1.2218$
- $TF-IDF = 4 \times 1.2218 \approx \mathbf{4.8872}$

Tính TF-IDF và Bài tập mở rộng

1. Công thức TF-IDF và Giải thích thông số (Chi tiết)

TF-IDF (Term Frequency - Inverse Document Frequency) là một phương pháp trọng số trong khai phá văn bản. Nó đánh giá mức độ quan trọng của một từ trong một văn bản so với toàn bộ tập tài liệu.

- **Công thức tổng quát:**

- Công thức tổng quát: $TF-IDF(t, d, D) = tf(t, d) \times idf(t, D)$
 - Giải thích các thành phần:
 - **t**: Từ cần tính trọng số.
 - **d**: Văn bản hiện tại.
 - **D**: Tập hợp toàn bộ các văn bản.
 - **tf(t, d)**: Tần suất của từ t trong văn bản d.
 - $tf(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$: Số lần từ t xuất hiện chia cho tổng số từ trong d.
 - Hoặc dùng logarit: $tf(t, d) = \log(1 + f_{t,d})$
 - **idf(t, D)**: Độ quan trọng của từ t trong toàn bộ tập D.
 - $idf(t, D) = \log\left(\frac{N}{df(t)}\right)$, trong đó:
 - **N**: Tổng số tài liệu.
 - **df(t)**: Số tài liệu có chứa từ t.
 - Có thể thêm smoothing: $idf(t, D) = \log\left(\frac{N}{df(t)+1}\right)$
-

2. Bài tập nâng cao với giải chi tiết

Câu 1: So sánh từ "AI" và "is"

Tập văn bản:

- **d1**: "AI is the future."
- **d2**: "AI is already here."
- **d3**: "AI and robotics."
- **d4**: "This is about innovation."

Yêu cầu: Tính TF-IDF cho từ "AI" và "is" trong d1. Dùng $tf =$ tần suất tương đối.

Lời giải:

- $tf("AI", d1) = 1 / 4 = 0.25; df("AI") = 3; N = 4 \Rightarrow idf = \log(4/3) \approx 0.1249 \Rightarrow TF-IDF("AI") = 0.25 \times 0.1249 \approx 0.0312$
- $tf("is", d1) = 1 / 4 = 0.25; df("is") = 3 \Rightarrow idf \text{ giống AI} \Rightarrow TF-IDF("is") = 0.0312 \Rightarrow$
Cả hai từ đều phổ biến, trọng số thấp.

Câu 2: Tính TF-IDF từ "vision" trong văn bản dài

Văn bản:

- $d1$ có 100 từ, trong đó "vision" xuất hiện 5 lần
- "vision" có mặt ở 4 tài liệu trong tổng số 500 tài liệu

Lời giải:

- $tf = 5 / 100 = 0.05$
- $idf = \log(500 / 4) = \log(125) \approx 2.0969$
- $TF-IDF = 0.05 \times 2.0969 \approx 0.1048$

Câu 3: Dạng logarit

Thông số:

- $f("model", d) = 7 \Rightarrow tf = \log(1 + 7) = \log(8) \approx 0.9031$
- $df = 50, N = 1000 \Rightarrow idf = \log(1000 / 50) = \log(20) \approx 1.3010$
- $TF-IDF = 0.9031 \times 1.3010 \approx 1.1747$

Câu 4: So sánh nhiều từ trong cùng tài liệu

Văn bản: "data mining uses data and models to find patterns"

Từ cần tính: "data", "models", "patterns"

- $tf("data") = 2/9 \approx 0.2222, df = 10$
- $tf("models") = 1/9 \approx 0.1111, df = 5$
- $tf("patterns") = 1/9 \approx 0.1111, df = 3$
- $N = 100$

Lời giải:

- $\text{idf}(\text{"data"}) = \log(100/10) = 1.0000 \Rightarrow \text{TF-IDF} = 0.2222 \times 1 = \mathbf{0.2222}$
- $\text{idf}(\text{"models"}) = \log(100/5) = \log(20) \approx 1.3010 \Rightarrow \text{TF-IDF} = 0.1111 \times 1.3010 \approx \mathbf{0.1446}$
- $\text{idf}(\text{"patterns"}) = \log(100/3) \approx 1.5229 \Rightarrow \text{TF-IDF} = 0.1111 \times 1.5229 \approx \mathbf{0.1692} \Rightarrow$
"patterns" mang thông tin phân biệt cao nhất.

3. Bài tập tự luyện mở rộng

Câu A: Một từ xuất hiện 6 lần trong tài liệu 200 từ, và trong 2 tài liệu trong tổng số 100. Tính TF-IDF (dùng tf thường và log idf).

Gợi ý giải:

- $\text{tf} = 6 / 200 = 0.03$
- $\text{idf} = \log(100 / 2) = \log(50) \approx 1.6989$
- $\text{TF-IDF} = 0.03 \times 1.6989 \approx \mathbf{0.0509}$

Câu B: So sánh TF-IDF của từ "neural" (xuất hiện 4 lần trong d1, có mặt ở 5/500 tài liệu) và "the" (xuất hiện 20 lần trong d1, có mặt ở 490/500 tài liệu).

Gợi ý giải:

- $\text{tf}(\text{"neural"}) = 4 / T, \text{tf}(\text{"the"}) = 20 / T$
- $\text{idf}(\text{"neural"}) = \log(500 / 5) = \log(100) = 2$
- $\text{idf}(\text{"the"}) = \log(500 / 490) = \log(1.02) \approx 0.0086 \Rightarrow \text{TF-IDF}(\text{"neural"})$ lớn hơn nhiều lần so với "the"

Câu A: Tính TF-IDF

Đề bài: Một từ xuất hiện 6 lần trong tài liệu 200 từ, và trong 2 tài liệu trong tổng số 100. Tính TF-IDF.

Lời giải chi tiết:

- **Bước 1: Tính TF**
 - Số lần xuất hiện của từ = 6

- Tổng số từ trong văn bản = 200
 -  $TF = 6 / 200 = \mathbf{0.03}$
 - **Bước 2: Tính IDF**
 - $df = 2$ (số tài liệu chứa từ)
 - $N = 100$ (tổng tài liệu)
 -  $IDF = \log_{10}(100 / 2) = \log_{10}(50) \approx \mathbf{1.6989}$
 - **Bước 3: TF-IDF**
 - $TF-IDF = 0.03 \times 1.6989 \approx \mathbf{0.05097}$
 -  Kết quả: $TF-IDF \approx \mathbf{0.051}$
-

Câu B: So sánh TF-IDF của hai từ

Đề bài: So sánh TF-IDF của "neural" và "the" trong cùng tài liệu d1. Biết:

- "neural": xuất hiện 4 lần trong d1, có mặt ở 5 tài liệu trong 500.
- "the": xuất hiện 20 lần trong d1, có mặt ở 490 tài liệu trong 500.

Lời giải chi tiết:

- **Giả sử tổng số từ trong tài liệu d1 là T.**
- **TF:**
 - $tf("neural") = 4 / T$
 - $tf("the") = 20 / T$
- **IDF:**
 - $idf("neural") = \log_{10}(500 / 5) = \log_{10}(100) = \mathbf{2}$
 - $idf("the") = \log_{10}(500 / 490) \approx \log_{10}(1.0204) \approx \mathbf{0.0086}$
- **TF-IDF:**

- $\text{TF-IDF}(\text{"neural"}) = (4 / T) \times 2 = \mathbf{8} / T$
- $\text{TF-IDF}(\text{"the"}) = (20 / T) \times 0.0086 \approx \mathbf{0.172} / T$

 **Kết luận:** TF-IDF của "neural" lớn hơn rất nhiều lần so với "the", chứng tỏ "neural" là từ đặc trưng cho tài liệu, còn "the" là từ phổ biến, không có khả năng phân biệt.