<u>Task 1</u>: Dựa trên bài đọc gợi ý trong phần Reading Materials, viết bài khảo sát gồm những nội dung sau:

a) Tóm tắt hệ thống hóa các bài toán cơ bản trong NLP, bao gồm định nghĩa cơ bản và ví dụ. Trong những bài toán này, bài toán nào phổ biến với ngôn ngữ Tiếng Việt?

Các bài toán cơ bản trong NLP:

- **Phân loại (Classification):**
    - Text Classification:
        - Định nghĩa: Gán nhãn củ đề cho câu hoặc tài liệu.
        - Ví dụ: Phân loại email có spam hay không.
    - Sentiment Analysis:
        - Định nghĩa: Xác định cực tính (tích cực, tiêu cực, trung tính) của văn bản.
        - Ví dụ: Đánh giá phản hồi "Dịch vụ tốt, hài lòng!" là tích cực.
- **Truy hồi và xếp hạng (Information Retrieval and Document Ranking):**
    - Sentence/Document Similarity:
        - Định nghĩa: Đo độ giống nhau giữa hai văn bản.
        - Ví dụ: So sánh hai câu "Máy bay hạ cánh" và "Máy bay đáp xuống sân bay" có nghĩa tương tự nhau.
    - Question Answering:
        - Định nghĩa: Trả lời câu hỏi được ghi dưới dạng ngôn ngữ tự nhiên.
        - Ví dụ:
            - Q: "Năm 2025, chủ tịch nước của Việt Nam là ai?"
            - A: "Ông Lương Cường"
- **Sinh văn bản (Text-to-Text Generation):**
    - Machine Translation:
        - Định nghĩa: Dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác.
        - Ví dụ: Dịch "Hello" -> "Xin chào"
    - Text Generation:
        - Định nghĩa: Tạo văn bản tự nhiên, giống như con người viết.
        - Ví dụ: Tự động ghi tiểu sử cho 1 người nào đó
    - Text Summarization:
        - Định nghĩa: Rút gọn nội dung bài viết nhưng vẫn giữ được ý chính.
        - Ví dụ:Tóm tắt một bài báo dài thành 1 đoạn văn ngắn
    - Text Simplification:
        - Định nghĩa: Làm cho văn bản dễ đọc và hiểu hơn nhưng vẫn giữ ý chính/
        - Ví dụ: Viết câu phức tạp thành câu đơn giản.
    - Lexical Normalization:
        - Định nghĩa: Chuẩn văn bản không chuẩn về dạng chuẩn.

        - Ví dụ: Chuẩn hoá các từ viết tắt “ko” -> “không”
    - Paraphrase Generation:
        - Định nghĩa: Sinh ra câu mới có cùng nghĩa nhưng sử dụng từ ngữ và cấu trúc khác.
        - Ví dụ: “Anh ấy lái xe đi chơi” -> “Anh ấy đi chơi bằng xe ô tô”.

- **Kiến thức và thực thể (Knowledge bases, entities and relations):**
    - Relation Extraction:
        - Định nghĩa: Trích xuất quan hệ ngữ nghĩa giữa các thực thể.
        - Ví dụ: Xác định “Hà Nội là thủ đô của Việt Nam” – quan hệ “thủ đô của”.
    - Relation prediction:
        - Định nghĩa: Dự đoán quan hệ giữa hai thực thể có tên trong văn bản.
        - Ví dụ: Cho cặp “Ông A”, “Ông B” và văn bản, xác định họ có quan hệ cha con.
    - Named Entity Recognition:
        - Định nghĩa: Gán nhãn loại thực thể cho từng từ trong văn bản.
        - Ví dụ:
            - “Nguyễn Văn A” -> PER (person)
            - “Hà Nội” -> LOC (location)
    - Entity Linking:
        - Định nghĩa: Gán thực thể đã nhận dạng vào một mục trong cơ sở tri thức (ví dụ Wikidata).
        - Ví dụ: “Paris” được nối tới trang Wikidata của Paris ở Pháp, không phải Paris ở Texas.

- **Chủ đề và từ khoá (Topics and Keywords):**
    - Topic Modeling:
        - Định nghĩa: Nhận diện chủ đề trừu tượng, tiềm ẩn bên trong toàn bộ văn bản
        - Ví dụ: Phân chủ đề cho tin tức: chính trị, thể thao, giáo dục.
    - Keyword Extraction:
        - Định nghĩa: Trích các từ/ cụm từ quan trọng nhất mô tả nội dung văn bản.
        - Ví dụ: Từ bài báo rút ra keyword “binary”, “classify”, “language”

- **Lý luận văn bản (Text Reasoning):**
    - Common Sense Reasoning:
        - Định nghĩa: Dùng tri thức thường thức để suy luận trên văn bản.
        - Ví dụ: “Anh ấy ăn bánh mì mỗi sáng” → ngầm hiểu là thức ăn chứ không phải đồ uống.
    - Natural Language Inference::

- Định nghĩa: Xác định câu giả thuyết có bị kéo theo, mâu thuẫn, hay trung tính với câu tiền đề.
- Ví dụ:
    - Tiền đề: "Con mèo đang ngủ trên ghế."
    - Giả thuyết: "Có con mèo trên ghế." → Entailment.

Trong những bài toán này, các bài toán phổ biến với ngôn ngữ Tiếng Việt: sentiment analysis, named entity recognition (NER), part of speech tagging (POS Tagging), word segmentation, intent detection and slot filling, machine translation, text classification, question answering...

b) Trong những bài toán trên, những bài toán nào có thể dùng LLM để giải quyết ? ví dụ: bằng cách prompt engineering.

-> Phần lớn các bài toán trong danh sách có thể được giải hoặc hỗ trợ bởi các mô hình ngôn ngữ lớn (LLM) thông qua kỹ thuật thiết kế prompt (prompt engineering), ở những mức độ khác nhau tùy theo bản chất nhiệm vụ. Hiệu quả thường tốt hơn khi người thiết kế nắm rõ đặc trưng bài toán, mục tiêu đầu ra, dạng dữ liệu và các ràng buộc, rồi tinh chỉnh prompt phù hợp (chỉ dẫn rõ ràng, ví dụ minh họa, ràng buộc định dạng hoặc chiến lược few-shot/chain-of-thought khi cần).

Ví dụ:

1. **Phân loại văn bản (Text Classification)**
   Prompt: "Phân loại câu sau vào nhóm chủ đề 'thể thao', 'chính trị' hoặc 'giáo dục': 'Đội tuyển Việt Nam vừa giành chiến thắng trước Thái Lan.'"
   Kết quả mong đợi: thể thao

2. **Phân tích cảm xúc (Sentiment Analysis)**
   Prompt: "Đánh giá cảm xúc của câu sau (tích cực, tiêu cực, trung tính): 'Sản phẩm này rất tốt, tôi rất hài lòng.'"
   Kết quả mong đợi: tích cực

3. **Trả lời câu hỏi (Question Answering)**
   Prompt: "Dựa vào đoạn sau, trả lời câu hỏi:
   Văn bản: 'Hà Nội là thủ đô của Việt Nam.'
   Câu hỏi: Thủ đô của Việt Nam là thành phố nào?"
   Kết quả mong đợi: Hà Nội

4. **Dịch máy (Machine Translation)**
   Prompt: "Dịch câu sau sang tiếng Anh: 'Xin chào, bạn khỏe không?'"
   Kết quả mong đợi: Hello, how are you?

5. **Tóm tắt văn bản (Text Summarization)**
   Prompt: "Tóm tắt bài báo sau thành 2 câu: [nội dung bài báo dài]"

Kết quả mong đợi: Tóm tắt ngắn gọn, đúng ý chính.

6. **Sinh văn bản (Text Generation)**
   Prompt: "Viết một đoạn văn giới thiệu về thành phố Đà Nẵng."
   Kết quả mong đợi: Một đoạn văn giới thiệu về Đà Nẵng.

7. **Viết lại câu (Paraphrase Generation)**
   Prompt: "Viết lại câu sau cho khác đi nhưng nghĩa không đổi: 'Tôi rất thích đọc sách.'"
   Kết quả mong đợi: Tôi cực kỳ yêu thích việc đọc sách.

8. **Điền thông tin (Slot Filling)**
   Prompt: "Trích xuất ngày và giờ từ câu: 'Tôi muốn đặt bàn ăn tối lúc 19h ngày 12/9.'"
   Kết quả mong đợi: ngày: 12/9, giờ: 19h

9. **Phát hiện tin giả (Fake News Detection)**
   Prompt: "Câu sau có phải là tin giả không: 'Cả nước sẽ bị cúp điện liên tục trong 3 ngày".
   Kết quả mong đợi: tin giả.

c) Trong những bài toán trên, bài toán nào có thể áp dụng trong việc xây dựng một ứng dụng chatbot ?

- Intent Detection: Nhận dạng ý định người dùng ("Đặt vé máy bay", "Hỏi thời tiết", "Đặt báo thức"…)
- Slot Filling: Trích xuất thông tin chi tiết từ câu lệnh ("Đặt lịch họp lúc 9h ngày mai", "Đặt phòng khách sạn tại Đà Nẵng từ 15/9 đến 17/9"…)
- Dialog Management: Quản lý trạng thái hội thoại, ghi nhớ ngữ cảnh, duy trì luồng đối thoại tự nhiên.
- Question Answering: Trả lời câu hỏi trực tiếp của người dùng
- Text Generation: Sinh câu trả lời tự nhiên, đa dạng cho người dùng.
- Paraphrase Generation: Đa dạng hóa cách trả lời, tránh lặp lại máy móc.
- Named Entity Recognition: Nhận dạng tên người, địa điểm, ngày tháng… để xử lý yêu cầu chính xác.

d) Khảo sát 1 số thư viện có sẵn giải quyết các bài toán này.

Nhóm 1: Xử lý văn bản, tiền xử lý, phân tích cơ bản

- **NLTK (Natural Language Toolkit):**

Dễ học, rất nhiều chức năng (tokenize, tách từ, gán nhãn từ loại, phân tích cú pháp, sentiment…). Tài liệu phong phú, phù hợp học thuật, nghiên cứu.
Hạn chế: Chậm, không tối ưu cho sản phẩm lớn, không hỗ trợ học sâu trực tiếp.

- **spaCy:**
  Tốc độ cao, dễ dùng, tích hợp mô hình học sâu, hỗ trợ nhiều ngôn ngữ, phù hợp cho ứng dụng thực tế (chatbot, extractor, NER…).
  Hạn chế: Ít linh hoạt hơn NLTK, một số tính năng nâng cao phải phát triển thêm.

- **Gensim:**
  Chuyên cho topic modeling, word embedding, semantic search (Word2Vec, Doc2Vec, FastText, LDA…), nhẹ, dễ sử dụng, phù hợp khai phá dữ liệu lớn.

Nhóm 2: Học sâu, mô hình lớn (Deep Learning, Transformers)

- **TensorFlow, PyTorch:**

  Nền tảng chung cho xây dựng mô hình học sâu (phân loại, sequence labeling, sinh văn bản, dịch máy, QA…). Linh hoạt, mạnh, phù hợp nghiên cứu và phát triển sản phẩm yêu cầu tùy biến cao.

- **Hugging Face Transformers:**
  Tập trung vào các mô hình transformer state-of-the-art (BERT, GPT, T5…). Hỗ trợ cực tốt cho text classification, QA, sinh văn bản, dịch máy, tóm tắt…, có model hub lớn, cộng đồng mạnh, dễ fine-tune và triển khai API.