**Câu 1: Mục tiêu của bài toán là gì? Xác định rõ input và output là gì?**

Trả lời:

* Mục tiêu của bài toán:  
  Xây dựng một hệ thống tóm tắt văn bản tự động sử dụng phương pháp trích xuất (extractive summarization) dựa trên thuật toán TextRank. Hệ thống sẽ giúp rút gọn nội dung văn bản dài thành một bản tóm tắt ngắn gọn, vẫn giữ được ý chính và thông tin quan trọng nhất.
* Input:  
  Một file văn bản ở định dạng XML, trong đó các câu được đóng gói trong thẻ <s>...</s>. File này chứa đoạn văn bản cần được tóm tắt.
* Output:  
  Một đoạn văn bản tóm tắt được trích xuất từ các câu trong file đầu vào, có độ dài ngắn hơn và thể hiện những câu quan trọng nhất dựa trên điểm xếp hạng PageRank.

**Câu 2: Phương pháp tiếp cận bài toán là gì? Mô tả ý tưởng chính của phương pháp tiếp cận?**

Trả lời:

* Phương pháp tiếp cận:  
  Bài toán tóm tắt văn bản được tiếp cận theo hướng trích xuất (extractive summarization), tức là chọn ra những câu quan trọng nhất từ văn bản gốc để làm bản tóm tắt thay vì viết lại nội dung (abstractive summarization).
* Ý tưởng chính:
  1. Tiền xử lý: Đọc và tách các câu trong file văn bản XML.
  2. Biểu diễn câu: Chuyển mỗi câu thành vector đặc trưng dựa trên TF-IDF, thể hiện mức độ quan trọng của các từ trong câu so với toàn bộ văn bản.
  3. Tính độ tương đồng: Dùng cosine similarity để tính độ tương đồng giữa các câu.
  4. Xây dựng đồ thị: Các câu được xem là nút trong đồ thị, các cạnh giữa câu được nối nếu độ tương đồng lớn hơn ngưỡng threshold.
  5. Xếp hạng câu: Áp dụng thuật toán PageRank trên đồ thị để tính điểm quan trọng cho mỗi câu.
  6. Chọn câu: Chọn những câu có điểm cao nhất để tạo thành bản tóm tắt ngắn gọn.

Phương pháp này tận dụng mối quan hệ giữa các câu trong văn bản để xác định độ quan trọng, từ đó trích xuất câu tiêu biểu nhất cho tóm tắt.

**Câu 3: Mô tả chi tiết các bước thực hiện của phương pháp đã tiếp cận ở trên**

Trả lời:

1. Đọc và trích xuất câu từ file XML:
   * Sử dụng biểu thức chính quy để tìm và lấy các câu nằm trong thẻ <s>...</s> từ file XML đầu vào.
   * Lưu các câu này vào danh sách để xử lý tiếp.
2. Tiền xử lý câu:
   * Tokenize từng câu, tức là tách câu thành các từ đơn (tokens), loại bỏ dấu câu và chuyển về chữ thường để chuẩn hóa dữ liệu.
3. Tính TF (Term Frequency) cho mỗi câu:
   * Đếm số lần xuất hiện của mỗi từ trong câu.
   * Tính tỷ lệ xuất hiện của từ trên tổng số từ trong câu, tạo thành vector TF cho câu đó.
4. Tính IDF (Inverse Document Frequency) cho toàn bộ câu:
   * Đếm số câu chứa từ đó.
   * Tính IDF dựa trên công thức: IDF(term) = log(N / (1 + số câu chứa term)) với N là tổng số câu.
5. Tính TF-IDF vectors:
   * Nhân TF của từng từ với IDF tương ứng để có vector đặc trưng cho mỗi câu.
6. Tính ma trận cosine similarity giữa các câu:
   * Dùng cosine similarity để đo độ giống nhau giữa từng cặp câu dựa trên TF-IDF vectors.
7. Xây dựng đồ thị câu:
   * Mỗi câu là một nút trong đồ thị.
   * Tạo cạnh giữa hai câu nếu cosine similarity của chúng lớn hơn hoặc bằng ngưỡng threshold.
8. Áp dụng thuật toán PageRank trên đồ thị:
   * Khởi tạo điểm PageRank đều cho tất cả câu.
   * Lặp cập nhật điểm dựa trên điểm của các câu liên kết, trọng số cạnh và hệ số giảm dần (damping factor).
   * Dừng khi điểm hội tụ hoặc đạt số vòng lặp tối đa.
9. Chọn các câu có điểm PageRank cao nhất:
   * Sắp xếp các câu theo điểm PageRank giảm dần.
   * Lấy ra số câu cần tóm tắt dựa trên số câu của bản tóm tắt mẫu (do giáo viên cung cấp) hoặc theo yêu cầu.
10. Tạo bản tóm tắt:
    * Ghép các câu đã chọn theo thứ tự xuất hiện trong văn bản gốc để tạo bản tóm tắt cuối cùng.

**Câu 4: Code được ≥ 5 đặc trưng biểu diễn dữ liệu / biểu diễn được văn bản đồ thị**

Trả lời:

Trong chương trình, tôi đã sử dụng các đặc trưng biểu diễn dữ liệu và mô hình đồ thị như sau:

1. Đặc trưng biểu diễn dữ liệu:
   * TF (Term Frequency): Tính tần suất xuất hiện của từng từ trong mỗi câu, thể hiện mức độ quan trọng cục bộ của từ trong câu.
   * IDF (Inverse Document Frequency): Đo độ hiếm của từ trên toàn bộ tập câu, giúp giảm trọng số từ xuất hiện nhiều lần không mang nhiều ý nghĩa.
   * TF-IDF Vector: Kết hợp TF và IDF để tạo vector đặc trưng cho mỗi câu, phản ánh sự quan trọng của từ trong câu và trong toàn văn bản.
   * Cosine Similarity: Đo độ tương đồng giữa hai câu dựa trên góc giữa hai vector TF-IDF, dùng để xây dựng các cạnh trong đồ thị.
   * Damping Factor (d) trong PageRank: Một tham số điều chỉnh ảnh hưởng của điểm truyền từ các nút lân cận trong thuật toán xếp hạng.
2. Biểu diễn văn bản dưới dạng đồ thị:
   * Các câu trong văn bản được biểu diễn như các nút (node) trong đồ thị.
   * Các cạnh (edge) nối giữa các nút nếu độ tương đồng cosine giữa hai câu vượt qua ngưỡng threshold.
   * Trọng số cạnh tương ứng với giá trị độ tương đồng giữa câu.

Việc biểu diễn văn bản theo cách này giúp áp dụng thuật toán PageRank để xếp hạng các câu, từ đó chọn ra những câu quan trọng nhất cho bản tóm tắt.

**Câu 5: Áp dụng được bất kỳ phương pháp phân lớp dữ liệu trong thư viện máy học / Xếp hạng được từ trong đồ thị theo mức độ quan trọng**

Trả lời:

* Trong chương trình, tôi sử dụng thuật toán PageRank, một thuật toán dựa trên lý thuyết đồ thị, để xếp hạng các câu trong văn bản dựa trên mức độ quan trọng của chúng. Thuật toán này hoạt động giống như phương pháp phân lớp dựa trên liên kết (link-based ranking) mà không cần phải huấn luyện dữ liệu như các mô hình máy học truyền thống.
* Thuật toán PageRank đánh giá mức độ quan trọng của từng câu (nút trong đồ thị) dựa trên các liên kết (cạnh có trọng số) từ các câu khác. Các câu được đánh giá cao hơn nếu chúng được nhiều câu quan trọng khác liên kết đến.
* Về phân lớp dữ liệu truyền thống, chương trình chưa áp dụng trực tiếp các mô hình học máy (như SVM, Random Forest, hay Logistic Regression), mà sử dụng phương pháp xếp hạng dựa trên đồ thị phù hợp với bài toán tóm tắt trích xuất.
* Nếu cần, có thể mở rộng chương trình để kết hợp mô hình phân lớp hoặc clustering nhằm cải thiện chất lượng tóm tắt.

**Câu 6: Lấy được tóm tắt văn bản**

Trả lời:

* Chương trình của tôi thực hiện tóm tắt văn bản trích xuất bằng cách chọn ra những câu quan trọng nhất từ văn bản gốc dựa trên điểm xếp hạng PageRank.
* Các bước chính để lấy tóm tắt:
  1. Tính điểm quan trọng cho từng câu thông qua thuật toán PageRank trên đồ thị câu được xây dựng từ độ tương đồng cosine.
  2. Sắp xếp các câu theo điểm PageRank giảm dần.
  3. Chọn số câu cần thiết (có thể dựa trên số câu trong bản tóm tắt mẫu hoặc theo yêu cầu người dùng).
  4. Ghép các câu đã chọn theo đúng thứ tự xuất hiện trong văn bản gốc để tạo bản tóm tắt hoàn chỉnh.
* Kết quả tóm tắt được hiển thị trực tiếp trong giao diện chương trình, giúp người dùng dễ dàng xem và so sánh với bản tóm tắt chuẩn.

**Câu 7: Nhận xét về kết quả đạt được: độ chính xác, ưu nhược điểm của phương pháp đang áp dụng**

Trả lời:

* Độ chính xác:  
  Phương pháp tóm tắt trích xuất sử dụng PageRank dựa trên đồ thị câu giúp chọn ra các câu có tính đại diện cao dựa trên mối quan hệ tương đồng giữa các câu. Kết quả tóm tắt thường giữ được nội dung chính và thông tin quan trọng của văn bản gốc, có độ chính xác tốt trong việc phản ánh ý chính.
* Ưu điểm:
  + Dễ triển khai, không cần dữ liệu huấn luyện phức tạp.
  + Tính toán dựa trên mối quan hệ nội bộ giữa các câu giúp tóm tắt có tính nhất quán và liên kết nội dung.
  + Có thể áp dụng với các loại văn bản đa dạng mà không cần điều chỉnh nhiều.
* Nhược điểm:
  + Chỉ chọn câu nguyên bản nên bản tóm tắt có thể dài hơn và không cô đọng như phương pháp tóm tắt trừu tượng (abstractive).
  + Phụ thuộc vào ngưỡng similarity (threshold) và số câu chọn, nếu chọn không hợp lý có thể bỏ sót thông tin quan trọng hoặc dư thừa câu không cần thiết.
  + Không xử lý tốt ngữ cảnh sâu hay các phép diễn giải phức tạp trong văn bản.
* Tổng kết:  
  Phương pháp này phù hợp để nhanh chóng tạo tóm tắt có nội dung chính xác, đặc biệt khi không có nhiều dữ liệu huấn luyện hoặc yêu cầu tài nguyên thấp. Tuy nhiên, nếu cần tóm tắt ngắn gọn và tự nhiên hơn, có thể cân nhắc áp dụng các kỹ thuật học sâu hoặc tóm tắt trừu tượng.

**Câu 8: Cải tiến phương pháp đang áp dụng, chẳng hạn bổ sung thêm đặc trưng biểu diễn dữ liệu hoặc thêm trọng số vào đồ thị**

**Trả lời:**

1. Bổ sung thêm đặc trưng biểu diễn dữ liệu:
   * Sử dụng các embedding từ mô hình ngôn ngữ hiện đại như Word2Vec, GloVe hoặc BERT để biểu diễn câu thay vì chỉ dùng TF-IDF. Embedding giúp nắm bắt ngữ cảnh và ý nghĩa sâu hơn của từ và câu, từ đó cải thiện độ chính xác trong việc tính toán độ tương đồng.
   * Kết hợp thêm các đặc trưng ngữ pháp, chẳng hạn như các thông tin về cấu trúc câu, vị trí câu trong đoạn văn, hoặc độ dài câu để đánh giá tầm quan trọng câu tốt hơn.
2. Điều chỉnh trọng số trong đồ thị:
   * Thay vì sử dụng chỉ cosine similarity, có thể kết hợp các trọng số khác như trọng số dựa trên tần suất từ khóa chính hoặc trọng số vị trí câu để làm tăng ảnh hưởng của các câu quan trọng hơn trong văn bản.
   * Áp dụng ngưỡng threshold động thay vì cố định, dựa trên phân bố similarity để tối ưu cấu trúc đồ thị phù hợp với từng văn bản.
3. Áp dụng các thuật toán xếp hạng khác hoặc kết hợp:
   * Kết hợp PageRank với các thuật toán học máy hoặc học sâu để cải thiện việc đánh giá mức độ quan trọng của câu.
   * Sử dụng kỹ thuật clustering để nhóm các câu tương đồng, giúp tránh lặp lại thông tin trong bản tóm tắt.
4. Tăng cường xử lý ngôn ngữ tự nhiên:
   * Thêm bước tiền xử lý nâng cao như loại bỏ từ dừng (stop words), chuẩn hóa từ (lemmatization), hoặc phân tích cú pháp để nâng cao chất lượng biểu diễn câu.