# XỬ LÝ CHUNKING CÁC DẠNG DỮ LIỆU: TỔNG QUAN VÀ PHƯƠNG PHÁP

## 1. Giới thiệu

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), chunking là một bước quan trọng giúp cải thiện khả năng truy xuất và xử lý thông tin từ các tài liệu lớn. Chunking, hay chia nhỏ tài liệu thành các đoạn nhỏ hơn, giúp các mô hình ngôn ngữ như Retrieval-Augmented Generation (RAG) có thể tiếp cận và xử lý dữ liệu một cách hiệu quả hơn. Các phương pháp chunking được áp dụng tùy theo loại hình tài liệu và mục tiêu xử lý, từ đó giúp hệ thống dễ dàng truy xuất thông tin và duy trì ngữ cảnh trong quá trình tạo phản hồi.

## 2. Tổng quan về Chunking

### 2.1. Định nghĩa Chunking

* Trong hệ thống Retrieval-Augmented Generation (RAG), chunking là quá trình chia nhỏ tài liệu thành các đoạn nhỏ hơn, gọi là "chunks", để cải thiện hiệu quả truy xuất thông tin và khả năng xử lý của các mô hình ngôn ngữ lớn (LLMs). Việc này giúp LLMs dễ dàng truy cập và xử lý thông tin liên quan từ các nguồn dữ liệu bên ngoài.

### 2.2. Một số bước để sử dụng chunking hiệu quả hơn

* Phân đoạn với chồng lấn (overlapping): Phân đoạn với một đoạn dữ liệu bị chồng giữa các phân đoạn để duy trì tính liên tục và tránh mất ngữ cảnh.
* Tích hợp tóm tắt hoặc siêu dữ liệu (metadata) vào các phân đoạn để tránh mất ngữ cảnh và truy vấn hiệu quả hơn.

### 2.3. Các dạng dữ liệu

* Dữ liệu phi cấu trúc: Loại dữ liệu này không có cấu trúc rõ ràng, bao gồm văn bản, hình ảnh, video, email và các tệp đa phương tiện khác.
* Dữ liệu bán cấu trúc: Dữ liệu này có một số cấu trúc nhưng không hoàn toàn theo định dạng cố định, như tệp XML hoặc JSON, file PDF,...
* Dữ liệu có cấu trúc: Đây là loại dữ liệu được tổ chức trong các bảng hoặc cơ sở dữ liệu với định dạng cố định, dễ dàng truy vấn và phân tích, như cơ sở dữ liệu Excel, Mysql,...
* Dữ liệu hỗn hợp: chứa nhiều loại dữ liệu đề cập ở trên.

## 3. Phương pháp Chunking với các loại tài liệu

### 3.1. Dữ liệu không có cấu trúc dạng văn bản (email, facebook posts,...)

* Phân đoạn với độ dài cố định: Phương pháp phân đoạn cố định, thường được gọi là phân đoạn ngây thơ (naive chunking), là cách đơn giản nhất và được sử dụng phổ biến nhất. Phương pháp này chia văn bản thành các đoạn có kích thước đồng đều dựa trên một số lượng ký tự xác định trước, với tùy chọn chồng lấn(overlapping) giữa các đoạn để bảo toàn ngữ cảnh.
* Phân đoạn đệ quy:
  + Chiến lược phân tách đệ quy khác với phương pháp phân tách cố định ở chỗ nó xem xét cấu trúc tự nhiên của văn bản. Thay vì cắt theo độ dài cố định, phương pháp này sử dụng một chuỗi các dấu phân tách, chẳng hạn như ngắt đoạn (\n\n), ngắt dòng (\n), khoảng trắng (" ") và chuỗi rỗng ("").
  + Phương pháp này hoạt động theo hệ thống phân cấp: đầu tiên, nó cố gắng tách văn bản bằng dấu phân tách lớn nhất (ngắt đoạn), và nếu đoạn kết quả vẫn quá dài, nó tiếp tục với dấu phân tách tiếp theo trong danh sách. Quá trình này tiếp diễn cho đến khi tất cả các đoạn đáp ứng giới hạn kích thước.
  + Ngoài ra, phương pháp này còn cố gắng hợp nhất các đoạn nhỏ hơn khi có thể để tránh phân mảnh không cần thiết, nhằm tạo ra các đoạn có kích thước gần với giới hạn nhất có thể.
* Phân đoạn theo ngữ nghĩa: Khác với các phương pháp dựa trên cấu trúc, phân đoạn theo ngữ nghĩa tập trung vào ý nghĩa vốn có của câu, sử dụng nhúng câu (sentence embeddings) để so sánh mối quan hệ ngữ nghĩa và nhóm các câu thành các đoạn có ý nghĩa chặt chẽ. Phương pháp này sử dụng các mô hình machine learning và nlp để thực hiện nhiệm vụ này.
* Phân đoạn bằng đại lý: Phân đoạn bằng đại lý là một chiến lược mạnh mẽ trong số các chiến lược này. Trong chiến lược này, sử dụng LLM như GPT để hoạt động như các tác nhân trong quy trình phân đoạn. Thay vì xác định thủ công cách phân đoạn nội dung, LLM chủ động sắp xếp hoặc phân chia thông tin theo đầu vào hiểu biết của nó. LLM xác định phương pháp tốt nhất để chia nội dung thành các phần có thể quản lý được, chịu ảnh hưởng của ngữ cảnh nhiệm vụ.
* Phân đoạn dữ liệu con: Tóm tắt toàn bộ tài liệu hoặc các phần lớn và đính kèm các bản tóm tắt này vào từng phần riêng lẻ dưới dạng siêu dữ liệu.
* Khác: phân đoạn theo câu, đoạn văn,...

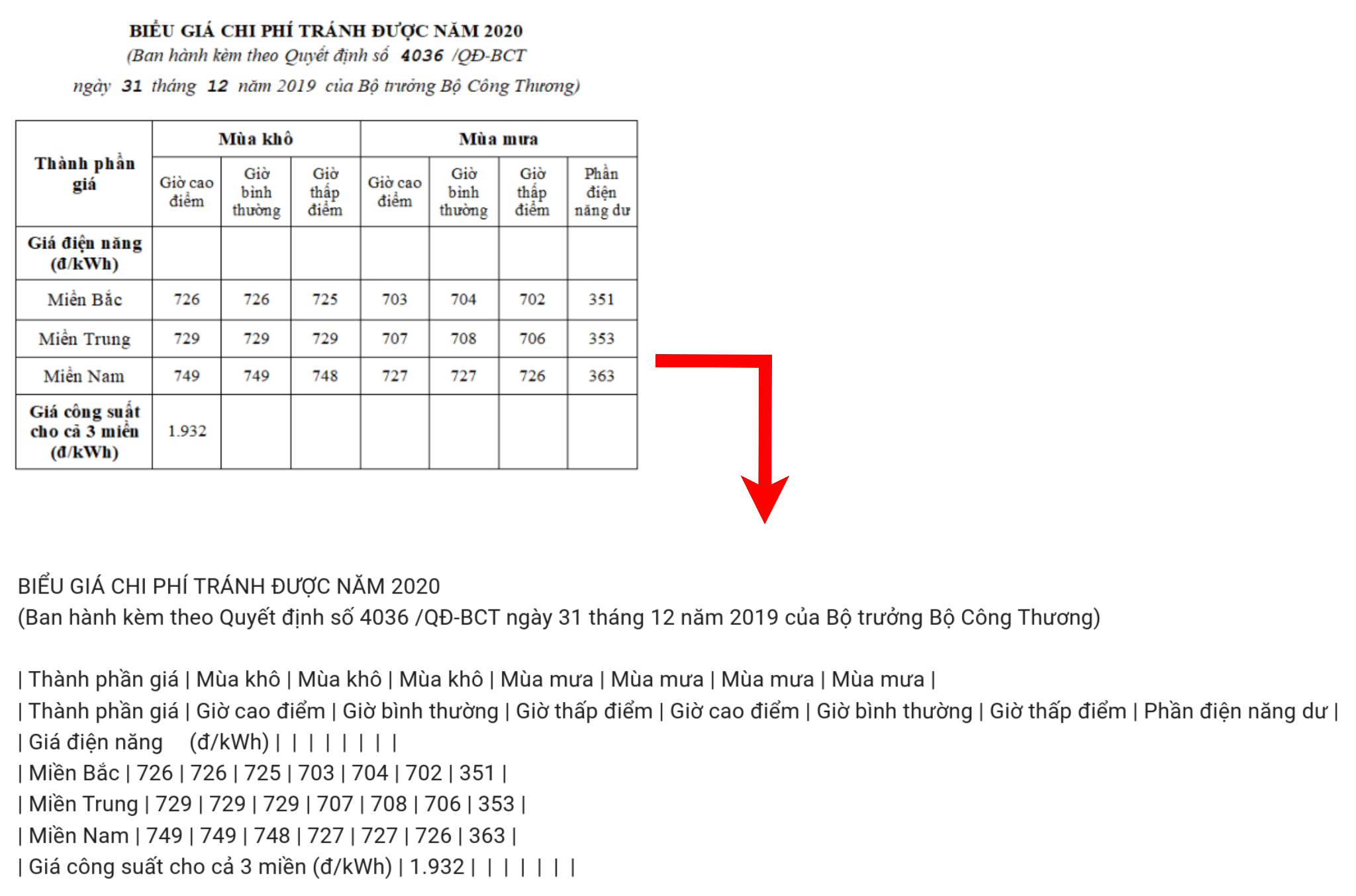
### 3.2. Dữ liệu bán cấu trúc

* Các phương pháp cho dữ liệu không có cấu trúc dạng văn bản, đặc biệt là phân đoạn đệ quy.
* Sử dụng Structural Chunkers như MarkdownHeaderTextSplitter, HTMLHeaderTextSplitter, LayoutPDFReader để phân đoạn dựa trên cấu trúc.

### 3.3 Dữ liệu dạng bảng, danh sách, số liệu thống kê.

Các bảng cần được chuẩn hoá thành Markdown hoặc XML để dễ dàng xử lý và truy vấn.

Ví dụ



Nếu bảng chứa nội dung quá dài, cần chia bảng ra thành nhiều đoạn nhỏ (chunk). Mỗi chunk nên chứa một phần nội dung có liên quan với nhau của bảng, và đầu mỗi chunk nên được gắn lại tiêu đề và tên cột của bảng để đảm bảo tính liên tục và dễ hiểu.



### 3.4: Tài liệu hình ảnh

Option 1:

* Sử dụng multimodal embedding (chẳng hạn như CLIP) để embed ảnh và text.

Option 2:

* Sử dụng Multimodal LLM (chẳng hạn như GPT-4, LLaVA, or FUYU-8b) để tạo text tóm tắt từ ảnh.
* Embed và retrieve các bản tóm tắt ảnh có reference tới raw image. Có thể sử dụng multi-vector retriever với Vector DB chẳng hạn như Chroma để lưu trữ raw text và images cùng với bản tóm tắt của chúng để retrieval.
* Đưa raw images và các đoạn text liên quan tới một mô hình Multimodal LLM để tổng hợp câu trả lời.

Link tham khảo: <https://viblo.asia/p/chatgpt-series-6-multimodal-rag-va-nhung-phuong-phap-duoc-nghien-cuu-de-cai-thien-chat-luong-he-thong-rag-aNj4vDKqL6r>

<https://masteringllm.medium.com/11-chunking-strategies-for-rag-simplified-visualized-df0dbec8e373>

<https://kdb.ai/learning-hub/articles/in-depth-review-of-chunking-methods/>

Nhận input là tài liệu dạng văn bản hoặc thư mục

Dùng các bước tiền xử lý, chunking, embedding, lưu vào vector store

Output: Vector database chứa các chunks + metadata

## Flow xây dựng lại pipeline chunking với Agent

### Target:

Dùng Agent (LLM) để phân loại loại tài liệu đầu vào rồi chunk theo loại.

Các bước cụ thể cần define

### Step 1: Nhận diện loại tài liệu

Dùng Agent để phân loại:

* Văn bản (PDF, DOCX, TXT)
* Bảng (Excel, HTML table)
* Bán cấu trúc (JSON, XML)
* Hình ảnh (image OCR, CLIP)

🔧 Công cụ:

* Sử dụng LLM để xem xét nội dung + metadata để nhận biết loại
* Hoặc rule-based: file extension + cấu trúc nội dung

### Step 2: Xác định phương pháp chunking theo loại tài liệu

Step 3: Chunking

Step 4: Output