REPORT: ASSIGNMENT END-TO-END - NLP-SYSTEM BUILDING

Nhóm 17

- Bùi Duy Hải - 22022575
- Lê Trung Hiếu - 22022576
- Nguyễn Lâm Tùng Bách - 22022640

Link Github: https://github.com/hieu7404/End-to-End–NLP-System

Tóm tắt

Báo cáo này trình bày quá trình phát triển một hệ thống Retrieval-Augmented Generation (RAG) cơ bản để trả lời câu hỏi (QA) dựa trên dữ liệu thu thập từ các nguồn công khai. Hệ thống bao gồm các bước crawl dữ liệu, xử lý văn bản, nhúng và tạo câu trả lời bằng mô hình Llama-3.2-1B-Instruct. Kết quả ban đầu đạt độ chính xác 32.1%, nhưng chưa hoàn thiện đầy đủ theo yêu cầu bài tập do giới hạn thời gian và tài nguyên.

1 Tạo dữ liệu

1.1 Biên soạn nguồn kiến thức

- Nguồn kiến thức được thu thập từ các URL công khai liệt kê trong file `data/data_source.csv`, sử dụng script `crawl_data.py`.
- Dữ liệu bao gồm văn bản từ các trang web của VNU và UET.

1.2 Trích xuất dữ liệu thô

- **Công cụ**:
  - Sử dụng `requests` và `BeautifulSoup` trong `crawl_data.py` để crawl dữ liệu từ các URL, với User-Agent giả lập để tránh bị chặn.
  - File `processing_data.py` làm sạch dữ liệu.
  - File `data_processor.py` chia đoạn văn bản và nhúng.
- **Quy trình**:
  - `crawl_data.py` tải nội dung từ các URL, lưu vào `data/all_data.txt`.
  - `processing_data.py` xử lý `all_data.txt` để tạo `data/data_clean.txt` với định dạng gọn gàng.
  - `data_processor.py` chia đoạn văn bản từ `data/data.txt` (được lọc từ `data_clean.txt`) thành các đoạn (chunks) với kích thước tối đa 256 token, nhúng bằng mô hình `bkai-foundation-models/vietnamese-bi-encoder` và lưu chỉ mục FAISS trong `data/faiss_index.bin` cùng danh sách đoạn trong `data/chunks.pkl`.

1.3 Chú thích dữ liệu

- Tập test được định nghĩa trong `data/questions.json`, chứa câu hỏi và tham chiếu (reference answers).
- Việc chú thích chỉ được thực hiện thủ công một phần, chưa đa dạng.
- Số lượng dữ liệu giới hạn ở câu hỏi trong `data/questions.json` và văn bản crawl từ link trong `data/data_source.csv`.

2 Chi tiết mô hình

2.1 Phương pháp

- Sử dụng `meta-llama/Llama-3.2-1B-Instruct` kết hợp với RAG.
- **Embedder**: `embedding.py` với mô hình `bkai-foundation-models/vietnamese-bi-encoder`.
- **Retriever**: FAISS trong `rag_system.py` để lấy top-k tài liệu liên quan.
- **Reader**: `llm_generator.py` tạo câu trả lời dựa trên prompt tăng cường từ RAG.

2.2 Lý do lựa chọn

- Mô hình Llama-3.2-1B được chọn vì khả năng truy cập từ Hugging Face và kích thước phù hợp với máy của bọn em có.

3 Kết quả

3.1 Số liệu từ dữ liệu test

- Hệ thống được chạy trên tập test trong `data/questions.json`.
- Kết quả từ `evaluate.py`:
  - **Custom Accuracy**: 32.1% - dựa trên khớp hoàn toàn hoặc một phần giữa tham chiếu và dự đoán trong `system_output.txt`.
- **Ví dụ**:
  - Câu hỏi: "Thời gian đào tạo chuẩn của ngành Kỹ thuật robot là bao lâu?"
  - Tham chiếu: "4,5 năm"
  - Dự đoán: "4,5 năm" → Đúng
  - Câu hỏi: "Trường Đại học Công nghệ có bao nhiêu giáo sư?"
  - Tham chiếu: "7"
  - Dự đoán: "7 giáo sư" → Đúng (khớp một phần)

4 Kết luận

Hệ thống RAG cơ bản đã được triển khai với khả năng crawl dữ liệu, xử lý văn bản và tạo câu trả lời. Tuy nhiên, hiệu suất còn hạn chế. Để cải thiện trong tương lai, chúng em dự kiến sẽ mở rộng nguồn dữ liệu từ nhiều nền tảng đa dạng hơn, tích hợp các tài nguyên tốt hơn để có thể sử dụng các mô hình lớn hơn.