Mục lục

1. Giới thiệu bài toán

Trong bối cảnh bùng nổ thông tin, các nền tảng tin tức trực tuyến như Google News hay Microsoft News cung cấp hàng trăm nghìn bài viết mỗi ngày. Điều này khiến người dùng gặp khó khăn trong việc tìm kiếm nội dung phù hợp với sở thích cá nhân, dẫn đến tình trạng quá tải thông tin.

Bài toán đặt ra: Xây dựng một hệ thống *gợi ý tin tức cá nhân hóa* (Personalized News Recommendation). Hệ thống này có nhiệm vụ dự đoán những bài báo mà một người dùng có khả năng quan tâm nhất, dựa trên lịch sử đọc tin và các hành vi tương tác trước đó của họ, nhằm nâng cao trải nghiệm người dùng.

So với các lĩnh vực gợi ý khác (phim ảnh, sản phẩm), bài toán gợi ý tin tức có những thách thức đặc thù:

- **Tính thời gian thực:** Tin tức thay đổi liên tục và nhanh chóng trở nên lỗi thời.
- **Vấn đề khởi đầu lạnh (Cold-start):** Các bài báo mới xuất hiện hàng ngày, và hệ thống cần có khả năng gợi ý chúng mà không cần nhiều dữ liệu tương tác ban đầu.
- **Sở thích đa dạng của người dùng:** Một người dùng có thể quan tâm đến nhiều lĩnh vực cùng lúc (ví dụ: thể thao, công nghệ, và chính trị).

Để giải quyết bài toán này trong khuôn khổ dự án, nhóm đề xuất sử dụng bộ dữ liệu chuẩn quốc tế **MIND (Microsoft News Dataset)** và xây dựng một mô hình gợi ý dựa trên học sâu (Deep Learning), có khả năng hiểu sâu ngữ nghĩa của bài viết và mô hình hóa hành vi phức tạp của người dùng.

2. Bộ dữ liệu và các chỉ số đánh giá

2.1 Bộ dữ liệu MIND

MIND (Microsoft News Dataset) là một bộ dữ liệu quy mô lớn, được xây dựng từ nhật ký hành vi người dùng trên nền tảng Microsoft News. Đây là nguồn tài nguyên tiêu chuẩn để nghiên cứu và phát triển các hệ thống gợi ý tin tức.

- **Nguồn gốc:** Dữ liệu được thu thập trong 6 tuần (từ 12/10 đến 22/11/2019).
- **Quy mô:**
  - Gần **1 triệu người dùng** ẩn danh.
  - Hơn **161,000 bài viết** bằng tiếng Anh.
  - Hơn **24 triệu lượt click** được ghi nhận.
- **Thông tin bài viết:** Mỗi bài viết bao gồm tiêu đề, tóm tắt, danh mục (category), danh mục con (subcategory), và các thực thể (entities) được liên kết với cơ sở tri thức WikiData.
- **Phân chia dữ liệu:** Bộ dữ liệu được chia sẵn thành ba tập: huấn luyện (training), kiểm thử (validation), và đánh giá (test).

Với quy mô và độ chi tiết cao, MIND cho phép xây dựng và đánh giá các mô hình học sâu phức tạp.

2.2 Chỉ số đánh giá

Để đánh giá hiệu quả của mô hình đề xuất, nhóm sẽ sử dụng các chỉ số phổ biến trong lĩnh vực hệ gợi ý:

- **AUC (Area Under ROC Curve):** Đo lường khả năng của mô hình trong việc phân biệt giữa một bài báo được người dùng click (tương tác dương) và một bài báo không được click (tương tác âm).
- **MRR (Mean Reciprocal Rank):** Đánh giá độ chính xác của vị trí xếp hạng. Chỉ số này tính giá trị nghịch đảo của thứ hạng của bài báo đầu tiên được click trong danh sách gợi ý.
- **nDCG@K (Normalized Discounted Cumulative Gain at K):** Đánh giá chất lượng của top-K gợi ý (ví dụ K=5, K=10). Chỉ số này xem xét cả mức độ liên quan và vị trí của các bài báo được gợi ý, trong đó các gợi ý đúng ở vị trí cao hơn sẽ được chấm điểm cao hơn.

3. Các mô hình hiện có

Bảng dưới đây tổng hợp một số mô hình tiên tiến đã được công bố trên bộ dữ liệu MIND, đây sẽ là cơ sở để nhóm tham khảo và phát triển phương pháp của riêng mình.

Model	**AUC**	**MRR**	**nDCG@5**	**nDCG@10**	**Ưu điểm nổi bật**
NAML	0.6686	0.3249	0.3524	0.4091	Học đa góc nhìn với attention, kết hợp tiêu đề, danh mục và embedding.
NRMS	0.6776	0.3305	0.3594	0.4163	Multi-head self-attention để mã hóa tin tức và lịch sử đọc.
NPA	0.6669	0.3224	0.3498	0.4068	Personalized attention, sử dụng negative sampling.
MINER	**0.7275**	0.3724	0.4102	0.4661	Mô hình đa sở thích, category-aware attention, regularization.
Fastformer	0.7268	**0.3745**	**0.4151**	**0.4684**	Attention hiệu quả O(n), phù hợp quy mô lớn.

4. Phương pháp đề xuất

Phần này trình bày kế hoạch xử lý dữ liệu và kiến trúc mô hình mà nhóm dự kiến sẽ phát triển để giải quyết bài toán gợi ý tin tức trên bộ dữ liệu MIND. Mục tiêu là xây dựng một mô hình có khả năng:

1. **Hiểu nội dung:** Biểu diễn được ngữ nghĩa, chủ đề và các thực thể quan trọng trong một bài viết.
2. **Hiểu người dùng:** Mô hình hóa được các sở thích đa dạng và thay đổi theo thời gian của người dùng dựa trên lịch sử đọc.
3. **Dự đoán chính xác:** Ước tính xác suất một người dùng sẽ click vào một bài viết ứng viên.

4.1 Quy trình xử lý dữ liệu

Để chuẩn bị dữ liệu cho việc huấn luyện mô hình học sâu, nhóm sẽ thực hiện một quy trình tiền xử lý gồm các bước sau:

4.1.1 Mã hóa văn bản (Text Encoding)

Mỗi bài viết sẽ được biểu diễn dưới dạng vector số học (embedding) để mô hình có thể xử lý.

- **Thành phần văn bản:** Tập trung vào **tiêu đề (title)** và **tóm tắt (abstract)**. Giới hạn độ dài (ví dụ: 30 token cho tiêu đề, 100 token cho tóm tắt) để cân bằng giữa thông tin và hiệu năng.
- **Word Embedding:** Nhóm sẽ xem xét hai phương án:
  - **Sử dụng mô hình ngôn ngữ lớn (PLM):** Dùng các mô hình như **BERT-base** để tạo ra các embedding ngữ cảnh hóa, có chất lượng cao (vector 768 chiều).
  - **Sử dụng embedding tĩnh:** Dùng các mô hình gọn nhẹ hơn như **GloVe** (300 chiều) hoặc **Word2Vec** trong trường hợp cần tối ưu tốc độ và tài nguyên.
- **Category Embedding:** Danh mục và danh mục con của mỗi bài viết sẽ được mã hóa bằng một lớp embedding riêng biệt để bổ sung thông tin về chủ đề.

4.1.2 Xử lý thực thể (Entity Processing)

Các thực thể (ví dụ: "Apple Inc.", "iPhone") chứa thông tin quan trọng giúp liên kết các bài viết.

- **Trích xuất thực thể:** Sử dụng danh sách thực thể đã được liên kết với WikiData có sẵn trong MIND.
- **Entity Embedding:** Biểu diễn mỗi thực thể bằng một vector. Nhóm sẽ thử nghiệm:
  - Sử dụng các embedding được huấn luyện sẵn từ đồ thị tri thức như **Wikipedia2Vec**.
  - Huấn luyện embedding từ đầu bằng các thuật toán như **TransE** trên đồ thị tri thức của WikiData.
- **Tích hợp thông tin:** Kết hợp embedding của từ và embedding của thực thể bằng cơ chế attention để mô hình có thể tập trung vào những thông tin quan trọng nhất.

4.1.3 Biểu diễn người dùng (User Representation)

Sở thích của người dùng sẽ được suy ra từ lịch sử các bài báo họ đã đọc.

- **Lịch sử đọc:** Lấy tối đa 50 bài báo gần nhất mà người dùng đã tương tác.
- **Mô hình hóa tuần tự:** Dự kiến sử dụng các mạng như **GRU** hoặc cơ chế **self-attention** để nắm bắt mối quan hệ tuần tự và sự phụ thuộc giữa các bài báo đã đọc.
- **Mô hình hóa đa sở thích:** Áp dụng các kỹ thuật từ mô hình MINER để biểu diễn người dùng bằng nhiều vector sở thích, thay vì chỉ một.

4.1.4 Chiến lược lấy mẫu (Sampling Strategy)

Để huấn luyện mô hình phân loại (click/không click), cần tạo ra các mẫu âm (negative samples).

- **Negative Sampling:** Với mỗi lượt click (mẫu dương), nhóm sẽ chọn ngẫu nhiên 4 bài báo khác mà người dùng không click trong cùng một phiên làm mẫu âm.
- **Hard Negative Mining:** Ưu tiên chọn các mẫu âm "khó"(ví dụ: bài báo cùng danh mục nhưng người dùng không click) để giúp mô hình học cách phân biệt tốt hơn.

4.2 Mô hình đề xuất của nhóm

Dựa trên việc phân tích các mô hình hiện có, nhóm nhận thấy rằng một mô hình hiệu quả cần kết hợp được các thế mạnh sau:

- Khả năng mô hình hóa **nhiều sở thích** của người dùng (như **MINER**).
- Khả năng khai thác **tri thức từ các thực thể** (như **DKN**).
- **Hiệu năng tính toán** cao để có thể áp dụng trong thực tế (như **Fastformer**).

Do đó, nhóm đề xuất xây dựng một mô hình **Hybrid (lai ghép)** với kiến trúc dự kiến như sau:

4.2.1 Kiến trúc tổng quan

- **Module mã hóa tin tức (News Encoder):**
  - **Nền tảng:** Sử dụng một kiến trúc hiệu quả như **Fastformer** để mã hóa tiêu đề và tóm tắt.
  - **Tích hợp tri thức:** Kết hợp thêm entity embedding theo phương pháp của **DKN**, sử dụng một cơ chế attention để tổng hợp thông tin từ văn bản và thực thể, tạo ra vector biểu diễn cuối cùng cho bài viết.
- **Module mã hóa người dùng (User Encoder):**
  - **Nền tảng:** Dựa trên ý tưởng của **MINER**, sử dụng **Capsule Network** hoặc **Multi-head Attention** trên lịch sử đọc tin để tạo ra nhiều vector sở thích $(I_1, I_2, ..., I_K)$ cho mỗi người dùng.
  - **Attention theo ngữ cảnh:** Khi có một bài viết ứng viên, mô hình sẽ dùng một cơ chế attention để xác định vector sở thích nào của người dùng là phù hợp nhất, từ đó tạo ra vector biểu diễn người dùng trong ngữ cảnh đó (u_{user}).
- **Module dự đoán (Prediction Layer):**
  - Tính toán điểm tương đồng giữa vector người dùng và vector bài viết (ví dụ: tích vô hướng - dot product).
  - Đưa điểm số qua hàm **sigmoid** để ra xác suất click cuối cùng.

$$\hat{y} = \sigma(u_{user}^{\top} v_{news})$$

4.2.2 Chiến lược huấn luyện

- **Hàm mất mát (Loss Function):** Sử dụng hàm **Binary Cross-Entropy** để so sánh xác suất dự đoán và nhãn thực tế (click=1, không click=0).
- **Thuật toán tối ưu (Optimizer):** Dự kiến sử dụng **AdamW**, một biến thể của Adam có hiệu quả tốt với các mô hình Transformer, với tốc độ học (learning rate) ban đầu là $\times 10^{-4}$. .
- **Điều chuẩn (Regularization):** Áp dụng **Dropout** (tỷ lệ 0.2) và **Weight Decay** để tránh overfitting.
- **Kích thước lô (Batch Size):** Lựa chọn trong khoảng 64-128, tùy thuộc vào bộ nhớ của GPU.

4.2.3 Ưu điểm của mô hình đề xuất

- **Toàn diện:** Kết hợp cả ba khía cạnh quan trọng: ngữ nghĩa văn bản, tri thức từ thực thể, và sở thích đa dạng của người dùng.
- **Hiệu quả:** Việc sử dụng nền tảng Fastformer giúp giảm độ phức tạp tính toán, có tiềm năng triển khai trong các hệ thống quy mô lớn.

- **Linh hoạt:** Kiến trúc module cho phép dễ dàng thử nghiệm và thay thế các thành phần (ví dụ: thay News Encoder bằng BERT).
- **Khả thi:** Toàn bộ mô hình có thể được xây dựng bằng các thư viện phổ biến như **PyTorch** và **Hugging Face Transformers**.

4.3 Chiến lược tối ưu hóa và triển khai

Bên cạnh việc xây dựng mô hình cốt lõi, nhóm cũng đề ra một số chiến lược để tối ưu hóa hiệu năng.

4.3.1 Tối ưu trong huấn luyện

- **Curriculum Learning:** Bắt đầu huấn luyện với các mẫu "dễ"(ví dụ: người dùng có lịch sử đọc dài) trước khi chuyển sang các mẫu khó hơn (người dùng mới).
- **Multi-task Learning (Tùy chọn):** Có thể thêm các nhiệm vụ phụ như dự đoán danh mục của bài viết để giúp mô hình học được các biểu diễn tổng quát hơn.

4.3.2 Tối ưu khi suy luận (Inference)

Đây là các kỹ thuật sẽ được xem xét nếu triển khai mô hình trong thực tế:

- **Lượng tử hóa mô hình (Model Quantization):** Chuyển đổi trọng số của mô hình từ kiểu dữ liệu FP32 sang FP16 hoặc INT8 để tăng tốc độ xử lý.
- **Caching:** Lưu lại các vector embedding của bài viết đã được tính toán trước để tái sử dụng, chỉ cần tính toán cho các bài viết mới.
- **Knowledge Distillation:** Huấn luyện một mô hình "học trò"(student model) nhỏ gọn hơn để bắt chước kết quả của mô hình lớn đã huấn luyện, nhằm giảm độ trễ khi phục vụ.

4.4 Tóm tắt quy trình tổng thể

Quy trình thực hiện dự án được tóm tắt như sau:

1. **Bước 1:** Tiền xử lý dữ liệu MIND, bao gồm mã hóa văn bản, thực thể và xây dựng lịch sử người dùng.
2. **Bước 2:** Cài đặt và xây dựng các thành phần của mô hình Hybrid đã đề xuất (News Encoder, User Encoder).
3. **Bước 3:** Huấn luyện mô hình trên tập training của MIND, tinh chỉnh các siêu tham số trên tập validation.
4. **Bước 4:** Đánh giá hiệu năng cuối cùng của mô hình trên tập test bằng các chỉ số AUC, MRR, và nDCG.

5. **Bước 5 (Đề xuất cho triển khai):** Xây dựng một quy trình hai giai đoạn: dùng một mô hình gọn nhẹ để lọc ra top-100 bài viết tiềm năng (recall), sau đó dùng mô hình Hybrid phức tạp hơn để xếp hạng lại (re-ranking) và đưa ra gợi ý cuối cùng.

5. Kết luận và hướng phát triển

Bản đề xuất này đã trình bày một kế hoạch chi tiết để xây dựng hệ thống gợi ý tin tức cá nhân hóa, một bài toán có ý nghĩa thực tiễn và nhiều thách thức thú vị.

Trong phạm vi đồ án, nhóm đặt mục tiêu:

- Phân tích sâu bài toán và các đặc thù của bộ dữ liệu MIND.
- Nghiên cứu và so sánh các phương pháp hiện có để rút ra các ý tưởng cốt lõi.
- Đề xuất và hiện thực hóa một mô hình lai ghép có tính khả thi cao, kết hợp các ưu điểm của những kiến trúc hàng đầu hiện nay.