**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN MÔN HỌC 2**

**Intent Detection and Slot Filling**

**Phạm Tiến Duy**

Duy.PT215335@sis.hust.edu.vn

**Ngành Khoa học máy tính**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | PGS. TS. Lê Thanh Hương  Chữ ký của GVHD |
|  |  |
| **Viện:** | Công nghệ thông tin và truyền thông |
| **HÀ NỘI, 7/2024** | |

**ĐỀ TÀI MÔN HỌC**

**MỤC LỤC**

[CHƯƠNG 1. MÔ TẢ ĐỀ TÀI THỰC HIỆN 3](#_Toc171710816)

[1.1 Giới thiệu chung đề tài 3](#_Toc171710817)

[1.2 Mô tả bộ dữ liệu 4](#_Toc171710818)

[1.2.1 Bộ ATIS 4](#_Toc171710819)

[1.2.2 Bộ SNIPS 4](#_Toc171710820)

[1.2.3 Bộ mixATIS 5](#_Toc171710821)

[1.2.4 Bộ mixSNIPS 5](#_Toc171710822)

[CHƯƠNG 2. CÁC PHƯƠNG PHÁP GIẢI QUYẾT 6](#_Toc171710823)

[2.1 Sử dụng bert cho bộ dữ liệu đơn ý định (ATIS, SNIPS) 6](#_Toc171710824)

[2.2 Sử dụng mô hình MISCA cho phân loại bộ dữ liệu đa ý định (mixATIS, mixSNIPS) 7](#_Toc171710825)

[2.3 Các phương pháp cải thiện accuracy mô hình 9](#_Toc171710826)

[2.3.1 Pre train Bert 9](#_Toc171710827)

[2.3.2 Tăng cường dữ liệu 9](#_Toc171710828)

[CHƯƠNG 3. CÁC THÍ NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC 12](#_Toc171710829)

[3.1 Sử dụng Bert để phân loại 12](#_Toc171710830)

[3.2 Sử dụng model MISCA để phân loại 13](#_Toc171710831)

[3.2.1 Dùng model MISCA với LSTM 13](#_Toc171710832)

[3.2.2 Dùng model MISCA với Bert 13](#_Toc171710833)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc171710834)

[4.1 Kết luận 16](#_Toc171710835)

[4.2 Hướng phát triển của đồ án trong tương lai 16](#_Toc171710836)

[TÀI LIỆU THAM KHẢO 17](#_Toc171710837)

# MÔ TẢ ĐỀ TÀI THỰC HIỆN

## Giới thiệu chung đề tài

**Intent Detection and Slot Filling** là hai thành phần quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), đặc biệt trong các hệ thống hội thoại như trợ lý ảo và chatbot.

**Intent Detection** là quá trình xác định mục đích hoặc ý định của người dùng khi họ tương tác với hệ thống. Ví dụ, khi người dùng gửi một câu hỏi hoặc yêu cầu, hệ thống sẽ phân tích câu đó để hiểu xem người dùng muốn làm gì. Ý định có thể là đặt một câu hỏi, yêu cầu thông tin, thực hiện một hành động cụ thể, v.v.

Các ví dụ về ý định có thể bao gồm:

* Đặt vé máy bay
* Tìm nhà hàng gần đây
* Kiểm tra thời tiết
* Đặt lịch hẹn

**Slot Filling** là quá trình trích xuất các thông tin cụ thể từ câu của người dùng, gọi là các "slot". Đây là các phần dữ liệu mà hệ thống cần để thực hiện hành động hoặc trả lời câu hỏi của người dùng. Các slot này thường là các giá trị như thời gian, địa điểm, tên, số lượng, v.v.

Ví dụ, trong câu "Tôi muốn đặt vé máy bay từ Hà Nội đến TP. HCM vào ngày 20 tháng 7," hệ thống cần trích xuất các thông tin sau:

* Nơi xuất phát: Hà Nội
* Nơi đến: TP. HCM
* Ngày: 20 tháng 7

Về ứng dụng Intent Detection và Slot Filling được sử dụng rộng rãi trong các ứng dụng như:

* Trợ lý ảo (như Siri, Google Assistant)
* Chatbot hỗ trợ khách hàng
* Hệ thống đặt chỗ trực tuyến
* Các ứng dụng dịch vụ thông minh khác

Các thành phần này giúp hệ thống hiểu và phản hồi chính xác nhu cầu của người dùng, nâng cao trải nghiệm người dùng và tự động hóa nhiều quy trình trong các lĩnh vực khác nhau.

## Mô tả bộ dữ liệu

### Bộ ATIS

ATIS (Hệ thống Thông tin Du lịch Hàng không) là một bộ dữ liệu gồm các bản ghi âm và các bản ghi chép thủ công tương ứng về con người yêu cầu thông tin chuyến bay trên các hệ thống truy vấn du lịch hàng không tự động. Bộ dữ liệu này bao gồm 17 danh mục ý định duy nhất. Phân chia ban đầu gồm có 4478, 500 và 893 câu nói thể hiện ý định được gán nhãn trong tập huấn luyện, phát triển và kiểm tra tương ứng.

Ví dụ 1 mẫu dữ liệu

A screenshot of a computer

Description automatically generated

Tập huấn luyện chứa 4478 câu nói, tập validation và tập kiểm tra mỗi tập chứa 893 câu nói, với kích cỡ từ điển là 943, số lượng slot là 129 và 26 intent

### Bộ SNIPS

Bộ dữ liệu đánh giá hiểu ngôn ngữ tự nhiên của SNIPS là một bộ dữ liệu gồm hơn 16,000 câu truy vấn được thu thập từ cộng đồng, phân bố trong 7 ý định người dùng với độ phức tạp khác nhau:

1. **SearchCreativeWork** (Ví dụ: Tìm cho tôi chương trình truyền hình I, Robot)
2. **GetWeather** (Ví dụ: Có gió ở Boston, MA bây giờ không?)
3. **BookRestaurant** (Ví dụ: Tôi muốn đặt một nhà hàng được đánh giá cao tại Paris vào đêm mai)
4. **PlayMusic** (Ví dụ: Phát bài hát cuối cùng của Beyoncé trên Spotify)
5. **AddToPlaylist** (Ví dụ: Thêm bài Diamonds vào danh sách phát roadtrip của tôi)
6. **RateBook** (Ví dụ: Đánh giá 6 sao cho cuốn sách Of Mice and Men)
7. **SearchScreeningEvent** (Ví dụ: Kiểm tra lịch chiếu phim Wonder Woman tại Paris)

Tập huấn luyện chứa 13,084 câu nói, tập validation và tập kiểm tra mỗi tập chứa 700 câu nói, với mỗi ý định có 100 câu truy vấn.

A screenshot of a test

Description automatically generated

### Bộ mixATIS

Bộ dữ liệu đa ý định được xây dựng từ bộ dữ liệu có ý định duy nhất Atis

A table with numbers and letters

Description automatically generated

### Bộ mixSNIPS

Bộ dữ liệu đa ý định được xây dựng từ bộ dữ liệu có ý định duy nhất Snips

A table with numbers and letters

Description automatically generated

# CÁC PHƯƠNG PHÁP GIẢI QUYẾT

## Sử dụng bert cho bộ dữ liệu đơn ý định (ATIS, SNIPS)

Phương pháp này dựa trên kiến trúc mô hình BERT, một mã hóa Transformer đa tầng hai chiều dựa trên mô hình Transformer gốc (Vaswani et al., 2017). Đầu vào của BERT là sự kết hợp của các nhúng WordPiece (Wu et al., 2016), các nhúng vị trí, và nhúng đoạn. Đặc biệt, đối với các nhiệm vụ phân loại và gắn thẻ đơn câu, nhúng đoạn không có sự phân biệt. Một nhúng phân loại đặc biệt ([CLS]) được chèn làm token đầu tiên và một token đặc biệt ([SEP]) được thêm vào làm token cuối cùng. Cho một chuỗi token đầu vào x = (, . . . , ), đầu ra của BERT là H = (, . . . , ).

Mô hình BERT được huấn luyện trước bằng hai chiến lược trên văn bản không được gán nhãn với quy mô lớn, bao gồm mô hình ngôn ngữ che mặt và dự đoán câu tiếp theo. Mô hình BERT đã được huấn luyện trước cung cấp một biểu diễn câu phụ thuộc vào ngữ cảnh mạnh mẽ và có thể được sử dụng cho các nhiệm vụ mục tiêu khác nhau, ví dụ như phân loại ý định và điền slot thông qua quy trình fine-tuning, tương tự như cách nó được sử dụng cho các nhiệm vụ NLP khác.

A diagram of a computer program

Description automatically generated with medium confidence

Để mở rộng BERT thành mô hình phân loại ý định và điền slot chung, dựa trên trạng thái ẩn của token đặc biệt đầu tiên ([CLS]), được ký hiệu là h1, ý định được dự đoán như sau:

Đối với điền slot, chúng ta đưa các trạng thái ẩn cuối cùng của các token khác , . . . , vào một lớp softmax để phân loại các nhãn điền slot. Để thực hiện quy trình này với việc mã hóa token WordPiece, mỗi từ đầu vào được chia thành các sub-token bằng WordPiece tokenizer và sử dụng trạng thái ẩn tương ứng với sub-token đầu tiên làm đầu vào cho bộ phân loại softmax.

Với n∈1...N

trong đó là trạng thái ẩn tương ứng với sub-token đầu tiên của từ xn.

Để mô hình chung phân loại ý định và điền slot, mục tiêu được sắp xếp như sau:

Mục tiêu học tập là tối đa hóa xác suất có điều kiện . Mô hình được fine-tuned end-to-end bằng cách giảm thiểu hàm mất mát cross-entropy.

Ngoài ra, mô hình còn thêm một lớp Conditional Random Field (CRF) để dự đoán nhãn slot, nhằm cải thiện hiệu suất điền slot bằng cách mô hình hóa các phụ thuộc giữa các nhãn slot cho các từ xung quanh.

## Sử dụng mô hình MISCA cho phân loại bộ dữ liệu đa ý định (mixATIS, mixSNIPS)

Mô hình MISCA (Multi-task Intent and Slot Co-Attention) được thiết kế để đồng thời giải quyết hai nhiệm vụ chính trong xử lý ngôn ngữ tự nhiên: phân loại ý định (intent classification) và điền slot (slot filling). Dưới đây là chi tiết về từng thành phần của mô hình:

**1. Bộ mã hóa câu (Utterance Encoders):**

Mô hình sử dụng hai loại bộ mã hóa cho mỗi câu đầu vào:

* **Bộ mã hóa chia sẻ nhiệm vụ (Task-shared encoder)**: Tạo ra các biểu diễn vector cho từng từ trong câu bằng cách kết hợp các nhúng từ vựng và nhúng từ ký tự. Đây là bước quan trọng để hiểu ngữ cảnh của từng từ trong câu.
* **Bộ mã hóa cụ thể nhiệm vụ (Task-specific encoder)**: Được sử dụng cho mỗi nhiệm vụ cụ thể như phân loại ý định và điền slot. Bộ mã hóa này tạo ra các vector đặc trưng cho từng từ trong câu, được sử dụng để xác định ý định và các nhãn slot.

**2. Attention cho nhãn (Label Attention):**

Cơ chế attention cho nhãn giúp mô hình tập trung vào các đặc trưng quan trọng của các nhãn ý định và slot trong câu. Mỗi nhãn có thể có mức độ quan trọng khác nhau đối với từng từ trong câu, và mô hình sử dụng attention để đảm bảo rằng các đặc trưng quan trọng này được phát hiện và sử dụng hiệu quả.

A diagram of a diagram

Description automatically generated

**3. Attention đồng thời ý định và điền slot (Intent-Slot Co-Attention):**

Đây là thành phần quan trọng nhất của mô hình MISCA. Attention đồng thời giữa ý định và điền slot cho phép mô hình học được sự tương tác giữa các nhiệm vụ này. Ví dụ, thông tin về ý định có thể giúp cải thiện việc dự đoán các nhãn slot, và ngược lại. Cơ chế attention này giúp mô hình tự động học các mối quan hệ phức tạp giữa ý định và slot, cải thiện độ chính xác của cả hai nhiệm vụ.

**4. Bộ giải mã (Decoders):**

* **Giải mã ý định (Intent Decoder)**: Mô hình dự đoán nhiều ý định dựa trên vector đặc trưng cuối cùng của câu và sử dụng một hàm kích hoạt để tính toán xác suất cho từng ý định. Mô hình cũng có thể dự đoán số lượng ý định dựa trên dữ liệu đào tạo.
* **Giải mã điền slot (Slot Decoder)**: Mô hình sử dụng CRF (Conditional Random Field) để dự đoán các nhãn slot dựa trên các vector đặc trưng cuối cùng của từng từ trong câu. Đây là một phương pháp hiệu quả để giải quyết vấn đề dán nhãn chuỗi dựa trên lược đồ BIO.

**5. Huấn luyện chung (Joint Training):**

Mô hình MISCA được huấn luyện chung để tối ưu hóa đồng thời cả hai nhiệm vụ: phân loại ý định và điền slot. Mục tiêu huấn luyện là tối đa hóa xác suất có điều kiện của việc dự đoán cả ý định và các nhãn slot trong câu. Mô hình sử dụng một hàm mất mát kết hợp, điều chỉnh bởi một hằng số λ để điều chỉnh sự quan trọng giữa hai nhiệm vụ.

MISCA là một mô hình mạnh mẽ cho xử lý ngôn ngữ tự nhiên, giúp cải thiện hiệu suất của các hệ thống chatbot và các ứng dụng NLP khác trong việc hiểu và phản hồi tự động dựa trên ngôn ngữ người dùng.

## Các phương pháp cải thiện accuracy mô hình

### Pre train Bert

Trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP), mô hình BERT (Bidirectional Encoder Representations from Transformers) đã chứng minh được khả năng mạnh mẽ trong việc nắm bắt ngữ cảnh và ý nghĩa ngôn ngữ. Tuy nhiên, khi áp dụng cho các tập dữ liệu có ngữ cảnh đặc thù hoặc chứa thuật ngữ chuyên ngành, hiệu suất của mô hình BERT gốc có thể không đạt được mức tối ưu. Để khắc phục vấn đề này, việc tiền huấn luyện (pre-train) BERT với bộ dữ liệu tùy chỉnh là một giải pháp hiệu quả.

**Cải thiện hiệu suất cho ngữ cảnh cụ thể**

- Ngôn ngữ chuyên ngành: Các văn bản thuộc lĩnh vực y học, luật, kỹ thuật thường chứa đựng các thuật ngữ và ngữ pháp phức tạp. BERT gốc, được huấn luyện trên tập dữ liệu phổ quát, có thể không nắm bắt được đầy đủ các đặc thù này. Tiền huấn luyện BERT trên bộ dữ liệu chứa nội dung chuyên ngành giúp mô hình cập nhật và hiểu rõ hơn các thuật ngữ và cấu trúc này.

- Ngôn ngữ hoặc phương ngữ địa phương: Việc làm việc với ngôn ngữ ít phổ biến hoặc các phương ngữ địa phương đòi hỏi mô hình phải hiểu rõ ngữ pháp, cấu trúc và từ vựng đặc trưng. Tiền huấn luyện mô hình trên bộ dữ liệu chứa các ngôn ngữ/phương ngữ này giúp cải thiện đáng kể khả năng hiểu và xử lý của mô hình.

**Tùy chỉnh quan trọng**

- Tùy biến theo loại dữ liệu: Các ứng dụng cụ thể có thể đòi hỏi xử lý các loại văn bản khác nhau như đoạn văn dài, cuộc hội thoại ngắn, câu hỏi và câu trả lời. Tiền huấn luyện BERT với loại dữ liệu tương ứng sẽ giúp tối ưu hóa mô hình cho công việc đó.

Việc tiền huấn luyện BERT với bộ dữ liệu tùy chỉnh mang lại nhiều lợi ích vượt trội, từ việc nâng cao độ chính xác đến khả năng ứng dụng rộng rãi hơn cho các công việc NLP phức tạp. Quy trình này, mặc dù đòi hỏi tài nguyên và thời gian, đã chứng minh được giá trị to lớn trong việc tạo ra các mô hình NLP hiệu quả và chính xác hơn, đáp ứng tốt yêu cầu của các ứng dụng thực tiễn

### Tăng cường dữ liệu

Tăng cường dữ liệu (Data Augmentation) là một kỹ thuật quan trọng trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) nhằm tạo ra các phiên bản bổ sung của dữ liệu huấn luyện hiện có. Mục tiêu chính của việc này là cải thiện độ chính xác và khả năng tổng quát hóa của mô hình bằng cách cung cấp cho nó nhiều mẫu hơn để học, đặc biệt khi dữ liệu gốc bị giới hạn hoặc bị thiếu cân bằng.

**Tăng số lượng dữ liệu huấn luyện**

- Chống Thiếu Dữ Liệu: Trong nhiều trường hợp, việc thu thập và gán nhãn dữ liệu có thể tốn kém và thời gian. Tăng cường dữ liệu giúp tạo ra nhiều mẫu huấn luyện hơn từ dữ liệu hiện có, qua đó tăng cường khối lượng dữ liệu mà không cần thu thập thêm.

- Cân Bằng Dữ Liệu: Một số lớp trong dữ liệu có thể bị thiếu cân bằng, dẫn tới mô hình bị thiên lệch khi huấn luyện. Kỹ thuật tăng cường dữ liệu giúp tạo ra các mẫu bổ sung cho các lớp ít xuất hiện, giúp cân bằng dữ liệu và cải thiện hiệu suất mô hình.

**Nâng cao khả năng tổng quát hóa**

- Tăng Đa Dạng Ngữ Liệu: Việc tạo ra các biến thể mới của dữ liệu huấn luyện giúp mô hình học được nhiều trường hợp khác nhau, giảm thiểu việc overfitting vào dữ liệu gốc.

- Cải Thiện Khả Năng Chịu Đựng Lỗi: Các phương pháp tăng cường dữ liệu như thêm nhiễu ngẫu nhiên, thay thế từ đồng nghĩa, và hoán đổi từ sẽ giúp mô hình trở nên linh hoạt hơn và ít nhạy cảm với các lỗi ngữ liệu.

**Tối ưu hóa các ứng dụng cụ thể**

- Nâng Cao Độ Chính Xác: Trong các tác vụ như phân loại văn bản, trích xuất thực thể hay sinh văn bản, tăng cường dữ liệu có thể cung cấp nhiều ngữ cảnh và trường hợp khác nhau, giúp mô hình hiểu sâu sắc hơn và cải thiện độ chính xác.

- Ứng Biến Với Các Biến Thể Kiểu Dữ Liệu: Đối với các ứng dụng yêu cầu phân tích các loại văn bản khác nhau, việc tăng cường dữ liệu giúp mô hình thích ứng và hoạt động tốt hơn trên các biến thể dữ liệu đa dạng.

**Kỹ thuật tăng cường dữ liệu**

Việc tăng cường dữ liệu trong NLP thường được thực hiện thông qua các kỹ thuật sau:

1. Thay thế từ đồng nghĩa (Synonym Replacement):

- Thay thế một số từ trong văn bản bằng các từ đồng nghĩa của chúng để tạo ra các phiên bản mới của văn bản.

2. Hoán đổi từ (Token Swapping):

- Hoán đổi vị trí của các từ trong câu để tạo ra các mẫu mới mà vẫn giữ nguyên nghĩa của câu.

3. Thêm nhiễu ngẫu nhiên (Random Noise Injection):

- Thêm nhiễu vào văn bản bằng cách thay đổi, thêm hoặc xóa các từ ngẫu nhiên để mô hình học cách chịu đựng lỗi.

5. Tạo câu bằng cách dịch ngược lại (Back Translation):

- Dịch văn bản từ ngôn ngữ gốc sang ngôn ngữ thứ hai rồi dịch ngược lại về ngôn ngữ ban đầu để tạo ra phiên bản mới của văn bản.

Tăng cường dữ liệu là một kỹ thuật mạnh mẽ giúp cải thiện đáng kể hiệu suất và khả năng tổng quát hóa của các mô hình xử lý ngôn ngữ tự nhiên. Bằng cách tạo ra các biến thể mới của dữ liệu huấn luyện ban đầu, kỹ thuật này giúp tăng cường số lượng mẫu, cân bằng dữ liệu và nâng cao khả năng chịu đựng lỗi của mô hình. Việc áp dụng đúng các kỹ thuật tăng cường dữ liệu và quy trình hiệu quả sẽ mang lại những cải thiện rõ rệt trong các ứng dụng NLP thực tế.

# CÁC THÍ NGHIỆM VÀ KẾT QUẢ ĐẠT ĐƯỢC

## Sử dụng Bert để phân loại

Bộ dataset Atis

Chạy model với 20 epoch, lr = 5e-4

A screenshot of a computer program

Description automatically generated

Bộ dataset Snips

Chạy model với 20 epoch, lr = 5e-4

A screenshot of a computer program

Description automatically generated

Kết quả chạy tương đương với trong paper

## Sử dụng model MISCA để phân loại

### Dùng model MISCA với LSTM

Bộ dataset mixAtis

Chạy model với 20 epoch, lr = 5e-4

A screenshot of a computer program

Description automatically generated

Bộ dataset mixSnips

Chạy model với 20 epoch, lr = 5e-4

### Dùng model MISCA với Bert

Bộ dataset mixAtis

Chạy model với 20 epoch, lr = 5e-4

A screenshot of a computer program

Description automatically generated

Kết quả chạy so với kết quả bài báo thấy intent\_f1 giảm mạnh gần 20%, slot\_f1 giảm 12%

Bộ dataset mixSnip

Chạy model với 20 epoch, lr = 5e-4

A screenshot of a computer code

Description automatically generated

Kết quả chạy so với kết quả bài báo thấy intent\_f1 tăng nhẹ 1% , slot\_f1 tăng nhẹ 2%

Pre train lại Bert với bộ dữ liệu train của bộ mixAtis

Với 50 epoch, lr=0.01

A graph with numbers and lines

Description automatically generated

Bộ dataset mixAtis

Chạy model với Bert được pre train 20 epoch, lr = 5e-4

A screenshot of a computer

Description automatically generated

Kết quả chạy so với kết quả thấy intent\_f1 tăng mạnh so với trước khi pre-train

Kết quả so với bài báo intent\_f1 tăng 11%, slot\_f1 thấp hơn 2%

Tăng cường dữ liệu với Back translate

Tăng gấp 2 dữ liệu và tăng thêm 1 số dữ liệu cho nhãn ít mẫu của bộ mixAtis

A screenshot of a computer code

Description automatically generated

Kết quả cho thấy f1 của intent và slot đều tang them 1 %

Kết quả so với bài báo intent\_f1 tăng 12%, slot\_f1 thấp hơn 1%

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Qua một vài thử nghiệm thực tế, em đã có thể làm quen với cách để triển khai huấn luyện một mô hình học sâu, đặc biệt là với một mô hình nặng như BERT. Thông qua những cách huấn luyện khác nhau, dù chỉ là tác vụ phân loại nhưng em đã thu được những kết quả khả quan và có thể sử dụng những kiến thức này để áp dụng cho những bài toán tương lai.

## Hướng phát triển của đồ án trong tương lai

Về phương hướng phát triển: Từ bài toán này, em muốn phát triển lên một bài toán phức tạp hơn như xây dựng 1 chatbot.

Link github: https://github.com/pduy395/IDSF

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "BERT for Joint Intent Classification and Slot Filling" Qian Chen∗ , Zhu Zhuo, Wen Wang 2019 |
| [2] | "MISCA: A Joint Model for Multiple Intent Detection and Slot Filling with Intent-Slot Co-Attention" Thinh Pham and Chi Tran and Dat Quoc Nguyen |
| [3] | https://d2l.ai/chapter\_natural-language-processing-pretraining |