NGUYỄN TUẤN DŨNG

HỆ THỐNG THÔNG TIN

2025

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

NGUYỄN TUẤN DŨNG

NGHIÊN CỨU KỸ THUẬT PHÂN LOẠI CÂU TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN BẰNG MÔ HÌNH HỌC SÂU

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội – 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

NGUYỄN TUẤN DŨNG

NGHIÊN CỨU KỸ THUẬT PHÂN LOẠI CÂU TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN BẰNG MÔ HÌNH HỌC SÂU

Ngành: Hệ thống thông tin
Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN:
1. TS. NGUYỄN MẠNH CƯỜNG

Hà Nội – 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đề án tốt nghiệp thạc sĩ ngành Hệ thống thông tin với đề tài "**Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu**" là do tôi nghiên cứu, tổng hợp và thực hiện dưới sự hướng dẫn của TS. Nguyễn Mạnh Cường.

Toàn bộ nội dung của đề án, những điều được trình bày là của chính cá nhân tôi hoặc là được tham khảo, tổng hợp từ nhiều nguồn tài liệu khác nhau. Tất cả các tài liệu tham khảo, tổng hợp đều được trích xuất nguồn gốc rõ ràng. Các số liệu, kết quả được nêu trong đề án là trung thực và được nghiên cứu thực nghiệm.

Học viên thực hiện

Nguyễn Tuấn Dũng

LỜI CẢM ƠN

Đề án tốt nghiệp được hoàn thành dưới sự hướng dẫn tận tình, nghiêm khắc của **TS. Nguyễn Mạnh Cường**. Lời đầu tiên tôi xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới Thầy. Tôi đã nhận được sự hỗ trợ quý báu trong quá trình thực hiện đề tài, những kiến thức và lời khuyên hữu ích từ Thầy đã tạo động lực lớn cho tôi hoàn thành tốt nhiệm vụ của mình. Đặc biệt, những định hướng, nhận xét và góp ý của Thầy trong suốt quá trình nghiên cứu là những bài học vô cùng quý giá đối với tôi không chỉ trong việc hoàn thành đề án mà trong cả các hoạt động chuyên môn sau này.

Bên cạnh đó, tôi xin chân thành cảm ơn Thầy, Cô trường Đại học Công nghiệp Hà Nội đã tận tình dạy dỗ, trong đó phải kể đến Thầy Cô trong Khoa Công nghệ thông tin đã tạo điều kiện thuận lợi để tôi thực hiện đề án tốt nghiệp này. Tôi cũng xin cảm ơn các bạn học viên trong Khoa Công nghệ thông tin đã đóng góp ý kiến giúp tôi thực hiện đề án đạt hiệu quả hơn.

Đề án tốt nghiệp này đã giúp tôi rèn luyện kỹ năng tư duy phân tích, xử lý dữ liệu và trình bày thông tin một cách có logic và rõ ràng. Tôi hi vọng rằng những kiến thức và kinh nghiệm thu thập từ đề án tốt nghiệp này sẽ tiếp tục hỗ trợ tôi trong tương lai, không chỉ trong nghiên cứu mà còn trong sự nghiệp và cuộc sống.

Cuối cùng, tôi xin cảm ơn gia đình, bạn bè và những người đã luôn bên cạnh động viên, chia sẻ và hỗ trợ tôi trong quá trình học tập, nghiên cứu.

Học viên thực hiện

Nguyễn Tuấn Dũng

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
GPU	Graphics Processing Unit	Bộ xử lý đồ họa
GPT	Generative Pre-trained Transformer	Bộ biến đổi sinh trước được huấn luyện
GRU	Gated Recurrent Unit	Đơn vị hồi tiếp có cổng
LSTM	Long Short-Term Memory	Bộ nhớ dài ngắn hạn
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
ReLU	Rectified Linear Unit	Đơn vị chỉnh lưu tuyến tính
RNN	Recurrent Neural Network	Mạng nơ-ron hồi tiếp
SVM	Support Vector Machines	Máy vector hỗ trợ
TF-IDF	Term Frequency-Inverse Document Frequency	Tần suất xuất hiện nghịch đảo tài liệu

DANH MỤC CÁC BẢNG

DANH MỤC HÌNH ẢNH

MỞ ĐẦU

1. Giới thiệu

Trong bối cảnh công nghệ thông tin và trí tuệ nhân tạo ngày càng phát triển, xử lý ngôn ngữ tự nhiên (Natural language processing - NLP) đã trở thành một lĩnh vực nghiên cứu quan trọng và hấp dẫn. Một trong những thách thức lớn của xử lý ngôn ngữ tự nhiên là phân loại câu, một bước không thể thiếu để hiểu và xử lý ngôn ngữ của con người. Việc phân loại câu chính xác không chỉ giúp cải thiện hiệu suất của các hệ thống dịch máy, ứng dụng trò chuyện tự động, mà còn nâng cao khả năng phân tích cảm xúc, nhận diện ý định và nhiều ứng dụng khác.

Lý do chính để lựa chọn đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu" nằm ở tiềm năng to lớn của mô hình học sâu trong việc giải quyết các vấn đề phức tạp của xử lý ngôn ngữ tự nhiên. Mô hình học sâu, đặc biệt là mạng nơ-ron tích chập (CNN), đã chứng minh khả năng vượt trội trong việc xử lý dữ liệu văn bản và phân loại câu so với các phương pháp truyền thống.

Các lý do tôi lựa chọn đề tài này bao gồm:

- Thứ nhất, việc nghiên cứu và áp dụng các kỹ thuật học sâu trong phân loại câu giúp mở ra nhiều hướng nghiên cứu mới, góp phần đẩy mạnh sự phát triển của xử lý ngôn ngữ tự nhiên. Kỹ thuật học sâu có khả năng tự động học các đặc trưng quan trọng từ dữ liệu, thay vì phải dựa vào các phương pháp trích xuất đặc trưng thủ công, do đó có thể đạt được độ chính xác cao hơn và giảm thiểu công sức của các nhà nghiên cứu.
- Thứ hai, mô hình học sâu không chỉ cải thiện độ chính xác của việc phân loại câu mà còn tăng cường khả năng xử lý ngôn ngữ tự nhiên trong các ngữ cảnh khác nhau, từ đó tạo ra các ứng dụng thông minh và hiệu quả hơn. Ví dụ, trong các hệ thống hỗ trợ khách hàng tự động, việc phân loại chính

xác ý định của người dùng có thể giúp cung cấp các phản hồi chính xác và nhanh chóng hơn.

- Cuối cùng, nghiên cứu về kỹ thuật phân loại câu bằng mô hình học sâu không chỉ có ý nghĩa học thuật mà còn có giá trị thực tiễn cao. Kết quả của nghiên cứu này có thể được ứng dụng rộng rãi trong các ngành công nghiệp, từ dịch vụ khách hàng, giáo dục trực tuyến, đến y tế và quản lý thông tin. Điều này không chỉ giúp cải thiện hiệu suất và hiệu quả của các hệ thống hiện tại mà còn mở ra nhiều cơ hội mới cho sự phát triển của công nghệ.

Tóm lại, đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu" không chỉ đáp ứng nhu cầu nghiên cứu và ứng dụng thực tiễn mà còn góp phần quan trọng vào sự phát triển của lĩnh vực xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo.

2. Lý do chọn đề tài/Tính cấp thiết của đề tài

Trong thời đại công nghệ 4.0, xử lý ngôn ngữ tự nhiên (NLP) đã và đang trở thành một lĩnh vực nghiên cứu cốt lõi trong trí tuệ nhân tạo (AI) do sự gia tăng đáng kể của dữ liệu văn bản từ các nguồn khác nhau, như mạng xã hội, các hệ thống quản lý thông tin, và các dịch vụ trực tuyến. Việc phân loại câu, một nhiệm vụ quan trọng trong xử lý ngôn ngữ tự nhiên, đóng vai trò thiết yếu trong việc trích xuất thông tin, phân tích cảm xúc và tương tác giữa con người và máy tính. Từ đó, nghiên cứu và phát triển các kỹ thuật phân loại câu hiệu quả trở nên cấp bách hơn bao giờ hết.

Sự gia tăng của dữ liệu phi cấu trúc

Dữ liệu phi cấu trúc, đặc biệt là văn bản, đang phát triển nhanh chóng và chiếm một tỷ lệ lớn trong tổng số dữ liệu được tạo ra hàng ngày. Việc xử lý và phân tích hiệu quả loại dữ liệu này là một thách thức lớn đối với các nhà nghiên cứu và nhà phát triển. Kỹ thuật phân loại câu đóng vai trò quan trọng trong việc hiểu và xử lý ngữ nghĩa của văn bản, từ đó chuyển đổi dữ liệu phi cấu trúc thành thông tin có giá trị.

Yêu cầu của các ứng dụng thực tiễn

Các ứng dụng thực tiễn như ứng dụng trò chuyện tự động, hệ thống hỗ trợ khách hàng tự động, dịch máy, và các công cụ phân tích cảm xúc đều dựa vào việc phân loại câu để cải thiện khả năng tương tác và cung cấp trải nghiệm người dùng tốt hơn. Khả năng phân loại câu chính xác và nhanh chóng giúp các hệ thống này hoạt động hiệu quả hơn, đáp ứng yêu cầu ngày càng cao của người dùng trong việc xử lý và phân tích thông tin ngôn ngữ tự nhiên.

Tiềm năng của học sâu trong phân loại câu

Mô hình học sâu đã chứng minh tiềm năng to lớn trong việc nâng cao hiệu suất của các nhiệm vụ xử lý ngôn ngữ tự nhiên. Khả năng học các đặc trưng phức tạp từ dữ liệu mà không cần dựa vào các phương pháp trích xuất thủ công giúp mô hình học sâu vượt trội hơn so với các kỹ thuật truyền thống. Việc nghiên cứu và áp dụng mô hình học sâu trong phân loại câu không chỉ cải thiện độ chính xác mà còn mở ra nhiều cơ hội phát triển cho các ứng dụng trí tuệ nhân tạo.

Đáp ứng nhu cầu nghiên cứu và phát triển công nghệ

Nghiên cứu về kỹ thuật phân loại câu bằng mô hình học sâu không chỉ có ý nghĩa khoa học mà còn đóng góp quan trọng vào việc phát triển công nghệ, nâng cao năng lực cạnh tranh của các doanh nghiệp trong lĩnh vực công nghệ thông tin. Đồng thời, nghiên cứu này cũng góp phần tạo ra những bước đột phá mới trong việc xử lý ngôn ngữ tự nhiên, đáp ứng nhu cầu nghiên cứu và ứng dụng thực tiễn.

Giải quyết các thách thức ngôn ngữ

Ngôn ngữ tự nhiên đa dạng và phức tạp, mỗi ngôn ngữ có những đặc điểm riêng biệt cần được xử lý một cách tinh tế. Việc phát triển các kỹ thuật phân loại câu phù hợp với ngôn ngữ cụ thể là cần thiết để đảm bảo hiệu quả và chính xác trong các ứng dụng ngôn ngữ địa phương.

Đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu" không chỉ đáp ứng nhu cầu cấp thiết của thực tiễn và công

nghệ mà còn mở ra nhiều hướng nghiên cứu mới, góp phần vào sự phát triển bền vững của lĩnh vực xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo.

3. Tổng quan nghiên cứu

Nghiên cứu bao gồm các nội dung về bài toán phân loại câu sử dụng mô hình học sâu.

Khái niệm và tầm quan trọng của phân loại câu

Phân loại câu là một nhánh quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, tập trung vào việc phân loại các câu thành các nhóm hoặc nhãn khác nhau dựa trên nội dung và ngữ nghĩa. Nhiệm vụ này có ứng dụng rộng rãi trong nhiều lĩnh vực như phân tích cảm xúc, nhận diện ý định, dịch máy, và các hệ thống hỗ trợ khách hàng tự động. Việc phân loại câu chính xác giúp cải thiện hiệu suất của các ứng dụng này, tăng cường khả năng tương tác giữa con người và máy tính.

Các kỹ thuật truyền thống trong phân loại câu

Trước khi học sâu trở thành xu hướng chủ đạo, phân loại câu chủ yếu dựa vào các kỹ thuật máy học truyền thống như Naive Bayes, Support Vector Machines, và Decision Trees. Những kỹ thuật này thường yêu cầu bước trích xuất đặc trưng thủ công, sử dụng các phương pháp như bag-of-words hoặc TF-IDF để chuyển đổi văn bản thành định dạng số. Mặc dù đã đạt được một số thành công nhất định, các phương pháp này có những hạn chế nhất định, đặc biệt là trong việc xử lý ngữ nghĩa phức tạp và bối cảnh ngữ cảnh của câu.

Sự phát triển đột phá của học sâu trong xử lý ngôn ngữ tự nhiên

Với sự phát triển của trí tuệ nhân tạo và sức mạnh tính toán, học sâu đã trở thành một công cụ mạnh mẽ trong việc giải quyết các vấn đề phức tạp của xử lý ngôn ngữ tự nhiên, bao gồm phân loại câu. Mô hình học sâu như mạng nơ-ron tích chập (CNN) đã chứng tỏ hiệu quả vượt trội so với các kỹ thuật truyền thống nhờ khả năng tự động học các đặc trưng từ dữ liệu.

4. Mục tiêu của đề tài

Đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu" đặt ra các mục tiêu nghiên cứu cụ thể như sau:

- Khảo sát các phương pháp phân loại câu truyền thống và hiện đại: Phân tích sự phát triển của các kỹ thuật phân loại câu từ các phương pháp truyền thống như Naive Bayes, SVM cho đến các kỹ thuật hiện đại sử dụng học sâu cùng với các ứng dụng của chúng trong việc phân loại câu. Đánh giá sự hiệu quả của những mô hình này trong việc xử lý các ngữ cảnh và ngữ nghĩa phức tạp của văn bản.
- Sử dụng các kiến trúc học sâu tiên tiến để thiết kế và triển khai các mô hình phân loại câu.
- Đánh giá hiệu suất và tính ứng dụng của mô hình

5. Nội dung nghiên cứu

Đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu" được triển khai qua các nội dung nghiên cứu sau:

Tổng quan về xử lý ngôn ngữ tự nhiên và phân loại câu

- Giới thiệu tổng quan về xử lý ngôn ngữ tự nhiên: Trình bày khái quát về lĩnh vực xử lý ngôn ngữ tự nhiên, bao gồm các khái niệm cơ bản, ứng dụng và thách thức. Đặc biệt, nhấn mạnh vai trò của xử lý ngôn ngữ tự nhiên trong việc phát triển các hệ thống thông minh như dịch máy và phân tích cảm xúc.
- Phân loại câu trong xử lý ngôn ngữ tự nhiên: Nghiên cứu định nghĩa, tầm quan trọng và các ứng dụng thực tiễn của phân loại câu. Phân loại câu là bước nền tảng cho nhiều nhiệm vụ khác nhau trong xử lý ngôn ngữ tự nhiên, giúp hiểu ngữ nghĩa và ý định của văn bản.

Các phương pháp phân loại câu truyền thống

- Phương pháp thống kê và học máy truyền thống: Khảo sát các kỹ thuật phân loại câu truyền thống như Naive Bayes và Support Vector Machines (SVM). Trình bày ưu và nhược điểm của các phương pháp này.
- Trích xuất và biểu diễn đặc trưng: Nghiên cứu các phương pháp trích xuất và biểu diễn đặc trưng cho văn bản, như bag-of-words, và các kỹ thuật giảm chiều dữ liệu. Đánh giá hiệu quả của các phương pháp này trong việc hỗ trợ phân loại câu.

Học sâu trong phân loại câu

- Giới thiệu về học sâu: Trình bày tổng quan về học sâu và sự phát triển của nó trong xử lý ngôn ngữ tự nhiên. Giải thích cơ chế hoạt động và lợi ích của các mô hình học sâu so với các phương pháp truyền thống.
- Mạng nơ-ron tích chập (CNN): Nghiên cứu cách CNN được áp dụng trong phân loại câu. Trình bày kiến trúc của CNN và cách nó xử lý dữ liệu văn bản, nhấn mạnh khả năng trích xuất đặc trưng cục bộ hiệu quả từ văn bản.

Thực nghiệm và đánh giá mô hình

- Thiết lập môi trường thực nghiệm: Mô tả môi trường thực nghiệm, các công cụ thực nghiệm
- Huấn luyện và thử nghiệm mô hình: Triển khai các mô hình học sâu đã chọn trên bộ dữ liệu phù hợp
- Đánh giá hiệu suất mô hình: Sử dụng các chỉ số đánh giá mô hình

Khả năng ứng dụng thực tiễn

- Ứng dụng trong các lĩnh vực khác nhau: Nghiên cứu tiềm năng ứng dụng của mô hình phân loại câu trong các hệ thống thực tế như ứng dụng trò chuyện, dịch máy, phân tích cảm xúc, và các hệ thống hỗ trợ khách hàng tự động.
- Thách thức và giải pháp: Trình bày các thách thức gặp phải trong quá trình triển khai mô hình vào thực tiễn và đề xuất các giải pháp khả thi để khắc phục những thách thức này.

Kết quả nghiên cứu và đề xuất hướng phát triển

- Tổng kết kết quả nghiên cứu: Tóm tắt các kết quả đạt được trong nghiên cứu, nhấn mạnh các đóng góp mới cho lĩnh vực xử lý ngôn ngữ tự nhiên và học sâu.
- Đề xuất hướng phát triển tiếp theo: Đưa ra các khuyến nghị và hướng phát triển cho các nghiên cứu trong tương lai, tập trung vào việc cải thiện mô hình và mở rộng ứng dụng của phân loại câu trong các lĩnh vực khác nhau.

6. Phương pháp nghiên cứu

Đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu", chúng tôi sử dụng một phương pháp nghiên cứu kết hợp giữa lý thuyết và thực nghiệm để đảm bảo tính toàn diện và chính xác của nghiên cứu. Các bước phương pháp nghiên cứu được triển khai như sau:

Phương pháp nghiên cứu tài liệu

- Tìm hiểu tài liệu chuyên ngành: Thu thập và phân tích các tài liệu học thuật, bài báo khoa học, sách và các nghiên cứu trước đây liên quan đến phân loại câu trong xử lý ngôn ngữ tự nhiên (NLP) và học sâu.
- Khảo sát các kỹ thuật hiện đại: Nghiên cứu các mô hình học sâu tiên tiến như CNN,... Đánh giá ưu điểm, nhược điểm, và ứng dụng của mô hình trong việc phân loại câu.

Phương pháp thực nghiệm

- Chuẩn bị dữ liệu: Lựa chọn và tiền xử lý các bộ dữ liệu cần thiết cho quá trình huấn luyện và thử nghiệm.
- Huấn luyện mô hình: Sử dụng dữ liệu đã chuẩn bị để huấn luyện các mô hình học sâu.
- Thử nghiệm mô hình: Sau khi huấn luyện, tiến hành thử nghiệm mô hình trên bộ dữ liệu kiểm thử để đánh giá hiệu suất.

Phương pháp trình bày

- Tổng hợp và phân tích dữ liệu thu được từ quá trình thử nghiệm. Xây dựng các báo cáo chi tiết về hiệu suất mô hình, so sánh với các phương pháp khác, và những cải tiến đạt được. Trình bày rõ ràng các thông tin trên báo cáo.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

1.1.1. Khái niệm xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (Natural language processing - NLP) là một lĩnh vực phân ngành của khoa học máy tính, trí tuệ nhân tạo và ngôn ngữ học, tập trung vào việc tương tác giữa máy tính và ngôn ngữ tự nhiên của con người. Xử lý ngôn ngữ tự nhiên liên quan đến việc phát triển các thuật toán và mô hình để cho phép máy tính hiểu, phân tích, và tạo ra ngôn ngữ tự nhiên, từ đó thực hiện các nhiệm vụ như dịch máy, nhận dạng giọng nói, phân tích cảm xúc, và tương tác với người dùng thông qua văn bản hoặc lời nói [1].

Xử lý ngôn ngữ tự nhiên bao gồm nhiều nhiệm vụ khác nhau, mỗi nhiệm vụ nhằm giải quyết một khía cạnh cụ thể của ngôn ngữ tự nhiên. Các nhiệm vụ cơ bản trong xử lý ngôn ngữ tự nhiên bao gồm:

- **Phân tích cú pháp:** Xác định cấu trúc cú pháp của câu, nghĩa là cách các từ và cụm từ được sắp xếp và liên kết với nhau trong một câu. Phân tích cú pháp là cơ sở cho việc hiểu ngữ nghĩa của câu.
- **Phân tích ngữ nghĩa:** Tập trung vào việc hiểu ý nghĩa của các từ, cụm từ và câu. Phân tích ngữ nghĩa bao gồm việc nhận diện ngữ cảnh, phân giải đồng nghĩa, và hiểu nghĩa đa nghĩa của từ trong các tình huống khác nhau.
- **Phân loại câu:** Phân loại văn bản thành các danh mục hoặc lớp nhất định dựa trên nội dung. Ví dụ, phân loại email thành thư rác và thư không rác, hoặc phân loại các bài viết thành các chủ đề khác nhau.
- **Nhận dạng thực thể có tên:** Nhận diện và phân loại các thực thể trong văn bản như tên người, địa điểm, tổ chức, và các danh từ riêng khác. Đây là một bước quan trọng trong việc trích xuất thông tin từ văn bản.
- **Dịch máy:** Tự động dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. Dịch máy yêu cầu hiểu sâu về cả cú pháp và ngữ nghĩa của cả hai ngôn ngữ nguồn và đích.

- **Tóm tắt văn bản:** Tạo ra một bản tóm tắt ngắn gọn và súc tích từ một đoạn văn bản dài. Tóm tắt văn bản có thể là tóm tắt trích dẫn hoặc tóm tắt tổng hợp.
- **Nhận dạng giọng nói:** Chuyển đổi giọng nói thành văn bản. Đây là một trong những ứng dụng phổ biến nhất của xử lý ngôn ngữ tự nhiên, được sử dụng trong các trợ lý ảo như Siri, Alexa, và Google Assistant.
- **Sinh ngôn ngữ tự nhiên:** Tạo ra văn bản hoặc lời nói có ý nghĩa từ dữ liệu có cấu trúc. Sinh ngôn ngữ tự nhiên có thể được sử dụng để tự động tạo báo cáo, tóm tắt dữ liệu hoặc giao tiếp tự động.

1.1.2. Lịch sử phát triển của xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên là một lĩnh vực nghiên cứu liên ngành, với lịch sử phát triển kéo dài hàng thập kỷ. Sự tiến bộ của xử lý ngôn ngữ tự nhiên có thể được chia thành nhiều giai đoạn, mỗi giai đoạn phản ánh những bước tiến quan trọng trong công nghệ với những giai đoạn chính:

Giai đoạn khởi đầu và nền tảng lý thuyết (1950 - 1980)

- **Những năm 1950:** Xử lý ngôn ngữ tự nhiên có nguồn gốc từ những năm 1950, trên tạp chí Mind, Alan Turing đã xuất bản một bài báo có tựa đề "Computing Machinery and Intelligence" trong đó đề xuất cái mà hiện được gọi là bài kiểm tra Turing như một tiêu chí của trí thông minh, mặc dù vào thời điểm đó, nó không được nêu rõ là một vấn đề tách biệt với trí tuệ nhân tạo.
- **ELIZA (1966):** Một trong những chương trình máy tính nổi tiếng nhất trong giai đoạn này là ELIZA, được phát triển bởi Joseph Weizenbaum tại MIT. ELIZA sử dụng một tập hợp các quy tắc đơn giản để tạo ra các phản hồi ngôn ngữ tự nhiên, giả lập một nhà tâm lý học. ELIZA đôi khi cung cấp một tương tác giống con người đến kinh ngạc. Khi "bệnh nhân" vượt quá cơ sở kiến thức rất nhỏ, ELIZA có thể cung cấp một phản hồi chung chung.

Mặc dù ELIZA khá sơ khai, nó đã chứng minh tiềm năng của xử lý ngôn ngữ tự nhiên trong việc tương tác với con người.

- **Những năm 1980:** Giai đoạn này được xem là thời kỳ đỉnh cao của các phương pháp biểu tượng trong xử lý ngôn ngữ tự nhiên, với một số trọng tâm nghiên cứu chính trong giai đoạn này bao gồm:
    + **Phân tích cú pháp dựa trên quy tắc:** Đây là thời kỳ mà các phương pháp phân tích cú pháp dựa trên quy tắc nhận được nhiều sự chú ý, chẳng hạn như sự phát triển của HPSG (Head-Driven Phrase Structure Grammar), một dạng hiện thực hóa tính toán của ngữ pháp tạo sinh.
    + **Xử lý hình thái học:** Ví dụ như two-level morphology, một phương pháp xử lý hình thái học hai tầng để phân tích cấu trúc từ.
    + **Ngữ nghĩa học:** Các nghiên cứu về ngữ nghĩa, chẳng hạn như thuật toán Lesk algorithm, được phát triển để giải quyết vấn đề phân giải đa nghĩa của từ.
    + **Tham chiếu:** Nghiên cứu về tham chiếu, đặc biệt là trong lý thuyết trung tâm nhằm hiểu cách các đơn vị ngôn ngữ như danh từ và đại từ được sử dụng để duy trì tính gắn kết trong văn bản.
    + **Các lĩnh vực khác của hiểu ngôn ngữ tự nhiên:** Bao gồm lý thuyết cấu trúc tu từ, giúp hiểu và mô tả cách các phần của văn bản liên kết với nhau về mặt tu từ.

Ngoài ra, các hướng nghiên cứu khác cũng được tiếp tục trong giai đoạn này, chẳng hạn như sự phát triển của các ứng dụng trò chuyện, với ví dụ nổi bật là Racter và Jabberwacky.

Một phát triển quan trọng trong thời kỳ này, đóng vai trò là tiền đề cho bước ngoặt để chuyển sang các phương pháp thống kê trong xử lý ngôn ngữ tự nhiên vào những năm 1990, là sự gia tăng tầm quan trọng của việc đánh giá định lượng

trong nghiên cứu xử lý ngôn ngữ tự nhiên. Điều này dần dần dẫn đến việc áp dụng các phương pháp thống kê và học máy rộng rãi hơn trong các nghiên cứu và ứng dụng xử lý ngôn ngữ tự nhiên sau này.

Giai đoạn của các phương pháp thống kê trong xử lý ngôn ngữ tự nhiên (1990 - 2010)

Trước 1990, hầu hết các hệ thống xử lý ngôn ngữ tự nhiên dựa trên các bộ quy tắc phức tạp được viết thủ công, hầu hết các hệ thống phụ thuộc vào các tập ngữ liệu được phát triển đặc biệt cho các nhiệm vụ mà chúng thực hiện, điều này thường là một hạn chế lớn đối với sự thành công của các hệ thống này. Những quy tắc này thường rất khó mở rộng và duy trì, và hệ thống gặp khó khăn trong việc xử lý các biến đổi phức tạp của ngôn ngữ tự nhiên.

- **Thành công ban đầu của các phương pháp thống kê:** Nhiều thành công đáng chú ý của các phương pháp thống kê trong xử lý ngôn ngữ tự nhiên đã diễn ra trong lĩnh vực dịch máy, đặc biệt là nhờ công việc tại IBM Research, với các mô hình sắp xếp của IBM [2]. Những hệ thống này đã tận dụng được các tập ngữ liệu đa ngữ có sẵn, được tạo ra bởi các cơ quan chính phủ như Quốc hội Canada và Liên minh Châu Âu, do luật pháp yêu cầu dịch tất cả các thủ tục hành chính sang tất cả các ngôn ngữ chính thức của các hệ thống chính phủ tương ứng.
- **Sự phát triển của dữ liệu ngôn ngữ chưa được gán nhãn:** Với sự phát triển của web, ngày càng có nhiều dữ liệu ngôn ngữ thô (chưa được gán nhãn) trở nên sẵn có từ giữa những năm 1990s. Do đó, nghiên cứu ngày càng tập trung vào các thuật toán học không giám sát (unsupervised) và học bán giám sát (semi-supervised). Các thuật toán này có thể học từ dữ liệu chưa được gán nhãn với các câu trả lời mong muốn hoặc sử dụng kết hợp dữ liệu đã gán nhãn và chưa gán nhãn.

Thông thường, các nhiệm vụ này khó khăn hơn nhiều so với học có giám sát, và thường tạo ra kết quả kém chính xác hơn với cùng một lượng dữ liệu đầu

vào. Tuy nhiên, có một lượng lớn dữ liệu chưa gán nhãn có sẵn (bao gồm, ví dụ như toàn bộ nội dung của World Wide Web), điều này có thể bù đắp cho các kết quả kém nếu thuật toán được sử dụng có độ phức tạp thời gian đủ thấp để thực tế áp dụng.

Xử lý ngôn ngữ tự nhiên bằng mạng nơ-ron (Từ năm 2000 đến nay)

Đây là giai đoạn mà các mô hình mạng nơ-ron sâu và học biểu diễn đã thay đổi căn bản cách tiếp cận đối với các nhiệm vụ trong xử lý ngôn ngữ tự nhiên.

- **Trong năm 2003:** Một bước ngoặt quan trọng đã diễn ra khi mô hình n-gram, một trong những thuật toán thống kê tốt nhất thời đó, bị vượt qua bởi một mạng nơ-ron nhiều lớp (multi-layer perceptron) với một lớp ẩn và ngữ cảnh của một số từ nhất định, được huấn luyện trên tới 14 triệu từ bằng cách sử dụng một cụm máy tính (CPU cluster) trong ngữ cảnh mô hình hóa ngôn ngữ. Nghiên cứu này được thực hiện bởi Yoshua Bengio và các cộng sự [3], đánh dấu một bước tiến quan trọng trong việc áp dụng các mạng nơ-ron vào xử lý ngôn ngữ tự nhiên.
- **Năm 2010 đến nay:** Tomáš Mikolov, khi đó là nghiên cứu sinh Tiến sĩ tại Đại học Công nghệ Brno, cùng với các cộng sự [4] đã áp dụng một mạng nơ-ron hồi tiếp đơn giản với một lớp ẩn vào việc mô hình hóa ngôn ngữ. Công trình của Mikolov đã mở đường cho sự phát triển của Word2vec, một mô hình nổi tiếng giúp biểu diễn từ ngữ trong không gian vector. Mô hình này đã trở thành nền tảng cho nhiều nghiên cứu và ứng dụng xử lý ngôn ngữ tự nhiên, nhờ khả năng học từ các ngữ cảnh xung quanh và nắm bắt được các quan hệ ngữ nghĩa như tương đồng và đồng nghĩa.

Lịch sử phát triển của xử lý ngôn ngữ tự nhiên là sự không ngừng cải tiến của các phương pháp và công nghệ nhằm thu hẹp khoảng cách giữa ngôn ngữ của con người và khả năng học tập của máy tính. Từ những nỗ lực ban đầu với các hệ thống dựa trên quy tắc trong những năm 1950s đến 1980s, xử lý ngôn ngữ tự nhiên đã chuyển mình mạnh mẽ với sự ra đời của các mô hình thống kê trong những

năm 1990. Cuộc cách mạng này đã được thúc đẩy bởi sự gia tăng của sức mạnh tính toán và lượng dữ liệu khổng lồ, cùng với sự chuyển đổi từ các lý thuyết ngôn ngữ truyền thống sang việc sử dụng ngữ liệu thực tế.

Bước vào những năm 2000, xử lý ngôn ngữ tự nhiên đã bước sang một giai đoạn mới với sự xuất hiện của các mô hình mạng nơ-ron sâu, đặc biệt là các kỹ thuật học sâu và học biểu diễn. Những tiến bộ này, cùng với sự phát triển của các mô hình tiên tiến đã mở ra những khả năng mới trong việc xử lý và hiểu ngôn ngữ tự nhiên với độ chính xác và hiệu suất vượt trội.

1.1.3. Ứng dụng của xử lý ngôn ngữ tự nhiên

Với những khả năng mạnh mẽ của mình, xử lý ngôn ngữ tự nhiên đã trở thành một trong những lĩnh vực quan trọng nhất của trí tuệ nhân tạo, với nhiều ứng dụng thực tế trong các ngành công nghiệp khác nhau. Từ việc hỗ trợ con người trong giao tiếp, phân tích dữ liệu đến tự động hóa quy trình kinh doanh, xử lý ngôn ngữ tự nhiên đang ngày càng được tích hợp vào nhiều khía cạnh của cuộc sống và công việc. Dưới đây là những ứng dụng phổ biến và quan trọng của xử lý ngôn ngữ tự nhiên:

Dịch máy

Dịch máy là một trong những ứng dụng lâu đời và quan trọng nhất của xử lý ngôn ngữ tự nhiên. Dịch máy tự động giúp chuyển đổi văn bản hoặc lời nói từ ngôn ngữ này sang ngôn ngữ khác, đóng vai trò quan trọng trong việc kết nối các ngôn ngữ và văn hóa khác nhau trên toàn thế giới. Qua nhiều thập kỷ phát triển, dịch máy đã tiến bộ vượt bậc từ các phương pháp dựa trên quy tắc ban đầu đến các mô hình học sâu hiện đại.

Trong những năm 1950s, dịch máy chủ yếu dựa trên các hệ thống quy tắc, trong đó các quy tắc ngữ pháp và từ vựng được lập trình thủ công. Nguồn gốc của dịch máy bắt nguồn từ công trình của Al-Kindi , một nhà mật mã học người Ả Rập thế kỷ thứ chín, người đã phát triển các kỹ thuật dịch ngôn ngữ hệ thống, bao gồm phân tích mật mã, phân tích tần suất và xác suất và thống kê, được sử dụng

trong dịch máy hiện đại. Ý tưởng về dịch máy sau đó xuất hiện vào thế kỷ 17. Năm 1629, René Descartes đề xuất một ngôn ngữ phổ quát, với các ý tưởng tương đương ở các ngôn ngữ khác nhau chia sẻ một ký hiệu.

Ý tưởng sử dụng máy tính kỹ thuật số để dịch ngôn ngữ tự nhiên đã được đề xuất từ năm 1947. Điển hình nhất là dự án Georgetown-IBM năm 1954, trong đó một đoạn văn bản tiếng Nga được dịch sang tiếng Anh. Mặc dù đạt được một số thành công ban đầu, các hệ thống này nhanh chóng bộc lộ những hạn chế do sự phức tạp và tính không linh hoạt của ngôn ngữ tự nhiên.

Đến những năm 1990, các phương pháp dịch máy thống kê bắt đầu thay thế các hệ thống dựa trên quy tắc. Các phương pháp này dựa vào các mô hình xác suất để dự đoán các câu dịch dựa trên tần suất xuất hiện của từ và cụm từ trong các cặp ngôn ngữ song ngữ. Một ví dụ quan trọng là các mô hình sắp xếp của IBM, đặc biệt là trong dự án Europarl [5], tận dụng các tài liệu đã được dịch sẵn từ các cơ quan chính phủ như Liên minh Châu Âu.

Với sự xuất hiện của mạng nơ-ron và các mô hình học sâu vào đầu thập niên 2010, dịch máy đã tiến đến một bước tiến mới với sự ra đời của dịch máy nơ-ron. Thay vì dịch từng cụm từ hoặc câu một cách rời rạc, dịch máy nơ-ron sử dụng mạng nơ-ron sâu để mô hình hóa toàn bộ câu trong ngữ cảnh, giúp cải thiện đáng kể độ chính xác và mượt mà của bản dịch.

Ứng dụng thực tế của dịch máy:

- **Dịch văn bản:** Google Translate và các hệ thống dịch máy trực tuyến khác là ví dụ điển hình về ứng dụng của dịch máy trong việc dịch văn bản giữa hàng trăm ngôn ngữ khác nhau. Người dùng có thể nhanh chóng dịch các đoạn văn, trang web, hoặc thậm chí toàn bộ tài liệu chỉ trong vài giây.

Hình 1.1. Công cụ Google Translate

- **Dịch máy trong các doanh nghiệp:** Nhiều công ty sử dụng dịch máy để dịch tự động email, tài liệu nội bộ, và các tài liệu hướng dẫn. Điều này giúp tiết kiệm thời gian và chi phí, đặc biệt trong các tổ chức toàn cầu có nhu cầu giao tiếp đa ngôn ngữ.
- **Dịch ngôn ngữ trong thời gian thực:** Một ứng dụng quan trọng khác của dịch máy là dịch ngôn ngữ trong thời gian thực, chẳng hạn như trong các cuộc hội nghị quốc tế, các cuộc gọi video đa ngôn ngữ, hoặc khi đi du lịch. Dịch máy hỗ trợ dịch trực tiếp các cuộc hội thoại, giúp loại bỏ rào cản ngôn ngữ trong giao tiếp quốc tế.
- **Dịch ngôn ngữ trong giáo dục và nghiên cứu:** Trong lĩnh vực giáo dục và nghiên cứu, dịch máy giúp sinh viên và nhà nghiên cứu truy cập vào các tài liệu học thuật và nghiên cứu từ khắp nơi trên thế giới, bất kể ngôn ngữ gốc của tài liệu.

Nhận dạng giọng nói

Nhận dạng giọng nói là một trong những ứng dụng nổi bật của xử lý ngôn ngữ tự nhiên, cho phép chuyển đổi ngôn ngữ nói thành văn bản. Công nghệ này đã trở thành một phần không thể thiếu trong nhiều sản phẩm và dịch vụ hiện đại, từ trợ lý ảo đến hệ thống điều khiển bằng giọng nói nhằm đáp ứng yêu cầu ngày càng cao của con người.

Nhận dạng giọng nói đã trải qua nhiều giai đoạn phát triển, bắt đầu từ các hệ thống nhận dạng đơn giản chỉ với vài từ hoặc câu lệnh cố định vào những năm

1950. Đến những năm 1980 và 1990, các hệ thống nhận dạng giọng nói dựa trên mô hình thống kê, như Hidden Markov Models đã giúp tăng cường khả năng nhận dạng các chuỗi âm thanh dài hơn và đa dạng hơn. Tuy nhiên, phải đến đầu những năm 2010, với sự ra đời của các mô hình mạng nơ-ron và học sâu, nhận dạng giọng nói mới thực sự đạt được bước tiến đáng kể về độ chính xác và tính linh hoạt [6].

Ứng dụng của nhận dạng giọng nói:

- **Trợ lý ảo:** Nhận dạng giọng nói là công nghệ cốt lõi đằng sau các trợ lý ảo như Siri của Apple, Alexa của Amazon, Google Assistant, và Cortana của Microsoft. Các trợ lý này có khả năng nhận diện và xử lý các lệnh bằng giọng nói để thực hiện nhiều tác vụ như gửi tin nhắn, đặt báo thức, tìm kiếm thông tin, và điều khiển các thiết bị thông minh trong nhà.

Hình 1.2. Trợ lý ảo Siri nâng cao trải nghiệm người dùng

- **Hệ thống điều khiển bằng giọng nói:** Nhận dạng giọng nói được tích hợp vào các hệ thống điều khiển trong ô tô, thiết bị gia đình, và thiết bị y tế. Ví dụ, người lái xe có thể sử dụng lệnh giọng nói để điều chỉnh hệ thống giải trí, gọi điện thoại, hoặc thiết lập chỉ đường mà không cần phải rời mắt khỏi đường đi.

- **Giao tiếp với máy tính và thiết bị di động:** Trên các thiết bị di động và máy tính, nhận dạng giọng nói giúp người dùng thực hiện các tác vụ như soạn thảo văn bản, tìm kiếm trên web, và điều hướng ứng dụng chỉ bằng giọng nói. Điều này giúp nâng cao trải nghiệm người dùng và hỗ trợ những người có khuyết tật trong việc sử dụng công nghệ.
- **Tổng đài tự động:** Nhiều trung tâm dịch vụ khách hàng sử dụng hệ thống nhận dạng giọng nói để tự động hóa các cuộc gọi, giúp khách hàng giải quyết các vấn đề thông qua các hệ thống tự động mà không cần đến sự can thiệp của con người. Điều này giúp giảm chi phí vận hành và tăng tốc độ xử lý yêu cầu.

Phân tích cảm xúc

Phân tích cảm xúc, còn được gọi là khai thác ý kiến, là quá trình sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, học máy, và khai thác dữ liệu để xác định, trích xuất và phân loại cảm xúc hoặc thái độ của con người trong các đoạn văn bản. Đây là một trong những ứng dụng quan trọng và phổ biến nhất của xử lý ngôn ngữ tự nhiên, với nhiều ứng dụng trong kinh doanh, truyền thông xã hội, dịch vụ khách hàng, và nhiều lĩnh vực khác. Phân tích cảm xúc nhằm xác định thái độ (tích cực, tiêu cực, trung lập) hoặc cảm xúc (như vui vẻ, giận dữ, buồn bã) trong một văn bản nhất định. Văn bản có thể là bài đăng trên mạng xã hội, đánh giá sản phẩm, phản hồi của khách hàng, hoặc bất kỳ nội dung văn bản nào khác mà người dùng chia sẻ.

Phân tích cảm xúc trải qua nhiều giai đoạn phát triển với nhiều công nghệ được cải tiến không ngừng theo thời gian. Khởi đầu là phân tích cảm xúc dựa trên từ vựng, phương pháp này sử dụng các từ điển chứa các từ hoặc cụm từ được gán nhãn cảm xúc. Khi một văn bản được phân tích, hệ thống sẽ tra cứu các từ trong từ điển này để xác định cảm xúc của văn bản. Sau đó phát triển thành sử dụng các thuật toán như Naive Bayes, Support Vector Machines, và Decision trees để học

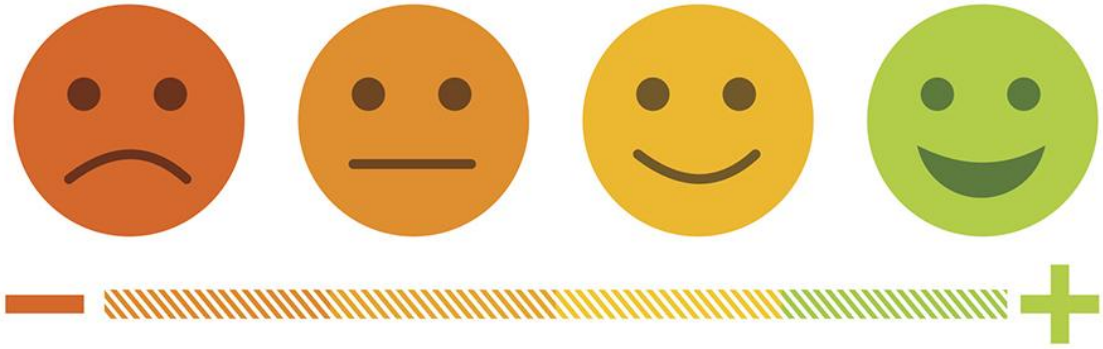
từ các tập dữ liệu gán nhãn trước đó. Các mô hình này dựa vào các đặc trưng như từ vựng, tần suất từ, và n-gram để dự đoán cảm xúc.

Ở thời điểm hiện tại, các mô hình học sâu đã mang lại bước tiến lớn trong phân tích cảm xúc. Các mô hình này có khả năng nắm bắt ngữ cảnh và mối quan hệ giữa các từ trong câu, giúp hiểu rõ hơn về cảm xúc được truyền tải trong văn bản.

Ứng dụng của phân tích cảm xúc:

- **Trong truyền thông và quản lý thương hiệu:** Các công ty có thể sử dụng phân tích cảm xúc để đánh giá hiệu quả của các chiến dịch truyền thông xã hội và quảng cáo. Bằng cách phân tích phản ứng của công chúng đối với một chiến dịch cụ thể, doanh nghiệp có thể đo lường tác động và điều chỉnh chiến lược trong thời gian thực. Đồng thời theo dõi cảm xúc của khách hàng về thương hiệu của họ trên các nền tảng xã hội và trong các bài đánh giá trực tuyến. Bằng cách nhận diện các xu hướng cảm xúc tích cực hay tiêu cực, doanh nghiệp có thể điều chỉnh chiến lược quảng bá sản phẩm và dịch vụ để đáp ứng nhu cầu và mong đợi của khách hàng.
- **Trong giáo dục:** Phân tích cảm xúc có thể được sử dụng để đánh giá phản hồi của học sinh đối với các khóa học, giáo viên, và nội dung học tập. Thông qua việc phân tích các phản hồi này, các trường học và tổ chức giáo dục có thể điều chỉnh chương trình giảng dạy và phương pháp giảng dạy để cải thiện trải nghiệm học tập. Phân tích cảm xúc có thể giúp giảng viên tăng khả năng nắm bắt trạng thái cảm xúc của học sinh qua các bài trả lời khảo sát. Điều này giúp giảng viên phát hiện sớm các vấn đề, chẳng hạn như học sinh có dấu hiệu gặp những vấn đề về tâm lý, từ đó có thể can thiệp kịp thời để tránh ảnh hưởng đến khả năng học tập.
- **Trong y tế:** Phân tích cảm xúc có thể được sử dụng để theo dõi sự tiến triển của bệnh nhân trong quá trình điều trị tâm lý, giúp xác định hiệu quả của các phương pháp điều trị và điều chỉnh phác đồ điều trị khi cần thiết.

- **Sáng tạo nội dung và giải trí:** Phân tích cảm xúc có thể được sử dụng để phân tích phản ứng của khán giả đối với các chương trình truyền hình, phim ảnh, hoặc nội dung trực tuyến. Điều này giúp các nhà sáng tạo nội dung hiểu rõ hơn về sở thích và phản ứng của khán giả, từ đó tối ưu hóa nội dung để tăng cường tương tác và sự hài lòng của khán giả.



Hình 1.3. Phân tích cảm xúc giúp nắm bắt nhu cầu của người dùng cuối tốt hơn

Ứng dụng trò chuyện tự động

Ứng dụng trò chuyện tự động và trợ lý ảo là hai trong số những ứng dụng mạnh mẽ của xử lý ngôn ngữ tự nhiên. Những công nghệ này đã cách mạng hóa cách con người tương tác với máy tính và thiết bị thông minh, mở ra kỷ nguyên mới của giao tiếp giữa người và máy. Trò chuyện tự động và trợ lý ảo hiện diện trong nhiều ứng dụng thực tế khác nhau, từ dịch vụ khách hàng đến điều khiển thiết bị thông minh trong nhà.

Ứng dụng trò chuyện tự động là các ứng dụng phần mềm thực hiện các cuộc trò chuyện với con người qua giao diện văn bản hoặc giọng nói. Ứng dụng trò chuyện tự động có thể được lập trình để trả lời các câu hỏi, hỗ trợ thực hiện các nhiệm vụ, hoặc thậm chí tương tác như một người bạn ảo.

Ứng dụng phổ biến nhất của ứng dụng trò chuyện tự động là cung cấp tổng đài tự động. Các doanh nghiệp sử dụng ứng dụng trò chuyện tự động để cung cấp dịch vụ hỗ trợ khách hàng liên tục, giải đáp các câu hỏi thường gặp, hỗ trợ đặt hàng, theo dõi đơn hàng, và giải quyết các vấn đề cơ bản trước khi điều chuyển

đến nhân viên kỹ thuật. Điều này giúp giảm chi phí nhân sự và cải thiện thời gian khách hàng cần bỏ ra khi sử dụng dịch vụ.

1.2. TỔNG QUAN VỀ CÁC MÔ HÌNH HỌC SÂU

1.2.1. Tổng quan về học sâu

Học sâu (Deep learning) là một nhánh của học máy (Machine learning), tập trung vào việc mô hình hóa các trừu tượng cao trong dữ liệu thông qua việc sử dụng các mạng nơ-ron nhân tạo có nhiều lớp. Các lớp này được tổ chức theo kiểu "sâu", tức là gồm nhiều tầng xử lý dữ liệu, mỗi tầng học cách biểu diễn dữ liệu ở mức độ trừu tượng khác nhau.

Một vài đặc điểm của học sâu:

- **Nhiều tầng:** Học sâu sử dụng các mạng nơ-ron sâu, trong đó dữ liệu đầu vào được truyền qua nhiều tầng nơ-ron. Mỗi tầng học một biểu diễn phức tạp hơn của dữ liệu, từ đó cho phép hệ thống nhận dạng các mẫu phức tạp hơn.
- **Học các yếu tố đặc trưng tự động:** Không giống như các phương pháp học máy truyền thống cần thiết kế đặc trưng thủ công, học sâu tự động học các đặc trưng từ dữ liệu thô mà không cần can thiệp nhiều từ con người. Điều này giúp cải thiện hiệu suất trong các nhiệm vụ phức tạp như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, và dịch máy.
- **Hiệu quả với dữ liệu phi cấu trúc:** Học sâu đặc biệt hiệu quả trong việc xử lý các loại dữ liệu phi cấu trúc như hình ảnh, âm thanh, và văn bản. Điều này đã mở ra nhiều ứng dụng trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, và nhận dạng giọng nói.
- **Khả năng mở rộng:** Các mạng nơ-ron sâu có thể được mở rộng để xử lý lượng lớn dữ liệu và các bài toán phức tạp. Khả năng tổng quát hóa cao giúp mô hình học sâu có thể áp dụng cho nhiều nhiệm vụ khác nhau mà không cần phải thay đổi cấu trúc cơ bản của mô hình.

- **Hiệu suất vượt trội:** Học sâu đã cho thấy hiệu suất vượt trội trong nhiều nhiệm vụ so với các phương pháp truyền thống.
- **Yêu cầu tài nguyên phần cứng:** Mặc dù học sâu rất mạnh mẽ, nó cũng đòi hỏi lượng lớn dữ liệu để huấn luyện và tài nguyên tính toán mạnh mẽ. Quá trình huấn luyện các mô hình học sâu thường rất tốn kém về thời gian và công sức.
- **Khả năng tối ưu liên tục:** Các mô hình học sâu có thể liên tục cải thiện và tối ưu hóa thông qua quá trình huấn luyện và điều chỉnh các tham số dựa trên dữ liệu mới. Điều này cho phép các hệ thống học sâu duy trì hiệu suất cao ngay cả khi điều kiện và dữ liệu thay đổi.

1.2.2. Lịch sử phát triển của học sâu

Học sâu đã trải qua quá trình phát triển nhiều thập kỷ từ những ý tưởng ban đầu đến sự bùng nổ mạnh mẽ hiện nay. Dưới đây là tổng quan về quá trình phát triển của lĩnh vực này.

Giai đoạn sơ khai (1940 – 1980)

Giai đoạn từ những năm 1940 đến 1980 là thời kỳ đặt nền móng cho lĩnh vực học sâu, với những ý tưởng ban đầu về mạng nơ-ron nhân tạo và các thuật toán cơ bản.

Năm 1943: Warren McCulloch và Walter Pitts đã giới thiệu mô hình toán học đầu tiên về nơ-ron nhân tạo trong bài báo "A Logical Calculus of Ideas Immanent in Nervous Activity". Mô hình này mô phỏng hoạt động của một nơ-ron sinh học bằng cách sử dụng các hàm số toán học, đánh dấu sự khởi đầu của lý thuyết mạng nơ-ron.

Năm 1958: Frank Rosenblatt, đã phát triển mô hình Perceptron, một loại mạng nơ-ron đơn giản có thể học cách phân loại các mẫu dữ liệu. Ông đã đề xuất một perceptron với ba lớp: Một lớp đầu vào, một lớp ẩn với trọng số ngẫu nhiên không học được và một lớp đầu ra. Sau đó, ông xuất bản một cuốn sách năm 1962 cũng giới thiệu các biến thể và thí nghiệm máy tính, bao gồm một phiên bản với

perceptron bốn lớp trong đó hai lớp cuối cùng có trọng số được học (và do đó là một mạng thần kinh đa lớp đúng nghĩa) Perceptron có thể học từ dữ liệu thông qua một thuật toán đơn giản, nhưng nó chỉ hoạt động tốt với các dữ liệu tuyến tính và không thể giải quyết các vấn đề phức tạp hơn và đặt nền móng cho các thành phần cơ bản của hệ thống học sâu ngày nay.

Năm 1969: Marvin Minsky và Seymour Papert đã chỉ ra những hạn chế của Perceptron trong cuốn sách "Perceptrons", đặc biệt là việc Perceptron không thể giải quyết các vấn đề không tuyến tính như bài toán XOR. Họ cũng chỉ ra rằng việc huấn luyện một mạng nơ-ron nhiều lớp là không khả thi với công nghệ thời đó, điều này dẫn đến sự giảm sút quan tâm đối với nghiên cứu mạng nơ-ron trong một thời gian dài.

Giai đoạn phát triển tiềm năng (1980 – 2010)

Giai đoạn từ những năm 1980 đến 2010 là thời kỳ phát triển tiềm năng của học sâu, khi các ý tưởng và công nghệ quan trọng được phát triển và dần hoàn thiện, tạo nền tảng cho sự bùng nổ của lĩnh vực này sau năm 2010. Trong giai đoạn này, một số mô hình và thuật toán cốt lõi của học sâu đã được đề xuất và cải tiến, dù chưa được ứng dụng rộng rãi do hạn chế về dữ liệu và sức mạnh tính toán.

Các thuật toán được phát triển:

- **Thuật toán lan truyền ngược (backpropagation):** Năm 1986 David Rumelhart, Geoffrey Hinton, và Ronald J. Williams đã phổ biến thuật toán backpropagation, một phương pháp hiệu quả để huấn luyện mạng nơ-ron nhiều lớp. Backpropagation là một cải tiến quan trọng cho phép điều chỉnh trọng số của các nơ-ron trong mạng dựa trên sai số đầu ra, từ đó cải thiện khả năng học của mạng. Mặc dù backpropagation đã tồn tại từ trước, nhưng chính nhờ công trình của họ mà nó mới được công nhận rộng rãi và trở thành tiêu chuẩn cho việc huấn luyện mạng nơ-ron
- **Deep belief networks:** Năm 2006, Geoffrey Hinton và các đồng sự đã giới thiệu Deep belief networks [7], một loại mô hình học sâu cho phép huấn

luyện các mạng nơ-ron với nhiều tầng ẩn một cách hiệu quả bằng cách sử dụng thuật toán "greedy layer-wise training." Deep belief networks đã mở ra khả năng huấn luyện các mạng nơ-ron rất sâu mà không cần quá phụ thuộc vào backpropagation, đồng thời làm giảm thiểu nguy cơ "vanishing gradient." Deep belief networks đã đóng vai trò quan trọng trong việc hồi sinh học sâu sau một thời gian bị lãng quên.

Học sâu trong giai đoạn này gặp những khó khăn. Hầu hết các mô hình học sâu cần lượng dữ liệu lớn để huấn luyện hiệu quả. Trong những năm 1980 và 1990, việc thu thập và xử lý lượng dữ liệu lớn vẫn còn khó khăn, làm hạn chế hiệu quả của các mô hình này. Sức mạnh tính toán cần thiết để huấn luyện các mạng nơ-ron sâu chưa đủ mạnh mẽ trong giai đoạn này. Điều này khiến cho việc huấn luyện các mô hình học sâu trở nên tốn kém và chậm chạp.

Giai đoạn 1980 - 2010 đã chứng kiến những bước tiến quan trọng trong lý thuyết và công nghệ của học sâu. Dù còn nhiều hạn chế, các công trình nghiên cứu trong giai đoạn này đã đặt nền móng cho sự bùng nổ của học sâu sau năm 2010, khi dữ liệu lớn và sức mạnh tính toán trở nên sẵn có hơn. Các ý tưởng và thuật toán được phát triển trong giai đoạn này đã được cải tiến và ứng dụng rộng rãi, dẫn đến những tiến bộ đột phá trong các lĩnh vực như thị giác máy

Bùng nổ mạnh mẽ của học sâu (2010 – hiện nay)

Giai đoạn từ năm 2010 đến nay đã chứng kiến sự bùng nổ mạnh mẽ của học sâu, đưa lĩnh vực này từ các phòng thí nghiệm nghiên cứu trở thành công nghệ cốt lõi trong nhiều ứng dụng thực tiễn. Những tiến bộ trong sức mạnh tính toán, sự sẵn có của dữ liệu lớn, và các đột phá trong kiến trúc mạng nơ-ron đã đóng vai trò quan trọng trong sự phát triển này.

Các ứng dụng của học sâu trong các giai đoạn trước đó tập trung vào các nghiên cứu của các nhà nghiên cứu và chưa hướng đến phần đông người dùng. Tuy nhiên cho tới hiện nay, học sâu đã được ứng dụng sâu rộng và đã có thể tiếp cận bởi đông đảo người dùng như:

- **Xử lý ngôn ngữ tự nhiên:** Dịch máy như Google Translate, GPT,... Những mô hình này có khả năng hiểu và dịch ngữ cảnh tốt hơn, giúp tạo ra các bản dịch tự nhiên và chính xác hơn. Các trợ lý ảo như Siri, Google Assistant,… Các mô hình xử lý ngôn ngữ tự nhiên giúp các trợ lý ảo có thể xử lý ngôn ngữ tự nhiên và cung cấp phản hồi chính xác. Bên cạnh đó Xử lý ngôn ngữ tự nhiên được ứng dụng để phân tích cảm xúc từ văn bản, giúp các doanh nghiệp hiểu rõ hơn về phản hồi của khách hàng thông qua các bài đăng trên mạng xã hội, đánh giá sản phẩm, hoặc phản hồi trực tiếp.
- **Thị giác máy tính:** Học sâu được sử dụng để nhận diện khuôn mặt trong các hệ thống bảo mật, mạng xã hội, và các ứng dụng nhận diện sinh trắc học khác. Được ứng dụng trên trên điện thoại thông minh và hệ thống giám sát an ninh sử dụng các mô hình học sâu để phát hiện và xác thực danh tính người dùng.
- **Y tế và chăm sóc sức khỏe:** Học sâu được sử dụng để phân tích các hình ảnh y tế để phát hiện sớm các bệnh như ung thư, tổn thương nội tạng, hoặc các bất thường khác. Học sâu cũng được sử dụng để phân tích dữ liệu di truyền, giúp phát hiện các gen liên quan đến các bệnh di truyền
- **Thương mại điện tử:** Các nền tảng thương mại điện tử sử dụng học sâu để phân tích hành vi mua sắm của người dùng và gợi ý các sản phẩm phù hợp. Hệ thống gợi ý sản phẩm giúp tăng cường trải nghiệm mua sắm và thúc đẩy doanh thu.

1.2.3. Các mô hình học sâu phổ biến

Mạng nơ-ron tích chập (Convolutional neural network - CNN)

Convolutional neural network (CNN) là một loại mạng nơ-ron sâu đặc biệt được thiết kế để xử lý dữ liệu dạng lưới. CNN được xây dựng dựa trên mô hình tổ chức của hệ thống thị giác của động vật, với khả năng tự động học các đặc trưng từ dữ liệu đầu vào mà không cần sự can thiệp thủ công.

CNN sử dụng các lớp tích chập (Convolutional layers) để trích xuất đặc trưng từ dữ liệu đầu vào. Các lớp này có khả năng phát hiện các mẫu nhỏ trong hình ảnh, chẳng hạn như cạnh, góc, và sau đó kết hợp chúng thành các đặc trưng phức tạp hơn qua các lớp tiếp theo.

Ứng dụng nổi bật:

- AlexNet (2012) [8]: Đây là mô hình CNN đã giành chiến thắng trong cuộc thi ImageNet năm 2012, đánh dấu bước ngoặt quan trọng trong việc áp dụng học sâu vào thị giác máy tính. AlexNet sử dụng nhiều lớp tích chập và gộp, kết hợp với kỹ thuật học sâu trên GPU để đạt được hiệu suất vượt trội.
- Residual networks - ResNet (2015) [9]: ResNet giải quyết vấn đề "Vanishing gradient" trong các mạng sâu bằng cách giới thiệu các khối residual, cho phép thông tin truyền qua các lớp mà không bị suy giảm. ResNet đã đạt được thành công lớn trong các nhiệm vụ thị giác máy tính và tiếp tục được sử dụng rộng rãi trong nhiều ứng dụng thực tế.

Mạng nơ-ron hồi tiếp (Recurrent neural networks - RNN)

Mạng nơ-ron hồi tiếp (RNN) là một loại mạng nơ-ron sâu được thiết kế đặc biệt để xử lý và phân tích dữ liệu tuần tự, chẳng hạn như chuỗi văn bản, âm thanh, hoặc dữ liệu thời gian. Khác với các mô hình mạng nơ-ron truyền thống, RNN có khả năng ghi nhớ thông tin từ các bước trước đó trong chuỗi dữ liệu, giúp chúng hiệu quả hơn trong việc dự đoán các sự kiện dựa trên ngữ cảnh trước đó.

RNN có cấu trúc đặc biệt với các vòng lặp bên trong, cho phép thông tin được lưu truyền từ bước này sang bước khác trong chuỗi dữ liệu, thông tin từ các bước trước được lưu trữ và sử dụng ở các bước tiếp theo. Điều này giúp RNN xử lý tốt các chuỗi dữ liệu có độ dài biến đổi. RNN đặc biệt hiệu quả trong các nhiệm vụ như dịch máy, nhận dạng giọng nói, và dự đoán chuỗi thời gian.

Vấn đề vanishing gradient: Một trong những vấn đề lớn nhất của RNN truyền thống là "vanishing gradient," xảy ra khi thông tin về các bước trước đó dần dần bị mất đi khi chuỗi dữ liệu quá dài, khiến việc học các phụ thuộc dài hạn

trở nên khó khăn. Điều này hạn chế hiệu suất của RNN trong các ứng dụng yêu cầu ghi nhớ thông tin lâu dài.

Các biến thể của RNN:

- **Long short-term memory (LSTM):** Để giải quyết vấn đề vanishing gradient, Sepp Hochreiter và Jürgen Schmidhuber đã phát triển LSTM vào năm 1997 [10]. LSTM là một biến thể của RNN được trang bị các cổng (gates) đặc biệt, cho phép nó kiểm soát luồng thông tin vào, lưu trữ, và truyền đi qua các bước thời gian. Nhờ vậy, LSTM có khả năng học và ghi nhớ các phụ thuộc dài hạn trong chuỗi dữ liệu một cách hiệu quả.
- **Gated recurrent unit (GRU):** GRU là một biến thể khác của RNN, được giới thiệu bởi Kyunghyun Cho và các cộng sự vào năm 2014 [11]. GRU có cấu trúc đơn giản hơn so với LSTM nhưng vẫn giải quyết được vấn đề vanishing gradient. GRU sử dụng ít tham số hơn và thường được chọn khi cần giảm thiểu độ phức tạp tính toán mà không làm giảm đáng kể hiệu suất.

Ứng dụng nổi bật:

RNN được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên, bao gồm các nhiệm vụ như phân loại văn bản, dịch máy, tạo sinh văn bản. Các mô hình như LSTM và GRU đã cải thiện đáng kể hiệu suất trong các ứng dụng này, đặc biệt trong dịch máy (như Google Translate) và tạo sinh văn bản (như GPT của OpenAI).

1.3. KẾT LUẬN CHƯƠNG 1

Chương 1 đã trình bày tổng quan về xử lý ngôn ngữ tự nhiên, đưa ra các nhiệm vụ chính của xử lý ngôn ngữ tự nhiên, và khái quát sự phát triển của xử lý ngôn ngữ tự nhiên qua từng giai đoạn phát triển. Mỗi giai đoạn từ khi phát triển các nền tảng lý thuyết đến khi áp dụng các quy tắc thống kê và sau đó là ứng dụng học sâu cho thấy xử lý ngôn ngữ tự nhiên được hình thành và phát triển trên một nền tảng vững chắc. Các ứng dụng của xử lý ngôn ngữ tự nhiên đã và đang làm

thay đổi nhiều ngành công nghiệp, từ chăm sóc sức khỏe, dịch vụ khách hàng, cho đến phân tích truyền thông xã hội, dịch máy, và nhận dạng giọng nói.

Chương 1 cũng đã trình bày về học sâu, một nhánh của học máy đã tạo ra những bước đột phá lớn trong xử lý ngôn ngữ tự nhiên nhờ khả năng xử lý dữ liệu phức tạp và học các biểu diễn đa tầng từ dữ liệu lớn. Học sâu đã tiến hóa qua nhiều giai đoạn, từ những khái niệm ban đầu về mạng nơ-ron, đến sự bùng nổ mạnh mẽ nhờ sự phát triển của phần cứng tính toán và các kỹ thuật huấn luyện hiệu quả hơn. Các mô hình học sâu đã trở thành trụ cột trong việc giải quyết các bài toán phức tạp trong xử lý ngôn ngữ tự nhiên.

CHƯƠNG 2. BÀI TOÁN PHÂN LOẠI CÂU TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN VÀ MÔ HÌNH HỌC SÂU

2.1. GIỚI THIỆU BÀI TOÁN PHÂN LOẠI CÂU TRONG XỬ LÝ NGÔN NGỮ TỰ NHIÊN

2.1.1. Tổng quan về phân loại câu

Phân loại câu (Sentence classification) là một nhánh của xử lý ngôn ngữ tự nhiên nhằm mục đích gán nhãn hoặc phân loại các câu dựa trên nội dung và ngữ nghĩa của chúng. Điều này có thể bao gồm việc xác định cảm xúc, loại câu hỏi, chủ đề, hoặc các thuộc tính khác liên quan đến ngữ cảnh của câu.

Phân loại câu có vai trò quan trọng trong nhiều ứng dụng thực tế, từ phân tích cảm xúc, hệ thống hỏi đáp, đến việc lọc tin rác và phân loại email. Ví dụ, trong phân tích cảm xúc, phân loại câu được sử dụng để xác định cảm xúc tích cực, tiêu cực, hoặc trung tính trong các bình luận trên mạng xã hội hoặc đánh giá sản phẩm.

2.1.2. Vai trò của phân loại câu trong xử lý ngôn ngữ tự nhiên

Phân loại câu là một trong những nhiệm vụ cốt lõi và quan trọng trong xử lý ngôn ngữ tự nhiên, vì nó có ảnh hưởng trực tiếp đến khả năng hiểu và xử lý ngôn ngữ của máy tính. Vai trò của phân loại câu trong xử lý ngôn ngữ tự nhiên được thể hiện rõ ràng qua các khía cạnh sau:

Tối ưu hóa trải nghiệm người dùng

- **Cải thiện giao tiếp:** Trong các ứng dụng như ứng dụng trò chuyện tự động hoặc trợ lý ảo, phân loại câu giúp hệ thống nhận biết ý định của người dùng dựa trên câu hỏi hoặc câu lệnh mà họ đưa ra. Điều này cho phép hệ thống phản hồi một cách chính xác và hiệu quả, cải thiện trải nghiệm người dùng.
- **Phân loại cảm xúc:** Phân loại câu có thể xác định cảm xúc trong một câu, giúp các hệ thống phân tích phản hồi của khách hàng hoặc bình luận trên mạng xã hội. Điều này hỗ trợ các doanh nghiệp hiểu rõ hơn về cảm xúc và nhu cầu của khách hàng.

Hỗ trợ tìm kiếm thông tin

- **Phân loại truy vấn:** Trong các hệ thống tìm kiếm thông tin, phân loại câu giúp phân loại và xử lý các truy vấn của người dùng một cách chính xác. Ví dụ, một hệ thống có thể phân loại một truy vấn là câu hỏi hoặc yêu cầu tìm kiếm thông tin, từ đó đưa ra kết quả tìm kiếm phù hợp.
- **Trả lời câu hỏi:** Phân loại câu đóng vai trò quan trọng trong các hệ thống trả lời câu hỏi, giúp hệ thống xác định loại câu hỏi (như câu hỏi trực tiếp, câu hỏi yêu cầu xác nhận, hoặc câu hỏi gợi ý) và cung cấp câu trả lời chính xác.

Hỗ trợ trong dịch máy và tóm tắt văn bản

- **Dịch máy:** Phân loại câu giúp cải thiện chất lượng dịch máy bằng cách phân loại đúng ngữ cảnh của câu, chẳng hạn như câu khẳng định, câu nghi vấn, hay câu mệnh lệnh. Điều này giúp hệ thống dịch chính xác hơn và giữ được ý nghĩa ngữ cảnh trong ngôn ngữ đích.
- **Tóm tắt văn bản:** Trong các hệ thống tóm tắt văn bản, phân loại câu có thể giúp xác định các câu quan trọng cần được giữ lại trong bản tóm tắt, từ đó giúp tạo ra các bản tóm tắt có ý nghĩa và phù hợp với mục đích của người dùng.

Phân tích dữ liệu văn bản

- **Phân loại chủ đề:** Phân loại câu giúp hệ thống tự động gán nhãn chủ đề cho các câu hoặc đoạn văn trong tài liệu, hỗ trợ việc phân tích và tổ chức dữ liệu văn bản lớn. Điều này rất hữu ích trong việc sắp xếp tài liệu, phân loại tin tức, hoặc quản lý nội dung trên mạng xã hội.
- **Phân tích xã hội học và hành vi người dùng:** Phân loại câu có thể được sử dụng để phân tích hành vi người dùng trên mạng xã hội, chẳng hạn như phát hiện các xu hướng, chủ đề nóng, và các cuộc thảo luận quan trọng. Điều này giúp các nhà nghiên cứu xã hội học và các doanh nghiệp hiểu rõ hơn về xu hướng xã hội và nhu cầu của người dùng.

Phân loại câu trong xử lý ngôn ngữ tự nhiên đóng vai trò nền tảng trong việc nâng cao hiệu suất và hiệu quả của các hệ thống xử lý ngôn ngữ. Nó giúp máy tính hiểu rõ hơn ngữ cảnh và ý định của người dùng, từ đó cải thiện chất lượng dịch vụ, tăng cường sự tương tác và hỗ trợ việc ra quyết định dựa trên dữ liệu văn bản một cách hiệu quả. Trong bối cảnh học sâu ngày càng phát triển, phân loại câu không chỉ giúp giải quyết các vấn đề kỹ thuật mà còn mở ra nhiều cơ hội ứng dụng mới trong nhiều lĩnh vực khác nhau.

2.2. PHƯƠNG PHÁP PHÂN LOẠI CÂU TRUYỀN THỐNG

2.2.1. Thuật toán Naive Bayes

Naive Bayes là một thuật toán phân loại dựa trên lý thuyết xác suất Bayes với giả định độc lập có điều kiện giữa các đặc trưng. Thuật toán này được gọi là "naive" (ngây thơ) vì nó giả định rằng các đặc trưng của dữ liệu đầu vào đều độc lập với nhau, tức là sự xuất hiện của một đặc trưng không ảnh hưởng đến sự xuất hiện của các đặc trưng khác. Dù giả định này thường không đúng trong thực tế, Naive Bayes vẫn hoạt động rất hiệu quả trong nhiều bài toán phân loại.

Naive Bayes là một thuật toán để xây dựng các bộ phân loại: các mô hình gán nhãn lớp cho các trường hợp cần phân lớp, được biểu diễn dưới dạng các vectơ giá trị thuộc tính, trong đó các nhãn lớp được rút ra từ một tập hợp hữu hạn. Không có một thuật toán duy nhất để huấn luyện các bộ phân loại như vậy, mà là một họ các thuật toán dựa trên một nguyên tắc chung: tất cả các bộ phân loại Naive Bayes đều giả định rằng giá trị của một thuộc tính cụ thể là độc lập với giá trị của bất kỳ thuộc tính nào khác.

Naive Bayes dựa trên công thức Bayes, được biểu diễn như sau:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)}$$

Trong đó:

$P(C|X)$: Xác suất của lớp C khi biết đặc trưng X.

$P(X|C)$: Xác suất của đặc trưng X khi biết lớp C.

$P(C)$: Xác suất tiên nghiệm của lớp C.

$P(X)$: Xác suất của đặc trưng X.

Ưu điểm của thuật toán Naive Bayes:

- **Hiệu quả tính toán:** Naive Bayes rất nhanh và có thể xử lý dữ liệu lớn một cách hiệu quả.
- **Yêu cầu dữ liệu huấn luyện ít:** Thuật toán không yêu cầu nhiều dữ liệu để đạt được độ chính xác tốt.
- **Hiệu quả với dữ liệu có tính độc lập:** Nếu các đặc trưng thực sự độc lập, Naive Bayes có thể đạt độ chính xác cao.

Nhược điểm:

- **Giả định độc lập:** Giả định rằng các đặc trưng là độc lập có thể không đúng trong nhiều trường hợp, dẫn đến kết quả không chính xác.
- **Không linh hoạt:** Naive Bayes có thể không hoạt động tốt với các bài toán có quan hệ phức tạp giữa các đặc trưng.

2.2.2. Mô hình Support Vector Machine

Support Vector Machine là một mô hình mạnh mẽ dùng cho cả phân loại và hồi quy, nhưng chủ yếu được sử dụng cho các bài toán phân loại. SVM hoạt động dựa trên nguyên tắc tìm kiếm siêu phẳng (hyperplane) tối ưu để phân tách dữ liệu thành các lớp khác nhau. Thuật toán này đặc biệt hiệu quả khi dữ liệu có sự phân tách rõ ràng hoặc khi làm việc với các không gian đa chiều.

SVM tìm cách tối đa hóa biên (margin) giữa hai lớp, tức là khoảng cách từ siêu phẳng phân tách đến các điểm dữ liệu gần nhất của mỗi lớp (các điểm này được gọi là các vector hỗ trợ - support vectors).

Siêu phẳng (Hyperplane): Trong không gian hai chiều, siêu phẳng là một đường thẳng. Trong không gian ba chiều, siêu phẳng là một mặt phẳng, và trong không gian nhiều chiều hơn, siêu phẳng là một không gian con với số chiều nhỏ hơn không gian tổng thể là một.

Biên (Margin): Là khoảng cách từ siêu phẳng tới điểm dữ liệu gần nhất của mỗi lớp. Mục tiêu của SVM là tìm siêu phẳng với biên lớn nhất có thể, vì một biên lớn hơn giúp giảm khả năng lỗi phân loại khi gặp dữ liệu mới.

Ưu điểm của Support Vector Machines:

- **Hiệu suất tốt với dữ liệu nhỏ:** SVM hoạt động tốt ngay cả khi số lượng mẫu nhỏ hơn số lượng đặc trưng, đặc biệt trong các không gian có số chiều cao.
- **Hiệu quả trong xử lý dữ liệu phi tuyến:** Nhờ Kernel Trick, SVM có thể xử lý tốt các bài toán phân loại phi tuyến tính.
- **Tránh overfitting:** Với soft margin, SVM có khả năng tổng quát hóa tốt và tránh được hiện tượng overfitting trong nhiều trường hợp.

Nhược điểm của Support Vector Machines:

- **Chi phí tính toán cao:** Đặc biệt khi xử lý với các tập dữ liệu lớn hoặc khi sử dụng các hàm kernel phức tạp, SVM có thể trở nên chậm chạp.
- **Khó khăn trong việc chọn kernel:** Việc lựa chọn hàm kernel và các tham số đi kèm có thể phức tạp và đòi hỏi thử nghiệm nhiều lần.
- **Không dễ mở rộng cho bài toán nhiều lớp:** SVM nguyên bản là một bộ phân loại nhị phân, và việc mở rộng nó cho bài toán phân loại nhiều lớp cần sử dụng các phương pháp bổ sung như "one-vs-one" hoặc "one-vs-all".

2.2.3. Mô hình Bag-of-Words

Bag-of-Words là một trong những kỹ thuật cơ bản và phổ biến nhất trong xử lý ngôn ngữ tự nhiên, được sử dụng để biểu diễn văn bản dưới dạng số liệu để phục vụ cho các thuật toán học máy, bao gồm cả phân loại câu. Bag-of-Words đơn giản hóa việc xử lý văn bản bằng cách biến đổi văn bản thành một tập hợp các từ (bag) mà không quan tâm đến thứ tự của chúng trong câu hay tài liệu.

Trong mô hình Bag-of-Words, một câu hoặc một tài liệu được biểu diễn bằng một vector có kích thước bằng số lượng từ khác nhau trong toàn bộ tập dữ

liệu. Mỗi vị trí trong vector đại diện cho một từ trong từ vựng, và giá trị tại mỗi vị trí là số lần xuất hiện của từ đó trong câu hoặc tài liệu.

Ví dụ, với câu "The cat sat on the mat", từ vựng sẽ bao gồm các từ "the", "cat", "sat", "on", và "mat". Vector Bag-of-Words cho câu này sẽ là [2, 1, 1, 1, 1], trong đó 2 là số lần xuất hiện của từ "the" và các giá trị 1 tương ứng với sự xuất hiện của các từ còn lại.

Quá trình xây dựng mô hình Bag-of-Words bao gồm các bước sau:

Bước 1: Tokenization: Tách văn bản thành các từ riêng biệt (tokens).

Bước 2: Xây dựng từ vựng: Tạo ra một danh sách các từ xuất hiện trong toàn bộ tập dữ liệu.

Bước 3: Vector hóa: Biểu diễn mỗi câu dưới dạng một vector số, trong đó mỗi giá trị thể hiện tần suất xuất hiện của từ tương ứng trong câu.

Ưu điểm của Bag-of-Words:

- **Đơn giản và dễ triển khai:** Bag-of-Words là một mô hình rất đơn giản, dễ hiểu và dễ triển khai.
- **Hiệu quả với văn bản ngắn:** Đối với các câu hoặc đoạn văn ngắn, Bag-of-Words có thể hoạt động hiệu quả trong việc trích xuất các đặc trưng đơn giản từ văn bản.

Nhược điểm của Bag-of-Words:

- **Không nắm bắt được ngữ cảnh:** Bag-of-Words không lưu giữ thông tin về thứ tự từ và ngữ cảnh của các từ trong câu, do đó có thể mất đi ý nghĩa quan trọng.
- **Vấn đề kích thước từ vựng**: Với các tập dữ liệu lớn, từ vựng có thể trở nên rất lớn, dẫn đến các vector rất dài và thưa, làm tăng chi phí tính toán và giảm hiệu quả của các thuật toán.
- **Không phân biệt được các từ đồng nghĩa:** Các từ đồng nghĩa hoặc những từ có liên quan ngữ nghĩa không được xử lý khác biệt, dẫn đến việc mất thông tin quan trọng trong quá trình phân loại.

2.2.4. TF-IDF (Term frequency-inverse document frequency)

TF-IDF là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên để biểu diễn văn bản dưới dạng số liệu nhằm phục vụ cho các thuật toán học máy, bao gồm cả phân loại câu. TF-IDF cải thiện mô hình Bag-of-Words bằng cách xem xét không chỉ tần suất xuất hiện của một từ trong một tài liệu cụ thể mà còn tần suất xuất hiện của từ đó trong toàn bộ tập dữ liệu. Điều này giúp TF-IDF phản ánh tốt hơn tầm quan trọng của từ trong văn bản, đặc biệt là trong các bài toán phân loại câu.

TF-IDF là sự kết hợp của hai thành phần chính: Term Frequency (TF) và Inverse document frequency (IDF):

- **Term frequency (TF):** Đây là tần suất xuất hiện của một từ trong một tài liệu, phản ánh mức độ quan trọng của từ đó trong tài liệu cụ thể.
- **Inverse document frequency (IDF):** Đây là một thước đo nhằm giảm tầm quan trọng của các từ xuất hiện phổ biến trong nhiều tài liệu, vì các từ này thường ít có giá trị phân biệt.

TF-IDF: Là tích của TF và IDF, phản ánh tầm quan trọng của một từ trong một tài liệu cụ thể, có cân nhắc đến sự xuất hiện của từ đó trong toàn bộ tập dữ liệu

Giả sử ta có hai câu:

Câu 1: "The cat sat on the mat."

Câu 2: "The dog sat on the log."

Sau khi tính toán TF-IDF, ta có thể nhận thấy rằng các từ phổ biến như "the", "on", "sat" sẽ có giá trị TF-IDF thấp hơn so với các từ "cat", "dog", "mat", "log". Điều này là do các từ phổ biến xuất hiện ở cả hai câu và do đó ít có giá trị phân biệt. Ngược lại, các từ như "cat" và "dog" sẽ có giá trị TF-IDF cao hơn, vì chúng chỉ xuất hiện trong một câu, giúp tăng khả năng phân biệt giữa các câu.

Ưu điểm của TF-IDF:

- **Giảm tầm quan trọng của các từ phổ biến:** TF-IDF giúp giảm thiểu ảnh hưởng của các từ phổ biến nhưng không mang nhiều thông tin, như các từ dừng.
- **Phản ánh tốt hơn tầm quan trọng của từ:** Bằng cách kết hợp TF và IDF, TF-IDF giúp xác định các từ có giá trị phân biệt cao trong tập dữ liệu.

Nhược điểm của TF-IDF:

- **Không nắm bắt được ngữ cảnh:** Giống như Bag-of-Words, TF-IDF không nắm bắt được ngữ cảnh và thứ tự từ trong câu.
- **Không phân biệt được từ đồng nghĩa:** TF-IDF không xử lý tốt các từ đồng nghĩa hoặc các từ có ngữ nghĩa tương tự.

2.2.5. N-grams

N-grams là một kỹ thuật phổ biến trong xử lý ngôn ngữ tự nhiên, được sử dụng để trích xuất và biểu diễn các mẫu ngôn ngữ trong văn bản. Trong N-grams, "N" biểu thị số lượng từ liên tiếp được nhóm lại với nhau để tạo thành một chuỗi. Ví dụ, với N=1, chúng ta có "unigram", với N=2, chúng ta có "bigram", và với N=3, chúng ta có "trigram". N-grams được sử dụng rộng rãi trong phân loại câu vì chúng cho phép nắm bắt được ngữ cảnh cục bộ và mối quan hệ giữa các từ liên tiếp trong văn bản.

N-grams được sử dụng trong phân loại câu để tạo ra các đặc trưng ngữ nghĩa từ văn bản. Khi sử dụng N-grams, các câu được chuyển đổi thành các vector đặc trưng, trong đó mỗi phần tử đại diện cho tần suất xuất hiện của một N-gram cụ thể. Các thuật toán học máy sau đó có thể sử dụng các vector này để phân loại câu.

Cách tạo N-grams từ một câu rất đơn giản. Ta chỉ cần trượt một cửa sổ có kích thước N qua câu, và tại mỗi bước, ghi lại chuỗi N từ liên tiếp.

Ví dụ, xét câu: "The cat sat on the mat". Với:

Unigram (N=1): "The", "cat", "sat", "on", "the", "mat".

Bigram (N=2): "The cat", "cat sat", "sat on", "on the", "the mat".

Trigram (N=3): "The cat sat", "cat sat on", "sat on the", "on the mat".

Ưu điểm của N-grams:

- **Giữ lại ngữ cảnh cục bộ:** N-grams giữ lại mối quan hệ giữa các từ liên tiếp, giúp cải thiện độ chính xác trong việc nắm bắt ý nghĩa của câu.
- **Đơn giản và hiệu quả:** Dễ dàng triển khai và sử dụng với nhiều thuật toán học máy, giúp nâng cao hiệu suất phân loại.
- **Khả năng phát hiện các cụm từ quan trọng:** Các cụm từ như "not bad", "very good" có thể mang ý nghĩa khác biệt và được N-grams phát hiện hiệu quả.

Nhược điểm của N-grams:

- **Kích thước vector lớn:** Khi giá trị N tăng lên, số lượng N-grams cũng tăng lên theo cấp số nhân, dẫn đến vector đặc trưng rất lớn và thưa (sparse).
- **Không nắm bắt được ngữ cảnh dài hạn:** N-grams chỉ nắm bắt được mối quan hệ trong một chuỗi N từ liên tiếp và không thể nắm bắt được ngữ cảnh toàn cục của câu.
- **Khả năng khái quát hóa kém:** Các cụm từ tương tự nhưng có thứ tự từ khác nhau (như "good weather" và "weather good") được coi là khác nhau trong mô hình N-grams.

2.3. PHƯƠNG PHÁP PHÂN LOẠI CÂU DỰA TRÊN HỌC SÂU

2.3.1. Mạng nơ-ron tích chập (Convolutional neural networks - CNN)

2.3.1.1. Khái niệm mạng nơ-ron tích chập

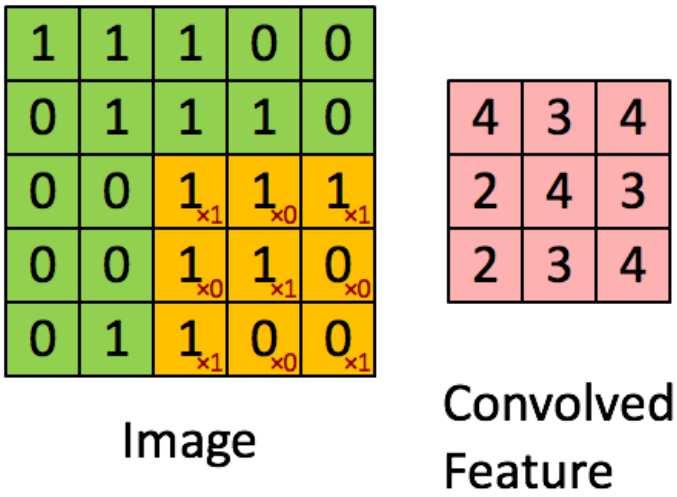
Mạng nơ-ron tích chập (Convolutional neural networks - CNN) là một mô hình học sâu mạnh mẽ, ban đầu CNN chủ yếu được sử dụng trong thị giác máy tính. CNN đã tạo nên những đột phá lớn trong phân loại hình ảnh và là cốt lõi của hầu hết các hệ thống thị giác máy tính ngày nay, từ việc gắn thẻ ảnh tự động của Facebook đến xe tự lái. Tuy nhiên trong những năm gần đây, CNN đã được điều chỉnh thành công để áp dụng trong nhiều tác vụ xử lý ngôn ngữ tự nhiên, bao gồm phân loại câu [12]. CNN trong phân loại câu có khả năng tự động trích xuất các

đặc trưng từ câu, giúp hệ thống đạt được hiệu suất cao trong việc phân loại các loại câu khác nhau.

2.3.1.2. Phép toán tích chập (Convolution)

Phép toán tích chập (Convolution) là bước quan trọng trong hoạt động của Mạng nơ-ron tích chập (CNN), được thiết kế để tự động trích xuất các đặc trưng từ dữ liệu đầu vào, chẳng hạn như hình ảnh hoặc chuỗi văn bản. Trong bối cảnh của CNN, phép toán này giúp phát hiện các đặc trưng cục bộ trong dữ liệu, chẳng hạn như các cạnh, góc, hoặc các mẫu phức tạp khác, bằng cách sử dụng các bộ lọc (filters) di chuyển qua toàn bộ đầu vào.

Quá trình: Tại mỗi bước, bộ lọc "trượt" qua các vị trí khác nhau của dữ liệu đầu vào. Tại mỗi vị trí, nó thực hiện phép nhân từng phần tử giữa bộ lọc và vùng tương ứng của dữ liệu đầu vào, sau đó cộng tất cả các giá trị lại (phép nhân chập).



Hình 2.1. Phép toán tích chập

2.3.1.3. Các hàm kích hoạt phổ biến

Hàm Sigmoid

Công thức: $f(x) = \frac{1}{1+e^{-x}}$

Hàm Sigmoid là một trong những hàm kích hoạt phổ biến trong mạng nơ-ron, đặc biệt trong các mô hình học máy trước đây như hồi quy logistic và các mạng nơ-ron đơn giản. Nó có vai trò biến đổi đầu vào thành một giá trị xác suất

nằm trong khoảng (0, 1), giúp mô hình xử lý và phân loại dữ liệu, đặc biệt là trong các bài toán phân loại nhị phân.

Ưu điểm của hàm Sigmoid:

- **Giá trị đầu ra dễ hiểu:** Đầu ra của hàm Sigmoid dễ diễn giải dưới dạng xác suất, điều này rất hữu ích trong các mô hình phân loại.
- **Tính liên tục và vi phân:** Hàm Sigmoid là liên tục và có đạo hàm rõ ràng, giúp việc tính toán gradient trong quá trình huấn luyện mô hình trở nên dễ dàng.

Hạn chế của hàm Sigmoid:

- **Vanishing gradient (Biến mất gradient):** Khi giá trị của x quá lớn hoặc quá nhỏ, gradient (đạo hàm) của hàm Sigmoid trở nên rất nhỏ (gần bằng 0). Điều này gây khó khăn trong quá trình huấn luyện các mạng nơ-ron sâu, vì các trọng số không được cập nhật hiệu quả, làm giảm tốc độ hội tụ của mô hình.
- **Độ bão hòa:** Khi đầu vào lớn hoặc nhỏ hơn một giá trị nhất định, đầu ra của hàm Sigmoid sẽ bão hòa ở gần 0 hoặc 1. Điều này làm giảm sự nhạy cảm của mô hình đối với thay đổi trong các giá trị đầu vào lớn, khiến việc học các đặc trưng phức tạp trở nên khó khăn hơn.
- **Giá trị đầu ra không đối xứng:** Hàm Sigmoid có giá trị đầu ra nằm trong khoảng (0, 1), điều này có thể gây ra vấn đề khi giá trị đầu ra cần phải đối xứng quanh gốc tọa độ. Hàm Tanh được sử dụng nhiều hơn trong các trường hợp này vì giá trị đầu ra của nó nằm trong khoảng (-1, 1).

Hàm ReLU

Công thức: $f(x) = \max(0, x)$

ReLU [13] (Rectified Linear Unit) là một hàm kích hoạt phổ biến được sử dụng rộng rãi trong các mạng nơ-ron hiện đại, đặc biệt là trong các mô hình học sâu như Convolutional Neural Networks (CNN). ReLU đã thay thế các hàm kích

hoạt truyền thống như Sigmoid và tanh trong nhiều ứng dụng do hiệu suất vượt trội của nó.

Ưu điểm của hàm ReLU:

- **Giảm thiểu vấn đề vanishing gradient:** Một trong những vấn đề lớn của các hàm kích hoạt như Sigmoid và tanh là vanishing gradient, tức là gradient trở nên rất nhỏ trong quá trình huấn luyện, dẫn đến việc cập nhật trọng số không hiệu quả. ReLU giải quyết vấn đề này bằng cách giữ cho gradient bằng 1 đối với các giá trị dương, giúp mô hình hội tụ nhanh hơn trong các mạng sâu.
- **Đơn giản và tính toán nhanh:** Hàm ReLU rất đơn giản về mặt tính toán, chỉ yêu cầu so sánh và trả về giá trị lớn hơn giữa 0 và giá trị đầu vào. Điều này làm cho nó tính toán nhanh hơn nhiều so với các hàm kích hoạt khác như Sigmoid và tanh, vốn đòi hỏi các phép tính lũy thừa và chia.
- **Thúc đẩy sự thưa thớt:** ReLU có khả năng đưa một phần đầu ra của nơ-ron về 0, điều này làm cho các mạng nơ-ron trở nên thưa thớt hơn. Sự thưa thớt này giúp tăng cường tính hiệu quả của mô hình, giảm thiểu sự phụ thuộc giữa các nơ-ron và giảm nguy cơ overfitting.

Hạn chế của hàm ReLU:

- **Dying ReLU:** Một vấn đề tiềm ẩn của ReLU là hiện tượng "Dying ReLU," xảy ra khi các nơ-ron có đầu ra bằng 0 trong một khoảng thời gian dài và không được cập nhật trọng số nữa, dẫn đến việc nơ-ron đó trở nên không hoạt động (không kích hoạt). Điều này có thể xảy ra nếu đầu vào của nơ-ron luôn là một số âm, hoặc nếu học suất quá lớn khiến nơ-ron không thể phục hồi từ trạng thái này.
- **Không đối xứng:** ReLU không đối xứng xung quanh gốc tọa độ (0), do đó không phải lúc nào cũng phù hợp cho các bài toán yêu cầu sự đối xứng giữa các giá trị âm và dương.

2.3.1.4. Kiến trúc của mạng nơ-ron tích chập

Dữ liệu đầu vào: Trong xử lý ngôn ngữ tự nhiên, dữ liệu đầu vào của CNN thường là một câu hoặc đoạn văn bản, được biểu diễn dưới dạng ma trận, trong đó mỗi từ hoặc token trong câu được mã hóa dưới dạng một vector nhúng (embedding vector). Những vector này thường được tiền huấn luyện từ các mô hình, hoặc sử dụng trực tiếp từ các mô hình ngôn ngữ. Kết quả là một ma trận hai chiều đại diện cho câu, trong đó mỗi hàng là một vector nhúng của một từ [14].

Lớp tích chập (Convolutional layer)

Chức năng: Lớp tích chập trong CNN quét qua các đoạn của câu (n-grams) bằng cách sử dụng các bộ lọc (filters) với kích thước cố định. Các bộ lọc này có thể được coi như các "cửa sổ trượt" di chuyển dọc theo chuỗi dữ liệu văn bản để trích xuất các đặc trưng cục bộ, chẳng hạn như các cụm từ hoặc ngữ cảnh liên quan.

Một bộ lọc là một ma trận nhỏ có kích thước cố định, chẳng hạn như 3 x 3, 5 x 5 hoặc thậm chí là 1 x n đối với dữ liệu chuỗi như văn bản. Bộ lọc này di chuyển trên toàn bộ dữ liệu đầu vào để thực hiện phép tích chập. Bộ lọc chứa các trọng số (weights) được học trong quá trình huấn luyện. Các trọng số này được nhân với các giá trị tương ứng trong một vùng nhỏ của dữ liệu đầu vào và tổng hợp để tạo ra giá trị đầu ra.

Stride: Đây là số bước mà bộ lọc di chuyển sau mỗi lần tích chập. Nếu stride lớn hơn 1, bộ lọc sẽ bỏ qua một số phần tử trong dữ liệu đầu vào, làm giảm kích thước của bản đồ đặc trưng.

Padding: Để kiểm soát kích thước của bản đồ đặc trưng, người ta thường thêm các giá trị 0 xung quanh biên của dữ liệu đầu vào (gọi là zero-padding). Điều này giúp giữ nguyên kích thước của bản đồ đặc trưng so với kích thước ban đầu của dữ liệu đầu vào.

Vai trò của lớp tích chập trong CNN:

- **Trích xuất đặc trưng cục bộ:** Lớp tích chập có khả năng phát hiện và trích xuất các đặc trưng cục bộ trong dữ liệu, chẳng hạn như các cạnh, góc, hoặc mẫu ngữ nghĩa trong một câu. Những đặc trưng này sau đó được sử dụng ở các lớp tiếp theo để xây dựng các đặc trưng phức tạp hơn.
- **Tính bất biến đối với biến đổi cục bộ:** CNN có khả năng nhận dạng các đặc trưng quan trọng ngay cả khi chúng bị dịch chuyển hoặc biến dạng nhẹ trong dữ liệu đầu vào. Lớp tích chập giúp duy trì tính bất biến này, làm cho mô hình mạnh mẽ hơn trong việc nhận dạng các đối tượng hoặc đặc trưng dưới nhiều biến đổi khác nhau.
- **Chia sẻ tham số:** Bộ lọc trong lớp tích chập chia sẻ cùng một bộ trọng số trên toàn bộ dữ liệu đầu vào, giúp giảm đáng kể số lượng tham số cần học so với các lớp kết nối đầy đủ (fully connected layers). Điều này giúp giảm thiểu nguy cơ overfitting và cải thiện hiệu suất tính toán.

Lớp kích hoạt (Activation layer)

Chức năng: Sau lớp tích chập, các giá trị trong bản đồ đặc trưng thường được đưa qua một hàm kích hoạt như ReLU để giới thiệu tính phi tuyến vào mô hình. Điều này giúp CNN học được các mối quan hệ phi tuyến giữa các từ trong câu, cải thiện khả năng phân loại.

Vai trò của lớp kích hoạt:

- **Tính phi tuyến:** Trong một mạng nơ-ron, nếu không có lớp kích hoạt, mạng sẽ chỉ là một tổ hợp tuyến tính của các lớp, bất kể số lượng các lớp ẩn có trong mạng. Điều này sẽ hạn chế khả năng của mạng trong việc học các mối quan hệ phức tạp. Lớp kích hoạt giới thiệu phi tuyến tính, giúp mạng có thể học được các hàm ánh xạ phi tuyến giữa đầu vào và đầu ra.
- **Kích hoạt nơ-ron:** Lớp kích hoạt quyết định giá trị nào từ một nơ-ron sẽ được truyền đi các lớp tiếp theo và giá trị nào sẽ bị loại bỏ. Điều này tương tự như việc quyết định xem nơ-ron có "bắn" tín hiệu hay không.

Lớp gộp (Pooling Layer)

Max Pooling: Thường được sử dụng trong CNN để chọn ra các đặc trưng quan trọng nhất từ các bản đồ đặc trưng. Lớp này giúp giảm kích thước của các bản đồ đặc trưng, từ đó giảm số lượng tham số và khối lượng tính toán, đồng thời tăng tính bất biến của mô hình đối với các biến đổi nhỏ trong dữ liệu.

Sau mỗi lớp tích chập, một lớp max pooling được thêm vào để giảm kích thước dữ liệu đầu ra, nhưng vẫn giữ lại các đặc trưng quan trọng nhất. Max pooling giúp chọn ra giá trị lớn nhất từ mỗi vùng tích chập, do đó giúp giảm độ phức tạp của mô hình và ngăn hiện tượng quá khớp. Lớp gộp quan trọng vì những lý do sau:

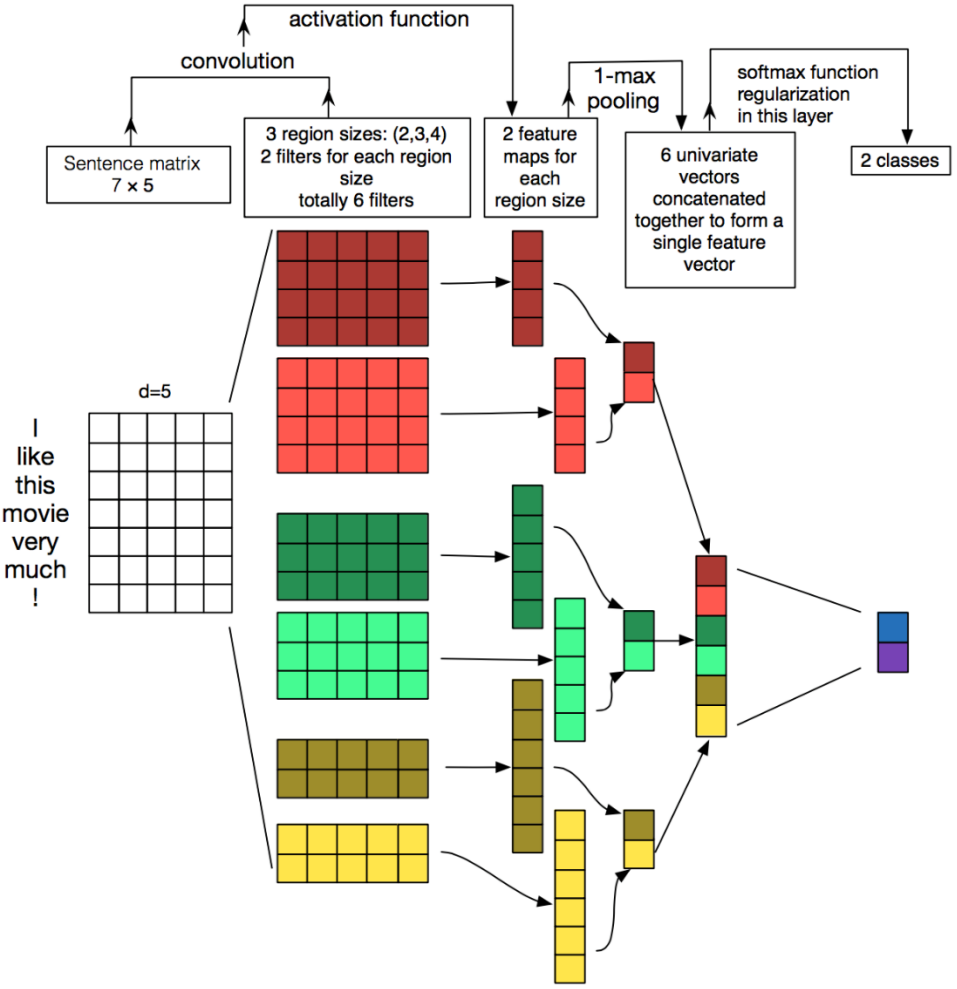
- **Giảm độ phức tạp:** Max pooling giúp giảm kích thước dữ liệu sau mỗi lớp tích chập mà vẫn giữ lại thông tin quan trọng nhất. Điều này giúp tăng hiệu quả tính toán và giảm thiểu việc mô hình học quá nhiều thông tin không cần thiết.
- **Tập trung vào đặc trưng mạnh nhất:** Max pooling đảm bảo rằng chỉ những đặc trưng mạnh nhất được giữ lại, giúp mô hình tập trung vào những gì quan trọng nhất cho quá trình dự đoán.

Lớp kết nối đầy đủ (Fully connected layer)

Chức năng: Sau khi qua lớp gộp, các bản đồ đặc trưng được làm phẳng và đưa qua các lớp kết nối đầy đủ để thực hiện phân loại. Lớp cuối cùng thường sử dụng hàm softmax để tính toán xác suất cho từng lớp đầu ra, giúp xác định loại câu mà hệ thống nhận diện.

CNN về cơ bản là nhiều lớp tích chập với các hàm kích hoạt phi tuyến tính như ReLU hoặc tanh áp dụng cho kết quả. Trong một mạng nơ-ron truyền thống, mỗi nơ-ron đầu vào được kết nối với mỗi nơ-ron đầu ra ở lớp tiếp theo. Điều đó cũng được gọi là lớp kết nối đầy đủ. Trong CNN, thay vào đó ta sử dụng tích chập trên lớp đầu vào để tính toán đầu ra. Điều này dẫn đến các kết nối cục bộ, trong đó mỗi vùng của đầu vào được kết nối với một nơ-ron ở đầu ra. Mỗi lớp áp dụng các bộ lọc khác nhau, thường là hàng trăm hoặc hàng nghìn và kết hợp kết quả

của chúng. Trong giai đoạn huấn luyện, CNN tự động học các giá trị của bộ lọc dựa trên nhiệm vụ người dùng mong muốn thực hiện.



Hình 2.2. Ví dụ về kiến trúc của một mạng nơ-ron tích chập

Hình ảnh trên mô tả ba kích thước vùng lọc: 2, 3 và 4, mỗi kích thước có 2 bộ lọc. Mỗi bộ lọc thực hiện tích chập trên ma trận câu và tạo ra các bản đồ đặc trưng (có độ dài biến đổi). Sau đó, thao tác pooling 1-max được thực hiện trên mỗi bản đồ, tức là số lớn nhất từ mỗi bản đồ đặc trưng được ghi lại. Do đó, một vectơ đặc trưng đơn biến được tạo ra từ tất cả sáu bản đồ, và 6 đặc trưng này được nối lại để tạo thành một vectơ đặc trưng cho lớp cuối cùng. Lớp softmax cuối cùng nhận vector đặc trưng này làm đầu vào và sử dụng nó để phân loại câu; ở đây chúng tôi giả sử phân loại nhị phân và do đó mô tả hai trạng thái đầu ra có thể.

2.3.2. Kỹ thuật Word2Vec

Word2Vec là một kỹ thuật học từ nhúng (word embedding) mạnh mẽ, được phát triển bởi nhóm nghiên cứu tại Google vào năm 2013 dưới sự dẫn dắt của Tomas Mikolov. Mục tiêu của Word2Vec là chuyển đổi các từ trong ngôn ngữ tự nhiên thành các vector số, trong đó các từ có nghĩa tương tự sẽ có vector gần nhau trong không gian vector [4]. Kỹ thuật này đã mở ra nhiều cơ hội mới trong xử lý ngôn ngữ tự nhiên nhờ khả năng biểu diễn ngữ nghĩa của từ một cách toán học.

Khi sử dụng đầu vào là văn bản cho mạng nơ-ron tích chập Convolutional Neural Networks - CNN, việc sử dụng Word2Vec là cần thiết vì các lý do sau:

- **CNN chỉ xử lý các giá trị số:** CNN là các mô hình học máy được thiết kế để xử lý các đầu vào dưới dạng ma trận số (như hình ảnh hoặc âm thanh). Do đó, khi đầu vào là văn bản, các từ cần được chuyển đổi thành các vector số để mô hình có thể xử lý được. Word2Vec thực hiện việc này bằng cách chuyển đổi mỗi từ trong văn bản thành một vector số trong không gian vector có chiều thấp hơn so với one-hot encoding truyền thống.
- **Biểu diễn ngữ nghĩa:** Word2Vec không chỉ đơn thuần chuyển từ thành số, mà nó còn đảm bảo rằng các từ có ý nghĩa tương tự nhau sẽ có các vector gần nhau trong không gian vector. Điều này giúp CNN có thể hiểu và xử lý ngữ nghĩa của từ trong văn bản. Ví dụ, các từ "vua" và "nữ hoàng" sẽ có các vector gần nhau, phản ánh mối quan hệ ngữ nghĩa giữa chúng.
- **Giảm chiều dữ liệu:** So với phương pháp one-hot encoding, nơi mỗi từ được biểu diễn bằng một vector có kích thước bằng kích thước của từ vựng, Word2Vec giúp giảm chiều dữ liệu đáng kể bằng cách mã hóa các từ thành các vector có kích thước nhỏ hơn nhiều, thường là 100-300 chiều. Điều này giúp giảm tải tính toán và tài nguyên cần thiết cho quá trình huấn luyện CNN.
- **Giảm tách biệt ngữ cảnh:** Một hạn chế lớn của one-hot encoding là mỗi từ được coi là một đơn vị độc lập, không liên quan đến các từ khác.

Word2Vec khắc phục điều này bằng cách tạo ra các vector từ nhúng mà trong đó ngữ nghĩa của từ được giữ lại và thể hiện mối quan hệ ngữ nghĩa với các từ khác, giúp CNN xử lý tốt hơn ngữ cảnh và ý nghĩa của các từ trong câu.

- **Tăng hiệu quả huấn luyện:** Việc sử dụng Word2Vec giúp mô hình CNN huấn luyện nhanh hơn và hiệu quả hơn, bởi vì các từ đã được biểu diễn dưới dạng vector có ý nghĩa, giảm bớt gánh nặng cho CNN trong việc tìm kiếm mối quan hệ giữa các từ từ đầu.

Word2Vec có hai mô hình chính để huấn luyện các vector từ: Continuous Bag of Words và Skip-Gram.

Continuous Bag of Words

Mục tiêu: Continuous Bag of Words dự đoán từ mục tiêu dựa trên ngữ cảnh xung quanh nó. Ngữ cảnh ở đây là các từ trước và sau từ mục tiêu trong một cửa sổ ngữ cảnh cố định.

Cách hoạt động: Continuous Bag of Words tính trung bình các vector của các từ ngữ cảnh rồi sử dụng vector trung bình này để dự đoán từ mục tiêu. Đây là một mô hình đơn giản và hiệu quả khi xử lý dữ liệu lớn, đặc biệt là khi các từ ngữ cảnh xuất hiện thường xuyên [4].

Skip-Gram

Mục tiêu: Skip-Gram làm ngược lại với Continuous Bag of Words. Nó sử dụng một từ trung tâm để dự đoán các từ ngữ cảnh xung quanh.

Cách hoạt động: Skip-Gram huấn luyện mô hình để tối đa hóa xác suất của từ ngữ cảnh dựa trên từ mục tiêu. Mô hình này hoạt động tốt khi xử lý các từ ít gặp, vì nó học được nhiều từ các từ ngữ cảnh hiếm.

Quá trình huấn luyện Word2Vec

- **Input layer:** Đầu vào của Word2Vec là một từ được mã hóa dưới dạng one-hot vector. Một từ trong một từ điển (vocabulary) có kích thước V sẽ được

biểu diễn bằng một vector V-chiều với chỉ một phần tử là 1 và các phần tử còn lại là 0.

- **Hidden layer:** Từ vector one-hot, một ma trận trọng số sẽ biến đổi vector này thành một vector nhúng (embedding vector) có kích thước nhỏ hơn nhiều, thường là 100-300 chiều.
- **Output layer:** Lớp đầu ra của Word2Vec là một lớp softmax, cung cấp xác suất của từng từ trong từ điển, dựa trên vector nhúng của từ đã cho.
- **Huấn luyện:** Mô hình Word2Vec được huấn luyện bằng cách tối ưu hóa một hàm mất mát (loss function) để điều chỉnh các vector từ sao cho các từ ngữ cảnh có liên quan sẽ có vector gần nhau trong không gian vector. Quá trình này thường sử dụng các kỹ thuật như Negative Sampling hoặc Hierarchical Softmax để giảm chi phí tính toán.

Các ứng dụng của Word2Vec

- **Phân tích ngữ nghĩa:** Word2Vec có thể nắm bắt các mối quan hệ ngữ nghĩa giữa các từ. Ví dụ, nó có thể nắm bắt các mối quan hệ như "vua" và "nữ hoàng" hoặc "Pháp" và "Paris" bằng cách biểu diễn chúng gần nhau trong không gian vector.
- **Hệ thống gợi ý:** Word2Vec được sử dụng trong các hệ thống gợi ý để hiểu rõ hơn về sở thích của người dùng thông qua phân tích văn bản, giúp tạo ra các gợi ý chính xác hơn.
- **Tìm kiếm thông tin:** Kỹ thuật này cũng được sử dụng để cải thiện các hệ thống tìm kiếm thông tin bằng cách hiểu ngữ nghĩa của các từ khóa tìm kiếm và trả về các kết quả liên quan chặt chẽ hơn.
- **Mô hình hóa ngôn ngữ:** Word2Vec có thể được sử dụng để cải thiện các mô hình ngôn ngữ trong việc dự đoán từ tiếp theo trong chuỗi văn bản.

Ưu điểm

- **Hiệu quả:** Word2Vec rất hiệu quả trong việc huấn luyện trên các tập dữ liệu lớn và có thể tạo ra các vector từ chất lượng cao.

- **Ngữ nghĩa:** Word2Vec có khả năng nắm bắt các mối quan hệ ngữ nghĩa giữa các từ, điều mà các phương pháp truyền thống khó làm được.

Hạn chế

- **Độc lập ngữ cảnh:** Word2Vec chỉ tạo ra một vector duy nhất cho mỗi từ, không phân biệt ngữ cảnh khác nhau của từ đó. Ví dụ, từ "bank" trong "river bank" và "financial bank" có cùng một vector, dù ngữ nghĩa của chúng khác nhau.
- **Yêu cầu tài nguyên:** Để huấn luyện Word2Vec hiệu quả, cần một lượng lớn dữ liệu văn bản và tài nguyên tính toán.

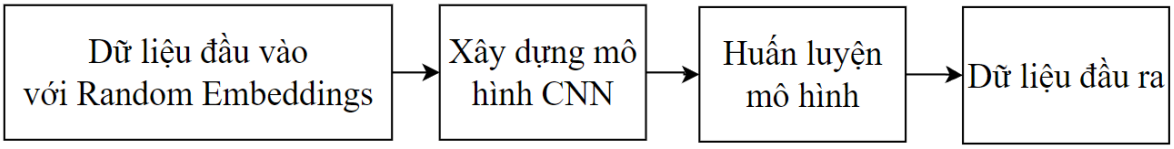
2.4. TRIỂN KHAI CÁC MÔ HÌNH PHÂN LOẠI CÂU DỰA TRÊN HỌC SÂU

2.4.1. Mô hình 1: CNN với Random embeddings

Mục tiêu của việc triển khai mô hình: Triển khai một mô hình phân loại câu. Đầu vào của mô hình là các câu văn bản đã được chuyển đổi thành vector ngẫu nhiên (Random embeddings). Đầu ra là nhãn phân loại của câu.

Trong mô hình CNN, các từ trong câu không thể được đưa trực tiếp vào mô hình vì các mô hình học sâu cần dữ liệu đầu vào dạng số. Do đó, các từ được chuyển đổi thành các vector số thông qua kỹ thuật embeddings.

Quy trình để triển khai mô hình như sau:



Hình 2.3. Quy trình triển khai mô hình CNN với Random embeddings

Xử lý dữ liệu đầu vào

Trong mô hình này, các từ trong tập dữ liệu được biểu diễn bằng các vector ngẫu nhiên với số chiều cố định (200 chiều). Mỗi từ sẽ được ánh xạ thành một vector số học có kích thước cố định. Mỗi từ sẽ được biểu diễn dưới dạng một vector có chiều thấp thay vì hàng ngàn hoặc hàng triệu chiều như các phương

pháp mã hóa one-hot. Khi học embeddings, các từ có nghĩa gần nhau trong ngữ cảnh sẽ có vector tương đối gần nhau trong không gian vector. Với random embeddings, vector khởi tạo ban đầu hoàn toàn ngẫu nhiên, và các vector này sẽ được điều chỉnh thông qua quá trình huấn luyện để trở thành các đại diện tốt hơn cho ngữ nghĩa của từ trong tập dữ liệu. Quy trình xử lý dữ liệu đầu vào bao gồm:

- **Làm sạch văn bản:** Văn bản được loại bỏ các ký tự không cần thiết, chuyển thành chữ thường, và loại bỏ các ký tự đặc biệt như dấu câu và URL.
- **Tokenization (Chia văn bản thành các từ):** Các câu trong văn bản được chia thành các từ hoặc token.
- **Biểu diễn văn bản dưới dạng số:** Sử dụng bộ tokenizer, mỗi từ trong câu được ánh xạ thành một số nguyên đại diện cho từ đó.
- **Padding và chuẩn hóa độ dài:** Các câu có độ dài khác nhau được chuẩn hóa về cùng một độ dài bằng cách thêm các giá trị 0 (padding) vào cuối các câu ngắn hơn. Điều này giúp đảm bảo mọi câu đều có cùng kích thước đầu vào.

Lớp tích chập (Convolutional layer)

CNN trong xử lý văn bản sử dụng các lớp tích chập (Convolutional layers) để học các đặc trưng cục bộ từ câu. Mỗi lớp tích chập áp dụng một bộ lọc (filter) lên các chuỗi vector từ để tìm kiếm các mẫu đặc trưng.

Bộ lọc: Trong mô hình này sử dụng ba lớp tích chập với các bộ lọc có kích thước khác nhau: 3, 4 và 5 từ. Các bộ lọc này trượt qua các đoạn văn bản để học các đặc trưng từ các cụm từ có độ dài tương ứng.

- Filter size = 3 sẽ học các mẫu đặc trưng từ các cụm 3 từ.
- Filter size = 4 sẽ học từ các cụm 4 từ, và tương tự cho filter size = 5.

Tích chập: Bộ lọc sẽ quét qua văn bản và thực hiện phép tích chập (convolution) để tạo ra một bản đồ đặc trưng (feature map), giúp mô hình nắm bắt được thông tin quan trọng từ các cụm từ nhỏ trong câu.

Lớp kích hoạt (Activation Layer)

Sau khi thực hiện tích chập, đầu ra sẽ được đưa qua lớp kích hoạt (Activation layer) để tạo ra tính phi tuyến, giúp mô hình có khả năng học các quan hệ phức tạp giữa các từ. Mô hình này sử dụng ReLU (Rectified linear unit) làm hàm kích hoạt. ReLU giúp mô hình giữ lại các giá trị dương và loại bỏ các giá trị âm trong kết quả tích chập, từ đó giúp giảm thiểu hiện tượng gradient biến mất (vanishing gradient) và tăng tốc độ hội tụ trong quá trình huấn luyện.

Lớp gộp (Pooling layer)

Lớp gộp (Pooling layer) có nhiệm vụ giảm kích thước của bản đồ đặc trưng (Feature map) sau tích chập, đồng thời giữ lại các thông tin quan trọng nhất. Mô hình này có thể sử dụng max pooling, chọn ra giá trị lớn nhất từ mỗi vùng được xem xét. Max pooling giúp chọn ra đặc trưng quan trọng nhất từ mỗi bộ lọc, giảm số lượng thông tin cần xử lý và giảm hiện tượng quá khớp (overfitting).

Việc giảm độ phức tạp của dữ liệu sau pooling giúp mô hình dễ dàng học các mối quan hệ sâu hơn mà không gặp khó khăn về quá tải dữ liệu.

Lớp kết nối đầy đủ (Fully connected layer)

Sau khi áp dụng các lớp tích chập và gộp, đầu ra sẽ được làm phẳng (flatten) thành một vector 1 chiều để đưa vào các lớp kết nối đầy đủ (Fully connected layer). Lớp kết nối đầy đủ có nhiệm vụ kết hợp các đặc trưng đã học được từ các lớp trước đó và tạo ra các kết quả phân loại.

Ở đây mô hình sử dụng một lớp Dense với 1 đơn vị, tương ứng với 2 nhãn phân loại câu trong bài toán phân loại câu. Lớp này sử dụng hàm kích hoạt Sigmoid, giúp chuyển đổi đầu ra thành các xác suất cho mỗi lớp.

Dropout: Để giảm thiểu overfitting, một lớp dropout được sử dụng để ngẫu nhiên bỏ qua một số nơ-ron trong quá trình huấn luyện.

Dữ liệu đầu ra

Mô hình này có 1 đơn vị đầu ra. Đơn vị này sử dụng hàm kích hoạt Sigmoid, giúp chuyển đổi đầu ra thành xác suất thuộc về một trong hai lớp phân loại câu tích, nếu xác suất lớn hơn 0.5, câu được phân loại là tích cực; ngược lại, nếu nhỏ

hơn 0.5, câu sẽ được phân loại là tiêu cực. giúp ta dễ dàng lựa chọn lớp có xác suất cao nhất làm kết quả dự đoán cuối cùng.

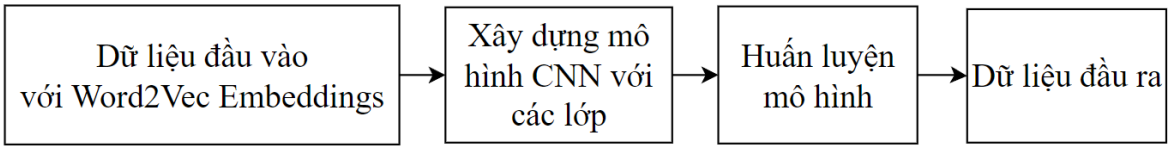
Hàm mất mát Binary cross-entropy: Để đo giá trị của hàm mục tiêu, hàm mất mát được sử dụng là binary cross-entropy, giúp đánh giá độ chính xác của dự đoán giữa hai lớp. Hàm này so sánh xác suất dự đoán với nhãn thực tế và tính toán lỗi để mô hình có thể tối ưu hóa qua từng bước huấn luyện.

2.4.2. Mô hình 2: CNN với Word2Vec embeddings

Mục tiêu của việc triển khai mô hình: Triển khai một mô hình phân loại câu. Đầu vào của mô hình là các câu văn bản đã được chuyển đổi thành vector từ Word2Vec. Đầu ra là nhãn phân loại của câu.

Mô hình CNN sử dụng Word2Vec embeddings là một cải tiến so với mô hình sử dụng random embeddings, thay vì khởi tạo các vector từ ngẫu nhiên, nó tận dụng các vector từ được huấn luyện trước đó bằng kỹ thuật Word2Vec. Mô hình này giúp cải thiện hiệu suất phân loại văn bản vì nó đã có kiến thức ngữ nghĩa cơ bản từ dữ liệu ngôn ngữ lớn.

Quy trình để triển khai mô hình như sau:



Hình 2.4. Quy trình triển khai mô hình CNN với Word2Vec embeddings

Xử lý dữ liệu đầu vào

Mô hình này sử dụng Word2Vec embeddings có thể được cập nhật và tinh chỉnh trong quá trình huấn luyện. Điều này giúp mô hình không chỉ dựa vào thông tin ngữ nghĩa từ các vector từ đã được học trước, mà còn có thể tối ưu hóa những vector từ này dựa trên dữ liệu huấn luyện cụ thể.

- **Embedding layer (Word2Vec):** Các vector từ ban đầu được khởi tạo từ tệp GloVe, sau đó chúng sẽ được mô hình điều chỉnh trong quá trình học.

Điều này giúp mô hình có thể tối ưu hóa các vector từ này sao cho phù hợp nhất với bài toán phân loại văn bản cụ thể.

- **Embedding matrix:** Mô hình này xây dựng một embedding matrix từ tệp GloVe và cho phép chúng được tinh chỉnh trong suốt quá trình học để nắm bắt ngữ cảnh và ý nghĩa của từ trong tập dữ liệu cụ thể.

Việc cho phép huấn luyện các vector từ này giúp mô hình có thể cập nhật các vector từ đối với những từ mà Word2Vec có thể chưa xử lý tốt hoặc có nghĩa đặc biệt trong bối cảnh dữ liệu cụ thể.

Lớp tích chập (Convolutional layer)

Giống như mô hình CNN với Random embeddings, mô hình này sử dụng các lớp tích chập (Convolutional layers) để học các đặc trưng từ các cụm từ trong câu.

Bộ lọc: Sử dụng ba bộ lọc với kích thước cửa sổ khác nhau: 3, 4, và 5 từ. Mỗi bộ lọc học các đặc trưng từ các cụm từ có độ dài tương ứng. Kích thước của các bộ lọc này giúp mô hình hiểu được các mối quan hệ ngữ nghĩa cục bộ giữa các từ trong các cụm từ ngắn.

Lớp kích hoạt (Activation layer), lớp gộp (Pooling layer), lớp kết nối đầy đủ (Fully connected layer), dữ liệu đầu ra

Các lớp này trương tự với mô hình 1 trước đó.

Mô hình CNN với Word2Vec embeddings được cho là mang lại sự linh hoạt và khả năng thích nghi cao hơn so với mô hình 1. Bằng cách cho phép cập nhật các vector từ trong quá trình huấn luyện, mô hình này có thể tối ưu hóa các embeddings để phù hợp hơn với ngữ cảnh cụ thể của bài toán phân loại câu. Điều này giúp cải thiện độ chính xác, đặc biệt là trong các trường hợp mà từ ngữ trong dữ liệu huấn luyện có sự khác biệt lớn so với dữ liệu huấn luyện ban đầu của Word2Vec.

2.5. KẾT LUẬN CHƯƠNG 2

Chương 2 đã cung cấp một cái nhìn toàn diện về bài toán phân loại câu trong xử lý ngôn ngữ tự nhiên và các phương pháp để giải quyết bài toán này, từ các phương pháp truyền thống đến các phương pháp dựa trên học sâu.

Phương pháp truyền thống trong phân loại câu như naive Bayes, Support Vector Machines, Bag-of-Words, TF-IDF, và N-grams đã được trình bày kỹ lưỡng. Mặc dù các phương pháp này đơn giản và có thể giải quyết các bài toán cơ bản trong phân loại câu, tuy nhiên chúng cũng gặp phải nhiều hạn chế, đặc biệt là trong việc xử lý ngữ nghĩa sâu và ngữ cảnh phức tạp của câu. Những phương pháp này thường gặp khó khăn trong việc nắm bắt các mối quan hệ ngữ nghĩa phức tạp và phụ thuộc nhiều vào các đặc trưng được trích xuất thủ công.

Phương pháp dựa trên học sâu đã chứng tỏ sự vượt trội trong việc phân loại câu nhờ khả năng tự động học các đặc trưng từ dữ liệu. Trong đó, Mạng nơ-ron tích chập (CNN) là một công cụ mạnh mẽ trong việc xử lý ngôn ngữ tự nhiên, đặc biệt là phân loại câu, nhờ khả năng trích xuất các đặc trưng cục bộ từ văn bản. Kỹ thuật Word2Vec cũng đóng vai trò quan trọng trong việc chuyển đổi từ ngữ thành các vector số có ngữ nghĩa, giúp các mô hình học sâu như CNN hiểu và xử lý văn bản một cách hiệu quả hơn. Cuối cùng, Chương 2 đã trình bày hai mô hình CNN bao gồm CNN với Random embeddings và CNN với Word2Vec embeddings, các mô hình này sẽ được thực nghiệm trong chương tiếp theo.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. THIẾT LẬP THỰC NGHIỆM

3.1.1. Bộ dữ liệu thực nghiệm

3.1.1.1. Bộ dữ liệu Movie Review (MR)

Bộ dữ liệu Movie Review (MR) [15] được thu thập từ các trang web đánh giá phim, gồm các bài đánh giá từ nhiều nguồn khác nhau. Tác giả bộ dữ liệu là Bo Pang và Lillian Lee từ Đại học Cornell. Bộ dữ liệu này được thiết kế để giúp các nhà nghiên cứu thử nghiệm và đánh giá các mô hình phân loại câu. Mục tiêu chính của việc sử dụng bộ dữ liệu này trong phân loại câu là xây dựng một mô hình có khả năng nhận diện và phân loại câu thành một trong hai nhãn phân loại tích cực hoặc tiêu cực từ từng câu độc lập.

Cấu trúc dữ liệu

Bộ dữ liệu bao gồm các đánh giá dưới dạng văn bản, trong đó cảm xúc của người viết thường được thể hiện ở mức câu hoặc đoạn văn bản. Dữ liệu được chia thành hai tệp "rt-polarity.pos" và "rt-polarity.neg", mỗi tệp chứa các câu đánh giá phim với nhãn cảm xúc tương ứng. Mỗi dòng trong các tệp này là một câu đánh giá, giúp dễ dàng đưa vào các mô hình phân loại câu mà không cần phải phân tách thủ công thành các câu con.

Trong đó:

- Tệp rt-polarity.neg: chứa các câu đánh giá phim với nhãn tiêu cực.
- Tệp rt-polarity.pos: chứa các câu đánh giá phim với nhãn tích cực.

Bảng 3.1. Mười dòng trong bộ dữ liệu Movie Review (MR)

Tệp	Dữ liệu
rt-polarity.neg	simplistic, silly and tedious
	not so much farcical as sour
	unfortunately the story and the actors are served with a hack script

	this 100-minute movie only has about 25 minutes of decent material
	solaris is a shapeless inconsequential move relying on the viewer to do most of the work
rt-polarity.pos	occasionally melodramatic, it's also extremely effective
	if this movie were a book, it would be a page-turner, you can't wait to see what happens next
	it's fun, splashy and entertainingly nasty
	deep intelligence and a warm, enveloping affection breathe out of every frame
	a movie that will thrill you, touch you and make you laugh as well

Đặc điểm của bộ dữ liệu Movie Review (MR)

Bộ dữ liệu có tổng cộng 10,662 dòng, mỗi dòng đã được phân loại vào nhãn tương ứng. Các câu đánh giá phim sử dụng ngôn ngữ tự nhiên, bao gồm từ ngữ mang tính cảm xúc, biểu cảm, và các phép ẩn dụ thường gặp trong các bài phê bình phim. Tính đa dạng trong cách diễn đạt của các câu đánh giá giúp mô hình học được nhiều ngữ cảnh cảm xúc phong phú.

Các mẫu trong bộ dữ liệu MR chủ yếu là các câu hoặc cụm từ ngắn trích từ các đánh giá phim, không phải các đoạn văn dài. Điều này giúp mô hình tập trung phân tích ở mức câu. MR có cấu trúc đơn giản, chỉ gồm các câu và nhãn cảm xúc, nên không cần nhiều thao tác tiền xử lý phức tạp như các bộ dữ liệu chứa văn bản dài hoặc đa ngữ.

Ưu điểm của bộ dữ liệu Movie Review (MR)

- **Dữ liệu đánh giá thực tế:** Bộ dữ liệu này gồm các bài đánh giá phim thực tế từ nhiều người, mang lại các sắc thái cảm xúc chân thực trong các đánh giá của người dùng.

- **Dễ huấn luyện và đánh giá:** Với nhãn nhị phân, việc huấn luyện và đánh giá mô hình trở nên đơn giản, nhanh chóng.
- **Cơ sở cho nghiên cứu mở rộng:** MR có thể được sử dụng kết hợp với các bộ dữ liệu khác như SST-2 để đánh giá hiệu suất của mô hình phân loại trên các dạng ngữ liệu cảm xúc khác nhau, từ đó nâng cao độ chính xác và khả năng khái quát của mô hình.

3.1.1.2. Bộ dữ liệu Stanford Sentiment Treebank (SST-2)

Bộ dữ liệu Stanford Sentiment Treebank (SST-2) [16] là một trong những bộ dữ liệu phổ biến nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên, đặc biệt là các bài toán phân loại câu. Đây là phiên bản khác từ bộ dữ liệu ban đầu của Stanford Sentiment Treebank (SST-1) với việc giảm số nhãn phân loại.

Cấu trúc dữ liệu

SST-2 cung cấp các câu độc lập và nhãn phân loại câu tương ứng. Mỗi câu là một nhận xét ngắn gọn hoặc đoạn văn ngắn. Các cột chính trong bộ dữ liệu bao gồm:

- **Idx:** Đây là mã định danh duy nhất cho mỗi câu.
- **Sentence:** Đây là nội dung thực tế của cụm từ hoặc câu từ bài đánh giá phim. Các cụm từ này có thể là một phần hoặc toàn bộ câu gốc trong bài đánh giá.
- **Label:** Đây là nhãn phân loại của từng cụm từ hoặc câu. Bộ dữ liệu phân loại thành hai nhãn phân loại khác nhau. Trong đó nhãn 0 là tiêu cực và nhãn 1 là tích cực.

Bảng 3.2. Năm dòng trong bộ dữ liệu Stanford Sentiment Treebank (SST-2)

idx	sentence	label
109	would have liked it more if it had just gone that one step further	1

112	takes a classic story , casts attractive and talented actors and uses a magnificent landscape to create a feature film that is wickedly fun to watch	1
41131	felt trapped and with no obvious escape for the entire 100 minutes	0
41206	would be rendered tedious by avary 's failure to construct a story with even a trace of dramatic interest	0
52079	with liberal doses of dark humor , gorgeous exterior photography , and a stable-full of solid performances	1

Đặc điểm của bộ dữ liệu SST-2

- **Kích thước:** Bộ dữ liệu có 67,348 câu kèm theo nhãn phân loại tương ứng
- **Từ vựng phong phú:** Các câu trong SST-2 được trích xuất từ các bài đánh giá phim thực tế, với ngữ cảnh phong phú và đa dạng về ngôn từ. Những câu này có thể là những nhận xét ngắn, độc lập hoặc đoạn cắt ngắn từ các câu dài trong đánh giá. Do lấy từ các đánh giá phim, từ ngữ trong SST-2 có nhiều sắc thái phức tạp, không chỉ bao gồm từ ngữ tích cực hay tiêu cực mà còn có các từ chỉ trạng thái trung gian, từ ngữ mỉa mai, hoặc cảm xúc đa chiều trong từng câu.

Ưu điểm của bộ dữ liệu SST-2

- **Cấu trúc đơn giản:** Mỗi mẫu dữ liệu trong SST-2 bao gồm một câu và một nhãn phân loại câu (tích cực hoặc tiêu cực). Điều này giúp cho việc tiền xử lý và xây dựng mô hình trở nên dễ dàng hơn.
- **Kích thước phù hợp:** SST-2 có kích thước phù hợp, đủ để huấn luyện các mô hình hiệu quả mà không quá tốn kém về tài nguyên tính toán.
- **Đa dạng chủ đề:** Các câu trong SST-2 bao phủ nhiều chủ đề khác nhau, giúp mô hình học được cách nhận diện cảm xúc trong các ngữ cảnh đa dạng.

- **Cân bằng nhãn:** Số lượng câu tích cực và tiêu cực trong bộ dữ liệu thường được phân bố tương đối cân bằng, giúp tránh tình trạng mô hình bị thiên lệch.
- **Thường được sử dụng để đánh giá mô hình:** SST-2 là một trong những bộ dữ liệu được sử dụng rộng rãi để so sánh hiệu suất của các mô hình phân loại câu khác nhau.

3.1.1.3. Bộ dữ liệu nhúng từ GloVe: Global vectors for word representation

Bộ dữ liệu nhúng từ "GloVe: Global vectors for word representation" [17] [18] là một tập các vector từ được huấn luyện trước với mục đích biểu diễn từ ngữ trong một không gian vector sao cho các từ có ngữ nghĩa tương tự nhau sẽ có vị trí gần nhau trong không gian này. GloVe là một trong những kỹ thuật phổ biến trong xử lý ngôn ngữ tự nhiên, giúp mô hình học máy hiểu được ngữ nghĩa của từ ngữ dựa trên ngữ cảnh. GloVe, được phát triển bởi các nhà nghiên cứu tại Stanford. Mô hình này được huấn luyện trên các tập dữ liệu lớn như Wikipedia, Common Crawl và các nguồn văn bản khác, nhằm học ra các vector từ có ngữ nghĩa tương quan với nhau. Mục tiêu của GloVe là tìm cách biểu diễn các từ thành vector sao cho khoảng cách giữa các vector phản ánh được mối quan hệ ngữ nghĩa giữa các từ.

Cấu trúc của bộ dữ liệu

Bộ dữ liệu nhúng từ GloVe bao gồm các tệp chứa các vector từ đã được huấn luyện trước, với mỗi từ được biểu diễn dưới dạng một vector có nhiều chiều (50, 100 hoặc 200 chiều). Mỗi tệp trong bộ dữ liệu tương ứng với một tập ngữ liệu khác nhau và số lượng chiều vector khác nhau, mỗi dòng trong tệp chứa một từ và một vector tương ứng. Có ba phiên bản với số chiều khác nhau: 50, 100, 200 chiều. Ví dụ cho một dòng: king 0.50451 0.68607 -0.59517 -0.022801…Trong đó, từ "king" được biểu diễn bởi một vector có nhiều chiều (tùy vào số chiều của bộ nhúng), mỗi giá trị trong vector đại diện cho một thành phần của từ trong không gian ngữ nghĩa.

Bảng 3.3. Năm dòng trong bộ dữ liệu nhúng từ "GloVe: Global vectors for word representation"

Từ	Vector nhúng từ
king	0.50451, 0.68607, -0.59517, -0.022801…
from	0.41037, 0.11342, 0.051524, -0.53833…
they	0.70835, -0.57361, 0.15375, -0.63335…
like	0.36808, 0.20834, -0.22319, 0.046283…
final	-0.41236, 0.6493, -0.55848, 0.86319…

Đặc điểm của bộ dữ liệu GloVe

- **Nhúng từ:** Mỗi từ trong bộ dữ liệu GloVe được biểu diễn bởi một vector số thực. Vector này cho phép mô hình máy học hiểu được ý nghĩa của từ dựa trên ngữ cảnh. GloVe học được các mối quan hệ ngữ nghĩa giữa các từ bằng cách sử dụng thống kê đồng xuất hiện của từ trong các tài liệu. Nói cách khác, từ nào xuất hiện cùng với các từ khác trong cùng ngữ cảnh sẽ có vector tương đồng. Bộ dữ liệu cung cấp các vector từ được huấn luyện trên nhiều nguồn văn bản khác nhau với quy mô lớn, bao gồm Wikipedia, Common Crawl và các tweet từ Twitter.
- **Số chiều của vector từ:** GloVe cung cấp các vector từ với nhiều số chiều khác nhau, thường là 50, 100 và 200 chiều. Số chiều càng cao, vector càng có khả năng biểu diễn nhiều đặc điểm ngữ nghĩa của từ hơn. Tuy nhiên, số chiều lớn hơn cũng làm tăng độ phức tạp tính toán. Vector 50 chiều phù hợp cho các ứng dụng đòi hỏi hiệu quả tính toán và không cần độ chính xác quá cao. Vector 200 chiều cung cấp biểu diễn ngữ nghĩa chi tiết hơn, thường được sử dụng cho các mô hình học sâu yêu cầu độ chính xác cao trong việc hiểu ngữ cảnh từ.
- **Tính chất của vector từ GloVe:** Vector từ trong GloVe có tính chất ngữ nghĩa toán học, nghĩa là ta có thể thực hiện các phép toán trên các vector để khám phá các mối quan hệ ngữ nghĩa giữa các từ.

- **Khả năng ứng dụng rộng rãi:** GloVe có thể được sử dụng làm lớp nhúng từ (embedding layer) trong các mô hình học máy và học sâu, giúp các mô hình này hiểu được ngữ nghĩa của từ ngữ mà không cần phải huấn luyện từ đầu. Bộ dữ liệu GloVe đặc biệt hữu ích trong các bài toán phân loại câu trong xử lý ngôn ngữ tự nhiên.

Ưu điểm của GloVe

- **Hiệu quả ngữ nghĩa:** GloVe học được các quan hệ ngữ nghĩa mạnh mẽ giữa các từ, và có thể nắm bắt được các mối quan hệ phức tạp như từ đồng nghĩa, trái nghĩa, hoặc quan hệ thứ bậc (ví dụ, "father" và "son").
- **Đa dạng nguồn dữ liệu:** Bộ dữ liệu GloVe được huấn luyện trên nhiều nguồn văn bản lớn, giúp mô hình có khả năng xử lý nhiều loại văn bản khác nhau, từ bài viết Wikipedia đến các bài đăng trên Twitter.
- **Dễ sử dụng:** Các vector từ đã được huấn luyện trước nên có thể dễ dàng tích hợp vào các mô hình học máy mà không cần phải tốn công sức huấn luyện lại từ đầu.

Thách thức khi sử dụng GloVe

- **Không cập nhật ngữ cảnh theo thời gian thực:** GloVe là mô hình nhúng từ tĩnh, tức là mỗi từ luôn có một vector cố định, không thay đổi theo ngữ cảnh. Điều này khiến GloVe không thể xử lý tốt các từ đa nghĩa (ví dụ: từ "bank" có thể có nghĩa là "ngân hàng" hoặc "bờ sông" tùy thuộc vào ngữ cảnh).
- **Kích thước lớn:** Đối với các phiên bản huấn luyện trên tập dữ liệu lớn, kích thước tệp dữ liệu rất lớn, có thể gây khó khăn trong việc lưu trữ và xử lý trên các hệ thống có tài nguyên hạn chế.

3.1.2. Công cụ phục vụ thực nghiệm

Trong đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu", việc sử dụng các công cụ và thư viện phần mềm là vô cùng quan trọng để xây dựng, huấn luyện, và đánh giá các mô hình học sâu.

Các công cụ này không chỉ hỗ trợ xử lý dữ liệu ngôn ngữ tự nhiên một cách hiệu quả mà còn tối ưu hóa quá trình huấn luyện mô hình, giúp cải thiện độ chính xác và hiệu suất. Phần này sẽ giới thiệu những công cụ chủ yếu được sử dụng trong quá trình thực nghiệm, từ các thư viện học sâu, xử lý ngôn ngữ tự nhiên đến các công cụ trực quan hóa và đánh giá kết quả.

Python là ngôn ngữ lập trình chính được sử dụng trong quá trình thực nghiệm của đề tài "Nghiên cứu kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng mô hình học sâu". Python đã trở thành một lựa chọn phổ biến trong các lĩnh vực liên quan đến trí tuệ nhân tạo và học máy nhờ cú pháp đơn giản, dễ tiếp cận, cùng với hệ sinh thái phong phú các thư viện hỗ trợ mạnh mẽ cho việc xử lý dữ liệu và triển khai các mô hình học sâu [19].

Python được lựa chọn cho đề tài này bởi các lý do sau:

- **Dễ học và sử dụng:** Python có cú pháp rõ ràng và dễ hiểu, giúp rút ngắn thời gian học tập và phát triển ứng dụng.
- **Thư viện đa dạng:** Python có một hệ sinh thái phong phú gồm nhiều thư viện chuyên dụng cho xử lý ngôn ngữ tự nhiên, học sâu, và trực quan hóa dữ liệu như NumPy, Pandas, TensorFlow, Keras, và Matplotlib. Các thư viện này đã được tích hợp đầy đủ các công cụ cần thiết để thực hiện các thao tác từ tiền xử lý dữ liệu đến huấn luyện mô hình.
- **Tương thích tốt với các nền tảng máy học:** Python hỗ trợ các nền tảng tính toán hiệu năng cao như GPU và TPU, giúp tăng tốc quá trình huấn luyện mô hình học sâu, đặc biệt là khi làm việc với các tập dữ liệu lớn và phức tạp

Trong đề tài này, Python đóng vai trò trung tâm trong toàn bộ quy trình thực nghiệm:

- **Tiền xử lý dữ liệu:** Python được sử dụng để làm sạch dữ liệu văn bản, token hóa, và tạo các biểu diễn vector cho các câu thông qua các công cụ

như Tokenizer trong Keras, Pandas để quản lý và thao tác dữ liệu, cũng như các hàm xử lý chuỗi cơ bản.

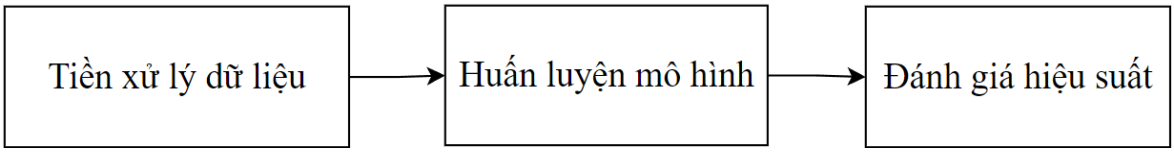
- **Xây dựng và huấn luyện mô hình học sâu:** Với sự hỗ trợ của TensorFlow và Keras, Python cung cấp các công cụ cần thiết để xây dựng mô hình học sâu như CNN. Các mô hình này được huấn luyện trên các tập dữ liệu ngôn ngữ tự nhiên để thực hiện nhiệm vụ phân loại câu.
- **Đánh giá và trực quan hóa kết quả:** Python cũng hỗ trợ đánh giá hiệu suất của các mô hình thông qua các công cụ tính toán độ chính xác, ma trận nhầm lẫn và trực quan hóa kết quả huấn luyện bằng các biểu đồ thông qua Matplotlib.

Các thư viện Python được sử dụng trong đề tài:

- **NumPy:** Thư viện này được sử dụng để làm việc với mảng và các phép toán ma trận. NumPy rất quan trọng khi thao tác với các dữ liệu dạng số.
- **Pandas:** Đây là một thư viện mạnh mẽ để xử lý và phân tích dữ liệu dạng bảng. Pandas được dùng để đọc, xử lý và phân loại dữ liệu đầu vào dưới dạng DataFrame
- **Scikit-learn:** Đây là thư viện chuẩn cho các tác vụ học máy, được sử dụng trong việc tính toán độ chính xác, ma trận nhầm lẫn, và phân chia dữ liệu. Scikit-learn hỗ trợ đánh giá hiệu suất mô hình và thực hiện các phép đo lường quan trọng trong quá trình phát triển mô hình.
- **TensorFlow và Keras:** TensorFlow là nền tảng mã nguồn mở mạnh mẽ cho học sâu, còn Keras là giao diện cấp cao của TensorFlow, giúp đơn giản hóa việc xây dựng và huấn luyện các mô hình học sâu. Trong dự án này, Keras được sử dụng để xây dựng các lớp mạng nơ-ron tích chập với các lớp như Embedding, Conv2D, MaxPooling, và Dense. TensorFlow giúp tối ưu hóa việc huấn luyện các mô hình này trên dữ liệu lớn.
- **Regularizers:** Các ràng buộc được sử dụng trong các lớp tích chập để tránh hiện tượng quá khớp (overfitting) bằng cách thêm các ràng buộc L2.

3.2. QUY TRÌNH THỰC NGHIỆM

Để triển khai thực nghiệm, đề tài thực hiện quy trình thực nghiệm gồm ba bước: tiền xử lý dữ liệu, huấn luyện các mô hình, và đánh giá hiệu suất. Quy trình này được thiết kế nhằm đảm bảo rằng dữ liệu đầu vào được chuẩn bị tốt nhất cho mô hình, các mô hình học sâu được xây dựng và tối ưu hóa hiệu quả, và kết quả cuối cùng được đánh giá khách quan và rõ ràng. Quy trình thực nghiệm được thể hiện qua sơ đồ dưới đây:



Hình 3.1. Sơ đồ quy trình thực nghiệm

3.2.1. Tiền xử lý dữ liệu

Trong các bài toán xử lý ngôn ngữ tự nhiên, tiền xử lý dữ liệu là bước quan trọng nhằm chuẩn bị dữ liệu thô thành dạng có thể sử dụng cho các mô hình học máy [4]. Dưới đây là các bước chi tiết về quy trình tiền xử lý dữ liệu cho bài toán phân loại câu:

Làm sạch dữ liệu

Bước đầu tiên là làm sạch các văn bản trong tập dữ liệu. Văn bản thô thường chứa các ký tự đặc biệt, dấu câu, URL và các khoảng trắng thừa. Việc loại bỏ các yếu tố này giúp giảm nhiễu cho mô hình. Cụ thể:

- Loại bỏ URL: Các đường dẫn trang web được loại bỏ hoặc thay thế để tránh mô hình bị ảnh hưởng bởi các từ không có ý nghĩa ngữ nghĩa.
- Loại bỏ ký tự đặc biệt và dấu câu: Những ký tự như dấu chấm, phẩy, và các ký tự đặc biệt khác không mang nhiều ý nghĩa ngữ nghĩa nên cần được loại bỏ.
- Chuyển thành chữ thường: Tất cả các từ được chuyển thành chữ thường để đảm bảo tính nhất quán, tránh phân biệt giữa từ viết hoa và viết thường.

Loại bỏ các nhãn mất cân bằng

Trong bài toán phân loại, dữ liệu có thể bị mất cân bằng nếu có sự chênh lệch lớn giữa số lượng các nhãn (lớp) khác nhau. Điều này dẫn đến mô hình bị thiên lệch và kém hiệu quả. Do đó, tập dữ liệu cần được cân bằng bằng cách lấy ngẫu nhiên số lượng mẫu giống nhau từ mỗi lớp nhãn. Việc cân bằng này đảm bảo rằng mô hình có cơ hội học tốt trên tất cả các nhãn.

Đánh giá độ dài câu

Văn bản đầu vào thường có độ dài không đồng đều, từ các câu ngắn chỉ vài từ cho đến các câu dài hơn nhiều. Trước khi chuyển đổi văn bản thành dạng số để đưa vào mô hình, cần xác định độ dài trung bình và tối đa của các câu. Điều này giúp quyết định chiều dài cố định của các chuỗi đầu vào. Các câu ngắn hơn độ dài này sẽ được thêm padding, trong khi các câu dài hơn sẽ bị cắt bớt.

Token hóa văn bản

Quá trình token hóa chuyển đổi văn bản thành chuỗi các chỉ số (token), mỗi chỉ số đại diện cho một từ trong từ điển. Trong các hệ thống xử lý ngôn ngữ tự nhiên, token hóa giúp biến các từ thành các số nguyên để mô hình có thể xử lý được. Từ điển từ này thường được tạo dựa trên tần suất xuất hiện của từ trong dữ liệu. Những từ ít xuất hiện có thể bị loại bỏ, chỉ giữ lại những từ phổ biến nhất để tối ưu hóa hiệu suất mô hình.

Padding chuỗi

Sau khi các câu đã được token hóa thành các chuỗi số, cần đảm bảo rằng tất cả các chuỗi có độ dài bằng nhau. Điều này được thực hiện thông qua quá trình padding. Với các câu ngắn hơn độ dài cố định, các giá trị 0 sẽ được thêm vào cuối chuỗi để đạt độ dài yêu cầu. Padding giúp mô hình có thể xử lý tất cả các câu với cùng một kích thước đầu vào, tránh việc gặp lỗi do các câu có độ dài khác nhau.

Chuyển đổi nhãn thành dạng số

Trong bài toán phân loại, các nhãn phân loại câu ban đầu là các giá trị rời rạc. Để mô hình có thể học tốt, các nhãn này cần được chuyển thành dạng số hoặc dạng one-hot encoding, trong đó mỗi nhãn được biểu diễn bằng một vector có giá

trị 1 tại vị trí tương ứng với lớp của nó. Việc này giúp mô hình học phân loại nhiều lớp hiệu quả hơn.

Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra

Cuối cùng, dữ liệu được chia thành hai tập riêng biệt: tập huấn luyện và tập kiểm tra. 90% dữ liệu được dùng để huấn luyện mô hình, trong khi phần còn lại được giữ lại để kiểm tra độ chính xác của mô hình trên dữ liệu mới mà nó chưa thấy trước đó. Điều này giúp đánh giá khả năng tổng quát hóa của mô hình sau khi huấn luyện.

Quy trình tiền xử lý dữ liệu bao gồm các bước như làm sạch văn bản, cân bằng dữ liệu, token hóa và padding các chuỗi, chuyển đổi nhãn, và chia dữ liệu thành tập huấn luyện và kiểm tra. Các bước này là rất cần thiết để đảm bảo dữ liệu được chuẩn bị ở dạng phù hợp nhất cho mô hình học sâu, từ đó nâng cao độ chính xác và hiệu quả của mô hình trong quá trình phân loại câu.

3.2.2. Huấn luyện các mô hình

3.2.2.1. Mô hình 1: CNN với Random embeddings

Quy trình để triển khai mô hình đã được trình bày ở Chương 2, dưới đây là các triển khai cho từng bước.

Xử lý dữ liệu đầu vào

Trước khi đưa dữ liệu văn bản vào mô hình, các câu văn bản cần được chuyển đổi thành các chuỗi số, nơi mỗi từ trong câu được ánh xạ thành một chỉ số duy nhất trong từ điển từ vựng.

- **Tokenization:** Quá trình này giúp mô hình hiểu các từ dưới dạng các số nguyên.
- **Padding:** Các câu thường có độ dài khác nhau, do đó cần chuẩn hóa về cùng một độ dài. Ở đây, tất cả các câu được điều chỉnh về chung độ dài của câu dài nhất thông qua quá trình padding (nếu câu ngắn hơn độ dài này, các từ ảo 0 được thêm vào cuối câu). Điều này giúp mô hình xử lý một kích thước đầu vào nhất quán. Ví dụ: Nếu câu có độ dài lớn nhất là 50 từ mà câu

hiện đang xử lý chỉ có 30 từ, ta sẽ thêm 20 số 0 vào cuối câu để đạt độ dài 50.

Sau khi xử lý xong, tập dữ liệu được chia thành 90% để huấn luyện và 10% để kiểm tra, đảm bảo mô hình có tập dữ liệu riêng để đánh giá sau khi huấn luyện.

Cấu hình của lớp nhúng:

- **input_dim:** Là số lượng từ có trong từ điển từ vựng (ở đây là 20,000 từ phổ biến nhất trong dữ liệu).
- **output_dim:** Số chiều của mỗi vector từ (200 chiều trong trường hợp này).
- **input_length**: Độ dài của mỗi câu (đã được padding), độ dài này là số từ dài nhất của một câu trong bộ dữ liệu.

Lớp tích chập (Convolutional layer)

Trong mô hình này, có 3 lớp tích chập với các kích thước cửa sổ khác nhau:

- Lớp tích chập đầu tiên có cửa sổ kích thước 3 từ.
- Lớp thứ hai có cửa sổ kích thước 4 từ.
- Lớp thứ ba có cửa sổ kích thước 5 từ.

Mỗi lớp tích chập có 100 bộ lọc (num_filters = 100), và hàm kích hoạt được sử dụng là ReLU (Rectified linear unit) để đưa tính phi tuyến vào mạng, giúp mô hình học được các đặc trưng phức tạp hơn.

Lớp gộp (Max pooling layer)

Sau khi áp dụng các lớp tích chập, để giảm kích thước dữ liệu đầu ra và giữ lại các đặc trưng quan trọng nhất, mô hình sử dụng các lớp Max Pooling. Kết quả của các lớp tích chập và gộp là ba ma trận đặc trưng từ các lớp tích chập khác nhau. Các ma trận này được gộp lại thành một tensor duy nhất, giúp mô hình học được các đặc trưng từ các cụm từ có kích thước khác nhau.

Lớp kết nối đầy đủ (Fully Connected Layer)

Sau khi qua các lớp tích chập và gộp, dữ liệu đầu ra là một tensor nhiều chiều. Để đưa dữ liệu này vào các lớp fully connected (kết nối đầy đủ), mô hình cần làm phẳng dữ liệu đầu ra thành một vector 1 chiều. Sau đó, một lớp Dropout

được thêm vào. Điều này giúp mạng không quá phụ thuộc vào một nhóm nơ-ron cụ thể.

Lớp Dense (Fully Connected): Sử dụng lớp Dense với 1 đầu ra, đại diện cho xác suất mà câu văn thuộc về lớp tích cực (1). Hàm kích hoạt Sigmoid được sử dụng để chuyển đổi đầu ra thành xác suất từ 0 đến 1.

Huấn luyện mô hình

Mô hình sau đó được biên dịch và huấn luyện [20]

- **Loss function:** Sử dụng hàm mất mát binary_crossentropy do mô hình chỉ có đầu ra là hai nhãn phân loại câu.
- **Optimizer:** Sử dụng bộ tối ưu hóa Adam [21], được biết đến với khả năng cập nhật trọng số hiệu quả trong các mạng nơ-ron sâu.
- **Metrics:** Độ chính xác (accuracy) được sử dụng làm thước đo để đánh giá mô hình.

Mô hình được huấn luyện với:

- **Batch size:** 32 (mỗi lần cập nhật trọng số mô hình sẽ xử lý 32 mẫu).
- **Validation split:** 10% dữ liệu huấn luyện được sử dụng để kiểm tra mô hình trong quá trình huấn luyện.

Kiến trúc mô hình

Bảng dưới đây cung cấp thông tin chi tiết về tên lớp, kích thước đầu ra, số lượng tham số và lớp mà nó kết nối đến.

Bảng 3.4. Kiến trúc mô hình CNN với Random embeddings

Lớp (loại)	**Kết nối đến**
InputLayer	
Embedding	InputLayer
Reshape	Embedding
Conv2D_1	Reshape
Conv2D_2	Reshape

Conv2D_3	Reshape
MaxPooling2D_1	Conv2D_1
MaxPooling2D_2	Conv2D_2
MaxPooling2D_3	Conv2D_3
Concatenate	MaxPooling2D_1, MaxPooling2D_2, MaxPooling2D_3
Flatten	Concatenate
Dropout	Flatten
Dense	Dropout

Trong đó:

- **Cột lớp (Loại):** Mỗi lớp trong mô hình được liệt kê tại đây, và ta có thể thấy rõ loại lớp đã sử dụng. Các lớp bao gồm:
  + InputLayer: Lớp đầu vào của mô hình.
  + Embedding: Lớp nhúng từ dùng để chuyển đổi các chỉ số từ thành các vector từ (200 chiều).
  + Reshape: Lớp thay đổi hình dạng đầu ra của lớp nhúng để phù hợp với Conv2D.
  + Conv2D: Các lớp tích chập (Convolutional Layers) với các bộ lọc có kích thước khác nhau.
  + MaxPooling2D: Các lớp gộp để giảm kích thước đầu ra.
  + Concatenate: Lớp gộp các đầu ra từ các lớp MaxPooling lại với nhau.
  + Flatten: Lớp làm phẳng đầu ra từ Conv2D và MaxPooling thành một vector 1 chiều.
  + Dropout: Lớp dropout để ngăn chặn overfitting.
  + Dense: Lớp fully connected với 1 đầu ra (nhãn phân loại).
- **Cột kết nối đến**: Cột này chỉ ra lớp đầu vào của lớp hiện tại.

Tham số huấn luyện mô hình

Các tham số huấn luyện mô hình được lựa chọn để đáp ứng yêu cầu mô hình có thể học được các đặc trưng quan trọng trong bài toán phân loại câu và đạt hiệu suất tốt trên tập kiểm thử.

Bảng 3.5. Các tham số huấn luyện mô hình

Tham số	**Giá trị**
Số lượng từ vựng tối đa	20,000
Số chiều của embedding	200
Số bộ lọc tích chập	100
Kích thước bộ lọc	(3,200), (4,200), (5,200)
Hàm kích hoạt	ReLU (cho lớp tích chập), Sigmoid (cho đầu ra)
l_2 Regularization	3
Hàm mất mát	Binary Crossentropy
Optimizer	Adam
Learning rate	0.001
Batch size	32
Số epoch	35

Trong đó:

- **Số lượng từ vựng tối đa**: Mô hình sử dụng 20,000 từ phổ biến nhất trong tập dữ liệu. Các từ không nằm trong danh sách này sẽ được thay thế bằng một token đặc biệt "unknown". Việc giới hạn này giúp mô hình tập trung vào các từ quan trọng nhất và giảm kích thước của lớp nhúng, giúp tối ưu bộ nhớ và tốc độ xử lý.
- **Số chiều của embedding**: Mỗi từ trong từ điển sẽ được ánh xạ thành một vector có 200 chiều. Giá trị 200 là mức phổ biến, đảm bảo mô hình có đủ

thông tin mà không bị quá tải, đồng thời cân bằng giữa hiệu suất và chất lượng biểu diễn.

- **Số bộ lọc tích chập**: Số lượng bộ lọc trong mỗi lớp tích chập là 100. Mỗi bộ lọc sẽ học một đặc trưng khác nhau của dữ liệu đầu vào. Việc chọn 100 bộ lọc giúp cân bằng giữa hiệu suất và chi phí tính toán.
- **Kích thước bộ lọc**: Ba loại kích thước bộ lọc cho phép quét qua lần lượt là 3, 4 hoặc 5 cụm từ một lúc. Các bộ lọc với kích thước khác nhau giúp mô hình học được đặc trưng ngữ cảnh từ các cụm từ có độ dài khác nhau.
- **Hàm kích hoạt**: ReLU cho các lớp tích chập giúp mô hình học nhanh hơn và giảm vấn đề vanishing gradient. Sigmoid được dùng ở đầu ra.
- **l_2 Regularization**: tham số này có giá trị là 3 làm giảm overfitting bằng cách phạt các trọng số lớn. Từ đó giúp mô hình tổng quát tốt hơn.
- **Hàm mất mát**: Binary Crossentropy là hàm mất mát được dùng phổ biến cho bài toán phân loại nhị phân. Nó đo lường sự khác biệt giữa giá trị thực tế và giá trị dự đoán của mô hình.
- **Optimizer**: Adam [21] là một thuật toán tối ưu hóa giúp điều chỉnh learning rate dựa trên độ dốc của hàm mất mát (gradient). Adam được thiết kế để hội tụ nhanh chóng, ổn định và hiệu quả với dữ liệu lớn. Bên cạnh đó Adam cũng làm việc tốt với dữ liệu có độ nhiễu cao, phù hợp với bài toán xử lý tự nhiên.
- **Learning rate**: Có giá trị là 0.001. Đây là tham số giúp xác định mức độ điều chỉnh trọng số sau mỗi lần cập nhật. Giá trị phù hợp giúp mô hình học ổn định hơn, tránh nhảy qua mất điểm tối ưu. Giá trị lớn có thể giúp hội tụ nhanh hơn nhưng dễ dẫn đến mất ổn định.
- **Batch size**: Có giá trị là 32. Mỗi lần cập nhật trọng số, mô hình sử dụng 32 mẫu dữ liệu thay vì cập nhật sau mỗi mẫu hoặc toàn bộ tập dữ liệu.
- **Số epoch**: Mô hình được huấn luyện qua 35 epoch. Một epoch là khi toàn bộ tập dữ liệu huấn luyện được đưa qua mô hình một lần. Nếu số epoch quá

thấp thì mô hình có thể chưa học đủ dẫn đến hiện tượng underfitting. Tuy nhiên nếu số epoch quá cao có thể khiến mô hình gặp tình trạng overfitting.

Lưu trữ kết quả

Sau khi huấn luyện xong, mô hình và lịch sử huấn luyện được lưu trữ lại. Độ chính xác (accuracy) được tính toán dựa trên sự so sánh giữa nhãn thực tế và nhãn dự đoán. Ma trận nhầm lẫn (confusion matrix) cũng được tính toán để xem mô hình dự đoán đúng và nhầm lẫn giữa các nhãn phân loại như thế nào.

Mô hình CNN với Random embeddings học các đặc trưng ngữ nghĩa từ văn bản bằng cách sử dụng các lớp tích chập và nhúng từ ngẫu nhiên. Sau quá trình huấn luyện, mô hình có thể phân loại các câu văn bản vào một trong hai nhãn nhãn phân loại câu.

3.2.2.2. Mô hình 2: CNN với Word2Vec embeddings

Mô hình này tương tự như Mô hình 1 nhưng thay vì sử dụng các vector nhúng ngẫu nhiên, nó sử dụng các vector nhúng từ đã được huấn luyện sẵn từ mô hình Word2Vec (dữ liệu nhúng từ GloVe). Trong mô hình này, các vector từ này có thể được huấn luyện. Điều này có nghĩa là các vector nhúng sẽ được cập nhật trong quá trình học của mô hình, giúp cải thiện khả năng biểu diễn của các từ trong ngữ cảnh của bài toán cụ thể. Dưới đây là các chi tiết của mô hình.

Quy trình để triển khai mô hình đã được trình bày ở Chương 2, dưới đây là các triển khai cho từng bước.

Xử lý dữ liệu đầu vào

Ma trận nhúng Word2Vec: Mô hình sử dụng vector nhúng đã được huấn luyện từ trước (như GloVe), với mỗi từ có một vector nhúng 200 chiều. Ma trận nhúng này chứa các vector nhúng cho tất cả các từ xuất hiện trong tập dữ liệu.

Lớp nhúng (Embedding layer)

Lớp nhúng Word2Vec: Lớp này ánh xạ các chỉ số từ của câu thành các vector nhúng có số chiều là 200. Lớp nhúng được khởi tạo với ma trận nhúng Word2Vec, trong đó mỗi từ có một vector biểu diễn với chiều dài 200.

Trainable embeddings: Việc cho phép các vector nhúng được huấn luyện đồng nghĩa với việc các vector này sẽ được cập nhật trong quá trình tối ưu hóa của mô hình. Điều này có nghĩa là qua từng epoch, các vector từ có thể điều chỉnh dựa trên cách mà chúng xuất hiện trong tập huấn luyện, giúp biểu diễn ngữ nghĩa tốt hơn cho bài toán phân loại câu.

Lớp tích chập (Convolutional layers), lớp kích hoạt (Activation layer), lớp gộp (Pooling layers), lớp kết nối đầy đủ (Fully connected layer), kiến trúc mô hình, tham số huấn luyện mô hình, huấn luyện mô hình

Các bước này tương tự mô hình 1.

3.3. KẾT QUẢ VÀ PHÂN TÍCH

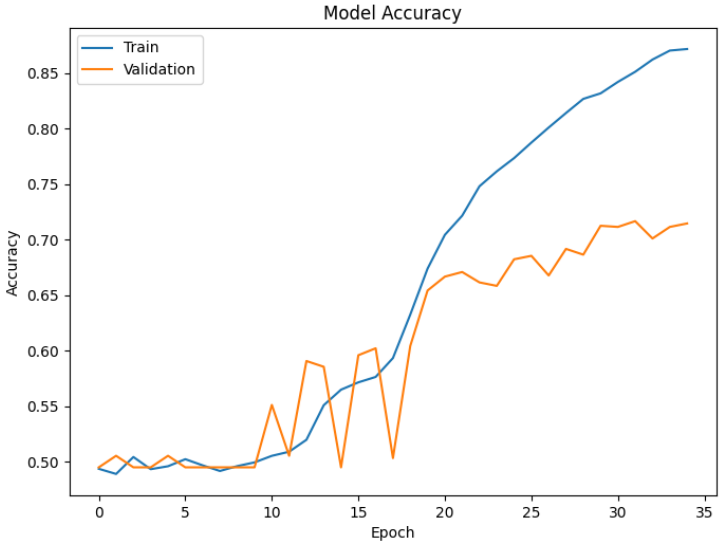
Sau khi thực nghiệm hai mô hình CNN với hai bộ dữ liệu Movie Review (MR) và Stanford Sentiment Treebank (SST-2) cho bài toán phân loại câu, các kết quả đã được ghi nhận và tổng hợp trong bảng sau, kết quả được tính bằng độ chính xác (Accuracy):

Bảng 3.6. Bảng tổng hợp kết huấn luyện các mô hình

Mô hình	Bộ dữ liệu MR	Bộ dữ liệu SST-2
CNN với Random embeddings	70.85%	90.45%
CNN với Word2Vec embeddings	74.79%	91.15%

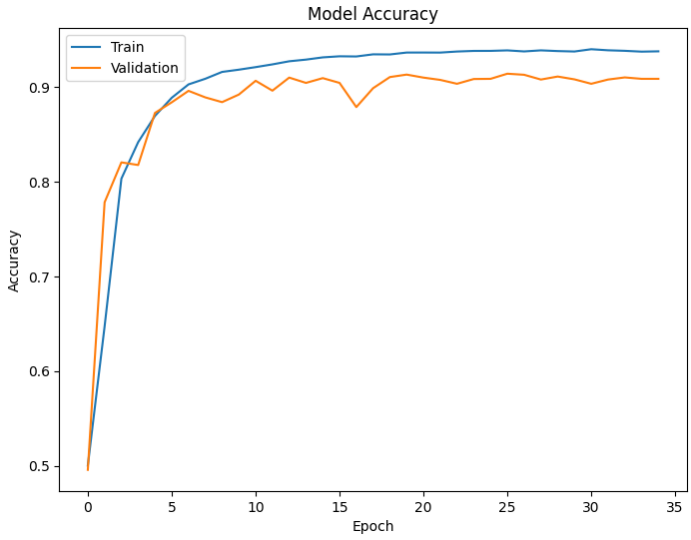
Mô hình 1: CNN với Random embeddings

Khi huấn luyện với bộ dữ liệu Movie Review (MR). Biểu đồ về độ chính xác (accuracy) cho thấy rằng mô hình đã học dần các đặc trưng của bộ dữ liệu qua các epoch và dần hội tụ. Độ chính xác đạt 70.85%. Đây là mức độ chính xác tương đối tốt cho mô hình với các nhúng từ ngẫu nhiên. Tuy nhiên trong giai đoạn đầu huấn luyện cũng chỉ ra rằng mô hình có gặp những khó khăn nhất định khi học các đặc trưng ngữ nghĩa từ dữ liệu Movie Review.



Hình 3.2. Quá trình huấn luyện mô hình CNN với Random embeddings trên bộ dữ liệu MR

Khi huấn luyện mô hình với bộ dữ liệu Stanford Sentiment Treebank (SST-2) cho kết quả tốt với độ chính xác 90.45%. Đường biểu diễn độ chính xác cho thấy mô hình học nhanh ngay từ những epoch đầu tiên và đạt trạng thái ổn định trong các epoch tiếp theo. Điều này cho thấy mô hình có khả năng tối ưu hóa tốt với dữ liệu và có thể trích xuất được các đặc trưng quan trọng từ dữ liệu đầu vào.

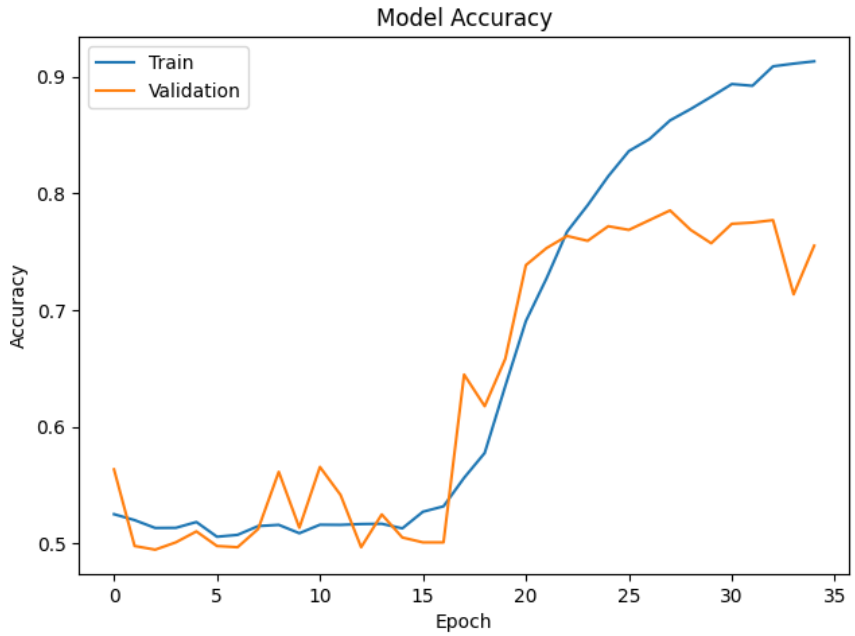


Hình 3.3. Quá trình huấn luyện mô hình CNN với Random embeddings trên bộ dữ liệu SST-2

Random embeddings không mang thông tin ngữ nghĩa sẵn có, do đó mô hình phải học từ đầu mà không có sự hỗ trợ từ các từ điển ngữ nghĩa đã được xây dựng trước.

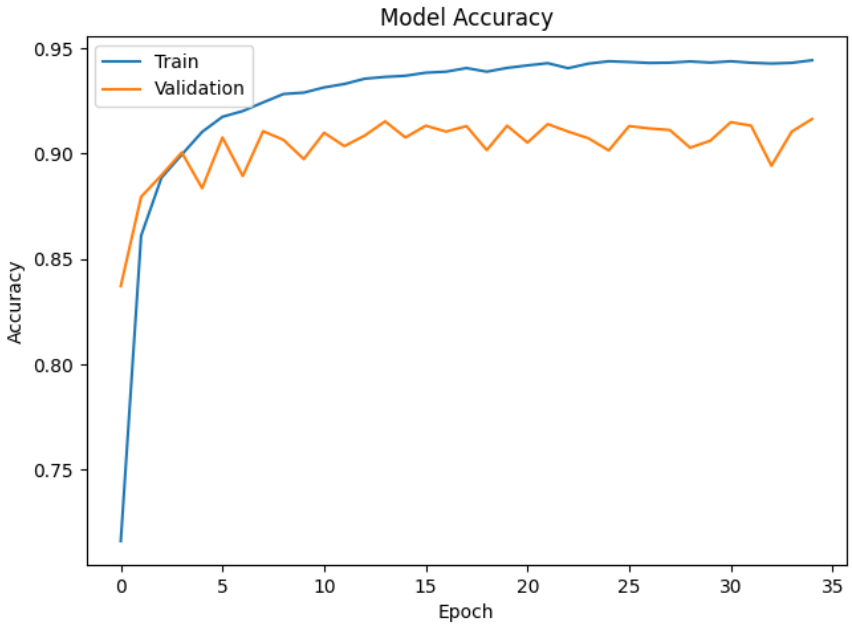
Mô hình 2: CNN với Word2Vec embeddings

Khi huấn luyện mô hình với bộ dữ liệu Movie Review (MR), kết quả độ chính xác đạt 74.79%. Kết quả này tốt hơn khi so với mô hình 1 nhờ khả năng huấn luyện nhúng từ giúp mô hình thích nghi tốt hơn với dữ liệu cụ thể. Đường biểu diễn độ chính xác cho thấy mô hình bắt đầu học chậm trong khoảng 15 epoch đầu tiên, tuy nhiên sau đó độ chính xác tăng mạnh Điều này phản ánh rằng mô hình đã tìm được cách học các đặc trưng ngữ nghĩa từ embeddings Word2Vec.



Hình 3.4. Quá trình huấn luyện mô hình CNN với Word2Vec embeddings trên bộ dữ liệu MR

Khi huấn luyện mô hình với bộ dữ liệu Stanford Sentiment Treebank (SST-2) đã đạt được độ chính xác 91.15%. Đây là một kết quả đáng chú ý, thể hiện rằng mô hình hoạt động tốt hơn so với mô hình trước đó.



Hình 3.5. Quá trình huấn luyện mô hình CNN với Word2Vec embeddings trên bộ dữ liệu SST-2

Trong quá trình huấn luyện mô hình CNN với Word2Vec embeddings, mô hình thể hiện sự cải thiện rõ rệt hơn trong quá trình huấn luyện, với sự gia tăng của độ chính xác qua các epoch và dần ổn định, cho thấy mô hình đang học tốt và cải thiện các trọng số của nhúng từ.

Tổng kết

Các mô hình CNN hoạt động hiệu quả trong việc học các đặc trưng ngữ nghĩa phức tạp từ dữ liệu văn bản. CNN có khả năng tự động hóa việc trích xuất các đặc trưng từ văn bản, điều mà các phương pháp truyền thống phải làm thủ công.

Các kết quả về độ chính xác của cả hai mô hình đều cho ra những kết quả tốt, trong đó thấy rõ sự khác biệt khi ứng dụng kỹ thuật Word2Vec embeddings giúp tận dụng tốt thông tin ngữ nghĩa, nhờ đó mô hình hoạt động hiệu quả hơn.

Kết quả của mô hình có ý nghĩa đối với nhiều ứng dụng xử lý ngôn ngữ tự nhiên trong thực tế. Một số tác động thực tiễn chính bao gồm:

- **Phân tích cảm xúc trong mạng xã hội**: Với độ chính xác cao trong việc phân loại câu, mô hình có thể được ứng dụng trong các hệ thống giám sát cảm xúc trên mạng xã hội hoặc các diễn đàn trực tuyến. Doanh nghiệp có thể sử dụng mô hình để đánh giá phản hồi của khách hàng về sản phẩm hoặc dịch vụ, giúp đưa ra các chiến lược kinh doanh phù hợp.
- **Gợi ý nội dung**: Các nền tảng mạng xã hội có thể tận dụng mô hình để hiểu phản hồi của người dùng, từ đó cá nhân hóa nội dung gợi ý dựa trên xu hướng cảm xúc của người dùng.
- **Cải thiện trợ lý ảo**: Mô hình phân loại câu giúp chatbot hiểu rõ hơn ý định của người dùng, từ đó phản hồi chính xác hơn.
- **Hỗ trợ hệ thống kiểm duyệt nội dung**: Các nền tảng truyền thông có thể sử dụng mô hình để phát hiện và ngăn chặn các bình luận tiêu cực, từ đó giúp đảm bảo môi trường an toàn trên các diễn đàn trực tuyến.

Những hạn chế còn tồn tại:

- **Overfitting:** Các mô hình có thể bị overfitting khi dữ liệu huấn luyện không đủ lớn. Khi huấn luyện hai mô hình với bộ dữ liệu MR đã xảy ra hiện tượng overfitting trong khi điều này không xảy ra với bộ dữ liệu SST-2, lý do là bộ dữ liệu MR có ít dữ liệu hơn đáng kể so với bộ dữ liệu SST-2.
- **Tiêu tốn tài nguyên:** Mô hình CNN yêu cầu nhiều tài nguyên tính toán hơn so với các phương pháp truyền thống, đặc biệt khi số lượng tham số lớn và cần tối ưu hóa nhiều lớp mạng.
- **Khả năng tổng quát hóa:** Mặc dù mô hình có thể đạt độ chính xác cao trên tập huấn luyện, khả năng tổng quát hóa khi áp dụng vào dữ liệu mới là thách thức, đặc biệt khi dữ liệu huấn luyện không đại diện cho toàn bộ không gian dữ liệu.

3.4. ĐÁNH GIÁ & ĐỀ XUẤT CẢI TIẾN

3.4.1. Đánh giá

Trong quá trình triển khai các mô hình học sâu cho bài toán phân loại câu, đề tài đã thực nghiệm với bốn mô hình khác nhau, bao gồm:

Mô hình 1: CNN với Random embeddings.

Mô hình 2: CNN với Word2Vec embeddings.

Kết quả thu được từ các mô hình này được đánh giá dựa trên độ chính xác (Accuracy). Đây là thước đo chính để đánh giá hiệu suất của mô hình, thể hiện tỷ lệ dự đoán đúng trên tổng số mẫu thử.

Cụ thể:

- **Mô hình 1 (Random embeddings):** Đây là mô hình cơ bản, sử dụng nhúng từ ngẫu nhiên. Mô hình này có hiệu suất kém hơn các mô hình khác do các vector từ được khởi tạo ngẫu nhiên không nắm bắt được ngữ nghĩa từ ngữ.
- **Mô hình 2 (Word2Vec embeddings):** Đây là mô hình đạt hiệu suất tốt trong bài toán phân loại câu. Việc sử dụng vector từ Word2Vec có thể huấn luyện cho phép mô hình học và điều chỉnh các vector này theo ngữ cảnh cụ thể của bài toán, giúp cải thiện khả năng dự đoán.

Mặc dù bài toán phân loại câu trong xử lý ngôn ngữ tự nhiên có yếu tố phức tạp với nhiều thách thức, nhưng đề tài đã xây dựng được nền tảng vững chắc và cung cấp các giải pháp hiệu quả. Những kết quả đạt được đã cho thấy hiệu quả vượt trội trong việc trích xuất và phân tích các đặc trưng ngữ nghĩa từ văn bản.

3.4.2. Đề xuất cải tiến

Tuy đề tài có những kết quả tốt sau khi thực nghiệm với mô hình nơ-ron tích chập (CNN) nhưng đề tài vẫn có nhiều tiềm năng phát triển hơn nữa với các hướng sau:

Tăng cường dữ liệu huấn luyện

Một trong những vấn đề thường gặp với các mô hình học sâu là thiếu dữ liệu huấn luyện đủ lớn và đa dạng, khiến cho mô hình không thể tổng quát tốt cho

dữ liệu kiểm thử. Có thể phát triển đề tài bằng cách sử dụng các tập dữ liệu lớn hơn, khai thác thêm các bộ dữ liệu công khai trong cùng lĩnh vực. Bên cạnh đó có thể kết hợp nhiều tập dữ liệu khác nhau, hoặc sử dụng học không giám sát để tận dụng dữ liệu chưa gán nhãn, giúp mô hình hiểu rõ hơn về các cấu trúc ngôn ngữ khác nhau.

Kết hợp nhiều kỹ thuật

Việc kết hợp nhiều kỹ thuật khác nhau có thể giúp mô hình học sâu nắm bắt tốt hơn các khía cạnh khác nhau của ngôn ngữ tự nhiên. Kết hợp CNN với LSTM hoặc GRU: Mô hình CNN mạnh về việc trích xuất các đặc trưng cục bộ trong ngữ cảnh nhỏ, nhưng hạn chế trong việc xử lý các chuỗi dài. Kết hợp CNN với LSTM hoặc GRU giúp mô hình học được cả đặc trưng cục bộ và ngữ cảnh dài hạn.

Tối ưu hóa quá trình huấn luyện

Tối ưu hóa các siêu tham số (hyperparameter) của mô hình như kích thước batch, số lượng epoch, tỷ lệ học (learning rate), kích thước vector nhúng (embedding size), và số lượng bộ lọc (filters) để tìm ra các giá trị tốt nhất cho hiệu suất mô hình. Sử dụng Grid search hoặc Random search để tìm kiếm các tham số tối ưu, thay vì chỉ thử nghiệm theo từng giá trị cố định.

Hướng phát triển lâu dài

Để mở rộng khả năng ứng dụng của mô hình, có thể thử nghiệm mô hình trên các ngôn ngữ khác ngoài tiếng Anh, từ đó phát triển các hệ thống đa ngôn ngữ mạnh mẽ hơn.

Thay vì chỉ phân loại câu đơn thuần, có thể phát triển mô hình để phân loại câu phức tạp, ngữ nghĩa đa chiều hoặc các tác vụ phân loại văn bản yêu cầu sự tinh tế hơn (Ví dụ: phân loại các đoạn hội thoại trong ứng dụng trò chuyện tự động).

3.5. KẾT LUẬN CHƯƠNG 3

Chương 3 đã trình bày chi tiết quá trình thực nghiệm và đánh giá hiệu suất của các mô hình học sâu cho bài toán phân loại câu trong xử lý ngôn ngữ tự nhiên. Cụ thể, các bước từ thiết lập thực nghiệm, tiền xử lý dữ liệu, xây dựng và huấn luyện mô hình, cho đến việc phân tích và đánh giá kết quả đã được thực hiện một cách có hệ thống.

Thiết lập thực nghiệm: Các mô hình được huấn luyện trên một bộ dữ liệu đảm bảo khả năng tổng quát hóa của mô hình khi áp dụng vào các bài toán thực tế. Bộ công cụ và phương pháp được lựa chọn kỹ lưỡng, phù hợp với bài toán và mục tiêu của đề tài.

Quy trình thực nghiệm: Quy trình tiền xử lý dữ liệu giúp chuẩn hóa và làm sạch dữ liệu đầu vào, tạo điều kiện thuận lợi cho các mô hình học sâu tiếp thu các đặc trưng của văn bản một cách hiệu quả. Các mô hình khác nhau được xây dựng và huấn luyện theo quy trình cụ thể.

Kết quả và phân tích: Kết quả thực nghiệm cho thấy sự khác biệt về hiệu suất giữa các mô hình, với mô hình CNN Word2Vec embeddings đạt hiệu suất tốt nhất. Kết quả này khẳng định tính khả thi và hiệu quả của mô hình học sâu trong việc phân loại câu so với các phương pháp truyền thống.

Đánh giá và đề xuất cải tiến: Dựa trên phân tích kết quả, một số hạn chế đã được xác định như hiện tượng overfitting trong một số mô hình. Đề xuất cải tiến đã được đưa ra, bao gồm việc kết hợp các mô hình học sâu khác như LSTM hoặc GRU, cũng như cải thiện quy trình tiền xử lý và kỹ thuật tăng cường dữ liệu.

KẾT LUẬN VÀ KHUYẾN NGHỊ

1. Kết luận chung

Đề án này đã tiến hành nghiên cứu và phân tích các kỹ thuật phân loại câu trong xử lý ngôn ngữ tự nhiên bằng cách sử dụng các mô hình học sâu. Qua việc nghiên cứu và ứng dụng các phương pháp truyền thống cũng như các mô hình hiện đại như mạng nơ-ron tích chập (CNN) và kỹ thuật Word2Vec, đề án đã xây dựng được một nền tảng vững chắc để phân loại câu một cách hiệu quả hơn.

Đề án này đã có những đóng góp trong việc ứng dụng kỹ thuật học sâu vào phân loại câu, từ đó mở rộng khả năng của xử lý ngôn ngữ tự nhiên trong các tác vụ phân tích ngôn ngữ. Dù CNN thường được sử dụng trong các bài toán thị giác máy tính, đề án này đã chỉ ra rằng CNN cũng có thể đạt được hiệu quả cao trong phân loại câu. Nhờ khả năng tự động trích xuất các đặc trưng từ dữ liệu đầu vào, CNN đã giúp cải thiện độ chính xác của mô hình. Qua quá trình thực nghiệm, đề án đã cung cấp những đánh giá cụ thể về các yếu tố ảnh hưởng đến hiệu suất của mô hình, từ cấu trúc mô hình đến kích thước của tập dữ liệu và kỹ thuật tiền xử lý dữ liệu và huấn luyện mô hình để có kết quả đầu ra.

2. Khuyến nghị về một số hướng nghiên cứu tiếp theo

Đề án về phân loại câu bằng mô hình học sâu trong xử lý ngôn ngữ tự nhiên vẫn còn nhiều hướng phát triển tiềm năng. Trong thời gian tới, nội dung nghiên cứu đề án có thể mở rộng theo hướng:

- **Khám phá các phương pháp tối ưu hóa mô hình:** Cải tiến các kỹ thuật huấn luyện, tinh chỉnh mô hình, và sử dụng các kỹ thuật học không giám sát hoặc bán giám sát có thể giúp nâng cao hiệu suất của các mô hình học sâu.
- **Tăng cường phân tích và mở rộng ứng dụng:** Mở rộng nghiên cứu sang các ngôn ngữ khác hoặc các loại văn bản khác nhau, đồng thời áp dụng các kỹ thuật phân loại câu vào các lĩnh vực khác như dịch máy, phân tích cảm xúc, và nhận dạng giọng nói.

- **Cải tiến mô hình:** Mặc dù CNN đã chứng minh hiệu quả, việc kết hợp CNN với các mô hình học sâu tiên tiến khác LSTM, có thể giúp mô hình xử lý tốt hơn các phụ thuộc dài hạn trong câu, từ đó cải thiện khả năng phân loại.

Tổng kết lại, đề án này đã chứng minh được tiềm năng to lớn của các mô hình học sâu trong việc phân loại câu trong xử lý ngôn ngữ tự nhiên. Mặc dù đã đạt được những thành công nhất định, vẫn còn nhiều cơ hội để mở rộng và phát triển đề tài trong tương lai, nhằm tối ưu hóa và áp dụng rộng rãi hơn các kỹ thuật này trong thực tiễn.

TÀI LIỆU THAM KHẢO

[1] Schopf, T., Arabi, K., & Matthes, F. (2023), "Exploring the Landscape of Natural Language Processing Research", Technical University of Munich, Department of Computer Science, Germany.

[2] Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L. and Roossin, P. S. (1990). "A Statistical Approach to Machine Translation". *Computational Linguistics*, 16(2), pp. 79-85.

[3] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2003), "A Neural Probabilistic Language Model", *Journal of Machine Learning Research*, 3, pp. 1137-1155.

[4] Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), "Efficient Estimation of Word Representations in Vector Space", *Proceedings of the International Conference on Learning Representations (ICLR)*.

[5] Koehn, P. (2005), "Europarl: A Parallel Corpus for Statistical Machine Translation", *Proceedings of the 10th Machine Translation Summit*, pp. 79-86.

[6] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. and Kingsbury, B. (2012), "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", *IEEE Signal Processing Magazine*, 29(6), pp. 82-97.

[7] Hinton, G. E., Osindero, S. and Teh, Y. W. (2006), "A Fast Learning Algorithm for Deep Belief Nets", *Neural Computation*, 18(7), pp. 1527-1554.

[8] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, 25, pp. 1097-1105.

[9] He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep Residual Learning for Image Recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.

[10] Hochreiter, S. and Schmidhuber, J. (1997), "Long Short-Term Memory", *Neural Computation*, 9(8), 1735-1780.

[11] Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014), "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", *arXiv preprint arXiv:1409.1259*.

[12] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998), "Gradient-based Learning Applied to Document Recognition", *Proceedings of the IEEE*, 86(11), pp. 2278-2324.

[13] Nair, V., & Hinton, G. E. (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines." *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807-814.

[14] Zhang, Y. and Wallace, B. (2015), "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification", arXiv preprint arXiv:1510.03820.

[15] https://www.cs.cornell.edu/people/pabo/movie-review-data/

[16] https://huggingface.co/datasets/stanfordnlp/sst2

[17] https://www.kaggle.com/datasets/rtatman/glove-global-vectors-for-word-representation

[18] Pennington, J., Socher, R., & Manning, C. (2014). "GloVe: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543. Association for Computational Linguistics, Doha, Qatar.

[19] Chollet, F. (2017). *Deep Learning with Python.* Manning Publications, New York.

[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." *Nature*, 521(7553), pp. 436-444.

[21] Kingma, D. P., & Ba, J. (2015). "Adam: A method for stochastic optimization." *International Conference on Learning Representations (ICLR)*.