Shape, square

Description automatically generated**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**

Logo

Description automatically generated

**TRÌ HOÀI LỘC – 20133063**

**NGUYỄN PHẠM NGỌC DUY – 20133031**

**Đề Tài:**

**XÂY DỰNG MÔ HÌNH PHÂN LOẠI QUAN HỆ THAM CHIẾU TRONG VĂN BẢN PHÁP QUY SỬ DỤNG LSTM**

**KHÓA LUẬN TỐT NGHIỆP NGÀNH KỸ THUẬT DỮ LIỆU**

**GIÁO VIÊN HƯỚNG DẪN**

**Th.S. TRẦN TRỌNG BÌNH**

**KHÓA 2020 – 2024**

*TP. Hồ Chí Minh, tháng 02 năm 2024*

Shape, square

Description automatically generated**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**

Logo

Description automatically generated

**TRÌ HOÀI LỘC – 20133063**

**NGUYỄN PHẠM NGỌC DUY – 20133031**

**Đề tài:**

**XÂY DỰNG MÔ HÌNH PHÂN LOẠI QUAN HỆ THAM CHIẾU TRONG VĂN BẢN PHÁP QUY SỬ DỤNG LSTM**

**KHÓA LUẬN TỐT NGHIỆP NGÀNH KỸ THUẬT DỮ LIỆU**

**GIÁO VIÊN HƯỚNG DẪN**

**THS. TRẦN TRỌNG BÌNH**

**KHÓA 2020 – 2024**

*TP. Hồ Chí Minh, tháng 02 năm 2024*

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CÔNG NGHỆ THÔNG TIN**  **\*\*\*\*\*\*** | **CỘNG HÒA XHCN VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*** |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: **Trì Hoài Lộc** MSSV 1: **20133063**

Họ và tên Sinh viên 2: **Nguyễn Phạm Ngọc Duy** MSSV 2: **20133031**

Chuyên ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **Xây dựng mô hình phân loài quan hệ tham chiếu trong văn bản pháp quy sử dụng LSTM**

Họ và tên giáo viên hướng dẫn: **ThS. Trần Trọng Bình**

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. Ưu điểm:

……………………………………………………………………………………………………………………………………………………………………………………

1. Khuyết điểm:

……………………………………………………………………………………………………………………………………………………………………………………

1. Đề nghị cho bảo vệ hay không? ………………………………………………….
2. Đánh giá loại: …………………………………………………………………….
3. Điểm: ……………………………………………………………………………..

*TP. Hồ Chí Minh, tháng … năm …*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CÔNG NGHỆ THÔNG TIN**  **\*\*\*\*\*\*** | **CỘNG HÒA XHCN VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*** |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên Sinh viên 1: **Trì Hoài Lộc** MSSV 1: **20133063**

Họ và tên Sinh viên 2: **Nguyễn Phạm Ngọc Duy** MSSV 2: **20133031**

Chuyên ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **Xây dựng mô hình phân loài quan hệ tham chiếu trong văn bản pháp quy sử dụng LSTM**

Họ và tên giáo viên phản biện: **TS. Nguyễn Thành Sơn**

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. Ưu điểm:

……………………………………………………………………………………………………………………………………………………………………………………

1. Khuyết điểm:

……………………………………………………………………………………………………………………………………………………………………………………

1. Đề nghị cho bảo vệ hay không? ………………………………………………….
2. Đánh giá loại: …………………………………………………………………….
3. Điểm: ……………………………………………………………………………..

*TP. Hồ Chí Minh, tháng … năm …*

Giảng viên phản biện

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

*Đầu tiên, có được môi trường học tập chất lượng và hiệu quả để chúng em có môi trường tốt nhất trong thời gian thực hiện nghiên cứu đề tài, chúng em xin gửi lời cảm ơn đến Ban giám hiệu trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.*

*Ngoài ra, chúng em xin gửi lời cảm ơn đến Ban chủ nhiệm khoa Công nghệ Thông tin và các Thầy/Cô khoa Công nghệ Thông tin đã tận tâm chỉ dạy chúng em trong suốt những năm tháng đại học vừa qua, để chúng em có cơ sở cũng như những kiến thức cần thiết để thực hiện đề tài trong suốt quá trình học tập tại trường.*

*Cuối cùng, chúng em xin gửi lời cảm ơn chân thành đến thầy Trần Trọng Bình - Giáo viên hướng dẫn Khóa luận tốt nghiệp – Khoa Công nghệ Thông tin – Chuyên ngành Kỹ thuật dữ liệu – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, đã quan tâm, hướng dẫn, góp ý và tận tình chỉ dạy chúng em trong suốt thời gian thực hiện nghiên cứu.*

*Đề tài của chúng em khó lòng tránh khỏi những hạn chế và sai sót nhất định, kính mong Thầy cô bỏ qua và chúng em mong nhận được những đóng góp và chỉ bảo thêm từ Thầy Cô để chúng em rút kinh nghiệm sau này.*

*Chúng em xin chân thành cảm ơn.*

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CÔNG NGHỆ THÔNG TIN**  **\*\*\*\*\*\*** | **CỘNG HÒA XHCN VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*** |

**ĐỀ CƯƠNG KHÓA LUẬN TỐT NGHIỆP**

Họ và tên SV thực hiện 1: **Trì Hoài Lộc** MSSV: **20133063**

Họ và tên SV thực hiện 2: **Nguyễn Phạm Ngọc Duy** MSSV: **20133031**

Thời gian làm tiểu luận: **Từ ngày 21/08/2023** Đến: **21/12/2023**

Chuyên ngành: **Kỹ Thuật Dữ Liệu**

Tên luận văn: **Xây dựng mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy sử dụng LSTM**

Giáo viên hướng dẫn: **ThS. Trần Trọng Bình**

**Nhiệm vụ của luận văn:**

1. Lý thuyết: **Nguyễn Phạm Ngọc Duy**
2. Thực hành: **Trì Hoài Lộc**

**KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thời gian** | **Công việc** | **Ghi chú** |
| 1 | 19/02 đến 25/02 | Xây dựng đề cương cho đề tài “Xây dựng mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy sử dụng LSTM” |  |
| 2 | 26/02 đến 03/03 | Viết lý thuyết phần mở đầu và tìm những tài liệu nghiên cứu có liên quan. |  |
| 3 | 03/03 đến 10/03 | Nghiên cứu những công trình có liên quan.  Thực hiện lấy dữ liệu từ trang web vbpl.vn |  |
| 4 | 11/03 đến 17/03 | Viết những hiểu biết của bản thân về những công trình nghiên cứu đã tham khảo.  Thực hiện lấy dữ liệu từ trang web thuvienphapluat.vn |  |
| 5 | 18/03 đến 24/03 | Bắt đầu viết chương 1. Xây dựng mã nguồn của các bước tiền xử lý dữ liệu. |  |
| 6 | 25/03 đến 31/3 | Hoàn thiện chương 1. Thử nghiệm với cơ chế embedding Word2Vec. |  |
| 7 | 01/04 đến 07/04 | Bước đầu định hình và xây dựng mô hình phân loại chủ đề văn bản. Tìm kiếm và thử nghiệm với những tập dữ liệu khác dùng thông số của thử nghiệm 1. |  |
| 8 | 08/04 đến 14/04 | Bắt đầu viết chương 2. Cải thiện các bước tiền xử lý, tiến hành thử nghiệm với thông số thử nghiệm 2. |  |
| 9 | 15/04 đến 28/04 | Hoàn thiện chương 2. Tiếp tục cải thiện các bước tiền xử lý, thay dổi cơ chế embedding từ Word2Vec sang GloVec và thử nghiệm với thông số thử nghiệm 3. |  |
| 10 | 29/04 đến 05/05 | Bắt đầu viết chương 3. Cải thiện các thông số trong quá trình huấn luyện mô hình và tiến hành thử nghiệm với thông số thử nghiệm 4. |  |
| 11 | 06/05 đến 12/05 | Tiếp tục điều chỉnh tham số và cải thiện các bước tiền xử lý, thử nghiệm với thông số thử nghiệm 5. |  |
| 12 | 13/05 đến 19/05 | Huấn luyện mô hình với các tập dữ liệu lớn hơn, thử nghiệm với thông số thử nghiệm 6. |  |
| 13 | 20/05 đến 26/05 | Chuyển đổi từ GloVec sang phương pháp embedding FastText. Nghiên cứu và thử nghiệm kết hợp mô hình với cơ chế attention. |  |
| 14 | 27/05 đến 02/06 | Thử nghiệm mô hình kết hợp cơ chế attention với FastText đã được huấn luyện trước, điều chình các tham số. |  |
| 15 | 03/06 đến 09/06 | Huấn luyên lại mô hình với FastText được huấn luyện lại từ tập dữ liệu. |  |
| 16 | 10/06 đến 16/06 | Hoàn thiện chương 3. Lựa cọn tham số phù hợp với mô hình. Phân mã nguồn thành 2 phần cho 2 tập dữ liệu khác nhau. |  |
| 17 | 17/06 đến 23/06 | Đánh giá mô hình thử nghiệm bằng chỉ số Accuracy và F1-Score. Hoàn thiện chương 4. |  |

|  |  |
| --- | --- |
|  | Ngày .... tháng 05 năm 2024 |
| Ý kiến của giáo viên hướng dẫn  ThS. Trần Trọng Bình | Người viết đề cương  Trì Hoài Lộc  Nguyễn Phạm Ngọc Duy |

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc154553394)

[CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN PHÂN LOẠI CHỦ ĐỀ VĂN BẢN 5](#_Toc154553395)

[1.1. Xử lý ngôn ngữ tự nhiên 5](#_Toc154553396)

[1.1.1. Giới thiệu xử lý ngôn ngữ tự nhiên 5](#_Toc154553397)

[1.1.2. Các bước xử lý 5](#_Toc154553398)

[1.2. Các kỹ thuật xử lý ngôn ngữ tự nhiên 6](#_Toc154553399)

[1.2.1. Tách từ (Tokenization) 6](#_Toc154553400)

[1.2.1.1. Word-based Tokenization 6](#_Toc154553401)

[1.2.1.2. Character-based Tokenization 6](#_Toc154553402)

[1.2.1.3. Subword-based Tokenization 7](#_Toc154553403)

[1.2.2. Word Embedding 7](#_Toc154553404)

[1.2.2.1. Word2Vec 8](#_Toc154553405)

[1.2.2.2. Glove (Global Vectors) 9](#_Toc154553406)

[1.2.2.3. FastText 10](#_Toc154553407)

[1.3. Bài toán phân loại chủ đề văn bản 11](#_Toc154553408)

[1.3.1. Khái niệm của bài toán phân loại chủ đề văn bản 11](#_Toc154553409)

[1.3.2. Thách thức trong bài toán 12](#_Toc154553410)

[1.4. Phương pháp đánh giá mô hình 12](#_Toc154553411)

[1.4.1. Độ chính xác Accuracy 12](#_Toc154553412)

[1.4.2. Confusion matrix 13](#_Toc154553413)

[1.4.3. Precision, Recall và F1-Score trong phân loại nhị phân và phân loại đa lớp 14](#_Toc154553414)

[CHƯƠNG 2: TỔNG QUAN VỀ CÁC MẠNG NƠ-RON HỌC SÂU 17](#_Toc154553415)

[2.1. Mạng Nơ-ron nhân tạo (Artificial Neural Network – ANN) 17](#_Toc154553416)

[2.1.1. Mạng nơ-ron nhân tạo 17](#_Toc154553417)

[2.1.2. Thành phần của mạng nơ-ron nhân tạo 17](#_Toc154553418)

[2.2. Mạng học sâu (Deep learning) 18](#_Toc154553419)

[2.2.1. Khái quát về mạng học sâu 18](#_Toc154553420)

[2.2.2 Các hàm kích hoạt 18](#_Toc154553421)

[2.2.2.1 Hàm sigmoid 18](#_Toc154553422)

[2.2.2.2 Hàm tanh 19](#_Toc154553423)

[2.2.2.3 Hàm ReLU 19](#_Toc154553424)

[2.2.2.4 Hàm leaky ReLU 20](#_Toc154553425)

[2.2.3 Biểu diễn mạng nơ-ron bằng vector và ma trận 20](#_Toc154553426)

[2.3. Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) 22](#_Toc154553427)

[2.3.1. Giới thiệu về RNN 22](#_Toc154553428)

[2.3.2. Phân loại bài toán RNN 23](#_Toc154553429)

[2.3.3. Ứng dụng của RNN 23](#_Toc154553430)

[2.3.4. Ưu điểm và hạn chế của RNN 24](#_Toc154553431)

[2.4. Mạng nơ-ron Dài - Ngắn hạn (Long Short-Term Neural Networks - LSTM) 24](#_Toc154553432)

[2.4.1. Giới thiệu về LSTM 24](#_Toc154553433)

[2.4.2. Cổng quên (Forget gate) 26](#_Toc154553434)

[2.4.3. Cổng đầu vào (Input gate) 26](#_Toc154553435)

[2.4.4. Cổng đầu ra (Output gate) 28](#_Toc154553436)

[CHƯƠNG 3: ỨNG DỤNG LSTM TRONG BÀI TOÁN LOẠI CHỦ ĐỀ VĂN BẢN 30](#_Toc154553437)

[3.1. Mô hình tổng quát của bài toán phân loại chủ đề văn bản 30](#_Toc154553438)

[3.2. Xây dựng mô hình 31](#_Toc154553439)

[3.2.1. Tiền xử lý dữ liệu (Pre-processing) 32](#_Toc154553440)

[3.2.1.1. Đưa các từ về dạng nguyên bản và làm sạch văn bản 32](#_Toc154553441)

[3.2.1.2. Loại bỏ từ dừng 33](#_Toc154553442)

[3.2.1.3. Tokenization 33](#_Toc154553443)

[3.2.2. Word Embedding 33](#_Toc154553444)

[3.2.3. Cơ chế Attention 34](#_Toc154553445)

[3.2.4. Huấn luyện mô hình bằng LSTM 35](#_Toc154553446)

[3.2 Thực nghiệm mô hình phân loại chủ đề văn bản 36](#_Toc154553447)

[3.2.1 Dữ liệu thực nghiệm 36](#_Toc154553448)

[3.2.1.1. Consumer Complaint Database 36](#_Toc154553449)

[3.2.1.2. COVID News Articles (2020 - 2022) 38](#_Toc154553450)

[3.2.2 Thực nghiệm mô hình 40](#_Toc154553451)

[3.2.3 Kết quả thực nghiệm 42](#_Toc154553452)

[CHƯƠNG 4: KẾT LUẬN 43](#_Toc154553453)

[4.1 Kết quả đạt được của đề tài 43](#_Toc154553454)

[4.2 Hạn chế 43](#_Toc154553455)

[4.3 Hướng phát triển 43](#_Toc154553456)

[TÀI LIỆU THAM KHẢO 44](#_Toc154553457)

[PHỤ LỤC 46](#_Toc154553458)

**DANH SÁCH HÌNH**

[Hình 1. 1 Kiến trúc CBOW 8](#_Toc154166889)

[Hình 1. 2 Kiến trúc Skip-gram 9](#_Toc154166890)

[Hình 1. 3 Minh họa bài toán phân loại chủ đề văn bản 11](#_Toc154166891)

[Hình 1. 4 Confusion Matrix cho phân lớp nhị phân 13](#_Toc154166892)

[Hình 1. 5 Confusion Matrix cho phân loại đa lớp 14](#_Toc154166893)

[Hình 2. 1 Cấu tạo của một Mạng nơ-ron đa tầng 18](#_Toc154077691)

[Hình 2. 2 Đồ thị hàm sigmoid 19](#_Toc154077692)

[Hình 2. 3 Đồ thị hàm tanh và đồ thị đạo hàm của hàm tanh 19](#_Toc154077693)

[Hình 2. 4 Đồ thị biểu diễn hàm ReLU 20](#_Toc154077694)

[Hình 2. 5 Đồ thị biểu diễn hàm Leaky ReLU 20](#_Toc154077695)

[Hình 2. 6 Kiến trúc mạng nơ-ron 3 tầng 21](#_Toc154077696)

[Hình 2. 7 Kiến trúc của 2 tầng liên tiếp nhau trong một mạng nơ-ron đa tầng 22](#_Toc154077697)

[Hình 2. 8 Các loại bài toán RNN 23](#_Toc154077698)

[Hình 2. 9 Kiến trúc 1 LSTM chứ 4 tầng ẩn (3 sigmoid và 1 tanh) 25](#_Toc154077699)

[Hình 2. 10 Đường đi của trạng thái ô (Cell state) trong mạng LSTM 25](#_Toc154077700)

[Hình 2. 11 Một tầng ẩn mang hàm sigmoid kết hợp với một toán tử nhân 25](#_Toc154077701)

[Hình 2. 12 Cổng quên (Forget gate) 26](#_Toc154077702)

[Hình 2. 13 Cổng đầu vào (Input gate) 27](#_Toc154077703)

[Hình 2. 14 Cổng đầu ra (Output gate) 28](#_Toc154077704)

[Hình 3. 1 Mô hình tổng quát bài toán phân loại chủ đề văn bản 30](#_Toc154553475)

[Hình 3. 2 Kiến trúc mô hình phân loại chủ đề văn bản của nhóm 32](#_Toc154553476)

[Hình 3. 3 Biểu đồ thống kê số lượng văn bản mỗi lớp của tập Consumer Complaint Database 37](#_Toc154553477)

[Hình 3. 4 Biểu đồ thống kê số lượng văn bản mỗi lớp đã qua xử lý lọc của tập Consumer Complaint Database 38](file:///E:\University\Filenam4\0.TieuLuanChuyenNganh\content\TLCN_NguyenPhamNgocDuy_20133031_TriHoaiLoc_20133063%20(1).docx#_Toc154553478)

[Hình 3. 5 Biểu đồ tần suất chiều dài văn bản của tập Consumer Complaint Database 38](file:///E:\University\Filenam4\0.TieuLuanChuyenNganh\content\TLCN_NguyenPhamNgocDuy_20133031_TriHoaiLoc_20133063%20(1).docx#_Toc154553479)

[Hình 3. 6 Biểu đồ thống kê số lượng văn bản mỗi lớp của tập COVID News Articles 39](file:///E:\University\Filenam4\0.TieuLuanChuyenNganh\content\TLCN_NguyenPhamNgocDuy_20133031_TriHoaiLoc_20133063%20(1).docx#_Toc154553480)

[Hình 3. 7 Biểu đồ thống kê số lượng văn bản mỗi lớp của tập COVID News Articles đã qua xử lý lọc 39](file:///E:\University\Filenam4\0.TieuLuanChuyenNganh\content\TLCN_NguyenPhamNgocDuy_20133031_TriHoaiLoc_20133063%20(1).docx#_Toc154553481)

[Hình 3. 8 Biểu đồ tần suất chiều dài văn bản của tập COVID News Articles 40](file:///E:\University\Filenam4\0.TieuLuanChuyenNganh\content\TLCN_NguyenPhamNgocDuy_20133031_TriHoaiLoc_20133063%20(1).docx#_Toc154553482)

**DANH SÁCH BẢNG**

[Bảng 3. 1 Chủ đề và số lượng mẫu dùng trong thực nghiệm 41](#_Toc154553693)

[Bảng 3. 2 Bảng các thử nghiệm với tham số đầu vào của 2 tập dữ liệu 41](#_Toc154553694)

[Bảng 3. 3 Bảng phân tích kết quả của các thử nghiệm 42](#_Toc154553695)

# MỞ ĐẦU

**1.  Tính cấp thiết của đề tài**

Ngày nay, hệ thống pháp luật ngày càng phức tạp và đồ sộ, dẫn đến nhu cầu trong việc tìm kiếm thông tin pháp lý phải chính xác và hiệu quả. Việc xác định chính xác các mối quan hệ tham chiếu giữa các văn bản pháp quy đóng vai trò then chốt trong việc thấu hiểu nội dung pháp luật và áp dụng một cách hiệu quả. Tuy nhiên, phương pháp phân loại thủ công tốn quá nhiều thời gian, công sức và tiềm ẩn sai sót, gây cản trở cho hoạt động nghiên cứu, tra cứu và áp dụng pháp luật. Do đó, chúng ta rất cần một công cụ dùng để phân loại văn bản theo một hoặc nhiều chủ đề riêng biệt là cần thiết nhằm giúp tiết kiệm thời gian, cải thiện hiệu suất và thuận tiện cho việc tìm kiếm cũng như trích suất những thông tin dùng trong mục đích nghiên cứu. Một số công trình nghiên cứu về việc phân loại chủ đề văn bản cũng xuất hiện trong những năm gần đây và nó vẫn là đề tài nóng cho đến hiện tại.

Với tính cấp thiết đã nêu như trên và cũng nhằm mục đích nghiên cứu, nhóm tiến hành tạo ra mô hình có thể phân loại được mối quan hệ tham chiếu trong văn bản pháp luật để tự động hóa quá trình phân loại quan hệ tham chiếu, giúp tiết kiệm thời gian, công sức và nâng cao tính hiệu quả trong việc tìm kiếm thông tin pháp lý.

Bài tiểu luận này sẽ tập trung vào cách nghiên cứu sử dụng mô hình mạng bộ nhớ dài - ngắn hạn (Long – Short Term Memory - LSTM) để xây dựng mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy.

**2.  Mục tiêu nghiên cứu**

Đề tài này tập trung vào các mục tiêu chính như sau:

Thứ nhất, chúng tôi nghiên cứu tập trung vào việc hiểu rõ bài toán phân loại văn bản từ các khía cạnh khác nhau, bao gồm cách con người sử dụng ngôn ngữ tự nhiên để giao tiếp và diễn đạt ý nghĩa từ đó áp dụng vào đề tài nghiên cứu của nhóm. Mô hình LSTM sẽ được huấn luyện trên tập dữ liệu đã gán nhãn, học cách nhận diện và kết hợp các đặc trưng từ văn bản để phân loại văn bản một cách chính xác.

Thứ hai, huấn luyện và đánh giá mô hình. Mô hình sử dụng tập dữ liệu đã gán nhãn để huấn luyện mô hình LSTM, sau đó sẽ đánh giá bằng cách phương pháp đánh giá chuẩn như precision, recall và F1-score để đảm bảo hiệu suất phân loại.

Thứ ba, tối ưu hóa và cải thiện mô hình. Dựa trên kết quả đánh giá, nghiên cứu sẽ tiến hành các bước tối ưu hóa mô hình LSTM để cải thiện khả năng phân loại và đảm bảo tính ổn định của mô hình.

**3.  Cách tiếp cận và phương pháp nghiên cứu**

**3.1. Đối tượng nghiên cứu**

Mô hình mạng bộ nhớ dài - ngắn hạn (LSTM*)*

**3.2. Phạm vi nghiên cứu**

Sử dụng dữ liệu lấy từ trang [*Thư viện pháp luật*](https://thuvienphapluat.vn/) để xây dựng mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy bằng LSTM.

**4.  Phân tích những công trình có liên quan**

Bài toán phân loại quan hệ tham chiếu trong văn bản pháp quy cũng là một trong những ứng dụng lĩnh vực phân loại văn bản trong xử lý ngôn ngữ tự nhiên - một bài toán kinh điển, tuy không quá mới mẻ nhưng tới hiện tại nó vẫn là một đề tài được bàn tán sôi nổi và chưa có dấu hiệu hạ nhiệt. Mục tiêu của bài toán là ánh xạ một đoạn văn bản với một chủ đề cụ thể, đó có thể là giáo dục, thể thao, giải trí, khoa học,.... Đối với đề tài là phân loại quan hệ tham chiếu trong văn bản pháp quy thì cũng là một trong những lĩnh vực phổ biến trong bài toán này. Đã có rất nhiều nghiên cứu về phân loại văn bản trong xử lý ngôn ngữ tự nhiên cũng như về chủ đề phân loại quan hệ tham chiếu trong văn bản pháp quy với nhiều phương pháp khác nhau.

Trong nghiên cứu của Minhui Liang và Tiansen Niu với chủ đề “*Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs*” [1] năm 2022 đã đề cập đến việc kết hợp giữa hai mô hình là CNN và Long- Short Term Memory hai chiều (BiLSTM) để phân loại văn bản. Cụ thể nghiên cứu này đã chỉ ra hạn chế của CNN trong việc phân loại văn bản với nội dung ngắn cho ra độ chính xác cao nhưng sẽ bị giảm với những văn bản có nội dung dài hơn. Để khắc phục những hạn chế của CNN trong việc xử lý các vấn đề về thông tin văn bản, các nhà nghiên cứu kết hợp LSTM hai chiều với CNN để mô hình hiểu rõ hơn nội dung và mục đích của văn bản bằng cách xem xét đầy đủ ngữ nghĩa phong phú có trong ngữ cảnh bằng cơ chế quên của LSTM. Ngoài ra còn đề cập đến việc cải tiến đầu vào của BiLSTM bằng cách kết hợp TF-IDF (Term Frequency – Inverse Document Frequency) và word-to-vector (word2vec) tạo ra một tập các từ đặc trưng với độ quan trọng cao và độ chính xác của phương pháp này mang lại rất khả quan.

Với đề tài phân loại quan hệ tham chiếu trong văn bản pháp quy thì cũng đã có một nghiên cứu có tên trùng với tên đề tài là “*Phân loại quan hệ tham chiếu trong văn bản pháp quy* ” [16] của Nguyễn Thị Thanh Thủy, Đặng Bảo Chiến, Triệu Khương Duy, Ngô Xuân Bách và Từ Minh Phương đã có làm về đề tài này. Trong bài nghiên cứu này trình bày đề xuất phương pháp phân loại quan hệ tham chiếu trong văn bản pháp quy có chứa thực thể tham chiếu. Cụ thể, giả sử cho một tập dữ liệu văn bản pháp quy D đã được xác định các thực thể tham chiếu. Xét A là một văn bản trong tập D, A có thể có một hoặc nhiều tham chiếu, được ký hiệu là Bi. Với mỗi tham chiếu Bi, xét đoạn văn bản chứa tham chiếu này. Mỗi đoạn văn bản trên sẽ được sử dụng làm đầu vào cho bài toán phân loại. Mục tiêu được đề cập ở đây là với mỗi thực thể tham chiếu Bi, cần phải xác định quan hệ giữa thực thể Bi với thực thể văn bản A đang xem xét, dựa trên các thông tin đầu vào từ đoạn văn bản chứa tham chiếu Bi. Đồng tác giả, bài nghiên cứu [17] cũng là bài nghiên cứu về lĩnh vực này, bài nghiên cứu kết hợp CRF và BiLSTM để trích xuất thực thể pháp luật tham chiếu trong văn bản pháp quy Việt Nam và đạt độ chính xác trên 95%.

Để trích xuất quan hệ tham chiếu, các bài [18, 19, 20] đề cập một phương pháp tiếp cận dựa trên luật kết hợp, với phương pháp này thường cần phải xác định trước luật mô tả cấu trúc của các thực thể liên quan. Người tạo ra luật này có yêu cầu về kiến thức, hiểu biết sâu sắc về những lĩnh vực liên quan nên đó cũng là một nhược điểm lớn là cần một chuyên gia.

Trong những năm gần đây, có nhiều nghiên cứu về trích xuất quan hệ dựa trên mô hình học sâu đang và cũng được quan tâm nhiều hơn do các mô hình này có khả năng tự học đặc trưng và đã thu được nhiều kết quả tốt. Ví dụ cho phương pháp này trong nghiên cứu [20, 21]. Tuy nhiên phương pháp này có điểm hạn chế là tốn khá nhiều thời gian và cần một tập dữ liệu huấn luyện đủ lớn.

Nhìn chung những nghiên cứu trên đều đề cập tới việc cải tiến hiệu suất của mô hình bằng cách làm tăng độ tin cậy của tập các đặc trưng của văn bản đầu vào và cho kết quả khả quan nhưng vẫn có hạn chế nhất định. Với những gì mà bài nghiên cứu [16, 17] đạt được, phương pháp được đề xuất trong bài nghiên cứu này mang lại tính tham khảo cao.

**5.  Kết quả dự kiến đạt được**

Trong bài tiểu luận này, chúng tôi mong muốn có thể hiểu hơn về mạng nơ-ron nhân tạo cũng như mạng bộ nhớ dài - ngắn hạn (Long – Short Term Memory - LSTM), vấn đề xử lý ngôn ngữ tự nhiên và phương pháp nhóm áp dụng trong đề tài. Thông qua đó xây dựng một mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy có hiệu suất cao.

# CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN PHÂN LOẠI QUAN HỆ THAM CHIẾU TRONG VĂN BẢN PHÁP QUY

## Giới thiệu chung về quan hệ dẫn chiếu trong văn bản pháp quy

### Khái niệm

Văn bản pháp quy là loại văn bản do cơ quan nhà nước ban hành để điều chỉnh các hoạt động trong xã hội. Có vai trò quan trọng trong việc đảm bảo trật tự xã hội, bảo vệ quyền và lợi ích hợp pháp của tổ chức và cá nhân ví dụ: Hiến pháp, Luật, Nghị định, Thông tư...

Quan hệ dẫn chiếu trong văn bản pháp quy là mối liên hệ giữa các văn bản quy phạm pháp luật với nhau. Mối liên hệ này được thể hiện qua việc một văn bản quy phạm pháp luật (văn bản dẫn chiếu) tham chiếu đến một hoặc nhiều văn bản quy phạm pháp luật khác (văn bản được dẫn chiếu) để làm cơ sở cho việc điều chỉnh các quan hệ xã hội.

Việc xác định mối quan hệ này giúp hệ thống pháp luật được thống nhất chặt chẽ với nhau hơn tránh các mâu thuẫn tồn tại trong các văn bản pháp quy. Ngoài ra còn hỗ trợ việc áp dụng pháp luật hiệu quả, chính xác hơn.

### Cách tiếp cận

Để hiểu và áp dụng hiệu quả các quy định về quan hệ dẫn chiếu trong văn bản pháp quy, bạn cần xác định rõ ràng rằng:

* Xác định loại quan hệ dẫn chiếu.
* Phân tích nội dung dẫn chiếu, xác định cụ thể điều, khoản, mục của văn bản pháp quy được dẫn chiếu.
* Phân tích nội dung quy định tại văn bản pháp quy được dẫn chiếu.
* So sánh và đối chiếu với văn bản pháp quy tham chiếu, xác định mối liên hệ giữa nội dung quy định tại văn bản pháp quy tham chiếu và văn bản pháp quy được tham chiếu, phân tích sự phù hợp, thống nhất giữa các quy định. xác định trường hợp mâu thuẫn, xung đột (nếu có) và cách giải quyết.
* Áp dụng vào thực tiễn: Sử dụng thông tin thu thập được để giải quyết các vấn đề pháp lý cụ thể.Đảm bảo tính chính xác, hợp lý và phù hợp với quy định của pháp luật.
* Khác với ba cách tiếp cận trong [16], cách tiếp cận của nhóm sẽ xoay quanh cách kết hợp: Thực thể A (thực thể tham chiếu) + Nội dung trước của thực thể B + Thực thể B (Những thực thể được tham chiếu đề cập trong thực thể A) + Nội dung phía sau của thực thể B.

## 1.2. Xử lý ngôn ngữ tự nhiên

### 1.2.1. Giới thiệu xử lý ngôn ngữ tự nhiên

Xử lý Ngôn Ngữ Tự Nhiên (Natural Language Processing - NLP) là một lĩnh vực đầy thú vị trong trí tuệ nhân tạo, tập trung vào việc nghiên cứu và phát triển các phương pháp và công nghệ để máy tính có khả năng hiểu, xử lý và tương tác với ngôn ngữ con người. NLP không chỉ đơn thuần là việc giúp máy tính xử lý ngôn ngữ, mà còn hướng tới mục tiêu làm cho máy tính hiểu và sử dụng ngôn ngữ một cách thông minh như con người [13].

### 1.2.2. Các bước xử lý

* **Phân tích hình thái** - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong tiếng Việt, khoảng trắng  được dùng để phân tách các tiếng (âm tiết) chứ không phải từ.
* **Phân tích cú pháp** - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.
* **Phân tích ngữ nghĩa** - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.
* **Tích hợp văn bản** - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau
* **Phân tích thực nghĩa** - Cấu trúc thể hiện điều được được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

## 1.3. Các kỹ thuật xử lý ngôn ngữ tự nhiên

### 1.3.1 Tách từ (Tokenization)

Tách từ (Tokenization) là quá trình tách một cụm từ, câu, đoạn văn, một hoặc nhiều tài liệu văn bản thành các đơn vị nhỏ hơn. Mỗi đơn vị nhỏ hơn này được gọi là tokens. Tokens có thể là bất cứ thứ gì - một từ (word), một từ phụ (sub-word) hoặc thậm chí là một ký tự (character).

#### 1.3.1.1. Word-based Tokenization

Đây là phương pháp được sử dụng rộng rãi nhất, đặc biệt là trong các ngôn ngữ có khoảng trắng hoặc dấu cách để phân tách các từ, ví dụ như tiếng Anh, tiếng Tây Ban Nha. Nó chia một đoạn văn bản thành các từ hoặc âm tiết (ví dụ tiếng Việt) dựa trên dấu phân cách. Dấu phân cách này có thể là khoảng trắng hoặc dấu câu.

Hạn chế của kỹ thuật này là nó dẫn đến một kho ngữ liệu khổng lồ và một lượng từ vựng lớn, khiến mô hình cồng kềnh hơn và đòi hỏi nhiều tài nguyên tính toán hơn. Bên cạnh đó, một hạn chế nữa là liên quan đến các từ sai chính tả. Nếu kho ngữ liệu có từ “knowledge” viết sai chính tả thành “knowldge”, mô hình sẽ gán token OOV (Out-Of-Vocabulary) cho từ sau đó.

#### 1.3.1.2. Character-based Tokenization

Với những hạn chế của **Word Tokenization** thì các nhà nghiên cứu đã đưa ra phương pháp tách từ dựa trên kí tự (**Character-based Tokenization**). Kỹ thuật này văn bản thô thành các ký tự riêng lẻ. Với ý tưởng là một ngôn ngữ có nhiều từ khác nhau nhưng có một số ký tự cố định. Điều này dẫn đến một lượng từ vựng rất nhỏ. Do đó, tokenization dựa trên ký tự sẽ sử dụng ít token hơn so với mã hóa dựa trên từ.

Một trong những lợi thế chính của mã hóa dựa trên ký tự là sẽ không có hoặc rất ít từ không xác định hoặc OOV. Do đó, nó có thể biểu diễn các từ chưa biết (những từ không được nhìn thấy trong quá trình huấn luyện) bằng cách biểu diễn cho mỗi ký tự. Một ưu điểm khác là các từ sai chính tả có thể được viết đúng chính tả lại, thay vì có thể đánh dấu chúng là mã thông báo OOV và làm mất thông tin.

Tuy nhiên đây không phải là phương pháp **tokenization** tốt nhất, một ký tự thường không mang bất kỳ ý nghĩa hoặc thông tin nào như một từ. Ngoài ra, tuy kỹ thuật này giúp giảm kích thước từ vựng nhưng lại làm tăng độ dài chuỗi trong mã hóa dựa trên ký tự. Mỗi từ được chia thành từng ký tự và do đó, chuỗi mã hóa dài hơn nhiều so với văn bản thô ban đầu.

#### 1.3.1.3. Subword-based Tokenization

Đây là một kỹ thuật cũng khá phổ biến được dùng trong **Tokenization**. Đây là một giải pháp nằm giữa **Word Tokenization** và **Character Tokenization**. Ý tưởng chính là giải quyết đồng thời các vấn đề của **Word Tokenization** (kích thước từ vựng rất lớn, có nhiều tokens OOV, sự khác biệt trong ý nghĩa của các từ rất giống nhau) và **Character Tokenization** (chuỗi rất dài và token riêng lẻ ít ý nghĩa hơn).Kỹ thuật **tokenization** dựa trên từ khóa phụ sử dụng các nguyên tắc sau.

* Không chia các từ thường dùng thành các từ phụ nhỏ hơn.
* Chia các từ hiếm thành các từ phụ có ý nghĩa.

**Tokenization** dựa trên từ khóa phụ cho phép mô hình có kích thước từ vựng phù hợp và cũng có thể học các biểu diễn độc lập theo ngữ cảnh có ý nghĩa. Mô hình thậm chí có thể xử lý một từ mà nó chưa từng thấy trước đây vì sự phân tách có thể dẫn đến các từ phụ đã biết.

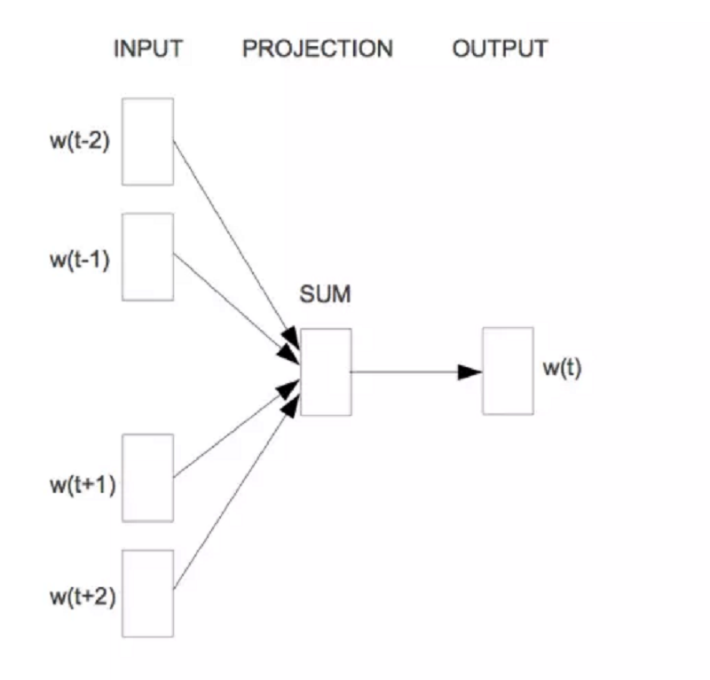
### 1.3.2. Word Embedding

**Word Embedding** ( Nhúng từ) là một kĩ thuật để biểu diễn từ dưới dạng các vector số học, nhưng điểm khác biệt đó là đây là các continuous vector và các vector này có thể học được. Bằng cách sử dụng các continuous vector này ta đã có thể thể hiện sự tương quan giữa các từ. Ví dụ trong câu “ I like listening to rock and piano music” cho dù ta có đổi chỗ “rock” và "piano" thì câu không thay đổi về mặt ngữ nghĩa nên hai từ này có sự tương quan về mặt ngữ nghĩa và cả về ngữ pháp khi cả hai đều là danh từ. Hoặc như trong câu “The leaf is green”, giữa "leaf" và "green" cũng có sự tương quan nhất định khi lá cây thường có màu xanh vì chất diệp lục có trong nó. Đã có nhiều thuật toán được đưa ra để xây dựng lên các vector đại diện cho các từ này. Sau đây là một số phương pháp được dùng phổ biến.

#### 1.2.2.1. Word2Vec

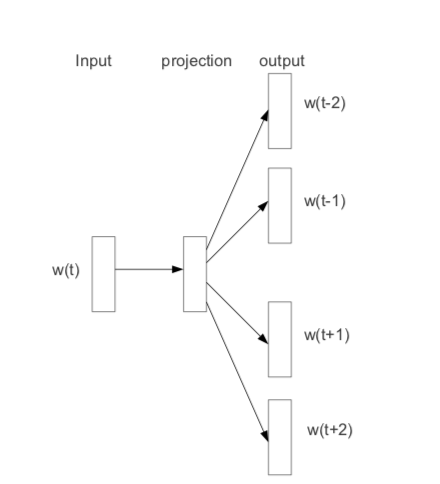
Word2Vec là một mô hình được sử dụng phổ biến nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) cũng như mạng nơ-ron được phát triển bởi một nhóm nghiên cứu tại Google vào năm 2013, dưới sự chỉ đạo của Tomas Mikolov. Word2Vec học hỏi các biểu diễn từ vựng thông qua qua việc dự đoán từ cụ thể bằng các từ ngữ cảnh dưới dạng các vector có chiều thấp trong không gian nhiều chiều dựa trên việc ánh xạ từ vựng vào các vector số thực. Có hai dạng mô hình word2vec:

* **Continuous Bag of Word (CBOW):** Ý tưởng chính của mô hình này là dự đoán từ mục tiêu dựa vào các từ ngữ cảnh xung quanh nó trong một phạm vi nhất định. Gọi từ mục tiêu ở đây là ***wi*** thì bằng cách sử dụng ***k*** từ trước và sau ***wi*** để dự đoán mà không quan tâm đến thứ tự các từ đó. Ví dụ với câu “I like dogs more than cats” và cho phạm vi k=2 thì mô hình sẽ cố dự đoán từ “dogs” dựa vào “I”, “like”, “more”, “than”.



**Hình 1. 1 Kiến trúc CBOW**

* **Skip-gram:** Ngược lại với **CBOW** thì **Skip-gram** lại dựa vào từ mục tiêu để tiến hành dự đoán những từ xung quanh nó. Cũng lấy ví dụ như phần CBOW thì nếu đưa từ “dogs” vào thì mô hình sẽ cố gắng dự đoán ra được các từ như “I”, “like”, “more” và “than”.



**Hình 1. 2 Kiến trúc Skip-gram**

#### 1.2.2.2. Glove (Global Vectors)

**GloVe (Global Vector)** là một trong những phương pháp mới để xây dựng vec-tơ từ (được giới thiệu vào năm 2014), nó thực chất được xây dựng dựa trên **Co-occurrence Matrix**. **GloVe** có bản chất là xác suất, ý tưởng xây dựng phương pháp này đến từ tỉ số sau:

(1)

Trong đó: ***P(k|i)*** là xác suất xuất hiện của từ ***k*** trong ngữ cảnh của từ ***i*** , tương tự với ***P(k|j)*** . Công thức của ***P(k|i)*** :

(2)

Với ***Xik*** là số lần xuất hiện của từ ***k*** trong ngữ cảnh của từ ***i*** (hoặc ngược lại). ***Xi*** là số lần xuất hiện của từ ***i*** trong ngữ cảnh của toàn bộ các từ còn lại ngoại trừ ***i***. (Các giá trị này chính là các mục nhập của ***Co-occurrence Matrix***)

Ý tưởng chính của ***GloVe*** làdựa vào độ tương tự ngữ nghĩa giữa hai từ ***i***, ***j*** có thể được xác định thông qua độ tương tự ngữ nghĩa giữa từ ***k*** với mỗi từ ***i***, ***j.*** Những từ ***k*** có tính xác định ngữ nghĩa tốt chính là những từ làm cho ***(1)*** >>1 hoặc xấp chỉ bằng 0. Ví dụ, nếu ***i*** là “table”, ***j*** là “cat” và ***k*** là “chair” thì ***(1)*** sẽ khá lớn do “chair” có nghĩa gần với “table” hơn là “cat”, ở trường hợp khác, nếu ta thay ***k*** là “ice cream” thì ***(1)*** sẽ xấp xỉ bằng 1 do “ice cream” hầu như chẳng lên quan gì tới “table” và “cat”.

#### 1.2.2.3. FastText

Trong **word2vec**, ta không trực tiếp sử dụng thông tin hình thái học. Trong cả mô hình skip-gram và túi từ liên tực (continuous *bag-of-word*), ta sử dụng các vector khác nhau để biểu diễn các từ ở các dạng khác nhau. Chẳng hạn, “walk” và “walks” được biểu diễn bởi hai vector khác nhau, trong khi mối quan hệ giữa hai vector đó không biểu thị trực tiếp trong mô hình. Từ quan điểm này, FastText đề xuất phương thức embedding từ con (*subword embedding*), thông qua việc thực hiện đưa thông tin hình thái học vào trong mô hình skip-gram trong word2vec.

Trong fastText, mỗi từ trung tâm được biểu diễn như một tập hợp của các từ con. Giả sử ta sử dụng từ “work” làm ví dụ để hiểu cách các từ tố được tạo thành. Trước hết, ta thêm một số ký tự đặc biệt “<” và “>” vào phần bắt đầu và kết thúc của từ để phân biệt các từ con được dùng làm tiền tố và hậu tố. Rồi ta sẽ xem từ này như một chuỗi các ký tự để trích xuất *n*-grams. Chẳng hạn, khi *n*=3, ta có thể nhận tất cả từ tố với chiều dài là 3:

“<wo”, “wor”, “ork”, “rk>”

Và từ con đặc biệt là “<work>”.

Với một từ *w*, ta ghi tập hợp của tất cả các từ con của nó với chiều dài từ 3 đến 6 và các từ con đặc biệt là G*w*. Do đó, từ điển này là tập hợp các từ con của tất cả các từ. Giả sử vector của từ con *g* trong từ điển này là **z***g*. Thì vector từ trung tâm **u***w* cho từ *w* trong mô hình skip-gram có thể biểu diễn là

(3)

## 1.3. Bài toán phân loại quan hệ tham chiếu trong văn bản pháp quy

### 1.3.1. Khái niệm của bài toán

Về mặt bản chất, bài toán phân loại quan hệ tham chiếu trong văn bản pháp quy là một trong những ứng dụng của bài toán phân loại văn bản trong xử lý ngôn ngữ tự nhiên. Bài toán phân loại văn bản là một bài toán xử lý ngôn ngữ tự nhiên cổ điển. Bài toán ánh xạ một văn bản vào một chủ đề cụ thể trong một tập hữu hạn các chủ đề đã biết dựa trên ngữ nghĩa của văn bản [14]. Việc phân lớp một cách tự động này để quá trình phân loại văn bản vào một chủ đề nào đó giúp cho việc sắp xếp, lưu trữ và truy vấn tài liệu dễ dàng hơn.

Về mặt bản chất, một văn bản là một tập các từ có liên quan với nhau để rồi tạo nên nội dung ngữ nghĩa của văn bản. Từ ngữ của một văn bản là đa dạng do tính đa dạng của ngôn ngữ (đồng nghĩa, đa nghĩa, từ vay mượn nước ngoài,…) và số lượng từ cần xét là lớn. Ở đây cần lưu ý rằng, một văn bản có thể có số lượng từ ngữ không nhiều, nhưng số lượng từ ngữ cần xét là rất nhiều vì phải bao hàm tất cả các từ của ngôn ngữ đang xét [14].

Phân loại quan hệ tham chiếu trong văn bản pháp quy thay vì phân loại dựa trên một thực thể như bài toán phân loại văn bản thông thường, bài toán sẽ phân loại dựa một thực thể tham chiếu với một hoặc nhiều thực thể được tham chiếu để xác định mối quan hệ giữa hai thực thể đó với nhau, đó có thể là *Căn cứ, Dẫn chiếu, Hết hiệu lực,…*.

Các đối tượng ở đây là *Hiến pháp, Luật, Thông tư, Nghị định, Quyết định, ....* Từ đó ta có thể chia bài toán làm hai phần:

* Trích xuất đối tượng được tham chiếu. Với những văn bản pháp quy được ban hành thường thì ta đã xác định được đối tượng tham chiếu ở đây chính là bản thân văn bản đó, đại diện cho văn bản này sẽ là tên của văn bản ban hành. Còn những đối tượng được đề cập tới trong văn bản sẽ là những đối tượng được tham chiếu.
* Sau khi đã trích xuất được các đối tượng thì tiến hành phân loại mối quan hệ giữa văn bản tham chiếu đó với các đối tượng được đề cập bên trong có mối quan hệ gì với nhau.

### 1.3.2. Thách thức trong bài toán

* Đa dạng về quan hệ tham chiếu vì số lượng văn bản pháp quy lớn và rất đa dạng.
* Vì sự đa dạng của văn bản pháp quy dẫn đến mối quan hệ giữa các thực thể có thể trở nên phức tạp, mơ hồ khó xác định rõ ràng.
* Luật là một ngành điển hình với những từ ngữ chuyên môn cao và phức tạp, muốn hiểu rõ các văn bản pháp quy thì cần có kiến thức và chuyên môn cao mới có thể làm được. Do vậy với những người tay ngang vào rất khó để xác định đúng được mối quan hệ giữa các thực thể trong các văn bản pháp quy với nhau.

## 1.4. Phương pháp đánh giá mô hình

Trong xuyên suốt quá trình huấn luyện mô hình học máy của một bài toán, để biết hiệu suất của một mô hình có đạt yêu cầu hay không chúng ta sẽ dùng các phương pháp độ đo phù hợp với bài toán dựa trên tập dữ liệu kiểm thử (tập test). Đối với bài toán phân lớp, hiêu suất của mô hình thường được đánh giá qua khả năng phân lớp cũng như độ chính xác dựa trên lớp mà mô hình dự đoán cho các điểm dữ liệu trong tập kiểm thử (tập test) với lớp thật sự của các điểm dữ liệu đó.

Có rất nhiều cách đánh giá cho mộp bài toán phân lớp, tùy vào những bài toán khác nhau mà chúng ta sử dụng những phương pháp khác nhau. Các phương pháp thường được sử dụng là: *Accuracy (Độ chính xác),* *Confusion* *matrix*, *Precision*-*Recall*, *F1*-*score*.

### 1.4.1. Độ chính xác Accuracy

Đây là một cách để đo lường tần suất thuật toán phân loại một điểm dữ liệu (data point) một cách chính xác. Độ chính xác (accuracy) là số điểm dữ liệu được dự đoán chính xác trong số tất cả các điểm dữ liệu.

Về mặt hình thức, độ chính xác (accuracy) có định nghĩa sau:

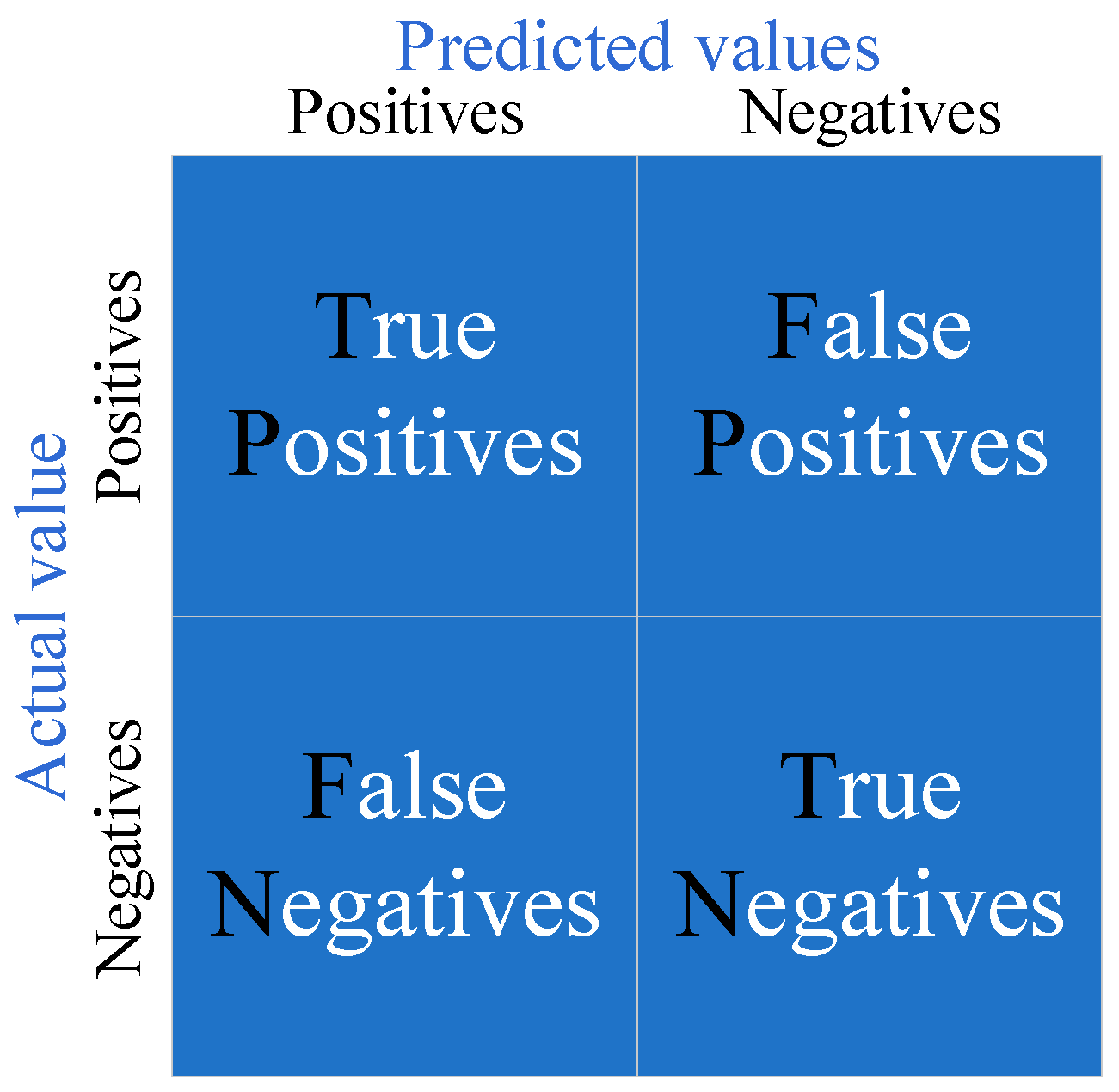
Accuracy =

### 1.4.2. Confusion matrix

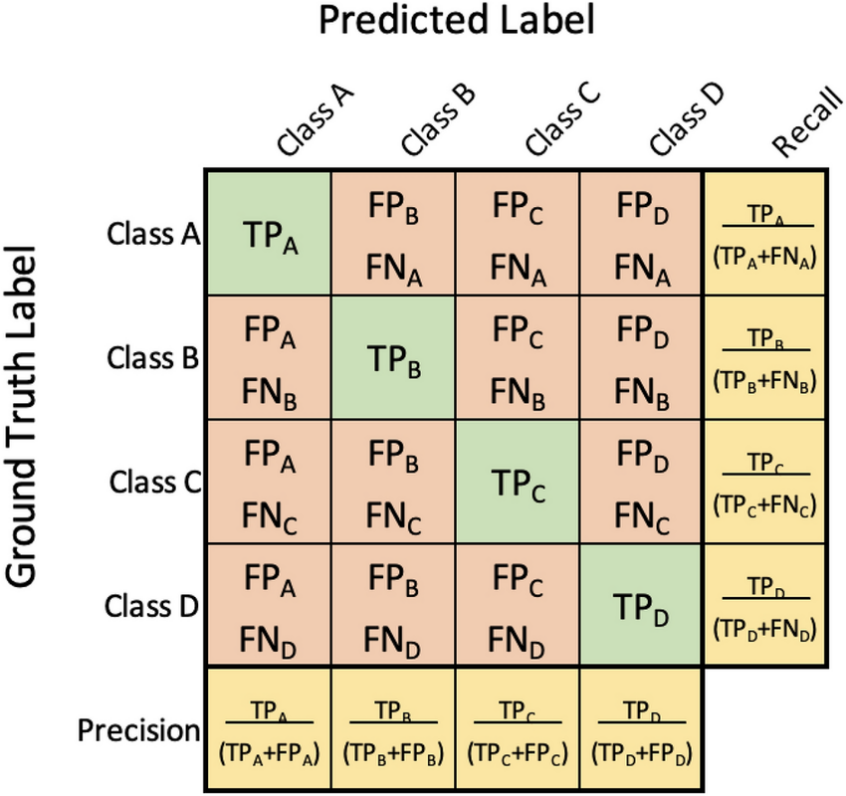
*Confusion matrix* cũng là một trong những phương pháp để đánh giá kết quả cho những bài toán dự đoán hay phân loại. Nó cung cấp một cái nhìn tổng quan về cách mô hình trên các lớp khác nhau. *Confusion matrix* thường được sắp xếp dưới dạng một ma trận *n* x *n* với n là số lớp phân loại, trong đó mỗi hàng của ma trận biểu thị lớp thực tế và mỗi cột biểu thị lớp được dự đoán.

Dựa vào Hình 1.4 và Hình 1.5, ta có các khái niệm cần chú ý:

* True Positive (TP): là trường hợp mô hình dự đoán một điểm dữ liệu thuộc lớp dương (positive), và thực tế nó cũng thuộc lớp dương.
* True Negative (TN): là trường hợp mô hình dự đoán một điểm dữ liệu thuộc lớp âm (negative), và thực tế nó cũng thuộc lớp âm.
* False Positive (FP): là trường hợp mô hình dự đoán một điểm dữ liệu thuộc lớp dương, nhưng thực tế không nó thuộc lớp dương.
* False Negative (FN): là trường hợp mô hình dự đoán một điểm dữ liệu thuộc lớp âm, nhưng thực tế nó không thuộc lớp âm.



**Hình 1. 3 Confusion Matrix cho phân lớp nhị phân**



**Hình 1. 4 Confusion Matrix cho phân loại đa lớp**

### 1.4.3. Precision, Recall và F1-Score trong phân loại nhị phân và phân loại đa lớp

*F1-score* là một độ đo đánh giá hiệu suất của một mô hình phân loại. Nó được sử dụng chủ yếu trong lĩnh vực Machine Learning và Data Science để đánh giá chất lượng của mô hình phân loại. *F1-score* kết hợp cả *Precision* và *Recall*, hai độ đo quan trọng trong việc đánh giá hiệu suất mô hình phân loại. Các chỉ số này có các cách tính khác nhau dựa vào bài toán là phân lớp nhị phân hay phân lớp đa lớp. Đối với phân lớp nhị phân là loại phân lớp dữ liệu chỉ thuộc 1 trong 2 lớp. Còn với phân lớp đa lớp là việc phân loại dữ liệu thuộc một lớp trong tổng số lớp (lớn hơn 2) mà tập dữ liệu có.

*Precision* (chính xác) là tỷ lệ giữa số lượng dự đoán đúng positive (true positive) và tổng số lượng positive được dự đoán (kết hợp true positive và false positive). Nếu mô hình có precision cao, nghĩa là mô hình dự đoán các positive case khá chính xác. Với bài toán phân loại nhị phân thì Precision có công thức như sau:

*Recall* (phục hồi) là tỷ lệ giữa số lượng dự đoán đúng positive và tổng số lượng positive trong dữ liệu thực tế (kết hợp true positive và false negative). Nếu mô hình có Recall cao, nghĩa là mô hình dự đoán đúng khá nhiều positive case. Công thứ tính Recall trong phân lớp nhị phân:

*F1-score* là sự kết hợp của *Precision* và *Recall*. *F1-score* có giá trị từ 0 đến 1, và giá trị càng cao càng tốt. Được tính bằng công thức:

*F1-score* =

Với bài toán phân loại đa lớp, ta có thể đưa về bài toán phân lớp nhị phân bằng cách xem xét từng lớp. Với mỗi lớp, ta coi dữ liệu thuộc lớp đó có nhãn là positive, tất cả các dữ liệu còn lại có nhãn là negative. Sau đó, giá trị *Precision*, *Recall* được áp dụng lên từng lớp. Với mỗi lớp, ta sẽ nhận được một cặp giá trị *Precision* và *Recall*. Tuy nhiên, với các bài toán có rất nhiều lớp dữ liệu, việc này đôi khi không khả thi. Thay vào đó, hai phép đánh giá dựa trên Precision và Recall áp dụng cho việc phân loại đa lớp được sử dụng là *Micro-average* và *Macro-average*.

Với *Micro-average* tính toán độ đo cho tất cả các mẫu và lớp, sau đó tính trung bình. Nó trực tiếp sử dụng số lượng True Positives (TP), True Negatives (TN), False Positives (FP), và False Negatives (FN) trên toàn bộ các lớp. *Prcesion* và *Recall* được tính như sau:

Trong đó, TPi, FPi , FNi lần lượt là TP, FP, FN của lớp *i*. Tức TP được tính là tổng của toàn bộ TP của mỗi lớp. Tương tự với FP và FN. *Micro-average F1-score* sẽ được tính tương tự *F1-score* cho phân lớp nhị phân nhưng sẽ dựa trên *Micro-average Precision* và *Micro-average Recall*

Còn với *Macro-average* sẽ tính toán độ đo cho từng lớp riêng lẻ và sau đó tính trung bình các giá trị này. Điều này đảm bảo rằng mỗi lớp được đối xử một cách công bằng, không phụ thuộc vào kích thước của lớp đó. Nói như vậy *Macro-average* *Precision* và *Macro-average* *Recall* được tính bằng cách trung bình cộng của các *Precision/ Recall* theo class cho tổng số class *C.* Công thức cụ thể như sau:

Cách tính *Macro-average F1-score* cũng tương tự *F1-score* cho phân lớp nhị phân nhưng sẽ dựa trên chỉ số *Macro-average Precision* và *Macro-average Recall*.

# CHƯƠNG 2: TỔNG QUAN VỀ CÁC MẠNG NƠ-RON HỌC SÂU

## 2.1. Mạng Nơ-ron nhân tạo (Artificial Neural Network – ANN)

### 2.1.1. Mạng nơ-ron nhân tạo

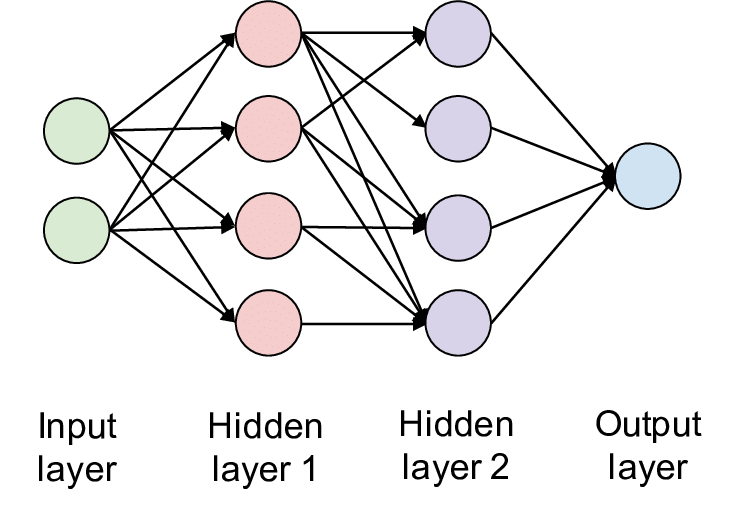
Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) về cơ bản có thể xem là một hệ thống tính toán. ANN cũng được đặt tên là "hệ thống thần kinh nhân tạo" hoặc "hệ thống xử lý phân tán song song" hoặc "hệ thống kết nối". Được lấy ý tưởng từ hệ thống mạng nơ-ron thần kinh của bộ não, mỗi tế bào thần kinh được kết nối với các tế bào thần kinh khác thông qua một liên kết kết nối. Mỗi liên kết kết nối được liên kết với một trọng số có thông tin về tín hiệu đầu vào. Đây là thông tin hữu ích nhất cho các tế bào thần kinh để giải quyết một vấn đề cụ thể vì trọng lượng thường kích thích hoặc ức chế tín hiệu đang được truyền đạt. Mỗi tế bào thần kinh có một trạng thái bên trong, được gọi là tín hiệu kích hoạt. Tín hiệu đầu ra, được tạo ra sau khi kết hợp các tín hiệu đầu vào và quy tắc kích hoạt, có thể được gửi đến các đơn vị khác.

### 2.1.2. Thành phần của mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo bao gồm nhiều lớp, mỗi lớp sẽ bao gồm nhiều nút khác nhau. Mỗi nút trong các lớp, hay còn có thể xem là một nơ-ron nhân tạo, được kết nối với nhau và mang những trọng số và ngưỡng có liên quan. Nếu như kết quả đầu ra của nút cao hơn ngưỡng được chỉ định thì nút đó sẽ được được kích hoạt và truyền dữ liệu đến cho lớp tiếp theo. Ngược lại, nếu dữ liệu đầu ra của nút không đạt được ngưỡng chỉ định thì sẽ không có dữ liệu nào được truyền đi.

Cụ thể, thành phần cấu tạo của một Mạng nơ-ron nhân tạo bao gồm:

* *Tầng dữ kiện (input layer)*: Là tầng đầu tiên của mạng, ở tầng này thể hiện các dữ kiện đầu vào.
* *Tầng kết quả (output layer)*: Tầng nằm ở vị trí cuối cùng trong mạng, tầng này thể hiện kết qủa đầu ra của mạng.
* *Tầng ẩn (hidden layer)*: Tầng này nằm ở giữa hai tầng trên, tầng này được dùng để tính toán và biến đổi dữ kiện ra đến kết quả.



**Hình 2. 1 Cấu tạo của một Mạng nơ-ron đa tầng**

## 2.2. Mạng học sâu (Deep learning)

### 2.2.1. Khái quát về mạng học sâu

Mạng học sâu có thể được xem như một mạng đa tầng với số tầng là khá nhiều, trong đó số lượng perceptron ở mỗi tầng có thể khác nhau và mỗi tầng có thể sử dụng một hàm kích hoạt khác nhau. Điều này khác với mạng đa tầng truyền thống ban đầu chỉ có một hàm kích hoạt dùng chung cho toàn mạng.

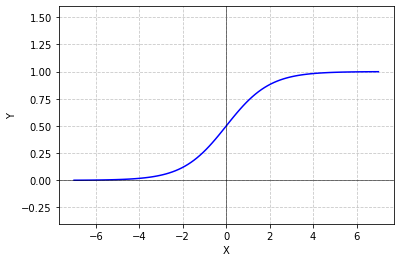
### 2.2.2 Các hàm kích hoạt

#### 2.2.2.1 Hàm sigmoid

Công thức:

(4)

Hàm sigmoid nhận vào một giá trị thực x và trả về một giá trị trong khoảng (0,1). Nếu x là một số thực âm rất nhỏ thì kết quả của hàm sigmoid sẽ tiệm cận 0, và ngược lại nếu x là một số dương rất lớn thì kết quả sẽ tiệm cận 1. Hình 2.2 là đồ thị biểu diễn cho hàm sigmoid.



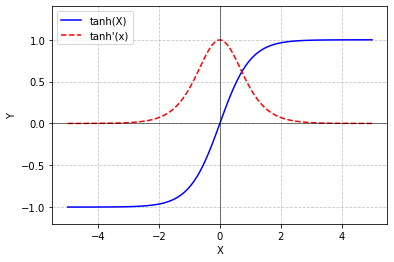
**Hình 2. 2 Đồ thị hàm sigmoid**

#### 2.2.2.2 Hàm tanh

Công thức:

(5)

Hàm tanh nhận vào một số thực và trả về một giá trị trong khoảng (-1, 1). Hình 2.3 là đồ thị của hàm tanh và đồ thị đạo hàm của hàm tanh.



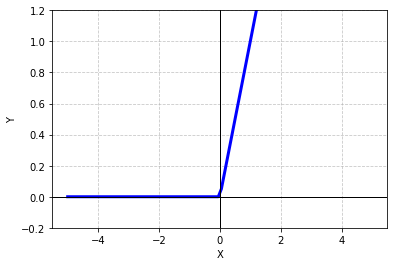
**Hình 2. 3 Đồ thị hàm tanh và đồ thị đạo hàm của hàm tanh**

#### 2.2.2.3 Hàm ReLU

Công thức:

(6)

Hàm ReLU là bộ lọc các giá trị đầu vào nhỏ hơn 0. Hình 2.4 là đồ thị biểu diễn hàm ReLU.



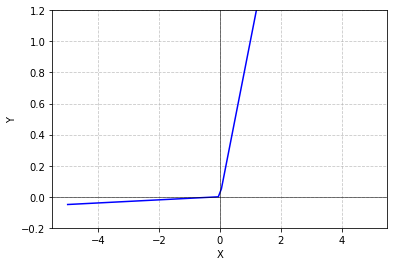
**Hình 2. 4 Đồ thị biểu diễn hàm ReLU**

#### 2.2.2.4 Hàm leaky ReLU

Công thức:

(7)

Leaky ReLU là một biến thể của ReLU để cố gắng xử lý hiện tượng Dying ReLU. Thay vì luôn trả về giá trị bằng 0 cho các giá trị âm, leaky ReLU tạo một đường xiên có độ dốc nhỏ. Hình 2.5 là đồ thị biểu diễn hàm Leaky ReLU.



**Hình 2. 5 Đồ thị biểu diễn hàm Leaky ReLU**

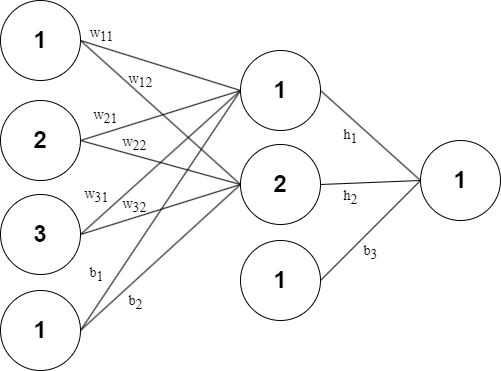
### 2.2.3 Biểu diễn mạng nơ-ron bằng vector và ma trận

Xem xét mạng nơ-ron gồm 3 tầng như hình 2.6. Khi đó, với là một hàm kích hoạt, ta có:

(8)

(9)

(10)



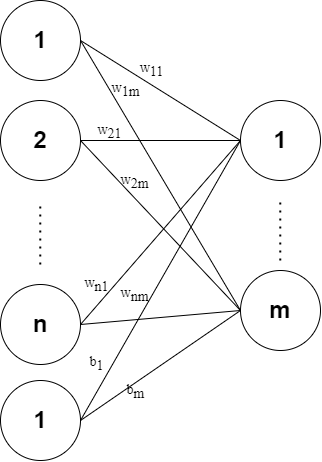
**Hình 2. 6 Kiến trúc mạng nơ-ron 3 tầng**

Quá trình tính toán ở mỗi perceptron sẽ được tính bằng cách lấy giá trị đầu ra của tầng trước đó hoặc là giá trị đầu vào nhân với trọng số tương ứng của nó, quá trình này sẽ được biểu diễn bằng kí hiệu toán học như sau:

(11)

Trong đó, num\_input là số lượng dữ liệu đầu vào được dùng để tính toán trong tầng hiện tại. Công thức này thực chất là 2 ma trận nhân với nhau, mỗi đầu ra của một perceptron là một giá trị vô hướng (scalar) và do mối quan hệ giữa 2 tầng liên tiếp là như nhau nên không mất tính tổng quát, chúng ta chỉ xét riêng trường hợp với 2 tầng.

Nếu dữ liệu đầu vào được biểu diễn thành 1 vector hay nói cách khác là một ma trận thì ma trận trọng số W sẽ xây dựng tương ứng với n hàng, m cột với là trọng số từ input I nối với một output j.



**Hình 2. 7 Kiến trúc của 2 tầng liên tiếp nhau trong một mạng nơ-ron đa tầng**

Dựa theo công thức (5), và xét là ma trận của các perceptron đầu ra thì sẽ có công thức như sau:

(12)

Việc biểu diễn mạng nơ ron dưới dạng ma trận như đã trình bày rất hữu ích cho việc tính toán khi thực hiện các phép lan truyền xuôi và lan truyền ngược.

## 2.3. Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN)

### 2.3.1. Giới thiệu về RNN

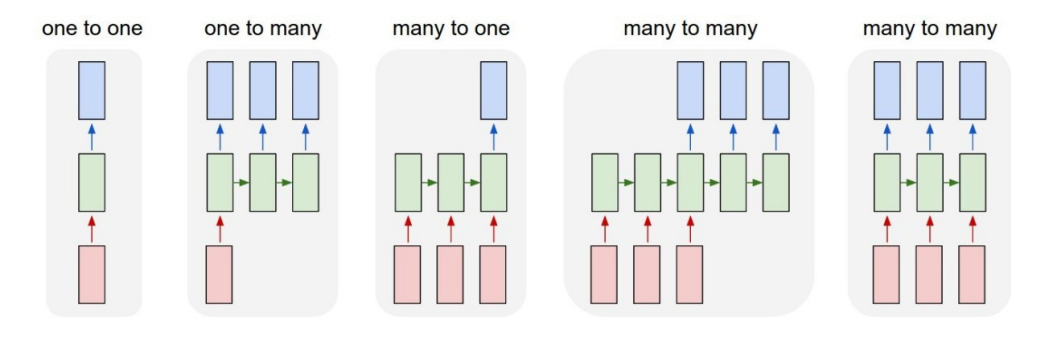
Recurrent Neural Network – RNN là một mạng no-ron nhân tạo dạng chuỗi và là một mạng nhân tạo hai chiều. Trong đó kết quả của một nút có thể ảnh hưởng đến đầu vào của nút tiếp theo nó.

Với các mạng thần kinh truyền thống, tất cả các đầu vào và đầu ra đều độc lập với nhau, nhưng trong trường hợp bắt buộc phải dự đoán từ tiếp theo của câu, các từ trước đó là bắt buộc và do đó cần phải nhớ các từ trước đó. Do đó, RNN ra đời, giải quyết vấn đề này với sự trợ giúp của Hidden Layer. Tính năng chính và quan trọng nhất của RNN là trạng thái Ẩn, ghi nhớ một số thông tin về một chuỗi. Trạng thái này còn được gọi là Trạng thái bộ nhớ vì nó ghi nhớ đầu vào trước đó vào mạng. Nó sử dụng các tham số giống nhau cho mỗi đầu vào vì nó thực hiện cùng một tác vụ trên tất cả các đầu vào hoặc các lớp ẩn để tạo ra đầu ra. Điều này làm giảm sự phức tạp của các tham số, không giống như các mạng thần kinh khác. Nhờ vào đó mà Mạng nơ-ron hồi quy được áp dụng nhiều vào các bài toán xử lý dữ liệu dạng chuỗi như: khởi tạo văn bản, mô hình hóa ngôn ngữ, nhận dạng giọng nói, phân loại văn bản, ….

### 2.3.2. Phân loại bài toán RNN

Bài toán RNN có bốn dạng bài toán khác nhau dựa vào đầu vào cũng như đầu ra của bài toán. Cụ thể bao gồm:

* **One to One:** Dạng bài toán này hoạt động như đa số mạng nơ-ron thông thường khác, gồm một đầu vào và một đầu ra.
* **One to Many:** Loại này sẽ gồm một đầu vào và nhiều đầu ra.
* **Many to One:** Đây là dạng bài toán gốm nhiều đầu vào nhưng đầu ra chỉ có một, bài toán phân loại văn bản cũng là một dạng bài toán Many to Many này.
* **Many to Many:** Bài toán này bao gồm nhiều đầu vào và nhiều đầu ra, bài toán tóm tắt văn bản hay bài toán dịch thuật cũng là một dạng của bài toán này.



**Hình 2. 8 Các loại bài toán RNN**

### 2.3.3. Ứng dụng của RNN

Như đã đề cập trước đó thì RNN được sinh ra để giải quyết các vấn đề về dữ liệu dạng chuỗi trong Deep learning vì vậy ứng dụng của RNN đa phần sẽ có liên quan đến các dữ liệu dạng này. Có thể kể đến như các ứng dụng điển hình sau:

* **Speech to text**: Chuyển giọng nói sang text.
* **Sentiment classification**: phân loại số sao cho các bình luận, ví dụ: input: “ứng dụng tốt”, output: 4 sao.
* **Machine translation**: Bài toán dịch tự động giữa các ngôn ngữ.
* **Video recognition**: Nhận diện hành động trong video.
* **Heart attack**: Dự đoán đột quỵ tim.

### 2.3.4. Ưu điểm và hạn chế của RNN

Với những gì đã đề cập ta có thể thấy được rằng RNN có một só ưu điểm là phù hợp với những bài toán có dữ liệu đầu vào thuộc dạng chuỗi có độ dài khác nhau và lưu trữ thông tin trước đó cho những nút tiếp theo.

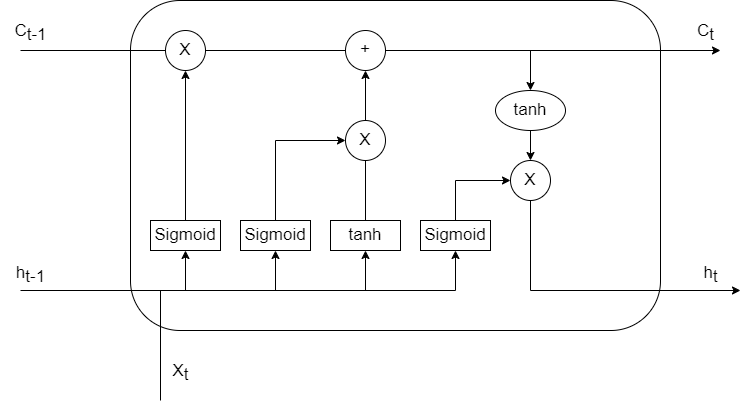
Dù vậy, RNN cũng có những hạn chế nhất định.

* Trong quá trình huấn luyện, thông tin có thể bị mất đi hoặc trở nên quá lớn, gây ra vấn đề trong việc học tốt trên các chuỗi dữ liệu dài. (vấn đề về vanishing/exploding gradient).
* RNN có thể khó huấn luyện hiệu quả, đặc biệt là với những chuỗi dữ liệu dài do bộ nhớ ngắn hạn.

## 2.4. Mạng nơ-ron Dài - Ngắn hạn (Long Short-Term Neural Networks - LSTM)

### 2.4.1. Giới thiệu về LSTM

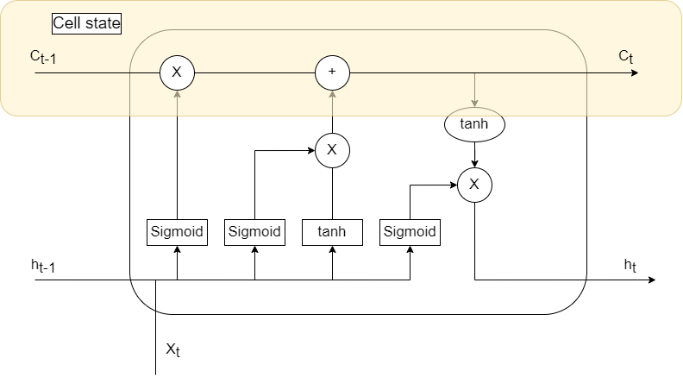
Mạng nơ-ron Dài - Ngắn hạn (Long Short-Term Memory - LSTM) là một dạng đặc biệt của mạng nơ-ron hồi quy (RNN) của hai tác giả Hochreiter và Schmidhuber, 1997 [10], được thiết kế để học hiệu quả sự phụ thuộc dài hạn nhằm giải quyết những hạn chế của RNN khi xử lý thông tin cần nhớ trong thời gian dài. Kiến trúc của LSTM là một dạng chuỗi các ô nhớ (Memory cells), mỗi ô nhớ sẽ có bốn tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt [11][12].



**Hình 2. 9 Kiến trúc 1 LSTM chứ 4 tầng ẩn (3 sigmoid và 1 tanh)**

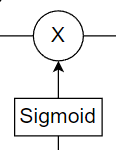
Cấu tạo của một ô nhớ LSTM gồm:

* Trạng thái ô (Cell state) được hiểu là một băng chuyền lưu trữ thông tin trải dài suốt toàn bộ chuỗi các ô nhớ [11].



**Hình 2. 10 Đường đi của trạng thái ô (Cell state) trong mạng LSTM**

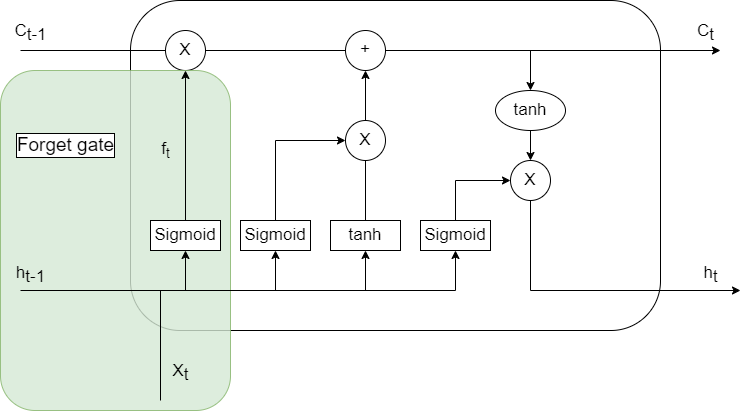
* Các cổng là các điểm kiểm soát thông tin của trạng thái ô, chúng được tạo ra bằng cách kết hợp một tầng ẩn và một toán tử như hình 2.11.



**Hình 2. 11 Một tầng ẩn mang hàm sigmoid kết hợp với một toán tử nhân**

### 2.4.2. Cổng quên (Forget gate)

Cổng quên sẽ quyết định thông tin cần loại bỏ hay giữ lại trong trạng thái ô (Cell state). Thông tin đầu vào sẽ đi qua tầng ẩn hàm sigmod . Nếu thể hiện giữ toàn bộ thông tin cũ, nếu thể hiện loại bỏ toàn bộ thông tin cũ trong trạng thái ô (cell state).



**Hình 2. 12 Cổng quên (Forget gate)**

Phương trình hàm sigmoid ở cổng quên:

(13)

Trong đó:

: Vector đầu ra của trạng thái ẩn trước đó tại thời điểm t-1

: Vector đầu vào tại thời điểm t

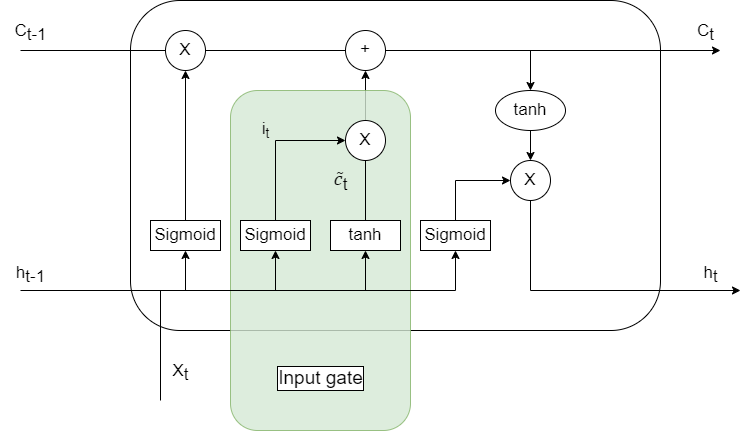
: Ma trận trọng số của cổng quên

: Vector độ lệch (bias) của cổng quên

: Cổng quên

### 2.4.3. Cổng đầu vào (Input gate)

Cổng đầu vào sẽ quyết định thông tin được thêm vào trạng thái ô (cell state). Thông tin đầu vào đi qua tầng ẩn hàm sigmoid để lọc thông tin cần thêm vào trạng thái ô và Cùng thông tin đầu vào ban đầu đi tiếp qua tầng ẩn hàm tanh với miền giá trị nằm trong khoảng [-1,1] tạo ra một vector thông tin mới . Kết hợp 2 tầng trên bằng toán tử nhân. Khi thì vector thông tin mới sẽ được thêm vào trạng thái ô, ngược lại nếu thì vector thông tin mới sẽ không được cập nhật vào trạng thái ô.



**Hình 2. 13 Cổng đầu vào (Input gate)**

Phương trình hàm sigmoid ở cổng đầu vào:

(14)

Trong đó:

: Vector đầu ra của trạng thái ẩn trước đó tại thời điểm t-1

: Vector đầu vào tại thời điểm t

: Ma trận trọng số của cổng đầu vào

: Vector độ lệch (bias) của cổng đầu vào

: Cổng đầu vào

Phương trình hàm tanh ở cổng đầu vào:

(15)

Trong đó:

: Vector đầu ra của trạng thái ẩn trước đó tại thời điểm t-1

: Vector đầu vào tại thời điểm t

: Ma trận trọng số của hàm tanh

: Vector độ lệch (bias) của hàm tanh

: Vector thông tin mới

Phương trình cập nhật trạng thái ô (cell state) ở thời điểm t:

(16)

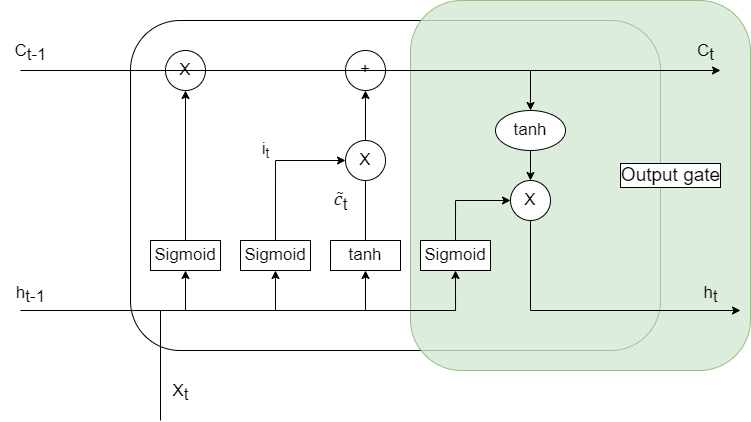
Trong đó:

: là trạng thái ô ở thời điểm t-1

: là trạng thái ô ở thời điểm t

### 2.4.4. Cổng đầu ra (Output gate)

Cổng đầu ra sẽ quyết định thông tin nào được truyền đi từ trạng thái ẩn tới đầu ra. Thông tin đầu vào đi qua tầng ẩn hàm sigmoid để đưa ra quyết định. Tiếp theo, trạng thái ô được đưa qua hàm tanh để chuyển giá trị về khoảng -1 và 1. Sau đó, nhân nó với để ra được trạng thái ẩn tại thời điểm hiện tại . Cuối cùng, trạng thái ô và trạng thái ẩn được đưa qua ô nhớ (memory cell) tiếp theo.



**Hình 2. 14 Cổng đầu ra (Output gate)**

Phương trình hàm sigmoid của cổng đầu ra:

(17)

Trong đó:

: Vector đầu ra của trạng thái ẩn trước đó tại thời điểm t-1

: Vector đầu vào tại thời điểm t

: Ma trận trọng số của cổng đầu ra

: Vector độ lệch (bias) của cổng đầu ra

: Cổng đầu ra

Phương trình trạng thái ẩn ở thời điểm t:

(18)

Trong đó: là trạng thái ẩn ở thời điểm t

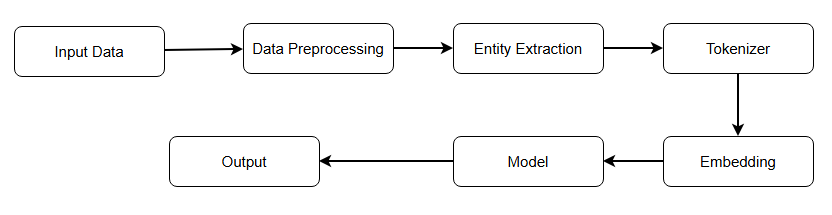
# **CHƯƠNG 3: ỨNG DỤNG LSTM TRONG BÀI TOÁN PHÂN LOẠI QUAN HỆ THAM CHIẾU TRONG VĂN BẢN PHÁP QUY**

## **3.1. Mô hình tổng quát của bài toán phân loại quan hệ tham chiếu trong văn bản pháp quy**

Mô hình tổng quát của bài toán phân loại quan hệ tham chiếu là phiên bản đơn giản của mô hình. Đầu vào là một đoạn văn bản, sau khi trải qua các bước tiền xử lý trong bài toán xử lý ngôn ngữ tự nhiên như chuẩn hóa bảng kí tự, loại bỏ phần dư, tách các âm tiết dính nhau…. Quá trình này được thực hiện trước khi trích xuất các đối tượng nhằm mục đích hạn chế các nhân tố gây nhiễu cho quá trình huấn luyện mô hình.

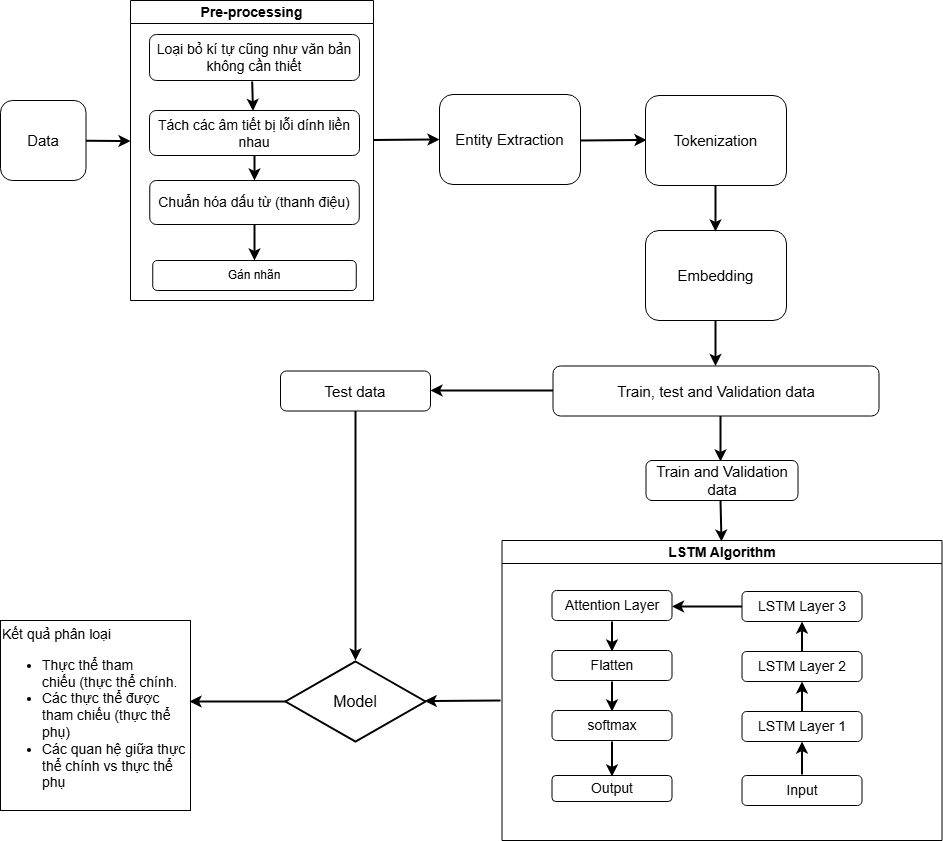
Sau tiền xử lý dữ liệu, dữ liệu được trích xuất ra các thông tin cần thiết như thực thể tham chiếu, thực thể được tham chiếu. Việc trích xuất thông tin này nhằm xác định các đối tượng để phân loại quan hệ cũng như những thông tin ngữ cảnh phục vụ cho quá trình huấn luyện cũng như phân loại.

Sau đó, dữ liệu sẽ được tokenize và biểu diễn dưới dạng các vector số sau đó tạo ma trận Embedding và làm đầu vào cho mô hình học máy để tiến hành huấn luyện. Cuối cùng thực hiện tính toán để cho đầu ra là một hoặc nhiều quan hệ giữa thực thể đang xét với một hoặc nhiều thực thể được tham chiếu.

***Hình 3.1: Mô hình tổng quát bài toán phân loại quan hệ tham chiếu của văn bản pháp lý***

## **3.2. Xây dựng mô hình phân loại quan hệ tham chiếu trong văn bản pháp quy**

Dựa vào mô hình tổng quát ở hình 3.1, Nhóm xây dựng một mô hình chi tiết các công việc theo trình tự thực hiện cho việc huấn luyện, với các phần trọng điểm gồm tiền xử lý dữ liệu, trích xuất thông tin và mô hình huấn luyện phân loại quan hệ trong văn bản pháp quy.

***Hình 3.2: Kiến trúc mô hình phân loại quan hệ trong văn bản pháp quy***

### 3.2.1. Xây dựng tập dữ liệu

Khi đã lấy được dữ liệu từ trang [*Thư viện pháp luật*](https://thuvienphapluat.vn/)sẽ tiến hành trích xuất các đối tượng bên trong văn bản, trong quá trình trích xuất sẽ có các bước xử lý về từ ngữ như chuẩn hóa bảng kí tự, loại bỏ các kí tự thừa, tách các âm tiết bị dính vào nhau trong quá trình lấy dữ liệu từ website.

Trong quá trình thực nghiệm, có 2 trường hợp mà nhóm đã thử nghiệm. Trường hợp một là giữ nguyên dấu “ / ” trong số hiệu của văn bản pháp luật và không chuẩn hóa các đối tượng về dạng chữ thường, lý do có trường hợp này xảy ra là vì dấu “ / ” là đặc trung cho số hiệu của văn bản pháp luật đang và việc không đưa các đối tượng như Luật, Hiến pháp, Nghị định ... về cùng một dạng chữ thường là nhằm làm nổi bậc thông tin cần chú ý trong quá trình học của mô hình.

Nhưng với trường hợp hai là loại bỏ dấu “ / ” trong số hiệu và đưa toàn bộ chuỗi đầu vào về dạng chữ thường lại cho kết quả huấn luyện tốt hơn trường hợp một. Nhóm tìm được lý do cho việc này là vì dấu “ / ” có thể được xem như các từ hoặc dấu hiệu riêng biệt, làm tăng nhiễu trong dữ liệu và khiến mô hình khó học hơn. Điều này có thể dẫn đến việc mô hình học sai hoặc không chính xác. Ngoài ra việc không đưa toàn bộ chuỗi đầu vào về dạng chữ thường khiến cho dữ liệu đề huấn luyện trở nên không đồng nhất, rõ ràng “ luật ”, “ Luật ”, “ LUẬT ” sẽ được xem như các từ khác nhau, dẫn đến việc mô hình học máy có thể hiểu sai ngữ nghĩa hoặc không nhận diện được các từ này là giống nhau về ngữ nghĩa làm cho kích thước từ vựng tăng lên một cách không cần thiết và giảm hiệu suất của mô hình.

### 3.2.2. Tokenization và Embedding

Sau khi đã xử lý chuỗi đầu vào thì tiến hành tách từ. Có thể hiểu ở đây là tách một câu thành một tập các từ của câu. Trong toàn bộ chuỗi đầu vào độ dài tối đa mà một chuỗi có là 1078 từ và thấp nhất là 81 từ. Trong đó, độ dài mà nhiều chuỗi đạt được nhất là 123 từ với 2.111 chuỗi trong tổng số 97.626 chuỗi. Vì phần này quá ít so với toàn bộ chuỗi đầu vào nếu lấy độ dài tối đa của một chuỗi là 123 làm chuẩn đầu vào nó sẽ khiến cho số điểm nhiễu trong dữ liệu tăng lên vì thiếu thông tin ngữ nghĩa cũng như điểm dữ liệu không đầy đủ. Do vậy nên nhóm chọn độ dài tiêu chuẩn cho một chuỗi đầu vào là giá trị trung bình của toàn bộ chuỗi đầu vào là 227 và tăng dần lên để xem xét hiệu suất của mô hình. Nhóm tiến hành tăng độ dài chuỗi lên từ từ và thấy độ dài 250 cho hiệu suất nhỉnh hơn 1-2% so với những độ dài thấp hơn, tiếp tục thử với độ dài lớn hơn như 260, 270, 280, 290 và 300 thì hiệu suất không bằng độ dài 250. Khi nhóm muốn tiến hành thử với độ dài 310 trở lên thì cầu hình máy không đáp ứng được dẫn đến lỗi. Do vậy nhóm chọn độ dài tiêu chuẩn cho chuỗi đầu vào là 250. Với những chuỗi có độ dài ngắn hơn thì sẽ thêm vào phần bị thiếu số 0 sao cho đạt được độ dài chuẩn.

Khi quá trình tách từ hoàn tất sẽ tiến hành xây dựng ma trận Embedding. Đến đây tiến hành tạo một ma trân embedding (ma trận nhúng) để biểu diễn từ vựng. Ma trận Embedding là một ma trận *VxD,* với V là số lượng từ vựng (số lượng từ vựng này ta sẽ lấy được trong quá trình tách từ) và D là số chiều của ma trận. Ma trận Embedding này biểu thị các từ và cụm từ dưới dạng các vector số học trong không gian nhiều chiều sao cho các từ tương tự về ngữ nghĩa thường có các vector tương tự hoặc gần nhau trong không gian này.

Để tạo ma trận Embedding, nhóm sử dụng FastText để tạo ma trận vì FastText là một trong những mô hình hỗ trợ rất tốt cho xử lý ngôn ngữ Tiếng Việt. Ngoài ra, FastText sử dụng n-gram để học vector embedding, giúp nó có thể xử lý tốt hơn các từ không xuất hiện trong tập dữ liệu đào tạo. Với những lĩnh vực cụ thể như Luật pháp, ta hoàn toàn có thể tự xây dựng một mô hình FastText riêng để tạo ma trận Embedding và nhóm chọn cách làm này. Thay vì dùng những mô hình đã được tạo sẵn, nhóm tạo ra FastText bằng thư viện từ của chính tập dữ liệu, bằng cách này mô hình sẽ có thể biểu diễn các từ chuyên ngành dễ dàng và chính xác hơn.

### 3.2.3. Cơ chế Attention

Attention là một cơ chế giúp mô hình có khả năng tập trung chọn lọc vào các phần quan trọng của chuỗi đầu vào và do đó, học được mối liên kết giữa chúng. Điều này giúp mô hình xử lý một cách hiệu quả với các câu đầu vào dài. Attention đã được chứng minh thông qua thực nghiệm là rất hiệu quả trong việc mô hình hóa các chuỗi như ngôn ngữ, đến mức chúng trở thành không thể thiếu.

Có rất nhiều loại attention nhưng đa phần thường được sử dụng trong kiến trúc Encoder-Decoder (có cả hai phần Encoder và Decoder). Nhưng đối với bài toán phân loại chủ đề văn bản là một bài toán đặc thù với mục tiêu là phân loại một đoạn văn bản vào một trong những nhãn đã cho trước thì sẽ chỉ sử dụng Encoder để biểu diễn văn bản đầu vào thành một vector số học để mô hình có thể hiểu được để thục hiện các phép tính toán và phân loại. Để phù hợp với bài toán đặt ra thì loại attention được nhóm sử dụng là self-attention (intra-attention) được giới thiệu trong bài báo [7][8], là một cơ chế attention chỉ dùng cho một câu. Có thể hiểu rằng ta sẽ tự tạo một ma trận với hàng và cột đều là cùng một câu để hiểu được những phần nào của câu sẽ liên quan đến nhau. Các bước của self-attention như sau:

* Bước 1: Bộ mã hóa (Encoder) tạo ra tập các annotations, hidden state hiện tại ht từ đầu vào.
* Bước 2: Thực hiện tính toán điểm attention ati (điểm attention của từ thứ i với hidden state hiện tại t) của tập các annotations với ht qua công thức

(19) [8]

* Bước 3:Dùng hàm softmax để chuẩn hóa các điểm attention thành các giá trị trọng số sti có giá trị nằm trong khoản từ 0 đến 1.

(20) [8]

* Bước 4: Các trọng số ở bước 3 cùng với tập annotations ở bước 1 sẽ được dùng để tạo nên context vector ct thông qua việc tính tổng trọng số của các annotations.
* Bước 5: Tiếp theo chúng ta tính các vector hidden và memory tạm thời được đánh trọng số lại ở bước 4.
* Bước 6: Sau đó ta tính các giá trị cell state và hidden state ở timestep t tương tự như mô hình LSTM chuẩn.
* Bước 7: Lặp lại các bước 2 đến 6 cho đến hết chuỗi.

### 3.2.4. Huấn luyện mô hình bằng LSTM

Sau khi đã tạo được ma trận Embedding thì tiến hành chia tập dữ liệu thành tập train và test, sau đó sẽ bắt đầu huấn luyện mô hình với tập train. Trong quá trình huấn luyện thì các dữ liệu đầu vào sẽ lần lượt đi qua các cổng của LSTM. Quá trình huấn luyện trong LSTM như sau:

* Đầu tiên dữ liệu đầu vào sẽ đi qua cổng quên để quyết định xem thông tin nào được giữ và thông tin nào bị bỏ lại. Thông tin đầu vào được cho vào hàm sigmoid. Đầu ra của hàm này đóng vai trò là mask để lọc thông tin từ trạng thái cell.
* Tiếp đến, hidden state ở nút trước đó được kết hợp với dữ liệu đầu vào tại nút hiện tại sẽ được truyền tiếp đến cổng đầu vào. Cổng này dùng để cập nhật bộ nhớ với các thông tin mới. Ở đây có xuất hiện 2 hàm sigmoid và hàm tanh. Tác dụng của chúng cũng như trên. Output từ hàm sigmoid sẽ có tác dụng lọc thông tin đã qua xử lý từ output hàm tanh. Từ hai cổng trên, ta đã có thể tính toán được giá trị của trạng thái cell hiện tại, từ đó truyền đi cho các từ phía sau.
* Cuối cùng, với thông tin của trạng thái cell và input hiện tại sẽ qua cổng đầu ra để quyết định output đầu ra. Trạng thái cell sau khi chỉnh sửa sẽ đi qua hàm tanh và input hiện tại thì được đi qua hàm sigmoid. Từ đây ta kết hợp 2 kết quả trên để có được kết quả đầu ra. là trạng thái ẩn (hidden state) của nút hiện tại.

## 3.3 Thực nghiệm mô hình phân loại chủ đề văn bản

### 3.3.1 Dữ liệu thực nghiệm

### 3.3.2 Thực nghiệm mô hình

### 3.3.3 Kết quả thực nghiệm

# CHƯƠNG 4: KẾT LUẬN

## 4.1 Kết quả đạt được của đề tài

## 4.2 Hạn chế

## 4.3 Hướng phát triển

# TÀI LIỆU THAM KHẢO

[1]. Minhui Liang & Tiansen Niu, “Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs” ScienceDirect, Nov. 02, 2022. [Online]. Available: [Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1877050922015058)

[2]. Cheng Liu & Xiaofang Wang, “Quality-related English text classification based on recurrent neural network” ScienceDirect, Nov. 25 2019, Accepted: Nov. 22 2019. [Online] Available: [Quality-related English text classification based on recurrent neural network - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1047320319303451)

[3]. Wen Zhang, Taketoshi Yoshida , Xijin Tang, “Text classification based on multi-word with support vector machine” ScienceDirect, April. 04, 2008. Accepted: March. 30, 2008. [Online]. Available: [Text classification based on multi-word with support vector machine - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0950705108000968)

[4]. Alper Kursat Uysal & Serkan Gunal, “The impact of preprocessing on text classification” ScienceDirect, Sep. 16, 2013. Accepted: Aug 28, 2013. [Online]. Available: [The impact of preprocessing on text classification - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0306457313000964).

[5]. Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes and Donald Brown, “Text Classification Algorithms: A Survey” arXiv, Apr. 23, 2019. Accepted: 17 April 2019. [Online]. Available: https://arxiv.org/abs/1904.08067

[6]. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, “Dive into Deep Learning” arXiv, Aug. 22, 2023. [Online]. Available: https://arxiv.org/abs/2106.11342

[7]. A. Vaswani *et al.*, “Attention Is All You Need.” arXiv, Aug. 02, 2023. Accessed: Dec. 09, 2023. [Online]. Available: <http://arxiv.org/abs/1706.03762>.

[8]. Jianpeng Cheng, Li Dong, Mirella Lapata, “Long Short-Term Memory-Networks for Machine Reading” arXiv, Sep. 20, 2016. [Online]. Available: [[1601.06733] Long Short-Term Memory-Networks for Machine Reading (arxiv.org)](https://arxiv.org/abs/1601.06733)

[9]. Berna Altınel, Murat Can Ganiz, “Semantic text classification: A survey of past and recent advances” ScienceDirect, Aug. 20, 2018. Accepted: Aug. 06, 2018. [Online]. Available: [Semantic text classification: A survey of past and recent advances - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0306457317305757)[10]. Sepp Hochreiter, Jürgen Schmidhuber, “Long Short-Term Memory” MIT Press Direct, Aug. 28, 1995. Accepted: Feb. 24, 1997. [Online]. Available: [Long Short-Term Memory | Neural Computation | MIT Press](https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext)

[11]. U.B. Mahadevaswamy, P. Swathi, “Sentiment Analysis using Bidirectional LSTM Network” ScienceDirect, Jan. 31, 2023. [Online]. Available: [Sentiment Analysis using Bidirectional LSTM Network - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1877050922024930)

[12]. Minhui Liang, Tiansen Niu, “Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs” ScienceDirect, Nov. 02, 2022. [Online]. Available: [Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1877050922015058)

[13]. Quản Thành Thơ, “Mạng nơ-ron nhân tạo: Từ hồi quy đến học sâu”. Việt Nam: NXB Đại học Quốc gia Thành phố Hồ Chí Minh, 2021.

[14]. Trần Cao Đệ, Phạm Nguyên Khang, “Phân loại văn bản với Máy học Vector hỗ trợ và Cây quyết định” Trường Đại học Cần Thơ, 2012.

[15]. Nguyễn Thị Thanh Thủy, Nguyễn Ngọc Điệp, “Một phương pháp trích xuất kết hợp thực thể và quan hệ tham chiếu trong văn bản pháp quy”. Việt Nam, 2021.

[16]. Nguyễn Thị Thanh Thủy, Đặng Bảo Chiến, Triệu Khương Duy, Ngô Xuân Bách, Từ Minh Phương , “Phân loại quan hệ tham chiếu trong văn bản pháp quy”. Việt Nam, 2021.

[17]. N. X. Bach, N. T. T. Thuy, D. B. Chien, T. K. Duy, T. M. Hien, and T.M Phuong. “Reference Extraction from Vietnamese Legal Documents”. In Proceedings of the Tenth International Symposium on Information and Communication Technology.

[18]. T.M. Phuong, D. Lee and K.H. Lee. “Learning rules to extract protein interactions from biomedical text”. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 148-158. Springer, Berlin, Heidelberg. 2003.

[19]. C. Zhang, X. Zhang, W. Jiang, Q. Shen and S. Zhang. “Rule-based extraction of spatial relations in natural language text”. In 2009 International Conference on Computational Intelligence and Software Engineering, pp. 1-4. IEEE. 2009.

[20]. K. Nebhi. “A rule-based relation extraction system using DBpedia and syntactic parsing”. In Proceedings of the NLP DBPEDIA-2013 Workshop co-located with the 12th International Semantic Web Conference (ISWC 2013). 2013.

[21]. D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao. “Relation classification via convolutional deep neural network”. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335-2344. 2014.

[22]. X. Jiang, Q. Wang, P. Li and B. Wang. “Relation extraction with multi-instance multi-label convolutional neural networks”. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1471-1480. 2016.

# PHỤ LỤC

[1]. Source Code:

[2]. Tập dữ liệu