HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHÍ MẠNH KIÊN

TÌM KIẾM VĂN BẢN PHÁP QUY SỬ DỤNG KỸ THUẬT HỌC SÂU

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng nghiên cứu)

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHÍ MẠNH KIÊN

TÌM KIẾM VĂN BẢN PHÁP QUY SỬ DỤNG KỸ THUẬT HỌC SÂU

CHUYÊN NGÀNH : **KHOA HỌC MÁY TÍNH**

MÃ SỐ: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC
GS. TS. TỪ MINH PHƯƠNG

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu và tìm hiểu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất cứ công trình nào khác.

Tác giả luận văn

Phí Mạnh Kiên

LỜI CẢM ƠN

Để hoàn thành được luận văn này, ngoài sự nghiên cứu và những cố gắng của bản thân, em xin gửi lời cảm ơn sâu sắc tới GS. TS. Từ Minh Phương, giảng viên trực tiếp hướng dẫn, tận tình chỉ bảo và định hướng cho em trong suốt quá trình nghiên cứu và thực hiện luận văn.

Em xin gửi lời cảm ơn chân thành cảm ơn tất cả các thầy cô giáo của Học viện Công nghệ Bưu chính Viễn thông đã giảng dạy và dìu dắt em trong suốt quá trình học tập tại trường từ khi còn học đại học cho đến cao học.

Cuối cùng, em xin gửi lời cảm ơn tới gia đình, bạn bè và những người đã luôn ở bên cổ vũ tinh thần, tạo điều kiện thuận lợi cho em để em có thể học tập tốt và hoàn thiện luận văn.

Dù đã cố gắng hết sức nhưng trong luận văn không thể tránh khỏi những sai sót, em mong nhận được sự góp ý để hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC BẢNG

DANH MỤC HÌNH ẢNH

DANH MỤC KÝ HIỆU CÁC CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
ASR	Automatic Speech Recognition	Nhận dạng tiếng nói tự động
BM25	Best Match - Okapi BM25	
CBOW	Continuous Bag Of Words	
CNN	Convolutional Neural Network	Mạng nơ-ron nhân chập
DNN	Deep Neural Network	Mạng nơ-ron nhiều lớp
FNN	Feed-forward Neural Network	Mạng nơ-ron truyền thẳng
GloVe	Global Vector	
GRU	Gate Recurrent Unit	
IR	Information Retrieval	Tìm kiếm thông tin
IRM	Information Retrieval Model	Mô hình tìm kiếm thông tin
LDA	Latent Dirichlet Allocation	Mô hình phát hiện chủ đề ẩn
LSA	Latent Semantic Analysis	
LSTM	Long-Short Term Memory	
MCMC	Markov-Chain Monte Carlo	
NATR	Neural Attentive Text Representation	
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
PLSA	Probabilistic Latent Semantic Analysis	
RNN	Recurrent Neural Networks	Mạng nơ-ron hồi quy
TF-IDF	Term Frequency - Inverted Document Frequency	Tần xuất từ - tần xuất văn bản nghịch đảo

MỞ ĐẦU

Ngày nay, trong kỉ nguyên kỹ thuật số, với sự bùng nổ của thông tin, số lượng các tài liệu điện tử do con người tạo ra ngày càng khổng lồ. Trong quá trình học tập, nghiên cứu hay làm việc, chúng ta cần tìm kiếm và đọc rất nhiều tài liệu để tìm được thông tin ta mong muốn. Việc này đôi khi mất nhiều thời gian, điển hình là trong lĩnh vực pháp luật. Một văn bản pháp luật thường có thể dài tới 15-20 trang hoặc thậm chí nhiều hơn. Một vụ việc có thể liên quan đến nhiều văn bản khác nhau. Các luật sư, nhân viên pháp lý... phải đọc rất nhiều văn bản và so sánh các điều, khoản trong đó với trường hợp đang xử lý. Theo một khảo sát năm 2013 tại Mỹ [19], trung bình, gần 47,3% số người được hỏi dành 15% thời gian, 36.6% số người dành 15-50% thời gian, 10.3% số người dành từ 50% thời gian trở lên mỗi tuần cho việc tìm kiếm và nghiên cứu văn bản pháp luật. Đây là một vấn đề thực tiễn, mang lại giá trị mà chúng ta cần giải quyết.

Bài toán tìm kiếm thông tin ra đời chính là để xử lý vấn đề trên. Nhiệm vụ chính của bài toán tìm kiếm thông tin là tìm kiếm các thông tin thoả mãn nhu cầu thông tin của người dùng. Người sử dụng của một hệ thống tìm kiếm thông tin không chỉ muốn tìm những văn bản có chứa những từ khóa trong câu truy vấn mà còn quan tâm tới việc thu nhận được những văn bản mang lại thông tin phù hợp với mục đích tìm kiếm.

Các hệ thống tìm kiếm thông tin thường biểu diễn văn bản và câu truy vấn dưới dạng các véc-tơ. Chất lượng biểu diễn văn bản và so sánh các véc-tơ biểu diễn có ảnh hưởng quan trọng tới kết quả. Gần đây, các kỹ thuật sử dụng học sâu cho thấy khả năng biểu diễn văn bản rất tốt trong xử lý ngôn ngữ tự nhiên nói chung và tìm kiếm thông tin văn bản nói riêng. Vì vậy, tôi chọn đề tài **"Tìm kiếm văn bản pháp quy sử dụng kỹ thuật học sâu"** cho luận văn của mình. Mục tiêu của luận văn là tìm hiểu các phương pháp biểu diễn văn bản và đề xuất mô hình sử dụng kỹ thuật học sâu ứng dụng trong tìm kiếm văn bản pháp quy tiếng Việt. Đầu vào của hệ thống là một câu hỏi về pháp luật. Đầu ra của hệ thống là văn bản pháp quy có liên quan, trả lời

được cho câu hỏi đó, cụ thể đến mức điều. Ví dụ, với câu hỏi "*Vợ chồng ly hôn tài sản chung được phân chia như thế nào?*" hệ thống sẽ trả về kết quả là: *Điều 59 Luật Hôn nhân và gia đình, Điều 7 Thông tư liên tịch hướng dẫn một số quy định của Luật Hôn nhân và gia đình*.

Nội dung luận văn được chia thành 3 chương như sau:

- **CHƯƠNG 1:** Bài toán tìm kiếm thông tin và các phương pháp biểu diễn văn bản: Trình bày tổng quan về bài toán tìm kiếm thông tin và các phương pháp biểu diễn văn bản phục vụ tìm kiếm, tìm kiếm thông tin.
- **CHƯƠNG 2:** Ứng dụng biểu diễn văn bản bằng mạng nơ-ron sâu trong tìm kiếm văn bản pháp quy: Giới thiệu về bài toán tìm kiếm văn bản pháp quy, trình bày phương pháp biểu diễn văn bản sử dụng mạng nơ-ron sâu.
- **CHƯƠNG 3:** Thử nghiệm và đánh giá: Mô tả quá trình xây dựng bộ dữ liệu và so sánh, đánh giá hiệu quả của mô hình đề xuất so với các phương pháp khác.

Các kết quả của luận văn đã được chấp nhận công bố tại hội nghị COLING 2020, hội nghị hạng A về xử lý ngôn ngữ tự nhiên.

CHƯƠNG 1. BÀI TOÁN TÌM KIẾM THÔNG TIN VÀ CÁC PHƯƠNG PHÁP BIỂU DIỄN VĂN BẢN

Chương này sẽ trình bày tổng quan về bài toán tìm kiếm thông tin nói chung và bài toán tìm kiếm văn bản pháp quy nói riêng, bao gồm khái niệm, kiến trúc hệ thống và mô hình tìm kiếm thông tin, cùng với các phương pháp biểu diễn văn bản phục vụ tìm kiếm.

1.1. Bài toán tìm kiếm thông tin

1.1.1. Tìm kiếm văn bản quy phạm pháp luật

Theo Bing Liu, tìm kiếm thông tin hay truy vấn thông tin (Information Retrieval – IR) là lĩnh vực nghiên cứu nhằm giúp người dùng tìm kiếm thông tin phù hợp với thông tin mình cần [15].

Theo Manning, tìm kiếm thông tin là việc tìm các tài liệu ở dạng phi cấu trúc (thường là văn bản) thỏa mãn một thông tin cần thiết trong một tập hợp dữ liệu lớn (thường được lưu trên máy tính) [18].

IR nghiên cứu cách thu thập, tổ chức, lưu trữ truy xuất và phân tán thông tin. Việc biểu diễn và tổ chức thông tin phải được thực hiện theo cách mà người dùng có thể truy cập được thông tin đáp ứng nhu cầu của mình.

Bài toán tìm kiếm thông tin

Input:

- Một tập tài *liệu lớn, ổn định.*
- Một nhu cầu thông tin thể hiện dưới dạng câu truy vấn (các từ khoá hoặc câu hỏi).

Output:

- Tìm tất cả tài liệu có *liên quan* đến câu truy vấn.

Trong đó, tài liệu ổn định ở đây có thể hiểu là tài liệu mà thao tác xóa, chỉnh sửa hoặc thêm mới trên nó ít khi xảy ra.

Những vấn đề cần giải quyết của bài toán tìm kiếm thông tin

- Biểu diễn tập tài liệu như thế nào?
- Biểu diễn nhu cầu thông tin của người dùng như thế nào?
- Bằng cách nào hệ thống có thể trả về những tài liệu có liên quan đến nhu cầu thông tin một cách có hiệu quả?
- Kết quả trả về được trình bày như thế nào?

Bài toán tìm kiếm văn bản pháp quy

Văn bản quy phạm pháp luật hay còn gọi là Văn bản pháp quy là một hình thức pháp luật thành văn được thể hiện qua các văn bản chứa được các quy phạm pháp luật do cơ quan hoặc cá nhân có thẩm quyền ban hành để điều chỉnh các quan hệ xã hội. Theo quy định của Luật Ban hành văn bản quy phạm pháp luật năm 2008 của Việt Nam thì Văn bản quy phạm pháp luật là văn bản do cơ quan nhà nước ban hành hoặc phối hợp ban hành theo thẩm quyền, hình thức, trình tự, thủ tục được quy định. Trong đó có quy tắc xử sự chung, có hiệu lực bắt buộc chung, được Nhà nước bảo đảm thực hiện để điều chỉnh các quan hệ xã hội.

Văn bản pháp quy có đặc điểm là thường dài, cấu trúc phức tạp, chia thành nhiều chương, điều, khoản… Một văn bản pháp luật thường có thể dài tới 15-20 trang hoặc thậm chí nhiều hơn. Một vụ việc có thể liên quan đến nhiều văn bản khác nhau. Các luật sư, nhân viên pháp lý... phải đọc rất nhiều văn bản và so sánh các điều, khoản trong đó với trường hợp đang xử lý. Việc này tốn rất nhiều thời gian, do vậy, nếu có một hệ thống giúp tìm kiếm và đưa ra được các điều khoản liên quan tới vụ việc đang xử lý sẽ giúp ích rất nhiều. Bài toán được phát biểu như sau:

- Đầu vào: Truy vấn của người dùng dưới dạng một câu hỏi.
- Đầu ra: Các điều khoản có liên quan, giúp trả lời được cho câu hỏi của người dùng.

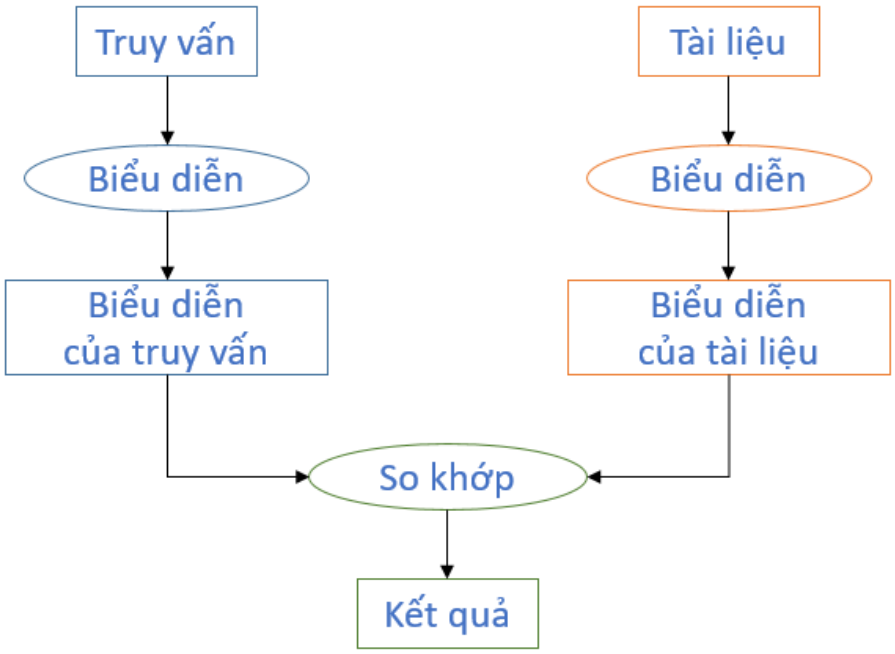
Ví dụ minh họa đầu vào và đầu ra của bài toán được mô tả bằng bảng bên dưới:

Bảng 1.1. Ví dụ minh họa bài toán tìm kiếm văn bản pháp quy.

Câu hỏi đầu vào	Con riêng có quyền hưởng thừa kế của bố đã mất không di chúc không?
Đầu ra	Điều 651 Bộ luật dân sự 2015
Nội dung điều luật	**Điều 651. Người thừa kế theo pháp luật** 1. Những người thừa kế theo pháp luật được quy định theo thứ tự sau đây: a) Hàng thừa kế thứ nhất gồm: vợ, chồng, cha đẻ, mẹ đẻ, cha nuôi, mẹ nuôi, con đẻ, con nuôi của người chết; b) Hàng thừa kế thứ hai gồm: ông nội, bà nội, ông ngoại, bà ngoại, anh ruột, chị ruột, em ruột của người chết; cháu ruột của người chết mà người chết là ông nội, bà nội, ông ngoại, bà ngoại; c) Hàng thừa kế thứ ba gồm: cụ nội, cụ ngoại của người chết; bác ruột, chú ruột, cậu ruột, cô ruột, dì ruột của người chết; cháu ruột của người chết mà người chết là bác ruột, chú ruột, cậu ruột, cô ruột, dì ruột; chắt ruột của người chết mà người chết là cụ nội, cụ ngoại. 2. Những người thừa kế cùng hàng được hưởng phần di sản bằng nhau. 3. Những người ở hàng thừa kế sau chỉ được hưởng thừa kế, nếu không còn ai ở hàng thừa kế trước do đã chết, không có quyền hưởng di sản, bị truất quyền hưởng di sản hoặc từ chối nhận di sản.

1.1.2. Hệ thống tìm kiếm và tìm kiếm thông tin

Hoạt động của một hệ thống tìm kiếm thông tin được mô tả trong Hình 1.1, bao gồm ba bước chính: biểu diễn văn bản, biểu diễn truy vấn và so khớp – đánh giá độ liên quan giữa văn bản và truy vấn.



Hình 1.1. Kiến trúc tổng quan của hệ thống tìm kiếm thông tin.

Truy vấn của người dùng thể hiện thông tin mà người đó cần, có thể thuộc một trong các dạng sau [15]:

- Truy vấn dạng từ khóa (Keyword queries): Người dùng thể hiện thông tin mình cần bằng một danh sách (ít nhất một) các từ khóa với mục đích tìm các tài liệu chứa một vài (ít nhất một) hoặc tất cả các từ khóa đó.
- Truy vấn dạng Boolean (Boolean queries): Người dùng có thể dùng các toán tử Boolean AND, OR và NOT để tạo các truy vấn phức tạp. Truy vấn sẽ bao gồm các từ khóa và các toán tử Boolean.
- Truy vấn dạng cụm từ (Phrase queries): Truy vấn gồm một chuỗi các từ tạo thành một cụm từ. Các tài liệu trả về phải chứa cả cụm từ đó.
- Truy vấn gần (Proximity queries): Là một phiên bản thoải mái hơn của truy vấn dạng cụm từ. Nó tìm kiếm các từ khóa trong truy vấn nằm gần nhau trong các tài liệu. Độ gần (closeness) được dùng như một yếu tố để xếp hạng các tài liệu trả về.

- Truy vấn dạng tài liệu (Full document queries): Khi truy vấn là toàn bộ một văn bản, người dùng muốn tìm những văn bản khác tương tự như văn bản trong truy vấn.
- Câu hỏi bằng ngôn ngữ tự nhiên (Natural language question): Người dùng thể hiện thông tin cần thiết dưới dạng một câu hỏi bằng ngôn ngữ tự nhiên, sau đó hệ thống tìm câu trả lời. Đây là trường hợp phức tạp nhất và cũng là lý tưởng nhất.

Mô hình tìm kiếm thông tin (Information Retrieval Model - IRM) quyết định tài liệu và truy vấn được biểu diễn như thế nào, cách xác định sự liên quan giữa một tài liệu với truy vấn của người dùng. Đây là thành phần quan trọng nhất trong hệ thống IR.

Mô hình tìm kiếm thông tin có thể được định nghĩa như sau [6]:

$$IRM = \{D, Q, F, R(q_k, d_j)\}$$

Trong đó:

- D (Document collection): Là tập hợp biểu diễn của các tài liệu.
- Q (Query collection): Là tập hợp biểu diễn các thông tin người dùng cần, còn được gọi là các truy vấn.
- F (Framework): Là phương pháp mô hình hóa việc biểu diễn tài liệu, truy vấn và mối quan hệ giữa chúng.
- R (Ranking function): Là hàm gán một số thực cho biểu diễn d_j của tài liệu j để thể hiện mức độ liên quan của nó với truy vấn q_k.

Việc biểu diễn văn bản và truy vấn đóng vai trò rất quan trọng, ảnh hưởng trực tiếp tới kết quả tìm kiếm của hệ thống. Phương pháp biểu diễn tốt cần trích xuất, sau đó chọn ra được các thông tin cần thiết để so khớp văn bản với truy vấn. Các phương pháp có thể dùng để biểu diễn văn bản bao gồm: biểu diễn sử dụng từ khóa, biểu diễn sử dụng chủ đề ẩn, biểu diễn sử dụng véc-tơ từ, biểu diễn sử dụng mạng nơ-ron sâu. Từng phương pháp sẽ được trình bày cụ thể trong các mục phía sau.

Sau khi có biểu diễn của câu truy vấn và các văn bản, hệ thống sẽ thực hiện quá trình so khớp, tính độ liên quan giữa các văn bản với truy vấn. Độ liên quan có thể được tính thông qua các hàm khoảng cách như Euclid, Cosine, hàm tích vô hướng hoặc thông qua một mạng nơ-ron. Các văn bản sẽ được xếp hạng dựa trên độ liên quan tới truy vấn và trả về cho người dùng.

1.2. Biểu diễn văn bản sử dụng từ khóa

1.2.1. TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF), là một thống kê số học phản ánh tầm quan trong của một từ (word) với một văn bản (document) trong tập các văn bản (corpus). Nó thường được dùng để làm trọng số trong việc thu thập thông tin và khai phá văn bản. Giá trị của TF-IDF tỉ lệ thuận với số lần xuất hiện của từ đó trong văn bản, tuy nhiên nó bị bù trừ bởi tần suất của nó trong tập tất cả các văn bản (corpus). Việc đó giúp loại bỏ những trường hợp mà một từ là từ phổ biến nhưng lại vô nghĩa ví dụ như các từ "*thì"*, "*là", "mà"* (người ta gọi những từ này là các từ dừng - stopwords).

TF-IDF là sự kết hợp của hai thống kê cục bộ - tổng quát là: tần suất của từ (term frequency – cục bộ) và tần suất nghịch đảo văn bản (inverse document frequency – tổng quát).

Các tham số trong TF-IDF:

- Term frequency: Tần số xuất hiện
- Inverse document frequency: Tần số nghịch đảo văn bản
- Document Length: Độ dài văn bản

Tần số xuất hiện

Yếu tố này đánh giá tần suất xuất hiện của từ trong văn bản. Càng xuất hiện nhiều, độ liên quan càng cao. Một văn bản xuất hiện từ khóa 5 lần sẽ liên quan nhiều hơn một văn bản mà từ khóa chỉ xuất hiện 1 lần. Tuy nhiên không thể nói rằng một văn bản xuất hiện từ khóa 6 lần thì liên quan gấp đôi một văn bản từ khóa xuất hiện

3 lần. Chính vì thế TF không còn được lấy trực tiếp, thay vào đó TF được tính theo công thức sau:

$$tf(t,d) = \sqrt{frequency}$$

tf của từ t trong văn bản d được tính bằng căn bậc hai của số lần t xuất hiện trong d.

Tần số nghịch đảo văn bản

Tần số nghịch đảo văn bản (Inverse Document Frequency) dùng để đánh giá độ đặc biệt của một từ dựa vào tần suất xuất hiện của từ trên toàn bộ tập các văn bản. Một từ xuất hiện ở nhiều văn bản thì sẽ ít có giá trị.

Ví dụ: Chúng ta muốn tìm kiếm luật sở hữu trí tuệ. Khi chúng ta tìm kiếm với từ khóa "luật" thì sẽ nhận được rất nhiều kết quả nhưng lại có rất ít kết quả chúng ta mong muốn. Còn khi chúng ta tìm kiếm với từ khóa "sở hữu trí tuệ" thì nhận được ít kết quả hơn nhưng chúng ta sẻ thấy rõ ràng các kết quả tìm kiếm sẽ sát với kết quả chúng ta mong muốn. Suy ra từ khóa "luật" sẽ có giá trị thấp hơn từ khóa "sở hữu trí tuệ ".

Inverse Document Frequency được tính như sau:

$$idf(t,d) = log\frac{|D|}{|d_t|}$$

Trong đó $|D|$ là tổng số văn bản trong tập dữ liệu, $|d_t|$ là số văn bản có chứa từ t.

Độ dài văn bản

Yếu tố này đánh giá độ dài của văn bản. Văn bản càng ngắn thì từ sẽ có giá trị càng cao và ngược lại. Điều này hoàn toàn dễ hiểu, chúng ta có thể thấy một từ xuất hiện trong tiêu đề sẽ có giá trị hơn rất nhiều cùng từ đó nhưng xuất hiện trong nội dung. Để thể hiện điều này ta dùng công thức:

$$norm(d) = \frac{1}{\sqrt{|d|}}$$

Trong đó $|d|$ là độ dài văn bản tính bằng tổng số từ.

Tổng hợp lại

$$tf - idf(t,d) = tf(t) \times idf(t,d) \times norm(d)$$

1.2.2. BM25

BM25 là hàm tính thứ hạng được các công cụ tìm kiếm sử dụng để xếp hạng các văn bản theo độ phù hợp với truy vấn nhất định. Hàm xếp hạng này dựa trên mô hình xác suất, được phát minh ra vào những năm 1970 – 1980. Phương pháp còn được gọi là Okapi BM25, vì lần đầu tiên công thức được sử dụng trong hệ thống tìm kiếm Okapi, được sáng lập tại trường đại học London những năm 1980 và 1990. [36]

Term frequency trong BM25

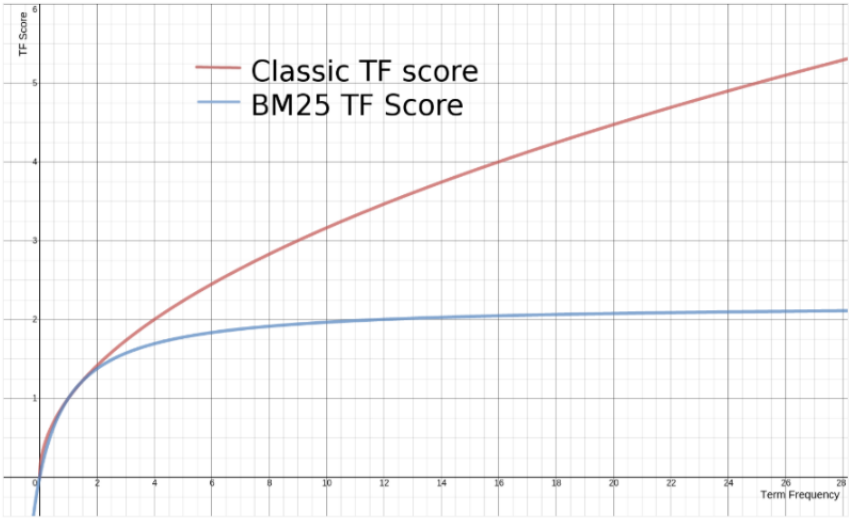
Đối với TF-IDF, giá trị của nó sẽ tăng vô hạn khi TF tăng lên. Để giảm tác động của TF thì BM25 đã chỉnh sửa công thức của TF lại, giới hạn tới một điểm cực đại, và chúng ta có thể tùy chỉnh giới hạn này bằng công thức:

$$\frac{(k+1) \times tf}{k + tf}$$

Trong đó k là hằng số, tf là số lần xuất hiện của từ trong văn bản.

k giúp giới hạn mức độ ảnh hưởng của một từ đơn lẻ trong truy vấn tới độ liên quan của một văn bản. Sự so sánh giữa ảnh hưởng của TF trong TF-IDF và BM25 có thể thấy ở Hình 1.2 bên dưới.

Thay đổi giá trị của k sẽ khiến độ dốc của đường cong ảnh hưởng của TF đến độ liên quan (đường màu xanh) thay đổi. Điều này ảnh hưởng đến việc một từ xuất hiện nhiều thêm sẽ làm tăng độ liên quan lên như thế nào. Đường cong tác động của TF lên độ liên quan tăng nhanh khi $TF \leq k$ và chậm dần khi $TF > k$. Trong Elasticsearch, k có giá trị mặc định là 1.2.



Hình 1.2. TF trong TF-IDF và BM2

Độ dài văn bản trong BM25

Công thức của TF-IDF chưa thực sự hoàn chỉnh, nó đúng với những văn bản có độ dài trung bình trong toàn bộ tập dữ liệu. Nếu độ dài văn bản quá ngắn hoặc quá dài so với độ dài trung bình, thì công thức trên sẽ cho kết quả thiếu chính xác.

Bởi vậy, người ta thêm vào trong công thức trên 2 tham số, một hằng số b và một giá trị độ dài L, công thức sẽ trở thành:

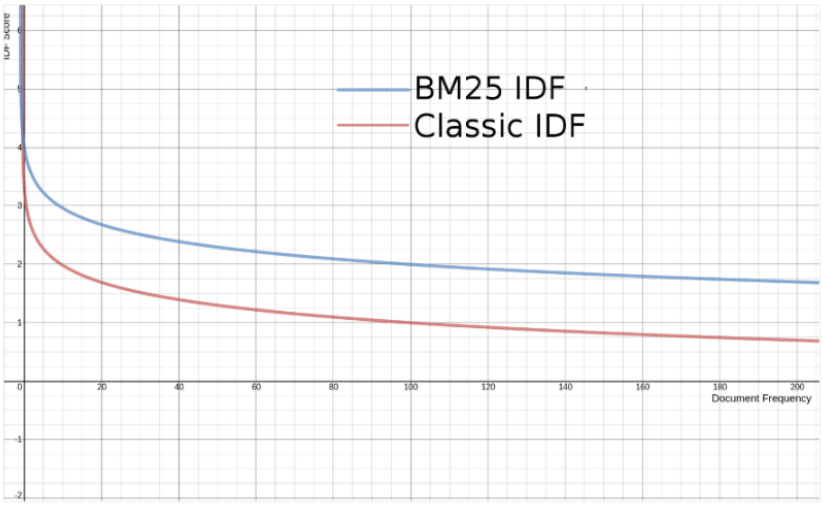
$$\frac{(k+1) \times tf}{k \times (1.0 - b + b \times L) + tf}$$

Trong đó:

- L là tỉ lệ giữa độ dài của văn bản đang xét so với độ dài trung bình của tất cả các văn bản.
- b là một hằng số

b càng lớn thì ảnh hưởng của độ dài của tài liệu so với độ dài trung bình càng được khuếch đại. Nếu đặt b thành 0, ảnh hưởng của tỷ lệ độ dài sẽ hoàn toàn bị vô hiệu và độ dài của tài liệu sẽ không ảnh hưởng đến điểm số. Theo mặc định, b có giá trị là 0.75 trong Elasticsearch.

Inverse Document Frequency trong BM25



Hình 1.3. IDF trong TF-IDF và BM25.

Biểu đồ Hình 1.3 cho thấy IDF trong BM25 khá giống IDF trong TF-IDF. Tuy nhiên BM25 đã chỉnh sửa công thức tính lại để thêm khả năng đưa ra điểm âm khi tần suất xuất hiện của từ trên toàn bộ tập văn bản rất cao.

$$idf_t = log \frac{1 + (D - d + 0.5)}{d + 0.5}$$

Trong đó:

- D: tổng số văn bản
- d: số lượng văn bản chứa từ t

1.3. Biểu diễn văn bản sử dụng chủ đề ẩn

1.3.1. Khái niệm mô hình Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation [4] [5] là một trong những mô hình phát hiện chủ đề ẩn thành công nhất hiện nay được phát triển bởi David Blei, Andrew Ng và Michael Jordan. Các văn bản được biểu diễn như một tập hợp các chủ đề, ví dụ một bài viết về bán kính hiển vi sẽ gồm hai chủ đề là: khoa học và kinh doanh. Các chủ đề lại được xem như là tập hợp của các từ, ví dụ chủ đề pháp luật thì các từ *"bộ luật"*,

"nghị định", "thông tư" sẽ có tần suất xuất hiện cao, còn các từ *"siêu thị", "nhà hàng", "tàu hỏa"* sẽ có tần suất thấp.

LDA là một mô hình tự sinh xác suất cho các bộ dữ liệu rời rạc như bộ văn bản ký tự. Bản chất của LDA là một mô hình Bayes phân cấp với 3 mức, trong đó mỗi một phần tử của bộ dữ liệu là một tập hợp hữu hạn của một tập các chủ đề nằm ẩn bên trong nội dung nhìn thấy được của thành phần đó. Trong khi đó, mỗi một chủ đề lần lượt được mô tả như là một tập hợp vô hạn trong một tập các xác suất chủ đề tiềm ẩn. Trong phạm vi của việc mô hình hóa dữ liệu dạng ký tự, xác suất chủ đề cung cấp một biểu diễn cụ thể của một văn bản.

Cách mô hình LDA hoạt động như sau: ban đầu coi tất cả văn bản trong bộ văn bản là rỗng, chưa có từ nào. Giả sử các văn bản đó là tập hợp của những chủ đề nào. Với mỗi một văn bản, chọn một chủ đề trong tập các chủ đề của văn bản đó, sau đó chọn một từ trong tập các từ của chủ đề được chọn, thực hiện hành động này cho đến khi phân phối xác xuất chủ đề đã đủ. Thực hiện chuỗi hành động trên với tất cả các văn bản trong bộ văn bản.

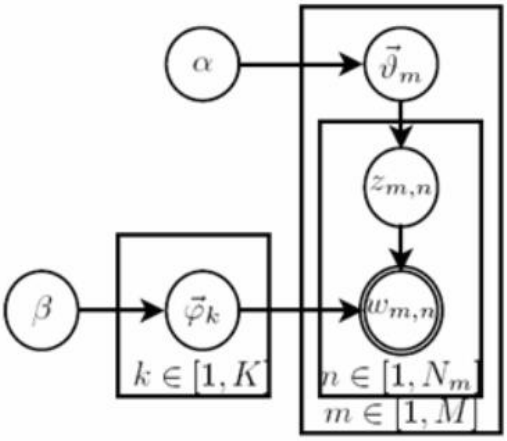
Tuy nhiên trong thực tế thì những thông tin biết được là ngược lại. Tức là, với một tập các văn bản, các văn bản này đã biết hết các từ có trong văn bản này. Việc phải làm bây giờ là phải ước lượng ngược lại tập các chủ đề có trong toàn bộ tập văn bản, tập hợp các từ của từng chủ đề, xác suất của từng từ trong chủ đề đó, và cuối cùng là phân phối xác suất chủ để có trong từng văn bản.

1.3.2. Tổng quan về mô hình sinh trong LDA

Với một tập các văn bản (*corpus*) gồm M văn bản ký hiệu là $D = \{d_1, d_2, \dots, d_M\}$, trong đó văn bản thứ m trong tập văn bản sẽ có N_m từ, các từ trong văn bản sẽ được lấy từ tập từ vựng của các thuật ngữ (term) $= \{t_1, t_2, \dots, t_V\}$,. Mục đích của LDA là tìm ra cấu trúc ẩn của các chủ đề (topic) hay các lĩnh vực (concept) trong văn bản. Các chủ đề và các lĩnh vực này phải nắm bắt được ý nghĩa của văn bản. Mặc dù các khái niệm chủ đề ẩn và lĩnh vực ẩn này đã được đề cập đến trong

các phương pháp LSA và PLSA, nhưng LDA sẽ cung cấp cho chúng ta một mô hình sinh hoàn chỉnh và cho kết quả tốt hơn so với các phương pháp được mô tả ở trên.

Quá trình sinh trong LDA được mô tả như sau: LDA sinh ra một luồng các từ quan sát được $w_{m,n}$ (là các từ có trong nội dung văn bản), được phân chia thành các văn bản. Với mỗi văn bản, một tỷ lệ chủ đề $\overrightarrow{\vartheta_m}$ sẽ được đưa ra, và từ đó, các từ đặc tả chủ đề được tạo ra. Nghĩa là, với mỗi từ, một chỉ số chỉ thi chủ đề $z_{m,n}$ được lấy mẫu theo các văn bản – tỷ lệ trộn cụ thể, và sau đó phân phối chủ đề tương ứng $\vec{\varphi}_{z_{m,n}}$ được sử dụng để sinh ra các từ. Các chủ đề $\vec{\varphi}_k$ sẽ được lấy mẫu một lần cho mọi văn bản trong tập văn bản D. Mô hình đồ họa, mô hình sinh hoàn chỉnh và sẽ được biểu diễn lần lượt tại các hình 1.2, 1.3.



Hình 1.4. Mô hình đồ họa của LDA.

Trong đó, các khối là các "đĩa" biểu diễn các bản sao. Đĩa ở ngoài biểu diễn các văn bản, đĩa ở trong biểu diễn việc lựa chọn các lựa chọn lặp lại của các chủ đề và các từ trong một văn bản.

⊡ "topic plate"
for all topics $k \in [1, K]$ **do**
  sample mixture components $\vec{\varphi}_k \sim \mathrm{Dir}(\vec{\beta})$
end for
⊡ "document plate":
for all documents $m \in [1, M]$ **do**
  sample mixture proportion $\vec{\vartheta}_m \sim \mathrm{Dir}(\vec{\alpha})$
  sample document length $N_m \sim \mathrm{Poiss}(\xi)$
  ⊡ "word plate":
  for all words $n \in [1, N_m]$ in document m **do**
    sample topic index $z_{m,n} \sim \mathrm{Mult}(\vec{\vartheta}_m)$
    sample term for word $w_{m,n} \sim \mathrm{Mult}(\vec{\varphi}_{z_{m,n}})$
  end for
end for

Hình 1.5. Mô hình sinh của Latent Dirichlet Allocation.

Trong đó: *Dir, Poiss, Mult* lần lượt là các phân phối Dirichlet, Poisson và Multinominal tương ứng.

1.3.3. Suy luận

Với một mô hình LDA đã cho, có thể thực hiện suy luận ra các chủ đề có trong một văn bản mới chưa có trong tập văn bản huấn huyện bằng một tiến trình lấy mẫu tương tự.

Nhiệm vụ cụ thể của việc suy luận này là từ một văn bản mới $\widetilde{\boldsymbol{m}}$, được biểu diễn bởi một véc-tơ các từ $\vec{\boldsymbol{w}}$, chúng ta phải đi ước lượng các xác suất hậu nghiệm của các chủ đề $\tilde{\vec{\boldsymbol{z}}}$ cho bởi véc-tơ các từ của câu truy vấn $\vec{\boldsymbol{w}}$ và mô hình LDA đã cho trước $\boldsymbol{L}(\underline{\boldsymbol{\Theta}}, \underline{\boldsymbol{\Phi}})$:

$$\boldsymbol{p}(\vec{\boldsymbol{z}}|\, \vec{\boldsymbol{w}}, \boldsymbol{L}) = \boldsymbol{p}\left(\tilde{\vec{\boldsymbol{z}}}, \widetilde{\vec{\boldsymbol{w}}}, \vec{\boldsymbol{w}}, \vec{\boldsymbol{z}}\right).$$

Để tìm ra các giá trị cần thiết cho văn bản mới, công thức lấy mẫu mới được sử dụng trong phần này là:

$$\boldsymbol{p}\left(\widetilde{\boldsymbol{z}_\iota} = \boldsymbol{k}|\, \widetilde{\overrightarrow{\boldsymbol{z}_{-\iota}}}, \vec{\boldsymbol{w}}; \overrightarrow{\boldsymbol{z}_{-\iota}}, \vec{\boldsymbol{w}}\right) = \frac{\boldsymbol{n}_{\boldsymbol{k}}^{(\boldsymbol{t})} + \widetilde{\boldsymbol{n}}_{\boldsymbol{k},-\boldsymbol{i}}^{(\boldsymbol{t})} + \boldsymbol{\beta}_{\boldsymbol{i}}}{\left[\sum_{\boldsymbol{v}=\boldsymbol{1}}^{\boldsymbol{V}} \boldsymbol{n}_{\boldsymbol{k}}^{(\boldsymbol{v})} + \widetilde{\boldsymbol{n}}_{\boldsymbol{k}}^{(\boldsymbol{v})} + \boldsymbol{\beta}_{\boldsymbol{v}}\right] - \boldsymbol{1}} \frac{\boldsymbol{n}_{\widetilde{\boldsymbol{m}}}^{(\boldsymbol{k},-\boldsymbol{i})} + \boldsymbol{\alpha}_{\boldsymbol{k}}}{\left[\sum_{\boldsymbol{z}=\boldsymbol{1}}^{\boldsymbol{K}} \boldsymbol{n}_{\boldsymbol{m}}^{(\boldsymbol{z})} + \boldsymbol{\alpha}_{\boldsymbol{z}}\right] - \boldsymbol{1}}$$

Trong đó biến $\vec{n}_k^{(t)}$ là một biến mới, biến này đếm các đối tượng quan sát được của các thuật ngữ và các chủ đề trong văn bản mới. Công thức này đưa ra một ví dụ đầy màu sắc về các hoạt động của việc lấy mẫu hậu nghiệm Gibbs.

Công thức tính phân phối chủ đề cho văn bản mới như sau:

$$\vartheta_{m,k} = \frac{n_{\tilde{m}}^{(k)} + \alpha_k}{\sum_{z=1}^{K} n_{\tilde{m}}^{(z)} + \alpha_z}$$

1.4. Biểu diễn văn bản sử dụng véc-tơ từ

1.4.1. Giới thiệu

Phương pháp biểu diễn văn bản bằng véc-tơ từ, hay biểu diễn bằng từ khóa phân tán, biểu diễn các từ dưới dạng véc-tơ có số chiều cố định và nhỏ hơn nhiều so với kích thước từ vựng. Giá trị của mỗi thành phần trong véc-tơ biểu diễn đều là số thực và có giá trị và thường khác 0 (không chỉ là 0 hay 1 như one-hot), do vậy cách biểu diễn này còn được gọi là biểu diễn đặc (dense) khác với biểu diễn thưa (sparse) kiểu one-hot.

Mô hình này hướng đến việc phân tích ngữ nghĩa của từ và biểu diễn quan hệ giữa các từ thông qua véc-tơ biểu diễn của chúng. Mỗi véc-tơ biểu diễn của từ bây giờ không phải là thể hiện số thứ tự của từ trong tập từ điển nữa, nó là véc-tơ đặc trưng của từ. Nhờ đó ta có thể giảm đáng kể số chiều cần thiết và hoàn toàn có thể xác định độ tương đồng ngữ nghĩa, trái nghĩa hay một số quan hệ khác của các từ dựa trên véc-tơ biểu diễn của chúng.

Đặc biệt cách biểu diễn này có thể thể hiện được một số quan hệ về ngữ pháp và ngữ nghĩa giữa các từ. Ví dụ quan hệ số ít – số nhiều (danh từ tiếng Anh), so sánh bằng – so sánh hơn (tính từ tiếng Anh), đồng nghĩa, trái nghĩa...

1.4.2. Các bước thực hiện

Cách biểu diễn của từ trong phương pháp này thu được thông qua tiến hành học máy (không giám sát) trên các mô hình ngôn ngữ mạng nơ-ron nhân tạo [21]

hoặc các mô hình giảm số chiều khác [24]. Người ta đưa vào mạng nơ-ron một tập dữ liệu huấn luyện lớn có độ bao quát rộng để xác định trọng số thích hợp nhất của các nơ-ron trong mạng. Cuối quá trình huấn luyện, sau khi đã xác định trọng số người ta đưa từng từ vào đầu vào của mạng và lấy kết quả là biểu diễn dạng véc-tơ của từ ở đầu ra.

Dữ liệu huấn luyện

Dữ liệu huấn luyện là một tập dữ liệu văn bản bình thường, có khối lượng đủ lớn. Không cần chuẩn hóa gì thêm nhưng phạm vi phủ của dữ liệu huấn luyện cần rộng.

Giả sử ta có một cửa sổ trượt (sliding window) có kích thước cố định di chuyển dọc theo một câu: từ ở giữa được là mục tiêu (target) và những từ ở bên trái và bên phải của nó trong cửa sổ trượt được gọi là các từ ngữ cảnh (context).

Ví dụ sau đây mô tả các cặp từ mục tiêu và ngữ cảnh dưới dạng mẫu huấn luyện được tạo bởi cửa sổ từ có kích thước là 5 trượt dọc theo câu "Nghị_định này có hiệu_lực thi_hành kể từ ngày ký."

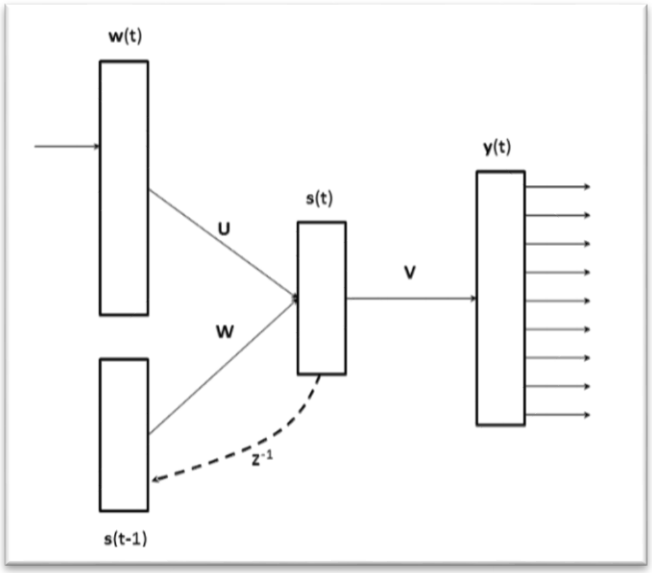
Bảng 1.2. Ví dụ về mẫu huấn luyện cho Skip-gram.

Cửa sổ trượt (kích thước = 5)	**Từ mục tiêu**	**Các từ ngữ cảnh**
[Nghị_định này có]	Nghị_định	này, có
[Nghị_định này có hiệu_lực]	này	Nghị_định, có, hiệu_lực
[Nghị_định này có hiệu_lực thi_hành]	có	Nghị_định, này, hiệu_lực, thi_hành
[này có hiệu_lực thi_hành kể]	hiệu_lực	này, có, thi_hành, kể
[có hiệu_lực thi_hành kể từ]	thi_hành	có, hiệu_lực, kể, từ
[hiệu_lực thi_hành kể từ ngày]	kể	hiệu_lực, thi_hành, từ, ngày
…	…	…
[từ ngày ký]	ký	từ, ngày

Mỗi cặp từ ngữ cảnh-mục tiêu được xem như một mẫu trong dữ liệu. Ví dụ, từ mục tiêu "có" trong ví dụ ở trên tạo ra bốn mẫu huấn luyện: ("có", "Nghị_định"), ("có", "này"), ("có", "hiệu_lực"), ("có", "thi_hành").

Thuật toán

Người ta đã từng sử dụng các mạng nơ-ron truyền thẳng (feed-forward) và mạng nơ-ron hồi quy để huấn luyện mô hình. Hiện tại thì phương pháp hồi quy được đánh giá là tốt hơn. Mô hình này gồm 3 lớp là: lớp đầu vào, lớp ẩn và ma trận trọng số hồi quy, lớp đầu ra.



Hình 1.6. Mô hình sử dụng mạng nơ-ron hồi quy.

Hình trên là sơ đồ của mô hình, trong đó $w(t)$ là từ đưa vào input ở thời điểm t, $y(t)$ là output tại thời điểm t, $s(t)$ là lớp ẩn tương ứng. $s(t)$ và $y(t)$ được tính theo công thức:

$$s(t) = f(Uw + Ws(t-1))$$

$$y(t) = g(Vs(t))$$

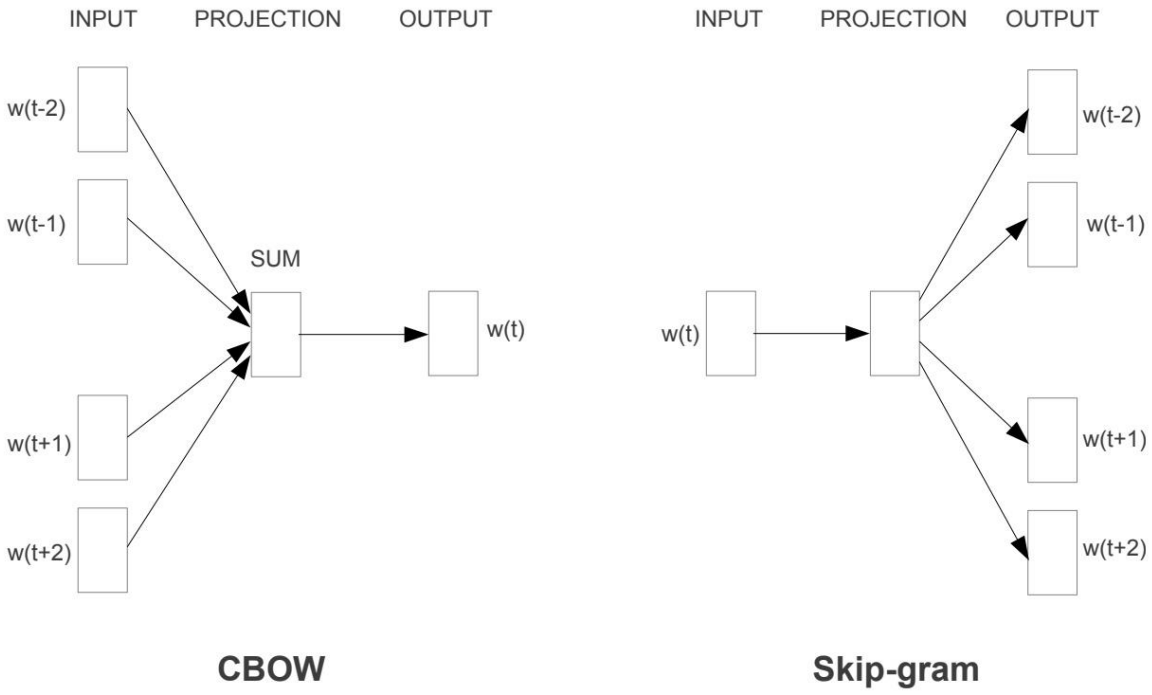
Trong đó:

$$f(z) = \frac{1}{1+e^{-z}}, g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

Đầu vào của lớp ẩn $s(t)$ không chỉ là từ được đưa vào ở đầu vào mà còn cả kết quả của lớp ẩn ở thời điểm trước đó.

Bản thân mô hình không chứa thông tin gì về ngữ pháp, do đó các biểu diễn véc-tơ của từ phản ánh các quan hệ về ngữ nghĩa và ngữ pháp hoàn toàn là do học từ dữ liệu.

Có 2 thuật toán học máy thường dùng trong việc học các biểu diễn từ của máy là CBOW (continuous bag of words) và Skip-gram. CBOW dựa vào các từ ngữ cảnh để dự đoán từ mục tiên, có thể hiểu đơn giản là nó dự đoán xác suất xuất hiện của từ thông qua ngữ cảnh xung quanh. Còn Skip-gram thì ngược lại, nó dự đoán các từ ngữ cảnh của từ mục tiêu đang xét, tức là nó dự đoán ngữ cảnh của từ.



Hình 1.7. Thuật toán Continuous bag of words và Skip-gram.

Ngoại trừ hai thuật toán nói trên, gần đây hơn (2014) nhóm nghiên cứu của đại học Stanford cũng giới thiệu thuật toán học máy GloVe (Global Vector) [24] cho phép đạt được véc-tơ từ với độ chính xác tốt hơn. Thuật toán này chú trọng các cặp từ cùng xuất hiện trong ngữ liệu để đạt được độ chính xác cao hơn về ngữ nghĩa. Ví dụ với ngữ liệu tiếng Anh, khảo sát trên các từ “ice” (băng) và steam (hơi nước) và các từ khác về tỉ lệ suất hiện đồng thời được kết quả như sau:

Bảng 1.3. Thống kê tỉ lệ xuất hiện đồng thời của các từ.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

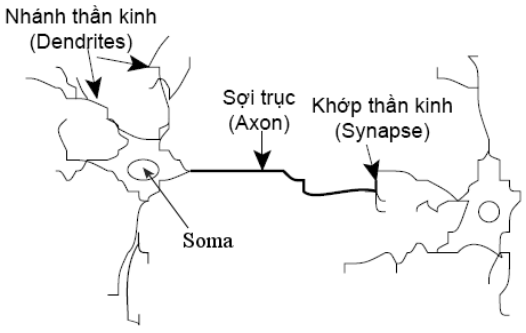
Theo đó xuất hiện đồng thời với "ice" nhiều nhất là "water" (nước) và "solid" (cứng). Còn "steam" xuất hiện đồng thời với "water" nhiều nhất sau đó là "gas" (hơi). Điều này phản ánh khá chính xác quan hệ gần về ngữ nghĩa của các từ trên.

1.5. Biểu diễn văn bản sử dụng mạng nơ-ron sâu

1.5.1. Giới thiệu về mạng nơ-ron nhân tạo

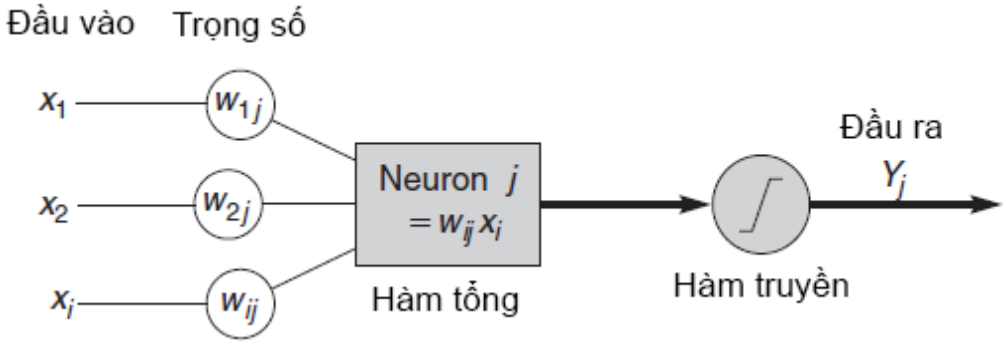
Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các nơ-ron được gắn kết để xử lý thông tin. ANN giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).

1.5.2. Cấu trúc và mô hình của một nơ-ron nhân tạo



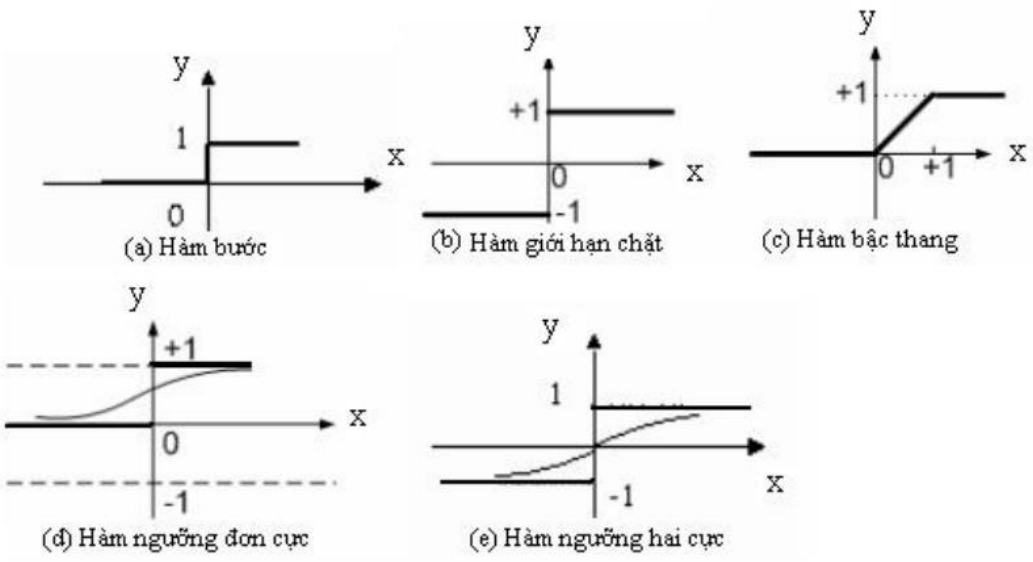
Hình 1.8. Mô hình một nơ-ron sinh học.

Mạng nơ-ron nhân tạo được lấy cảm hứng từ cách làm việc của bộ não con người. Các nơ-ron nhân tạo mô phỏng lại hoạt động của nơ-ron sinh học



Hình 1.9. Mô hình một nơ-ron nhân tạo.

Tương tự như nơ-ron sinh học, nơ-ron nhân tạo cũng nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả đến hàm truyền) và cho một tín hiệu đầu ra (là kết quả của hàm lan truyền).



Hình 1.10. Đồ thị các dạng hàm lan truyền.

Hàm lan truyền có thể có các dạng sau:

- Hàm bước: $y = \begin{cases} 1 \; khi \; x \geq 0 \\ 0 \; khi \; x < 0 \end{cases}$
- Hàm giới hạn chặt: $y = \begin{cases} 1 \; khi \; x \geq 0 \\ -1 \; khi \; x < 0 \end{cases}$

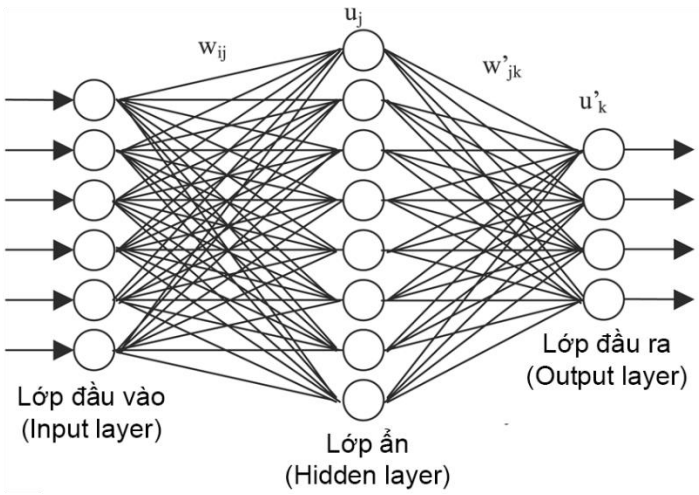
- Hàm bậc thang: $y = \begin{cases} 1\ khi\ x > 1 \\ x\ khi\ 0 \le x \le 1 \\ 0\ khi\ x < 0 \end{cases}$
- Hàm ngưỡng đơn cực: $y = \frac{1}{1+e^{-\lambda x}}\ với\ \lambda > 0$
- Hàm ngưỡng hai cực: $y = \frac{2}{1+e^{-\lambda x}} - 1\ với\ \lambda > 0$

Đồ thị các dạng hàm lan truyền được biểu diễn như Hình 1.10.

1.5.3. Cấu tạo và phương thức làm việc của mạng nơ-ron

Khi liên kết các đầu vào, đầu ra của nhiều nơ-ron với nhau, ta sẽ thu được một mạng nơ-ron. Việc ghép nối các nơ-ron trong mạng với nhau có thể theo nguyên tắc bất kỳ. Vì mạng nơ-ron là một hệ truyền đạt và xử lý tín hiệu nên có thể phân biệt các loại nơ-ron khác nhau. Các nơ-ron có đầu vào nhận thông tin từ môi trường bên ngoài khác với các nơ-ron có đầu vào được nối với các nơ-ron khác trong mạng, chúng được phân biệt với nhau qua véc-tơ hàm trọng số đầu vào w.

Nguyên lý cấu tạo chung của mạng nơ-ron gồm nhiều lớp, mỗi lớp bao gồm nhiều nơ-ron có cùng chức năng trong mạng. Thông thường một mạng nơ-ron sẽ bao gồm: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer). Trong đó có thể có nhiều lớp ẩn.



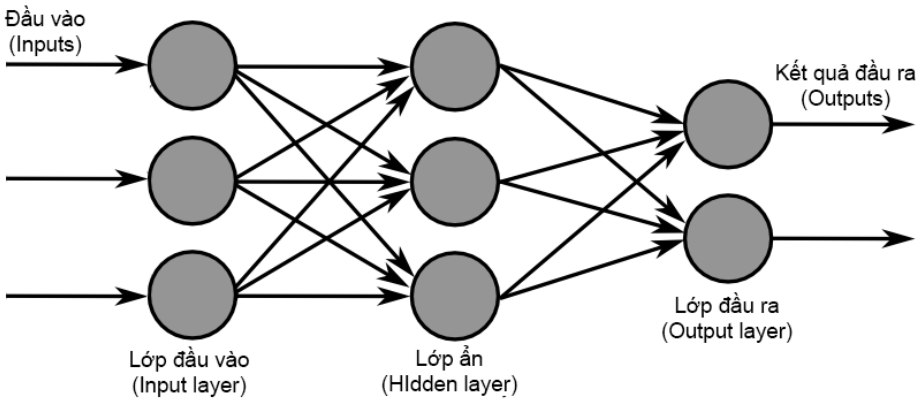
Hình 1.11. Mô hình cấu tạo của một mạng nơ-ron cơ bản.

Khi mới được hình thành thì mạng nơ-ron chưa có tri thức, tri thức của mạng sẽ được hình thành dần dần sau một quá trình học. Mạng nơ-ron được học bằng cách đưa vào những kích thích và mạng hình thành những đáp ứng tương ứng. Những đáp ứng tương ứng phù hợp với từng loại kích thích sẽ được lưu trữ. Giai đoạn này gọi là giai đoạn học hay huấn luyện của mạng. Khi đã hình thành tri thức mạng, mạng có thể giải quyết các vấn đề một cách đúng đắn. Đó có thể là những vấn đề ứng dụng rất khác nhau, được giải quyết chủ yếu dựa trên sự tổ chức hợp nhất giữa các thông tin đầu vào của mạng và các đáp ứng đầu ra.

1.5.4. Phân loại mạng nơ-ron

1.5.4.1. *Mạng nơ-ron truyền thẳng (Feed-forward Neural Network - FNN)*

Mạng nơ-ron truyền thẳng là kiến trúc mạng nơ-ron được sử dụng phổ biến. Đúng như tên của nó, các giá trị sẽ đi thẳng từ lớp đầu vào tới lớp đầu ra chứ không có chiều quay ngược lại (khác với mạng nơ-ron hồi quy được trình bày ở phần sau).

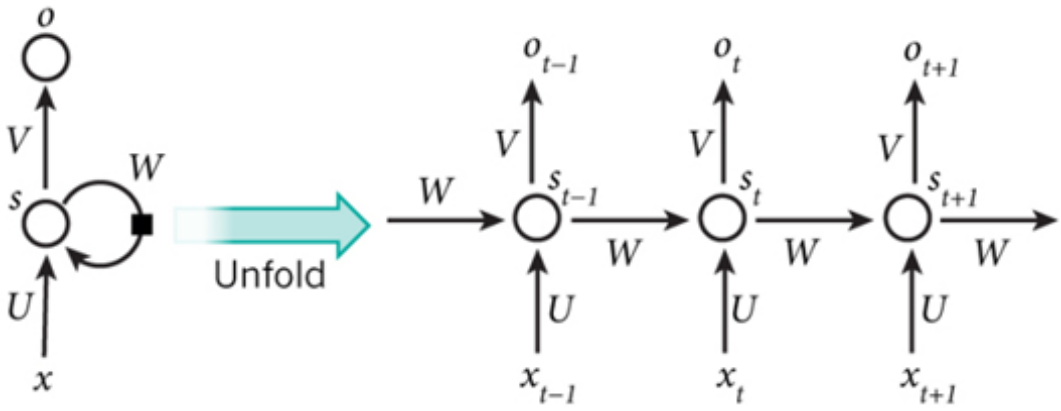


Hình 1.12. Mô hình mạng nơ-ro truyền thẳng.

1.5.4.2. *Mạng nơ-ron hồi quy (Recurrent Neural Networks – RNN)*

Ý tưởng về mạng nơ-ron hồi quy xuất phát từ mục đích muốn chuyển hóa một chuỗi các đầu vào thành chuỗi kết quả đầu ra, trong đó các thành phần trong chuỗi đều ảnh hưởng tới nhau. Ví dụ đối với bài toán chat bot, đầu vào là một câu (gồm nhiều từ và rõ ràng các từ phải liên quan tới nhau), mỗi từ được biểu diễn bằng một véc-tơ và ta mong muốn sử dụng mạng nơ-ron để ghi nhớ ngữ nghĩa của câu đó.

Mạng nơ-ron truyền thẳng - FNN đã đề cập ở trên không thể làm được điều này vì đầu vào của FNN chỉ là một bản ghi và các bản ghi khác nhau hoàn toàn không ảnh hưởng lẫn nhau. Nhưng mạng nơ-ron hồi quy có thể làm được điều này.



Hình 1.13. Mô hình mạng nơ-ron hồi quy.

1.5.5. Các mạng nơ-ron sâu

Mạng nơ-ron sâu Deep Neural Networks - DNN) là một mạng nơ-ron nhân tạo với nhiều lớp ẩn ẩn giữa lớp đầu vào và lớp đầu ra. Các mạng nơ-ron sâu có thể mô hình mối quan hệ phi tuyến tính phức tạp.

Các mạng nơ-ron sâu thường được thiết kế như các mạng nơ-ron truyền thẳng (Feed Forward Neural Network - FNN), những nghiên cứu gần đây đã áp dụng thành công kiến trúc học sâu đối với các mạng nơ-ron hồi quy, mạng nơ-ron LSTM cho các ứng dụng chẳng hạn như mô hình hóa ngôn ngữ. Các mạng nơ-ron nhân chập (Convolutional Neural Network - CNN) được sử dụng trong thị giác máy tính và thành công của chúng đã được ghi nhận. Gần đây hơn, các mạng nơ-ron nhân chập đã được áp dụng để mô hình hóa âm thanh cho nhận dạng giọng nói tự động (Automatic Speech Recognition - ASR).

Một mạng nơ-ron sâu có thể được huấn luyện bằng thuật toán lan truyền ngược. Các cập nhật trọng số có thể được thực hiện thông qua phương pháp *gradient descendent* sử dụng biểu thức sau:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}} + \xi(t)$$

Trong đó η là tốc độ học, C là hàm chi phí, ξ là một số ngẫu nhiên (stochastic term). Việc lựa chọn hàm chi phí phụ thuộc vào phương pháp học của mạng nơ-ron (có giám sát, không có giám sát hay tăng cường…) và hàm kích hoạt. Ví dụ, khi thực hiện học có giám sát với một bài toán phân loại nhiều lớp, ta thường chọn hàm kích hoạt là softmax và hàm chi phí là hàm entropy chéo (cross entropy).

Hàm softmax được định nghĩa như sau: Với đầu vào là véc-tơ có K phần tử z, hàm softmax sẽ cho ra một véc-tơ $\sigma(z)$ gồm K phần tử có giá trị trong khoảng (0; 1) và tổng các phần tử này bằng 1.

$$\sigma(z) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; với \; j = 1..K$$

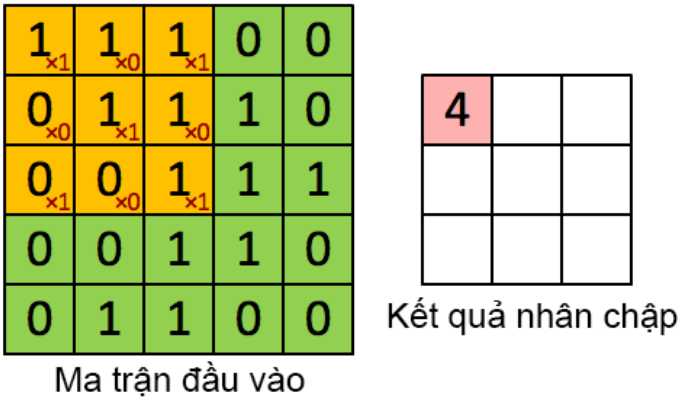
Hàm entropy chéo (cross entropy) được định nghĩa như sau:

$$C = -\sum_{j} d_j \log(p_j)$$

Trong đó d_j thể hiện xác suất mục tiêu của đơn vị đầu ra j và p_j là xác xuất đầu ra cho j sau khi áp dụng hàm kích hoạt.

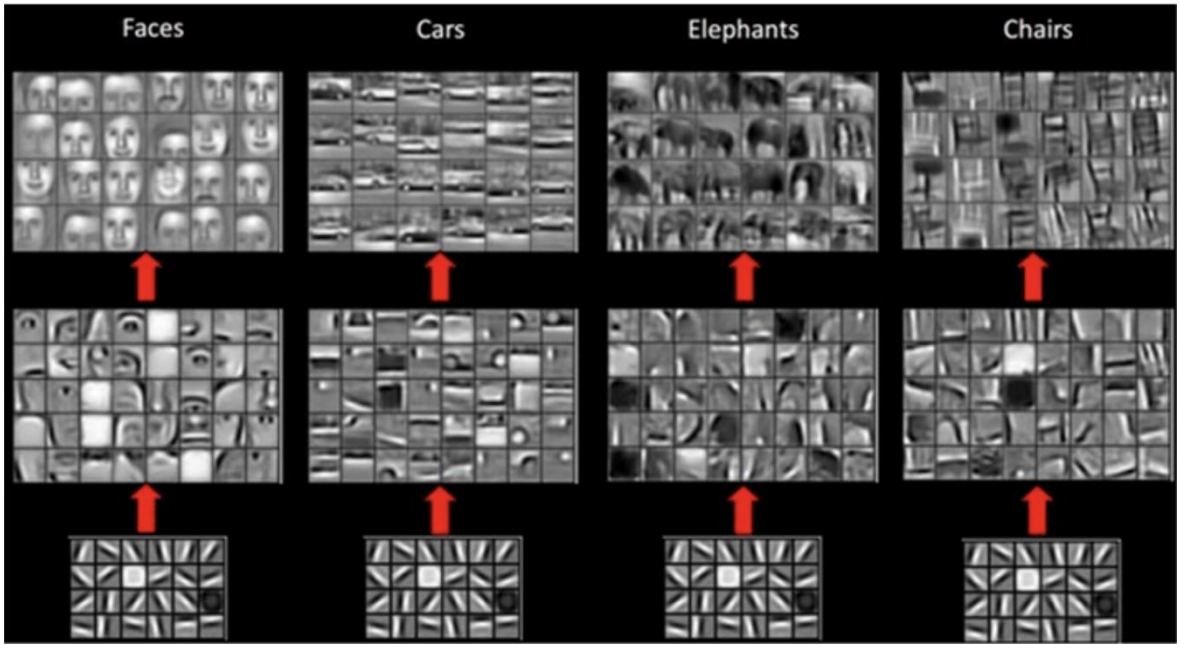
Mạng nơ-ron nhân chập

Nhân chập là một hàm của sổ trượt (sliding window function) được áp dụng trên một ma trận. Cửa sổ trượt, được gọi là một nhân (kernel), bộ lọc (filter) hoặc bộ dò (detector), là một ma trận vuông có cấp lẻ, các phần tử của nó là các trọng số. Ma trận cửa sổ sẽ được dịch chuyển lần lượt trên khắp ma trận gốc. Tâm của cửa sổ trượt sẽ được đặt trùng lên vị trí đang được tính nhân chập. Phép nhân chập sẽ tính tổng các tích của các phần tử trong ma trận cửa sổ với phần tử nằm bên dưới nó.

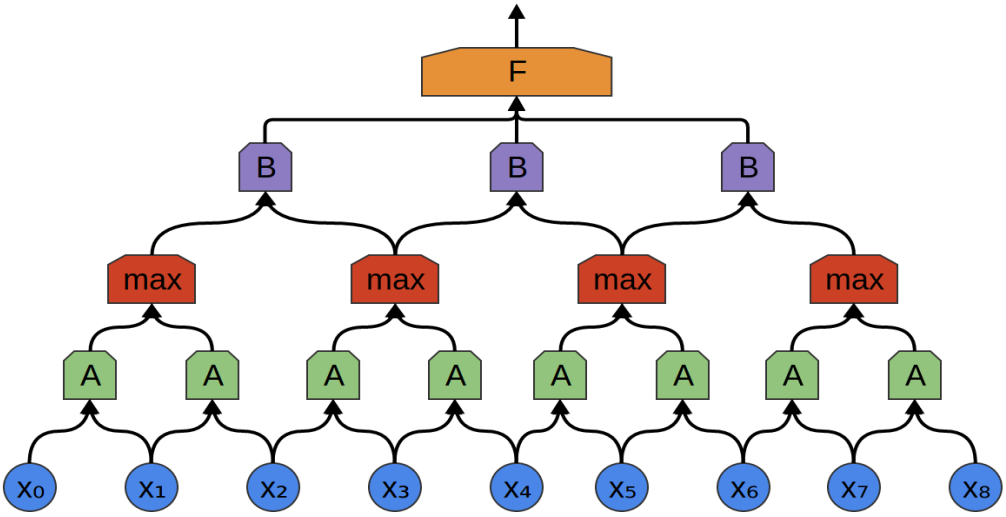


Hình 1.14. Minh họa phép nhân chập.

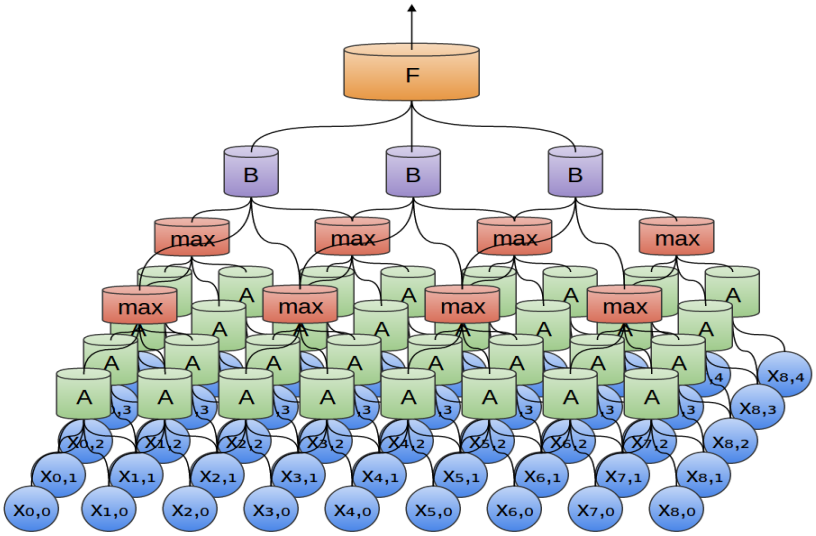
Mạng nơ-ron nhân chập là một dạng đặc biệt của mạng nơ-ron nhiều lớp. Trong mạng các lớp nhân chập (convolution layer) kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU hay tanh để tạo ra thông tin trừu tượng hơn cho các lớp tiếp theo [2] [23].



Hình 1.15. Các đặc trưng học được của một mạng nơ-ron nhân chập [23].



Hình 1.16. Kiến trúc cơ bản của mạng nơ-ron nhân chập một chiều



Hình 1.17. Kiến trúc cơ bản của mạng nơ-ron nhân chập hai chiều

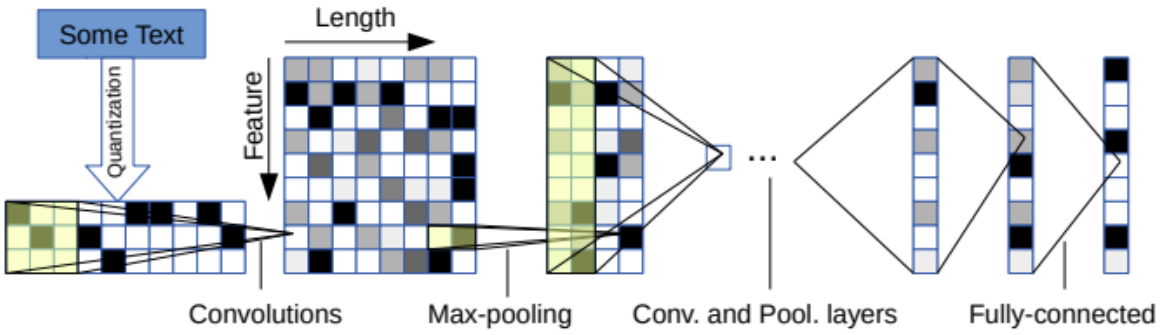
Trong mô hình mạng nơ-ron truyền thẳng truyền thống, các layer kết nối trực tiếp với nhau thông qua véc-tơ trọng số w (weighted vector). Các layer này còn được gọi là có kết nối đầy đủ (fully connected layer) hay affine layer.

Trong mô hình CNN, layer liên kết được với nhau thông qua cơ chế nhân chập. Layer tiếp theo là kết quả nhân chập từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở lớp tiếp theo sinh ra từ filter áp dụng lên một vùng cục bộ của lớp trước đó.

Mỗi lớp như vậy được áp dụng các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số lớp khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Tuy nhiên, ta sẽ không đi sâu vào khái niệm của các layer này.

Trong suốt quá trình huấn luyện, CNN tự học để nhận ra các đường cạnh từ các điểm ảnh trong lớp đầu tiên. Tiếp theo, nó sẽ học để nhận biết được các hình khối đơn giản từ các đường, cạnh trong lớp tiếp sau… cho đến việc nhận diện được các thực thể ở mức trừu tượng cao hơn. Lớp cuối cùng là lớp được sử dụng dể trích xuất các kết quả nhận diện cao nhất.

CNN được áp dụng trong các tác vụ như phân loại câu [14] [13] [31], phân tích cảm xúc, quan điểm [22], tìm kiếm theo ngữ nghĩa [10] [25], nhận dạng tiếng nói [1].



Hình 1.18. Mô hình CNN trong nghiên cứu [31].

1.5.6. Biểu diễn văn bản sử dụng mạng nơ-ron

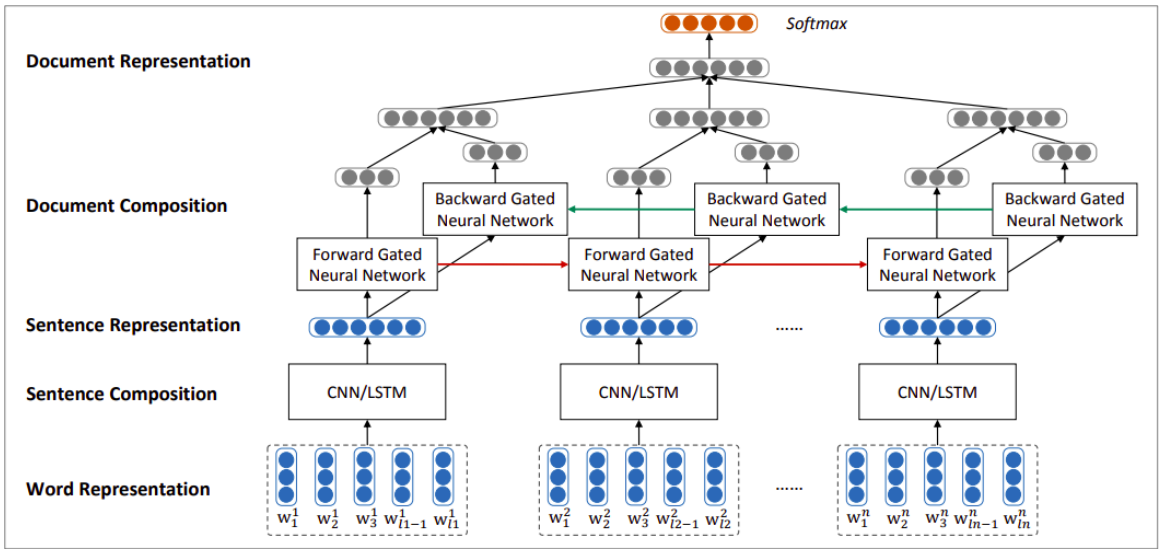
Trong nghiên cứu [14], tác giả đã thử nghiệm và đánh giá một kiến trúc CNN trên các bộ dữ liệu phân loại khác nhau, chủ yếu là phân loại cảm xúc và phân loại chủ đề. Kiến trúc CNN đã đạt được kết quả rất tốt trên các tập dữ liệu. Mạng được dùng trong nghiên cứu khá đơn giản. Lớp đầu vào và một câu bao gồm các véc-tơ biểu diễn của từ được nối với nhau. Theo sau là một lớp nhân chập với nhiều filter, sau đó là một lớp max-pooling và cuối cùng là bộ phân loại softmax. Tác giả cũng thực nghiệm với cả véc-tơ biểu diễn từ tĩnh (không được thay đổi trong quá trình huấn

luyện) và véc-tơ biểu diễn từ động (được thay đổi trong quá trình huấn luyện). Kiến trúc tương tự nhưng phức tạp hơn cũng đã được đề xuất trong nghiên cứu [13]. Nghiên cứu [28] sử dụng kiến trúc tương tự, bổ sung thêm một lớp để thực hiện phân cụm ngữ nghĩa cho câu.

Nghiên cứu [12] huấn luyện một CNN không sử dụng véc-tơ từ như word2vec hay GloVe mà áp dụng phép nhân chập trực tiếp lên véc-tơ one-hot. Trong nghiên cứu [11], tác giả đã mở rộng mô hình với một "region embedding" không giám sát được học bởi một CNN dự đoán ngữ nghĩa của một vùng chữ. Cách tiếp cận trong hai nghiên cứu này hoạt động tốt với các đoạn chữ dài (ví dụ như nhận xét về phim), nhưng chưa thực sự hiệu quả với các đoạn chữ ngắn (như tweet).

Một trường hợp sử dụng thú vị khác của CNN có thể được tìm thấy ở nghiên cứu [8] và [25] từ Microsoft Research. Hai nghiên cứu này mô tả các học biểu diễn ngữ nghĩa (semantically meaningful representations) của câu để có thể dùng trong tìm kiếm thông tin.

Nghiên cứu [26] đề xuất mô hình biểu diễn một văn bản thành một véc-tơ. Véc-tơ này sau đó được sử dụng để phân loại cảm xúc cho văn bản. Mô hình bao gồm hai mô đun chính là Sentence Composition và Document Composition.



Hình 1.19. Mô hình trong nghiên cứu [26].

Sentence Composition học biểu diễn cho câu từ biểu diễn của các từ. Mỗi từ được biểu diễn bởi một véc-tơ có giá trị là các số thực (word embedding). Tất cả các véc-tơ từ của câu được xếp chồng lên nhau tạo thành một ma trận (embedding matrix). Tác giả thử nghiệm cả CNN và LSTM để học biểu diễn của câu trong mô đun này (Kết quả thực nghiệm với LSTM trong nghiên cứu cho kết quả tốt hơn).

Véc-tơ biểu diễn câu thu được từ Sentence Composition sẽ được đưa vào Document Composition để tính toán biểu diễn cho cả văn bản. Trong mô đun này, tác giả sử dụng một mạng nơ-ron hồi quy gọi là Gated Recurrent Neural Network. Kết quả thu được cuối cùng là một véc-tơ biểu diễn cho văn bản, được sử dụng để phân lại cảm xúc cho văn bản đó.

1.6. Kết luận chương

Trong phần đầu của chương này, luận văn đã trình bày tổng quan về bài toán tìm kiếm thông tin nói chung và bài toán tìm kiếm văn bản pháp quy nói riêng, bao gồm khái, kiến trúc hệ thống và mô hình tìm kiếm thông tin.

Chương này cũng đã trình bày về các phương pháp biểu diễn văn bản bao gồm: biểu diễn sử dụng từ khóa, biểu diễn sử dụng chủ đề ẩn, biểu diễn sử dụng véc-tơ từ, biểu diễn sử dụng mạng nơ-ron sâu. Trong đó, phương pháp biểu diễn sử dụng từ khóa còn nhiều hạn chế, chưa biểu diễn được tốt về ngữ nghĩa, phương pháp biểu sử dụng mạng nơ-ron sâu đang cho thấy hiệu quả cao trong các nghiên cứu gần đây.

CHƯƠNG 2. ỨNG DỤNG BIỂU DIỄN VĂN BẢN BẰNG MẠNG NƠ-RON SÂU TRONG TÌM KIẾM VĂN BẢN PHÁP QUY

Chương này sẽ đề xuất phương pháp biểu diễn văn bản sử dụng mạng nơ-ron nhân chập kết hợp với cơ chế Attention áp dụng cho bài toán tìm kiếm văn bản pháp quy.

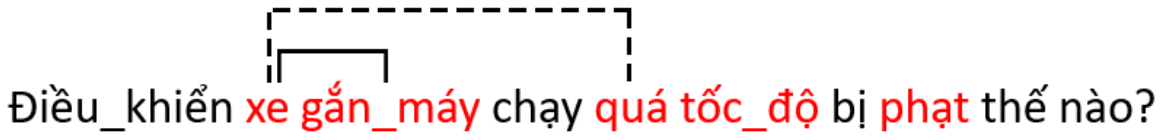
2.1. Ý tưởng

Mỗi điều luật được coi như một văn bản cần tìm kiếm. Tổng quan ý tưởng của phương pháp hai khâu chính. Đầu tiên là biểu điều luật và truy vấn dưới dạng véc-tơ. Sau đó dùng hàm tích vô hướng để so khớp, ước tính độ liên quan giữa chúng.

Mô hình gồm hai mô-đun chính là Mô-đun Biểu diễn truy vấn (Query Encoder) và Mô-đun Biểu diễn điều luật (Article Encoder). Hai mô-đun này sẽ được mô tả chi tiết hơn ở các mục phía sau trong chương này.

Trong mỗi mô-đun, mạng nơ-ron nhân chập sẽ được dùng để ghi nhận các thông tin ngữ cảnh. Sau đó cơ chế Attention sẽ được áp dụng để tính toán các biểu diễn của truy vấn hoặc điều luật.

Cơ chế Attention được lấy cảm hứng từ cách con người chỉ tập trung chú ý vào một số vùng trên ảnh khi nhìn hoặc một số từ trong câu khi đọc.



Hình 2.1. Ví dụ về cách con người chú ý vào một số từ trong câu.

Khi đọc một câu chúng ta cũng chú ý hơn vào một số từ khi hiểu ý nghĩa của câu đó. Trong câu ví dụ ở hình ở trên, ta sẽ chú ý hơn vào các từ “xe”, “gắn máy”, “quá”, “tốc độ”, “phạt”.

Ví dụ trên cũng thể hiện mối liên hệ giữa các từ trong một câu. Khi gặp từ "xe" chúng ta sẽ mong bắt gặp một từ mô tả loại xe ngay sau đó. Do vậy, từ "xe" sẽ liên hệ chặt chẽ với từ "gắn máy" hơn từ "quá" hoặc các từ khác trong câu.

Tóm lại, Attention trong học sâu có thể được hiểu theo nghĩa rộng là một véc-tơ chứa trọng số thể hiện độ quan trọng của các thành phần khi biểu diễn ý nghĩa của đối tượng chứa các thành phần đó hoặc để suy luận một thành phần khác (ví dụ như các điểm ảnh hay các từ trong câu).

Các loại cơ chế Attention

Đầu tiên, cơ chế Attention được ra đời để phục vụ bài toán dịch máy. Sau khi đạt được kết quả tốt, nó đã được mở rộng phạm vi áp dụng sang thị giác máy tính. Nhiều loại cơ chế Attention khác nhau đã ra đời.

Bảng 2.1. Hàm alignment score trong các cơ chế attention.

Tên	Hàm alignment score
Content-base attention [9]	$score(s_t, h_i) = cosine[s_t, h_i]$
Additive attention [3]	$score(s_t, h_i) = v_a^T \tanh(W_a[s_t; h_i])$
Location-Base [17]	$score(s_t, h_i) = softmax(W_a s_t)$
General [17]	$score(s_t, h_i) = s_t^T W_a h_i$
Dot-Product [17]	$score(s_t, h_i) = s_t^T h_i$
Scaled Dot-Product [27]	$score(s_t, h_i) = \frac{s_t^T h_i}{\sqrt{n}}$

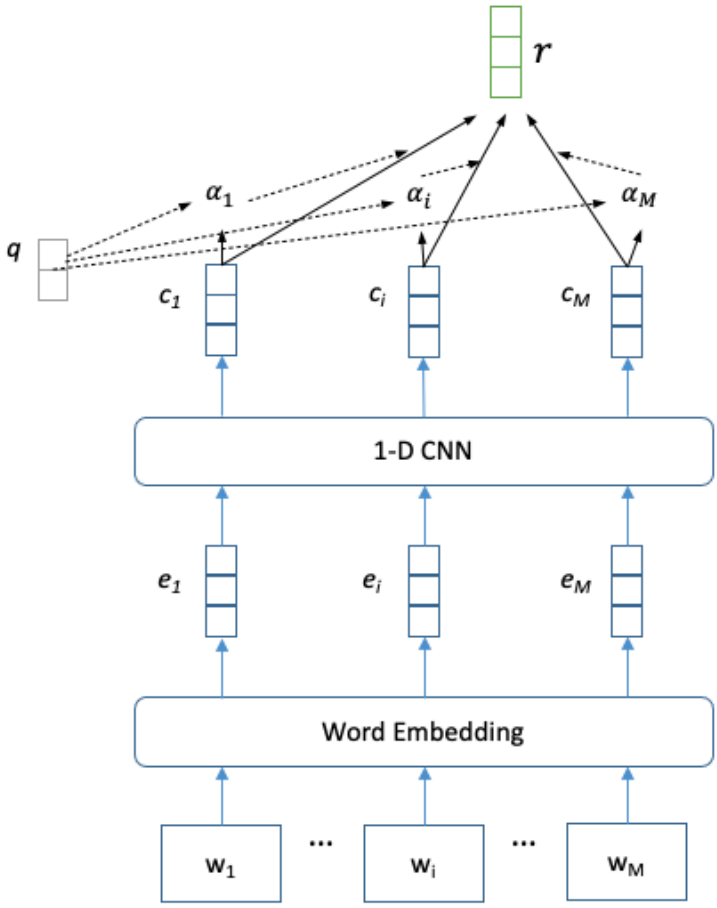
Bảng 2.2. Các loại cơ chế attention.

Tên	**Định nghĩa**
Self-Attention [7]	Liên hệ các vị trí khác nhau của cùng một chuỗi đầu vào. Về mặt lý thuyết, self-attention có thể áp dụng bất kỳ hàm tính điểm nào ở trên, nhưng chỉ cần thay thế chuỗi mục tiêu bằng cùng một chuỗi đầu vào.

Global/Soft [30]	Sử dụng toàn bộ không gian trạng thái đầu vào.
Local/Hard [17] [30]	Sử dụng một phần của không gian trạng thái đầu vào, ví dụ một vùng của hình ảnh đầu vào.

2.2. Mô-đun Biểu diễn truy vấn

Mô-đun này biến đổi truy vấn thành véc-tơ biểu diễn. Kiến trúc của nó được mô tả ở Hình 2.2, bao gồm ba lớp: word embedding, lớp nhân chập (Convolutional Neural Network - CNN) và attention.



Hình 2.2. Kiến trúc của Mô-đun Biểu diễn truy vấn.

Word embedding biến đổi các từ trong truy vấn thành véc-tơ thông qua một ma trận ánh xạ. Các véc-tơ từ sau quá trình huấn luyện có thể phản ánh quan hệ ngữ nghĩa giữa các từ. Kí hiệu truy vấn đầu vào có dạng $[w_1, w_2, \dots, w_M]$ với M là chiều dài của truy vấn, lớp word embedding sẽ cho ra chuỗi véc-tơ $[e_1, e_2, \dots, e_M]$.

Lớp word embedding hoạt động như sau: Đầu tiên nó sẽ tạo ra một ma trận embedding kích thước $|V| \times d$. Trong đó $|V|$ là kích thước từ điển, là kích thước véc-tơ từ mong muốn. Các phần tử trong ma trận được khởi tạo một cách ngẫu nhiên và được học trong quá trình huấn luyện mạng. Sau khi huấn luyện, véc-tơ từ của từ thứ i trong tập từ điển chính là hàng thứ i của ma trận embedding.

Lớp nhân chập có nhiệm vụ ghi nhận những ngữ cảnh cục bộ (local context) xung quanh các từ. Ngữ cảnh này quan trọng trong việc tính toán biểu diễn của cả chuỗi. Ví dụ với chuỗi "Điều_khiển xe gắn_máy chạy quá tốc_độ bị phạt thế nào?", các từ ngữ cảnh của "gắn_máy" là "xe" và "chạy" hữu ích cho việc hiểu rằng đó là một loại phương tiện giao thông. Ngữ cảnh c_i của từ i được tính bởi công thức:

$$c_i = ReLU(F \times e_{(i-K):(i+K)} + b_t)$$

Trong đó

- $e_{(i-K):(i+K)}$ là các véc-tơ từ bắt đầu từ vị trí $(i-K)$ tới $(i+K)$
- $F \in \mathbb{R}^{N_f \times (2K+1)D}$ và $b_t \in \mathbb{R}^{N_f}$ là nhân/bộ lọc (kernel/filter) và bias của CNN, N_f là số filter, $2K+1$ là kích thước cửa sổ.

Trong một chuỗi, mỗi từ có đóng góp khác nhau vào ý nghĩa của cả chuỗi. Dựa trên quan sát đó, mô hình sử dụng lớp attention để tính trọng số cho từng từ. Trọng số α_i của từ thứ i được tính như sau:

$$a_i = q^T \tanh(V \times c_i + v)$$
$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^{M} \exp(a_j)}$$

Trong đó q là attention query vector.

Biểu diễn cuối cùng của truy vấn được tính bằng tổng của c_i với trọng số α_i được tính theo công thức:

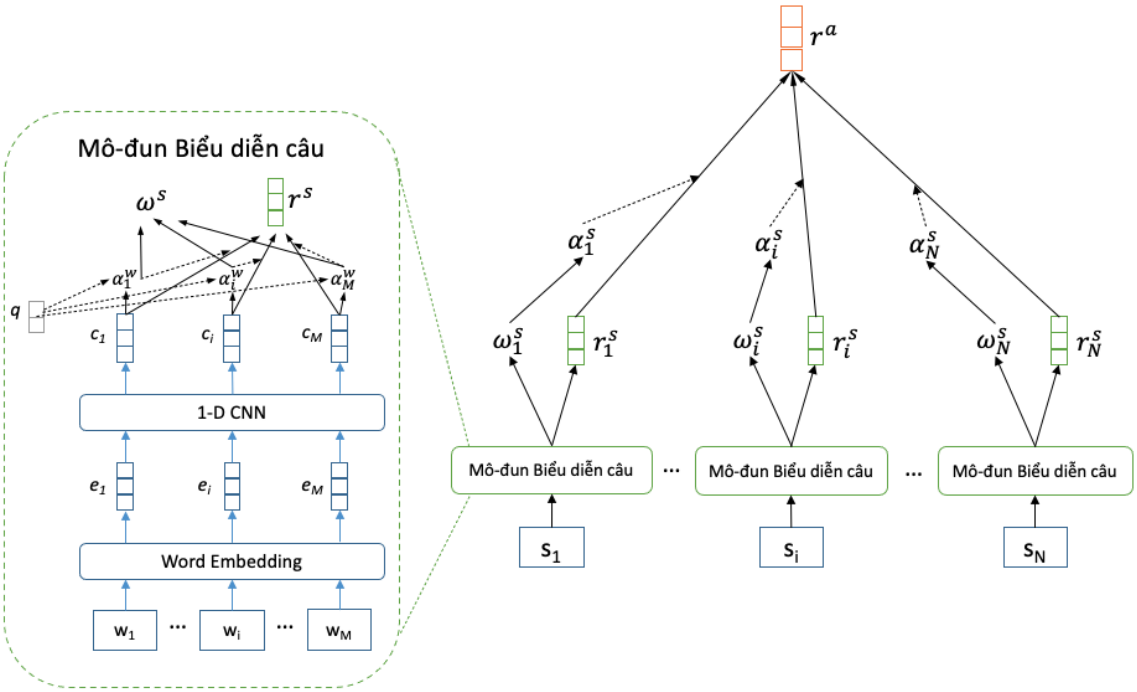
$$r = \sum_{j=1}^{M} \alpha_j c_j$$

2.3. Mô-đun Biểu diễn điều luật

Mô-đun này biến đổi điều luật dưới dạng một đoạn văn thành một véc-tơ. Kiến trúc của nó được mô tả ở Hình 2.3.

Thay vì xem như một chuỗi dài, mô hình xem điều luật như một đoạn văn tạo thành bởi các câu và sẽ được biểu diễn bằng kiến trúc phân cấp. Đầu tiên các câu sẽ được biểu diễn thành véc-tơ r^s bởi Mô-đun Biểu diễn câu (Sentence Encoder). Thành phần này có kiến trúc giống với Mô-đun Biểu diễn truy vấn (Query Encoder).

Tương tự như các từ trong một chuỗi, mỗi câu cũng có đóng góp khác nhau vào ý nghĩa chung của cả điều luật. Do vậy mô hình cũng dùng một lớp attention để tính trọng số cho từng câu.



Hình 2.3. Kiến trúc của Mô-đun Biểu diễn điều luật.

Một câu có chứa nhiều từ quan trọng hơn thì sẽ mang nhiều ý nghĩa hơn. Nhưng khi qua hàm softmax của Mô-đun Biểu diễn câu, trọng số của các từ được đưa về dạng có tổng bằng 1. Do đó, một câu không có từ nào quan trọng thì trọng số của

các từ trong câu đó cũng được gán giá trị. Nếu dùng véc-tơ biểu diễn các câu lấy từ đầu ra của Mô-đun Biểu diễn câu để tính trọng số cho các câu trong điều luật sẽ không làm nổi bật được các câu quan trọng. Thay vào đó, mô hình tính trọng số của các câu thông qua tổng trọng số của các từ khi chưa qua softmax và chuẩn hóa bằng cách chia cho độ dài của câu.

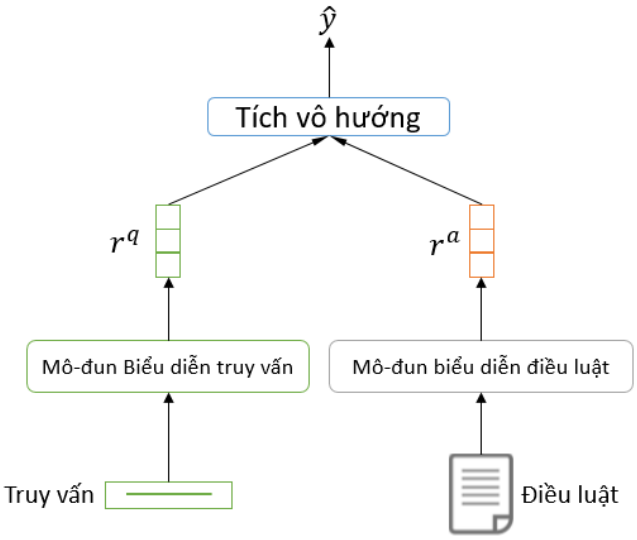
Các trọng số này sau đó được đưa qua hàm sparsemax [20] thay vì softmax. Lí do là vì hàm sparsemax sẽ đánh trọng số cao hơn (so với softmax) cho các câu quan trọng và có thể đánh trọng số bằng 0 cho các câu kém quan trọng. Điều này sẽ giúp véc-tơ biểu diễn của điều luật tốt hơn.

Véc-tơ r^a biểu diễn cho điều luật sẽ được tính bởi các công thức sau:

$$\omega^s = \frac{\sum_i a_i^w}{|s|}$$
$$\alpha_i^s = sparsemax(\omega_i^s)$$
$$r^a = \sum_{j=1}^{N} \alpha_j^s r_j^s$$

2.4. So khớp, tính độ liên quan

Hình 2.4 mô tả cách hệ thống tính độ liên quan giữa một điều luật và một truy vấn.



Hình 2.4. Tính độ liên quan giữa một điều luật và một truy vấn.

Độ liên quan giữa một điều luật và một truy vấn được tính bằng tích vô hướng giữa hai véc-tơ biểu diễn của chúng.

Hệ thống được huấn luyện bằng kỹ thuật "*negative sampling*". Hệ thống gán nhãn các điều luật liên quan tới một truy vấn là "*positive*", các điều luật không liên quan là "*negative*". Với mỗi điều luật liên quan, hệ thống chọn mẫu K điều luật không liên quan. Hệ thống sẽ học để phân loại K + 1 điều luật này là liên quan tới truy vấn hay không.

2.5. Kết luận chương

Chương này đã đề xuất phương pháp biểu diễn văn bản sử dụng mạng nơ-ron nhân chập kết hợp với cơ chế Attention áp dụng cho bài toán tìm kiếm văn bản pháp quy.

Chương tiếp theo sẽ trình bày quá trình thu thập, xây dựng dữ liệu, hệ thống và thử nghiệm, đánh giá phương pháp đã đề xuất.

CHƯƠNG 3. THỬ NGHIỆM VÀ ĐÁNH GIÁ

Chương này sẽ trình bày quá trình xây dựng tập dữ liệu văn bản quy, câu hỏi về lĩnh vực pháp luật, việc ứng dụng phương pháp biểu diễn văn bản sử dụng mạng nơ-ron nhân chập kết hợp với cơ chế Attention để xây dựng hệ thống tìm văn bản pháp quy. Cuối cùng là phần thực nghiệm, so sánh với các phương pháp khác.

3.1. Xây dựng tập dữ liệu văn bản pháp quy và câu hỏi

3.1.1. Xây dựng tập dữ liệu văn bản pháp quy tiếng Việt

Tập dữ liệu văn bản pháp quy sử dụng trong luận văn được thu thập từ trang vbpl.vn. Các văn bản được thu thập là các văn bản còn hiệu lực và thuộc các loại sau: bộ luật, luật, nghị định, thông tư, thông tư liên tịch.

Các văn bản được thu thập bằng cách bóc tách dữ liệu HTML từ trang web. Mỗi văn bản được bóc tách thành các điều và các thông tin kèm theo, sau đó được lưu vào MongoDB. Các thông tin kèm theo được cho bởi bảng bên dưới.

Bảng 3.1. Các thông tin di kèm văn bản.

Trên trường	Ý nghĩa
ten_van_ban	Tên văn bản
so_hieu	Số hiệu văn bản
ngay_ban_hanh	Ngày ban hành văn bản
ngay_hieu_luc	Ngày văn bản có hiệu lực
noi_ban_hanh	Nơi ban hành văn bản
nguoi_ky	Người ký văn bản
link	Đường dẫn của văn bản trên web

Tổng cộng đã thu thập được **8586** văn bản, chia thành **117545** điều.

3.1.2. Xây dựng tập câu hỏi và câu trả lời chuẩn

Các câu hỏi được thu thập từ các trang hỏi đáp pháp luật [32][33][34]. Mỗi câu hỏi ban đầu gồm có tiêu đề câu hỏi, chi tiết câu hỏi và câu trả lời. Do phần tiêu đề đã thể hiện được nội dung chính của câu hỏi, còn phần chi tiết câu hỏi đôi khi chứa nhiều thông tin nhiễu, không cần thiết hoặc không rõ ý định của câu hỏi nên chỉ chọn phần tiêu đề truy vấn. Các văn bản và điều, khoản liên quan được bóc tác từ phần trả lời bằng cách thủ công. Sau đó, người có chuyên môn về pháp luật sẽ thực hiện xem xét lại các điều khoản này đã thỏa mãn câu hỏi chưa, có còn hiệu lực không. Nếu chưa đáp ứng được thì sẽ thay bằng văn bản, điều, khoản khác.

Tập câu hỏi cuối cùng gồm 2925 câu, mỗi câu hỏi có thể có một hoặc nhiều cách hỏi khác nhau sau đây được gọi là truy vấn. Tổng cộng có **5922** truy vấn.

Dưới đây là bảng một số thống kê về bộ câu hỏi:

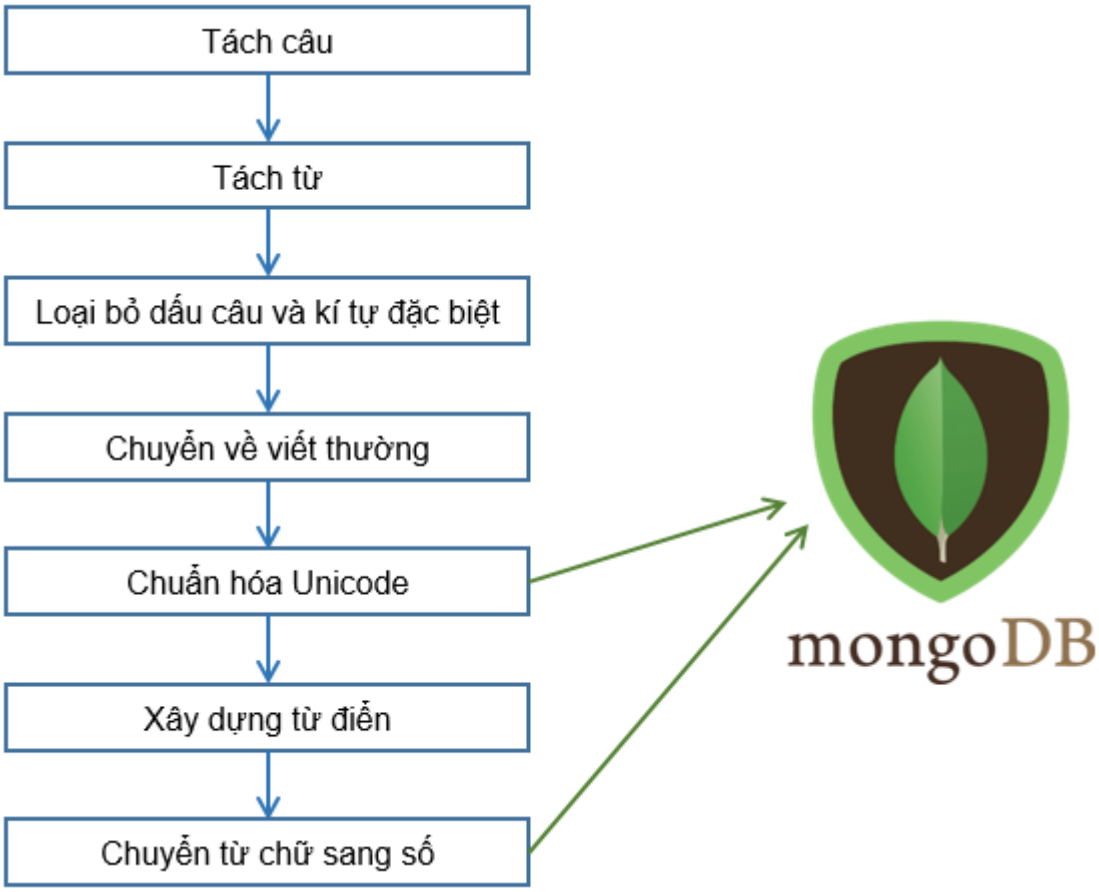
Bảng 3.2. Một số thống kê về bộ câu hỏi.

Tiêu chí	Giá trị nhỏ nhất	Giá trị lớn nhất	Trung bình
Độ dài câu hỏi theo số từ	2	36	12.5
Độ dài câu hỏi theo số âm tiết	4	45	17.3
Số văn bản liên quan tới một câu hỏi	1	4	1.19
Số điều liên quan tới một câu hỏi	1	11	1.6

3.2. Xây dựng hệ thống

3.2.1. Tiền xử lý dữ liệu

Đầu tiên, các điều luật sẽ được tách thành các câu. Sau đó mỗi câu sẽ được tách thành các từ. Hai bước này được thực hiện sử dụng thư viện hàm `sent_tokenize` và `word_tokenize` của thư viện Underthesea [35].



Hình 3.1. Các bước tiền xử lý dữ liệu.

Tiếp theo dữ liệu được loại bỏ các dấu câu và kí tự đặc biệt. Sau đó, dữ liệu sẽ được chuyển tất cả các ký tự viết hoa thành viết thường. Điều này giúp giảm số từ vựng trong tập từ dùng để biểu diễn văn bản, vì các từ giống nhau nhưng khác nhau do viết hoa hay viết thường sẽ không bị tính thành hai từ. Ví dụ từ "Cha" và "cha" sẽ được tính là giống nhau do chuyển hết thành chữ thường. Tập từ điển nhỏ hơn giúp chương trình chạy nhanh hơn.

Dữ liệu thu thập được chứa cả kí tự dạng Unicode đựng sẵn và Unicode tổ hợp. Máy tính sẽ hiểu một kí tự có ý nghĩa giống nhau ở hai dạng mã hóa là khác nhau nên sẽ phải thực hiện bước chuẩn hóa. Tất cả các kí tự ở dạng Unicode tổ hợp sẽ được ánh xạ sang dạng Unicode đựng sẵn. Phương pháp TF-IDF và BM25 có thể sử dụng được dữ liệu sau bước chuẩn hóa nên dữ liệu này được lưu trữ vào cơ sở dữ liệu.

Với phương pháp dùng mạng nơ-ron, dữ liệu sẽ phải được tiền xử lý thêm. Cụ thể, bước tiếp theo là xây dựng một tập từ điển. Tập từ điển này ánh xạ mỗi từ thành một số nguyên. Cần phải làm như vậy vì khi đưa vào mạng nơ-ron, dữ liệu phải ở dạng số. Bước cuối cùng là chuyển dữ liệu từ chuỗi các từ thành chuỗi các số nguyên và lưu vào cơ sở dữ liệu.

3.2.2. Xây dựng hệ thống tìm kiếm sử dụng phương pháp TF-IDF và BM25

Hệ thống này được xây dựng sử dụng Elasticsearch. Elasticsearch cho phép lưu trữ dữ liệu và tạo chỉ mục theo phương pháp biểu diễn TF-IDF và BM25. Mỗi điều sau khi tiền xử lý được lưu thành một bản ghi trong Elasticsearch, bao gồm các trường như bảng sau:

Bảng 3.3. Các trường của một bản ghi trong Elasticsearch.

Tên trường	Ý nghĩa
so_hieu	Số hiệu văn bản
ten_van_ban	Tên văn bản đã được tiền xử lý
ten_van_ban_raw	Tên văn bản khi chưa tiền xử lý
ten_dieu	Tên điều
tieu_de	Tiêu đề của điều đã được tiền xử lý
tieu_de_raw	Tiều đề của điều khi chưa tiền xử lý
noi_dung	Nội dung của điều đã được tiền xử lý
noi_dung_raw	Nội dung của điều khi chưa tiền xử lý
link	Đường dẫn tới văn bản trên trang web

Các trường ten_van_ban, tieu_de, noi_dung được đánh chỉ mục theo phương pháp TF-IDF và BM25 để phục vụ tìm kiếm. Khi nhận được truy vấn, hệ thống sẽ tiền xử lý rồi sử dụng API của Elasticsearch để tìm kiếm theo phương pháp tương ứng.

3.2.3. Xây dựng hệ thống tìm kiếm sử dụng phương pháp biểu diễn văn bản bằng mạng CNN kết hợp với cơ chế Attention

3.2.3.1. *Huấn luyện*

Hệ thống này sẽ lấy kết quả tìm kiếm bằng phương pháp BM25 dựa trên Elasticsearch làm đầu vào rồi dùng mô hình mạng nơ-ron để xếp hạng lại kết quả.

Mô hình được huấn luyện dựa trên kỹ thuật negative sampling. Gọi các điều liên quan đến câu truy vấn là positive, các điều không liên quan là negative. Các positive chính là các điều trong phần trả lời của câu hỏi tương ứng trong tập dữ liệu câu hỏi. Với mỗi positive của từng câu truy vấn, chọn ra K negative. Cho mô hình dự đoán nhãn của K + 1 điều này là liên quan hay không liên quan tới truy vấn.

K điều negative được chọn từ các điều được xếp hạng cao nhất trong kết quả trả về khi tìm kiếm bằng phương pháp BM25 kết hợp với chọn ngẫu nhiên các điều không liên quan trong cơ sở dữ liệu.

Chuẩn bị dữ liệu huấn luyện:

Dữ liệu đã được tiền xử lý sau bước chuyển từ chữ sang số được dùng để tạo dữ liệu huấn luyện cho mạng nơ-ron. Gọi QUERY_LEN là độ dài của câu truy vấn, SENTENCE_LEN là độ dài một câu trong điều, NUM_SENTENCES là số câu trong một điều, hệ thống thực hiện các việc:

- Các câu truy vấn có độ dài nhỏ hơn QUERY_LEN sẽ được thêm các số 0 vào cuối cho đủ độ dài bằng QUERY_LEN.
- Các câu truy vấn có độ dài lớn hơn QUERY_LEN sẽ cắt bớt cho đủ độ dài bằng QUERY_LEN.
- Các câu có độ dài nhỏ hơn SENTENCE_LEN sẽ được thêm các số 0 vào cuối cho đủ độ dài bằng SENTENCE_LEN.
- Các câu có độ dài lớn hơn SENTENCE_LEN sẽ cắt bớt cho đủ độ dài bằng SENTENCE_LEN.

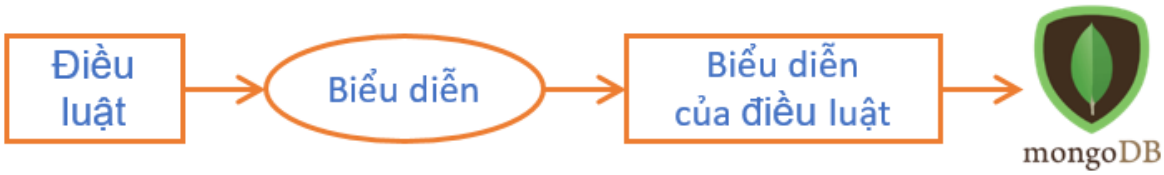
- Các điều có số câu nhỏ hơn NUM_SENTENCES sẽ được thêm các chuỗi số 0 vào cuối cho đủ số câu bằng NUM_SENTENCES.
- Các điều có số câu lớn hơn NUM_SENTENCES sẽ được cắt bớt cho đủ số câu bằng NUM_SENTENCES.

Một ví dụ liệu huấn luyện sẽ bao gồm

- X: câu truy vấn, 1 positive, K negative
- Y: nhãn tương ứng cho các điều

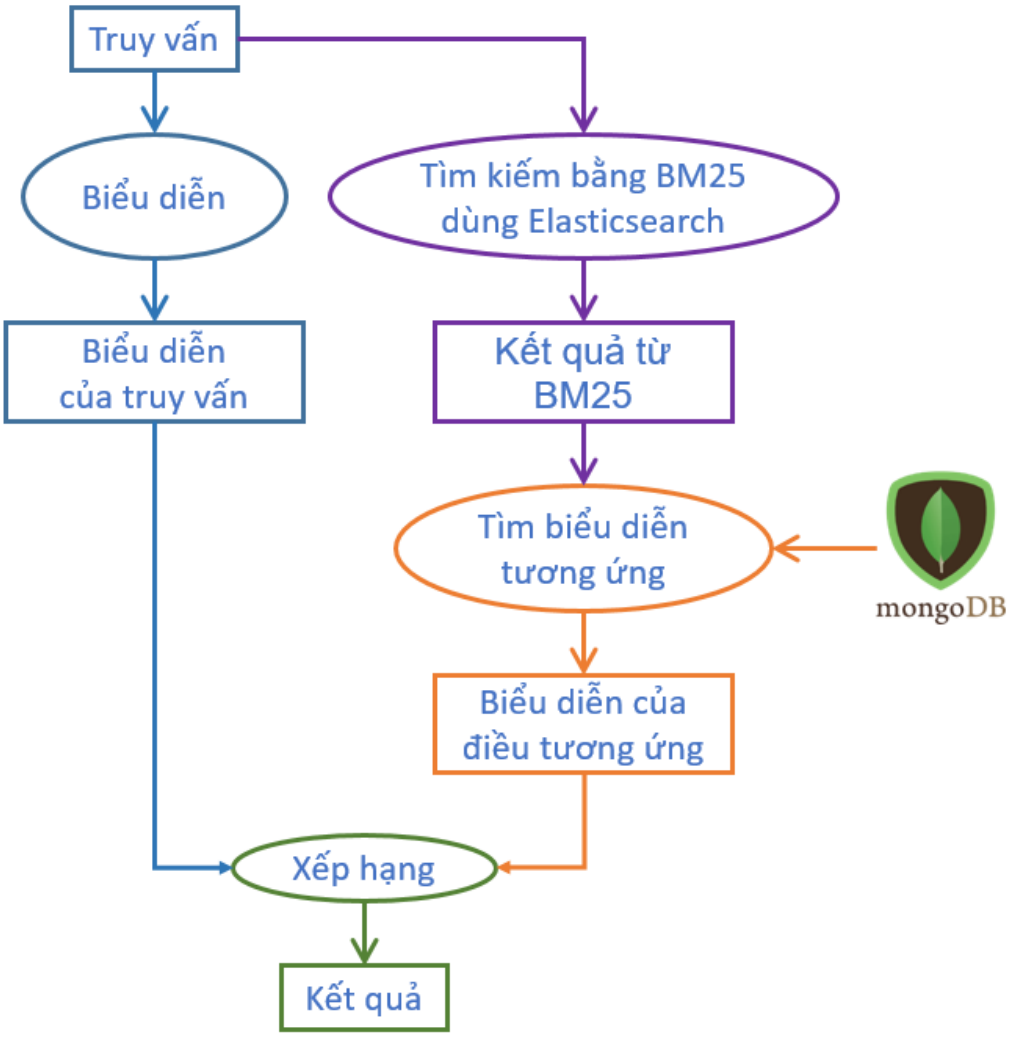
3.2.3.2. *Tìm kiếm*

Các điều luật trong tập dữ liệu văn bản pháp quy sẽ được tính toán biểu diễn sử dụng mô-đun Biểu diễn điều luật đã được huấn luyện. Sau đó, véc-tơ biểu diễn sẽ được lưu kèm với số hiệu văn bản và tên điều để có thể sử dụng được dễ dàng.



Hình 3.2. Lưu trữ biểu diễn của các điều luật.

Khi nhận một truy vấn, hệ thống sẽ biểu diễn nó thành véc-tơ bằng mô-đun biểu diễn truy vấn. Sau đó, hệ thống thực hiện tìm kiếm bằng phương pháp BM25 trong Elasticsearch để thu được một tập kết quả. Hệ thống sẽ tìm kiếm biểu diễn đã được lưu sẵn của các điều tương ứng trong tập kết quả đó. Tiếp theo, độ tương đồng giữa biểu diễn của câu truy vấn và từng điều sẽ được tính rồi dùng để xếp hạng và cho ra kết quả cuối cùng.



Hình 3.3. Quá trình tìm kiếm khi nhận một truy vấn.

3.3. Phương pháp đánh giá

3.3.1. Recall

Đánh giá thứ nhất sử dụng độ đo recall@k [16], ở đây k được chọn bằng 20.

$$recall@k = \frac{1}{|Q|}\sum_{q \in Q} \frac{|predict(q) \cap relevance(q)|}{|relevance(q)|}$$

Trong đó:

- k là số kết quả trả về, ở đây k được chọn bằng 20
- $predict(q)$ là các điều cho ra bởi hệ thống
- $relevance(q)$ là các điều thật sự liên quan tới truy vấn

3.3.2. NDCG

Đánh giá thứ hai sử dụng độ đo NDCG@k [29], ở đây k được chọn bằng 20.

Coi các điều liên quan có độ liên quan bằng 1, các điều không liên quan có độ liên quan bằng 0. Ta có

$$DCG = \sum_{i=1}^{k} \frac{relevance_i}{log_2(i+1)}$$

Trong đó:

- k là số kết quả trả về, ở đây k được chọn bằng 20
- $relevance_i$ là độ liên quan của điều thứ i trong tập kết quả trả về

Gọi $iDCG$ là DCG trong trường hợp lý tưởng của một truy vấn, tức là các điều liên quan được xếp trên cùng.

$$iDCG = \sum_{i=1}^{|groundtruth|} \frac{1}{log_2(i+1)}$$

Trong đó $|groundtruth|$ là số điều liên quan của câu truy vấn đó.

$NDCG$ trung bình cho tập dữ liệu kiểm tra tính bằng:

$$NDCG@k = \frac{DCG(q)}{iDCG(q)}$$

3.4. Kết quả thực nghiệm

Tập câu truy vấn được chia thành 2 phần: 90% dùng để huấn luyện mô hình mạng nơ-ron và 10% dùng để đánh giá các phương pháp.

Đầu tiên là thử nghiệm so sánh hiệu quả của phương pháp sử dụng mạng nơ-ron nhân chập kết hợp với cơ chế Attention, sau đây sẽ được gọi là NATR (Neural Attentive Text Representation), với phương pháp dùng TF-IDF và BM25. Tiếp theo sẽ là thực nghiệm so sánh hiệu quả khi thay đổi các tham số trong NATR. Cuối cùng là thực nghiệm kết hợp điểm của phương pháp BM25 và NATR khi xếp hạng.

Mô hình mạng nơ-ron trong phương pháp NART trong các thực nghiệm đều được huấn luyện trên Google Colab với GPU Tesla P100-PCIE-16GB.

3.4.1. Thực nghiệm so sánh hiệu quả của các phương pháp

Trong thực nghiệm này, hệ thống NATR được huấn luyện với 1 điều positive đi kèm 15 điều negative từ Elasticsearch và 15 điều negative. Khi tìm kiếm, hệ thống NATR ấy 1000 kết quả trả về từ Elasticsearch để xếp hạng lại. Kết quả được cho bởi bảng sau:

Bảng 3.4. So sánh hiệu quả các phương pháp.

Phương pháp	**Recall@20**	**NDCG@20**
TF-IDF	0.4716	0.3537
BM25	0.5593	0.3755
NATR	**0.7261**	**0.4642**

Hình 3.4. So sánh hiệu quả các phương pháp.

Thực nghiệm đã cho thấy NATR cho hiệu quả tốt hơn hẳn TF-IDF và BM25 cả về Recall@20 và NDCG@20. Điều này cho thấy mô hình đề xuất có khả năng biểu diễn truy vấn và điều luật tốt hơn.

3.4.2. Thực nghiệm hiệu quả khi thay đổi các tham số

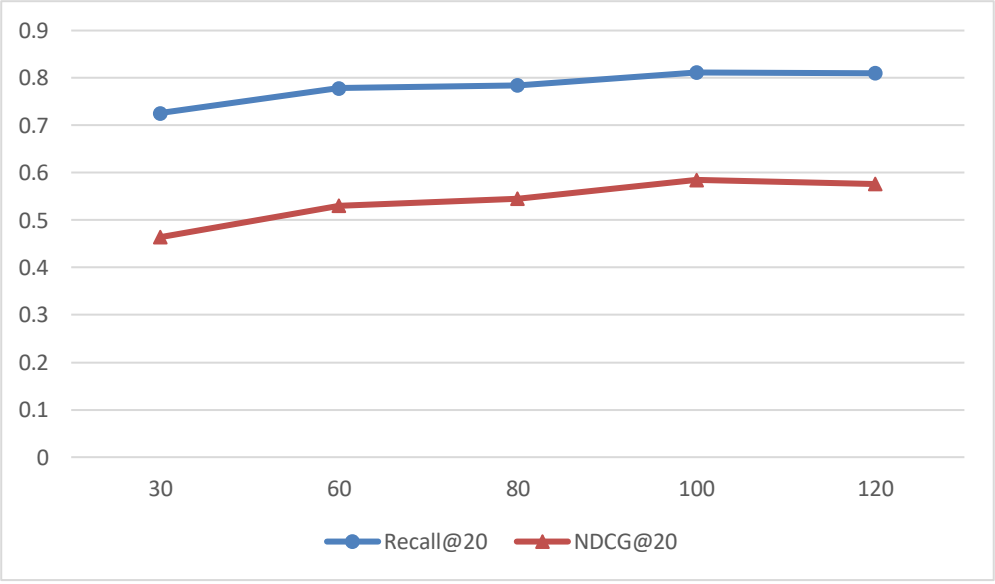
Trong thực nghiệm này, các tham số được thay đổi để đánh giá tác động lên hiệu quả của hệ thống NATR. Các tham số được thực nghiệm bao gồm:

- K: Số điều negative trong dữ liệu huấn luyện, một nửa lấy từ kết quả trả về của Elasticsearch, một nửa được lấy ngẫu nhiên
- N: Số kết quả trả về từ Elasticsearch dùng để xếp hạng lại khi tìm kiếm.

Kết quả thay đổi tham số K khi huấn luyện và cố định tham số N = 1000 khi tìm kiếm được cho bởi bảng sau:

Bảng 3.5. Kết quả khi thay đổi tham số K

K	Recall@20	NDCG@20	Thời gian huấn luyện
30	0.7261	0.4642	3 giờ 24 phút
60	0.7785	0.5305	6 giờ 20 phút
80	0.7842	0.5452	8 giờ 49 phút
100	**0.8115**	**0.5849**	10 giờ 50 phút
120	0.8103	0.5766	13 giờ 39 phút



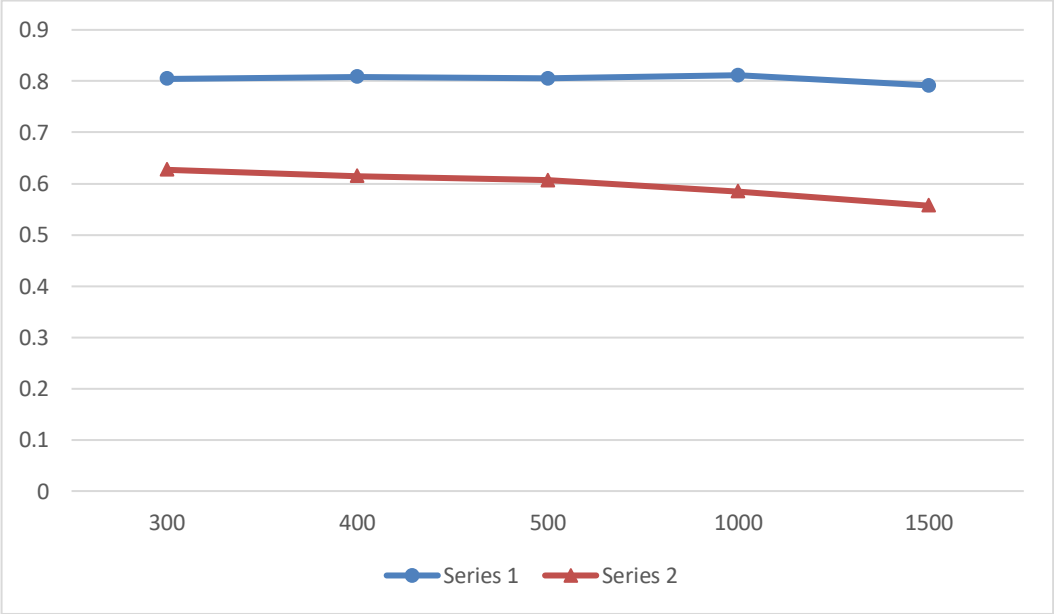
Hình 3.5. Kết quả khi thay đổi tham số K.

Thực nghiệm này cho thấy khi tăng số điều negative trong một ví dụ huấn luyện đến ngưỡng K = 100, kết quả có xu hướng tốt lên. Khi tăng K lên 120 phải giảm batch size để có thể huấn luyện trên GPU nên kết quả bị ảnh hưởng và kém đi.

Kết quả khi cố định K = 100 khi huấn luyện và thay đổi tham số N khi tìm kiếm được cho bởi bảng sau:

Bảng 3.6. Kết quả khi thay đổi tham số N

N	Recall@20	NDCG@20
300	0.8049	**0.6269**
400	0.8084	0.6147
500	0.8051	0.6063
1000	**0.8115**	0.5849
1500	0.7917	0.5569



Hình 3.6. Kết quả khi thay đổi tham số N.

Thực nghiệm này cho thấy tăng số kết quả trả về từ Elasticsearch dùng để xếp hạng lại khi tìm kiếm Recall@20 thay đổi không nhiều, trong khi đó NDCG@20 có xu hướng giảm. Nguyên nhân là do khi dùng càng nhiều kết quả trả về từ Elasticsearch dùng để xếp hạng lại thì kết quả cuối cùng càng có khả năng bị nhiễu, nên các điều liên quan có thể bị xếp hạng thấp hơn làm NDCG@20 giảm.

3.4.3. Thực nghiệm kết hợp điểm của BM25 và NATR

Trong thực nghiệm này, điểm của phương pháp BM25 và NATR sẽ được kết hợp với nhau để xếp hạng lại các điều trả về từ Elasticsearch. Các điều sẽ được sắp xếp theo thứ tự điểm của phương pháp BM25 từ cao đến thấp. Điều xếp thứ nhất được tính N điểm, xếp thứ 2 được tính N – 1 điểm, ..., điều xếp cuối cùng được 1 điểm. Tương tự với phương pháp NATR. Điểm kết hợp sẽ được tính theo công thức:

$$score = w \times BM25\_score + (1 - w) \times NATR\_score$$

Trong đó:

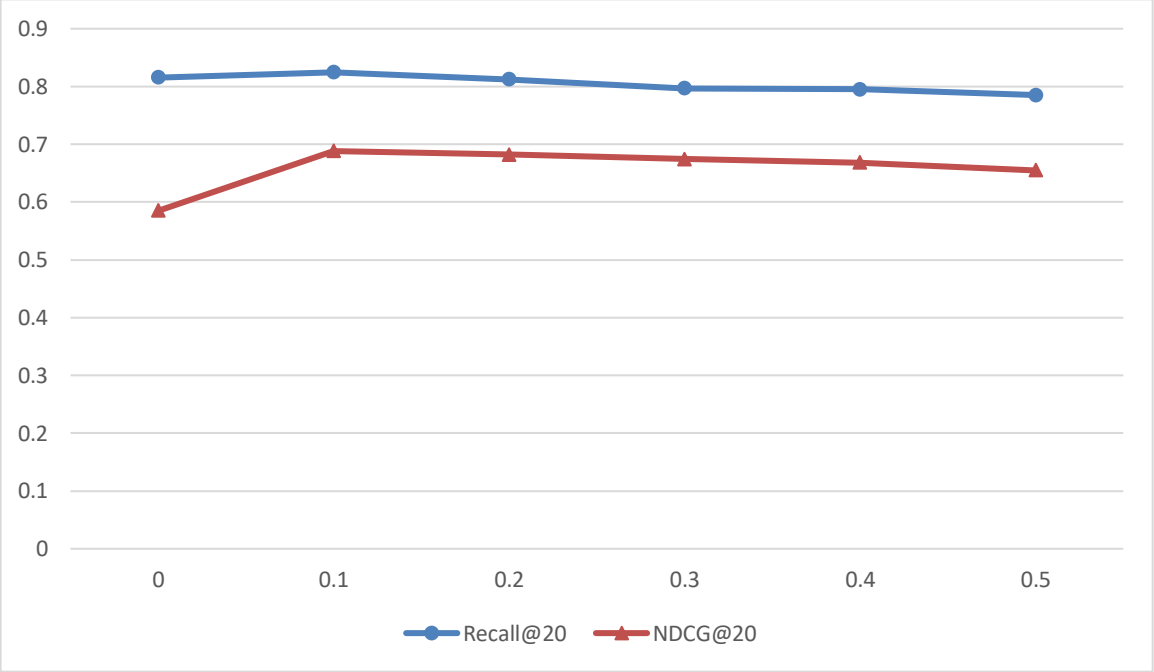
- $score$ là điểm kết hợp
- $BM25\_score$ là điểm của phương pháp BM25 trả về từ Elasicsearch
- $NATR\_score$ là điểm của phương pháp NATR
- w là trọng số điểm của phương pháp BM25 trả về từ Elasicsearch

Kết quả thực nghiệm khi cố định K = 100, N = 1000 và thay đổi tham số w được cho bởi bảng sau:

Bảng 3.7. Kết quả khi thay đổi tham số w.

w	**Recall@20**	**NDCG@20**
0.0	0.8155	0.5849
0.1	**0.8245**	**0.6882**
0.2	0.8122	0.6821
0.3	0.7970	0.6741

0.4	0.7954	0.6682
0.5	0.7852	0.6547



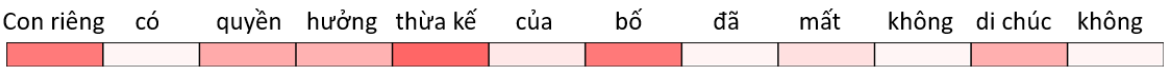
Hình 3.7. Kết quả khi thay đổi tham số w.

Thực nghiệm này cho thấy $w = 0.1$ là lựa chọn tốt nhất để kết hợp điểm của BM25 và NART. Nó cho kết quả tốt hơn chỉ dùng điểm của NATR. Khi tiếp tục tăng w lên thì kết quả có xu hướng xấu đi.

3.4.4. Hình ảnh hóa trọng số Attention

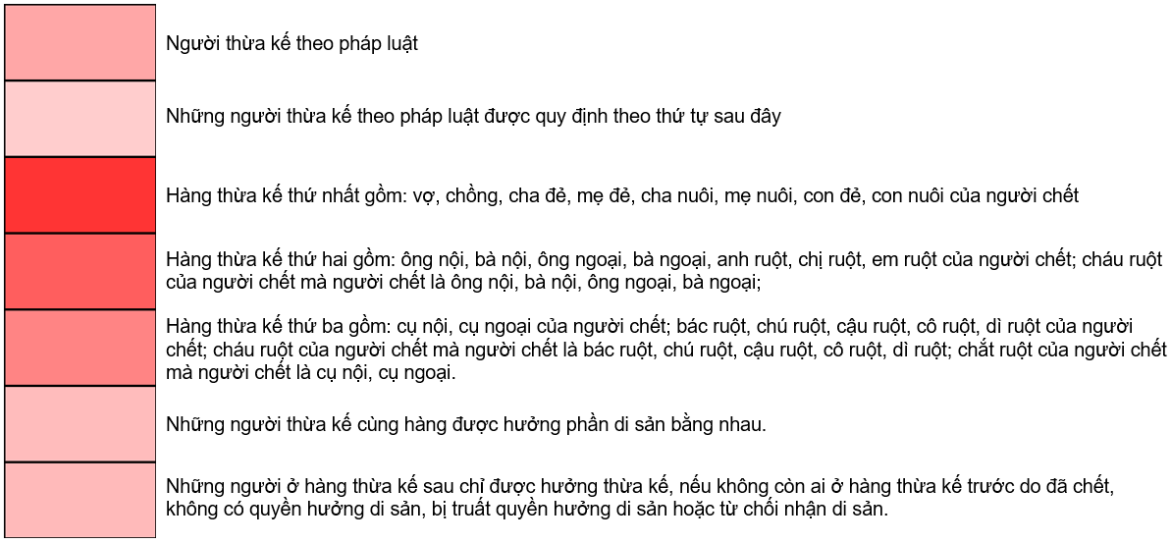
Dưới đây là hình ảnh mô tả trọng số Attention của phương pháp NATR khi biểu diễn câu truy vấn và điều luật. Màu sắc đậm hơn thể hiện trọng số cao hơn.

Với câu truy vấn, mỗi trọng số tương ứng với một từ. Có thể thấy mô hình đánh trọng số cao vào các từ quan trọng như "*con riêng*", "*thừa kế*", "*bố*".



Hình 3.8. Hình ảnh hóa trọng số Attention của truy vấn.

Với điều luật, mỗi trọng số được gán cho một câu. Có thể thấy mô hình coi câu thứ ba là quan trọng nhất và đánh trọng số cao nhất cho nó.



Hình 3.9. Hình ảnh hóa trọng số Attention của điều luật

3.5. Kết luận chương

Chương này đã trình bày quá trình xây dựng bộ dữ liệu văn bản và câu hỏi pháp quy. Tiếp theo đó là trình bày quá trình áp dụng các phương pháp biểu diễn văn bản để xây dựng hệ thống tìm kiếm văn bản pháp quy và thực so sánh hiệu quả dựa trên bộ dữ liệu đã xây dựng.

Quá trình thực nghiệm đã cho thấy phương pháp biểu diễn văn bản sử dụng mạng nơ-ron nhân chập kết hợp với cơ chế Attention được đề xuất đã cho kết quả tốt hơn các phương pháp hiện có như TF-IDF, BM25.

KẾT LUẬN

Luận văn tập trung nghiên cứu các phương pháp biểu diễn văn bản phục vụ truy xuất, tìm kiếm thông tin và đã đạt được một số kết quả sau:

- Trình bày các phương pháp biểu diễn văn bản
- Đề xuất phương pháp biểu diễn văn bản sử dụng mạng nơ-ron nhân chập và cơ chế Attention.
- Xây dựng bộ dữ liệu văn bản và câu hỏi pháp quy, áp dụng một số phương pháp biểu diễn văn bản để xây dựng hệ thông tìm kiếm thông tin, thử nghiệm và đánh giá các phương pháp đó.
- Kết quả của luận văn đã được chấp nhận công bố tại hội nghị COLING 2020.

Trong tương lai, luận văn có thể tiếp tục được nghiên cứu theo hướng ứng dụng xây dựng hệ thống truy xuất văn bản trong một chủ đề xác định.

TÀI LIỆU THAM KHẢO

[1] O.Abdel-Hamid, A.Mohamed, H. Jiang, Deng, L, G. Penn, and D. Yu (2014), "Convolutional Neural Networks for Speech Recognition.", *The IEEE/Audio ON the ACM Transactions, speech, and language processing, vol 22, no 10*, trang 1533-1545.

[2] Saad Albawi, Tareq Abed Mohammed (2017), "Understanding of a Convolutional Neural Network", *International Conference on Engineering and Technology (ICET)*.

[3] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio (2015), "Neural Machine Translation by Jointly Learning to Align and Translate"

[4] István Bíró (2009), "Document Classification with Latent Dirichlet Allocation".

[5] Blei, D.M., Ng, A.Y. and Jornal, M.I. (2003), "Latent Dirichlet Allocation", Journal of Machine Learning Research 3, trang 993-1022.

[6] Stefano Ceri et al (2013), *Web Information Retrieval*, trang 5.

[7] Jianpeng Cheng, Li Dong, Mirella Lapata (2016), "Long Short-Term Memory-Networks for Machine Reading".

[8] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, Li Deng (2014), "Modeling Interestingness with Deep Neural Networks".

[9] Alex Graves, Greg Wayne, Ivo Danihelka (2014), "Neural Turing Machines".

[10] B. Hu, Z. Lu, H. Li , and Q. Chen (2014), "Convolutional neural network architectures for matching natural language sentences.", *Advances in neural information processing systems*, trang 2042 -2050

[11] Rie Johnson, Tong Zhang (2015), "Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding".

[12] Rie Johnson, Tong Zhang (2015), "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks".

[13] Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom (2014), "A Convolutional Neural Network for Modelling Sentences".

[14] Yoon Kim (2014), "Convolutional Neural Networks for Sentence Classification" *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

[15] Bing Liu (2011), *Web Data Mining, 2nd Edition*, trang 212-215.

[16] Bing Liu (2011). *Web Data Mining, 2nd Edition*, trang 223.

[17] Minh-Thang Luong, Hieu Pham, Christopher D. Manning (2015), "Effective Approaches to Attention-based Neural Machine Translation"

[18] Christopher D. Manning et al (2009), "An Introduction to Information Retrieval".

[19] Susan Nevelow Mart et al (2013), "A Study of Attorneys' Legal Research Practices and Opinions of New Associates' Research Skills", trang 8.

[20] André F. T. Martins, Ramón Fernandez Astudillo (2016), "From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification".

[21] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean (2013), "Efficient Estimation of Word Representations in Vector Space".

[22] A. Mukherjee and B. Liu (2012), "Aspect extraction through semi-supervised modeling", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, trang 339 - 348.

[23] Keiron O'Shea and Ryan Nash (2015), "An Introduction to Convolutional Neural Networks".

[24] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014), "GloVe: Global Vectors for Word Representation", Computer Science Department, Stanford University, Stanford, CA 94305.

[25] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, Gregoire Mesnil (2014), "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval", *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management ACM*, trang 101-110.

[26] Duyu Tang, Bing Qin, Ting Liu (2015), "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification".

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2017), "Attention Is All You Need"

[28] Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, Hongwei Hao (2015). "Semantic Clustering and Convolutional Neural Network for Short Text Categorization".

[29] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, Wei Chen (2013), "A Theoretical Analysis of NDCG Type Ranking Measures"

[30] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio (2015), "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention".

[31] Xiang Zhang, Junbo Zhao, Yann LeCun (2016), "Character-level Convolutional Networks for Text Classification".

[32] https://hdpl.moj.gov.vn/Pages/home.aspx

[33] http://hethongphapluat.com/hoi-dap-phap-luat.html

[34] https://hoidapphapluat.net/

[35] https://github.com/undertheseanlp/underthesea

[36] https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables