ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN VIẾT HẠNH

NGHIÊN CỨU TÓM TẮT VĂN BẢN TỰ ĐỘNG VÀ ỨNG DỤNG

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

HÀ NỘI – 2018

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

NGUYỄN VIẾT HẠNH

NGHIÊN CỨU TÓM TẮT VĂN BẢN TỰ ĐỘNG VÀ ỨNG DỤNG

Ngành: Công nghệ thông tin
Chuyên ngành: Kỹ thuật phần mềm
Mã số: 60480103

LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN VĂN VINH

HÀ NỘI - 2018

LỜI CAM ĐOAN

Tôi là Nguyễn Viết Hạnh, học viên lớp Kỹ Thuật Phần Mềm K22 xin cam đoan báo cáo luận văn này được viết bởi tôi dưới sự hướng dẫn của thầy giáo, tiến sỹ Nguyễn Văn Vinh. Tất cả các kết quả đạt được trong luận văn này là quá trình tìm hiểu, nghiên cứu của riêng tôi. Trong toàn bộ nội dung của luận văn, những điều được trình bày là kết quả của cá nhân tôi hoặc là được tổng hợp từ nhiều nguồn tài liệu khác. Các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày ….. tháng …… năm 2018

Người cam đoan

Nguyễn Viết Hạnh

LỜI CẢM ƠN

Lời đầu tiên, tôi xin bày tỏ sự cảm ơn chân thành đối với Thầy giáo TS. Nguyễn Văn Vinh – giáo viên hướng dẫn trực tiếp của tôi. Thầy Vinh đã cho tôi những gợi ý và chỉ dẫn quý báu trong quá trình nghiên cứu và hoàn thiện luận văn thạc sĩ.

Tôi cũng xin gửi lời cảm ơn tới các thầy cô trong khoa Công nghệ thông tin, trường Đại học Công Nghệ, Đại học Quốc gia Hà Nội đã hướng dẫn, chỉ bảo và tạo điều kiện cho chúng tôi học tập và nghiên cứu tại trường trong suốt thời gian qua.

Tôi cũng xin được cảm ơn gia đình, những người thân, các đồng nghiệp và bạn bè tôi đã quan tâm, động viên, giới thiệu các tài liệu hữu ích trong thời gian học tập và nghiên cứu luận văn tốt nghiệp.

Mặc dù đã cố gắng hoàn thành luận văn nhưng chắc chắn sẽ không tránh khỏi những sai sót, tôi kính mong nhận được sự thông cảm và chỉ bảo của các thầy cô và các bạn.

Tôi xin chân thành cảm ơn!

MỤC LỤC

BẢNG CÁC TỪ VIẾT TẮT

STT	Từ viết tắt và thuật ngữ	Từ/Cụm từ đầy đủ	Giải thích
1	ANN	Artificial Neural Network	Mạng nơ ron nhân tạo được nghiên cứu ra từ hệ thống thần kinh của con người, giống như bộ não để xử lý thông tin.
2	LSTM	Long-Short Term Memory	Kiến trúc mạng học sâu cải tiến của RNN, giải quyết hạn chế của mạng RNN với các bài toán cần xử lý dữ liệu theo thời gian đòi hỏi trạng thái nhớ trung gian.
3	NLTK	Natural Language Toolkit	Một công cụ xử lý ngôn ngữ tự nhiên mạnh trên môi trường Python
4	RNN	Recurrent Neural Network	Mạng nơ ron hồi tiếp sử dụng để xử lý thông tin có tính chuỗi tuần tự
5	ROUGE	Recall Oriented Understudy for Gist Evaluation	Phương pháp đánh giá độ chính xác của văn bản tóm tắt
6	TF-IDF	Term Frequency -Inverse Document Frequency	Một phương pháp đánh giá độ quan trọng của các từ trong các văn bản.

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

Với sự phát triển mạnh mẽ của công nghệ thông tin và mạng máy tính, lượng tài liệu văn bản khổng lồ được tạo ra với nhiều mục đích sử dụng khác nhau khiến cho việc đọc hiểu và trích lược các thông tin cần thiết trong khối tri thức đồ sộ này tốn rất nhiều thời gian và chi phí (đặc biệt là chi phí cho hạ tầng và truyền dẫn thông tin đáp ứng yêu cầu cho một số lượng ngày càng nhiều các thiết bị cầm tay). Để tăng hiệu quả cũng như dễ dàng hơn trong việc tiếp nhận thông tin của người dùng, nhiều nghiên cứu về khai phá dữ liệu và xử lý ngôn ngữ tự nhiên đã được thực hiện. Một trong những nghiên cứu quan trọng đóng vai trò then chốt đó tóm tắt văn bản tự động.

Bài toán tóm tắt văn bản tiếng Việt cũng được nghiên cứu và áp dụng nhiều kỹ thuật như đối với tiếng Anh; tuy nhiên, tóm tắt văn bản nói riêng và xử lý ngôn ngữ tự nhiên nói chung áp dụng cho tiếng Việt gặp nhiều thách thức hơn. Sở dĩ là vì tiếng Việt với đặc trưng là tiếng đơn âm và có thanh điệu nên việc tách từ, tách các thành phần ngữ nghĩa trong câu tiếng Việt đòi hỏi xử lý phức tạp hơn so với xử lý câu tiếng Anh, thêm vào đó, không có nhiều kho dữ liệu tiếng Việt được chuẩn hóa và công bố.

Trong luận văn này, chúng tôi tập trung nghiên cứu tóm tắt văn bản tự động theo hướng tóm lược, các mô hình kiến trúc mạng học sâu và các kỹ thuật xử lý những thách thức trong tóm tắt văn bản. Bố cục của luận văn được tổ chức thành như sau:

Chương 1: Khái quát bài toán tóm tắt văn bản giới thiệu tổng quan về bài toán tóm tắt văn bản tự động, định nghĩa và các hướng tiếp cận.

Chương 2: Một số nghiên cứu về tóm tắt văn bản giới thiệu một số phương pháp, kỹ thuật đã được nghiên cứu và áp dụng cho bài toán tóm tắt văn bản tự động.

Chương 3: Mạng nơ ron nhân tạo giới thiệu cơ sở lý thuyết và cách hoạt động của các mô hình kiến trúc từ mạng ANN tới RNN và LSTM.

Chương 4: Xây dựng hệ thống tóm tắt văn bản theo hướng tóm lược trình bày mô hình bài toán tóm tắt văn bản tự động, quy trình thực hiện giải quyết bài toán trong luận văn, các xây dựng mô hình học sâu dựa trên kiến trúc mạng LSTM áp dụng cho bài toán tóm tắt văn bản.

Chương 5: Thử nghiệm và đánh giá trình bày quá trình thử nghiệm mô hình đã xây dựng cho tập dữ liệu tiếng Anh và tiếng Việt và thực hiện đánh giá độ chính xác của mô hình bằng phương pháp ROUGE.

Kết luận: phần này tổng kết các đóng góp và kết quả đạt được trong quá trình nghiên cứu và thực hiện luận văn, cũng như hướng phát triển trong tương lai để hoàn thiện hơn kết quả nghiên cứu.

CHƯƠNG 1: KHÁI QUÁT BÀI TOÁN TÓM TẮT VĂN BẢN

Cùng với sự tăng trưởng mạnh mẽ của mạng Internet, con người ngày càng bị quá tải bởi khối lượng lớn các thông tin và tài liệu trực tuyến. Điều này đã thúc đẩy rất nhiều nghiên cứu về tóm tắt văn bản tự động. Theo Radev và cộng sự [25] một tóm tắt được định nghĩa như là một văn bản được tạo từ một hoặc nhiều văn bản, truyền đạt các thông tin quan trọng từ các văn bản gốc, văn bản tóm tắt không dài hơn hơn 50% độ dài văn bản gốc và thông thường bản tóm tắt có độ dài khá ngắn, ngắn hơn nhiều so với 50% độ dài văn bản gốc.

1.1. Bài toán tóm tắt văn bản tự động

Tóm tắt văn bản tự động là tác vụ để tạo ra một tóm tắt chính xác và hợp ngữ pháp trong khi vẫn giữ được các thông tin chính và ý nghĩa của văn bản gốc. Trong các năm gần đây, có rất nhiều hướng tiếp cận đã được nghiên cứu cho tóm tắt văn bản tự động và đã được áp dụng rộng rãi trong nhiều lĩnh vực. Ví dụ, máy tìm kiếm sinh ra các trích đoạn như là các bản xem trước của tài liệu [2], các website tin tức sinh ra các đoạn mô tả ngắn gọn cho bài viết (thường là tiêu đề của bài viết) [20].

Mục tiêu của tóm tắt văn bản là tạo ra bản tóm tắt giống như cách con người tóm tắt, đây là bài toán đầy thách thức, bởi vì khi con người thực hiện tóm tắt một văn bản, chúng ta thường đọc toàn bộ nội dung rồi dựa trên sự hiểu biết và cảm thụ của mình để viết lại một đoạn tóm tắt nhằm làm nổi bật các ý chính của văn bản gốc. Nhưng vì máy tính khó có thể có được tri thức và khả năng ngôn ngữ như của con người, nên việc thực hiện tóm tắt văn bản tự động là một công việc phức tạp.

1.2. Các hướng tiếp cận tóm tắt văn bản.

Nhìn chung, có hai hướng tiếp cận cho tóm tắt văn bản tự động là trích chọn (extraction) và tóm lược (abstraction). Theo [32], tóm tắt văn bản có thể được phân loại dựa trên đầu vào (đơn hay đa văn bản), mục đích (tổng quát, theo lĩnh vực cụ thể, hay dựa trên truy vấn) và loại đầu ra (trích chọn hay tóm lược).

Phương pháp tóm tắt trích chọn thực hiện đánh giá các phần quan trọng của văn bản và đưa chúng một cách nguyên bản vào bản tóm tắt, do đó, phương pháp này chỉ phụ thuộc vào việc trích chọn các câu từ văn bản gốc dựa trên việc xếp hạng mức độ liên quan của các cụm từ để chỉ chọn những cụm từ liên quan nhất tới nội dung của tài liệu gốc. Trong khi đó, phương pháp tóm tắt tóm lược nhằm tạo ra văn

bản tóm tắt mới có thể không gồm các từ hay các cụm từ trong văn bản gốc. Nó cố gắng hiểu và đánh giá văn bản sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên tiên tiến để tạo ra một văn bản ngắn hơn, truyền đạt được những thông tin quan trọng nhất từ văn bản gốc. Mặc dù các tóm tắt được con người thực hiện thường không giống như trích chọn, song hầu hết các nghiên cứu về tóm tắt văn bản hiện tại vẫn tập trung vào tóm tắt bằng phương pháp trích chọn vì về cơ bản các tóm tắt sinh bởi phương pháp trích chọn cho kết quả tốt hơn so với tóm tắt bằng phương pháp tóm lược. Điều này là bởi vì phương pháp tóm tắt bằng tóm lược phải đối mặt với các vấn đề như thể hện ngữ nghĩa, suy luận và sinh ngôn ngữ tự nhiên, các vấn đề này phức tạp hơn nhiều lần so với việc trích chọn câu. Hướng tiếp cận tóm tắt bằng tóm lược khó hơn so với tóm tắt bằng trích chọn, song phương pháp này được kỳ vọng có thể tạo ra được các văn bản tóm tắt giống như cách con người thực hiện.

CHƯƠNG 2: MỘT SỐ NGHIÊN CỨU VỀ TÓM TẮT VĂN BẢN

2.1. Tóm tắt văn bản theo hướng trích chọn.

Như đã đề cập trong chương 1, các kỹ thuật tóm tắt bằng trích chọn sinh ra các đoạn tóm tắt bằng cách chọn một tập các câu trong văn bản gốc. Các đoạn tóm tắt này chứa các câu quan trọng nhất của đầu vào. Đầu vào có thể là đơn văn bản hoặc đa văn bản. Trong khuôn khổ của luận văn này, đầu vào của bài toán tóm tắt văn bản là đơn văn bản.

Các hệ thống tóm tắt văn bản theo hướng trích chọn thường gồm các tác vụ: xây dựng một đại diện trung gian (intermediate representation) của văn bản đầu vào thể hiện các đặc điểm chính của văn bản; tính điểm (xếp hạng) các câu dựa trên đại diện trung gian đã xây dựng; chọn các câu đưa vào tóm tắt [23].

Mỗi hệ thống tóm tắt văn bản tạo ra một số đại diện trung gian của văn bản mà nó sẽ thực hiện tóm tắt và tìm các nội dung nổi bật dựa trên đại diện trung gian này. Có hai hướng tiếp cận dựa trên dại diện trung gian là chủ đề đại diện (topic representation) và các đặc trưng đại diện (indicator representation). Các phương pháp dựa trên chủ đề đại diện biến đổi văn bản đầu vào thành một đại diện trung gian và tìm kiếm các chủ đề được thảo luận trong văn bản. Kỹ thuật tóm tắt dựa trên chủ đề đại diện tiêu biểu là phương pháp tiếp cận dựa trên tần xuất (frequency). Phương pháp dựa trên các đặc trưng đại diện thực hiện mô tả các câu trong văn bản như một danh sách các đặc trưng quan trọng chẳng hạn như độ dài câu, vị trí của câu trong tài liệu hay câu có chứa những cụm từ nhất định.

Khi các đại diện trung gian đã được tạo ra, một điểm số thể hiện mức độ quan trọng sẽ được gán cho mỗi câu. Đối với phương pháp dựa trên chủ đề đại diện, điểm số của một câu thể hiện mức độ giải thích của câu đối với một vài chủ đề quan trọng nhất của văn bản. Trong hầu hết các phương pháp dựa trên đặc trưng đại diện, điểm số được tính bằng tổng hợp các dấu hiệu từ các đặc trưng khác nhau. Các kỹ thuật học máy thường được sử dụng để tìm trọng số cho các đặc trưng.

Cuối cùng hệ thống tóm tắt sẽ lựa chọn các câu quan trọng nhất để tạo ra bản tóm tắt. Có thể áp dụng các thuật toán tham lam để chọn các câu quan trọng nhất từ văn bản gốc, hoặc biến việc lựa chọn câu thành một bài toán tối ưu trong đó xem xét ràng buộc tối đa hóa tầm quan trọng tổng thể và sự gắn kết ngữ nghĩa trong khi tối

thiểu hóa sự dư thừa. Có nhiều yếu tố khác cần được cân nhắc khi lựa chọn các câu quan trọng, ví dụ ngữ cảnh của bản tóm tắt hay loại tài liệu cần tóm tắt (bài báo tin tức, email, báo cáo khoa học). Các tiêu chí này có thể trở thành các trọng số bổ sung cho việc lựa chọn các câu quan trọng đưa vào bản tóm tắt.

2.1.1. Phương pháp chủ đề đại diện dựa trên tần xuất

2.1.1.1. Word probability

Xác suất của từ (word probability) là dạng đơn giản nhất sử dụng tần xuất trên văn bản đầu vào như là một chỉ số quan trọng. Phương pháp này khá phụ thuộc vào độ dài của văn bản đầu vào, ví dụ, một từ xuất hiện ba lần trong một văn bản 10 từ có thể là từ quan trọng song có thể nó là một từ bình thường trong văn bản 1000 từ.

Xác suất của một từ w: p(w) được tính dựa trên số lần xuất hiện của từ w, n(w), trong toàn bộ các từ thuộc văn bản đầu vào N.

$$P(w) = n(w)/N \tag{2.1}$$

Hệ thống SumBasic [18] được phát triển dựa trên ý tưởng sử dụng xác suất của từ để tính toán câu quan trọng. Với mỗi câu Sj trong văn bản đầu vào, nó gán một trọng số bằng xác suất trung bình của các từ chứa nội dung trong câu (một danh sách các từ không mang thông tin – stop words – sẽ bị loại khỏi quá trình đánh trọng số):

$$\text{Weight(Sj)} = \frac{\sum_{wi \in Sj} p(wi)}{|\{wi \,|wi \in Sj\}|} \tag{2.2}$$

Tiếp theo nó sẽ chọn các câu có điểm số tốt nhất gồm những từ có xác suất cao nhất. Bước này đảm bảo rằng các từ có xác suất cao nhất đại diện cho chủ đề của văn bản đầu vào sẽ được đưa vào bản tóm tắt. Sau khi chọn một câu đưa vào tóm tắt, xác suất của mỗi từ trong câu được hiệu chỉnh:

$$p_{new}\ (w_i) = p_{old}\ (w_i)^2 \tag{2.3}$$

Việc hiệu chỉnh này thể hiện rằng xác suất một từ xuất hiện hai lần trong bản tóm tắt là thấp hơn so với xác suất từ xuất hiện chỉ một lần. Quá trình lặp lại cho đến khi đạt được độ dài cần thiết của văn bản tóm tắt.

2.1.1.2. Phương pháp TF-IDF

Phương pháp dựa trên xác suất của từ phụ thuộc vào danh sách stop word để loại bỏ các từ không quan trọng khỏi bản tóm tắt. Việc quyết định từ nào sẽ đưa vào danh sách stop word sẽ ảnh hưởng tới hiệu năng của phương pháp word probability. Phương pháp TF-IDF (Term Frequency - Inverse Document Frequency) đã được nghiên cứu phát triển để giải quyết hạn chế của phương pháp xác suất từ. Phương pháp này sẽ đánh giá độ quan trọng của một từ bằng cách đánh trọng số cho từ. Các từ quan trọng trong văn bản sẽ được đánh trọng số cao, còn các từ phổ biến trong rất nhiều tài liệu (common words) sẽ được đánh trọng số thấp để loại bỏ khỏi danh sách đánh giá lựa chọn đưa vào văn bản tóm tắt. Trọng số của mỗi từ trong tài liệu d được tính như sau:

$$\text{Weight(w)} = f_d(w) * log \frac{D}{f_D(w)} \quad (2.4)$$

Trong đó, $f_d(w)$ là term frequency của từ w trong tài liệu d, $f_D(w)$ là số tài liệu chứa từ w và D là tổng số tài liệu. Như vậy, các từ xuất hiện trong hầu hết các tài liệu sẽ có giá trị IDF gần bằng 0. Trọng số TF*IDF của từ là một chỉ số tốt để đánh giá mức độ quan trọng.

2.1.2. Phương pháp đặc trưng đại diện

Phương pháp đặc trưng đại diện nhằm mô hình các đại diện của văn bản dựa trên một tập các đặc trưng và sử dụng chúng để xếp hạng các câu của văn bản đầu vào. Các phương pháp dựa trên đồ thị và kỹ thuật học máy thường được sử dụng để quyết định mức độ quan trọng của các câu sẽ đưa vào văn bản tóm tắt.

2.1.2.1. Phương pháp đồ thị cho tóm tắt văn bản

Phương pháp dựa trên đồ thị thể hiện văn bản như là một đồ thị liên thông. Các câu tạo thành các đỉnh của đồ thị và các cạnh giữa các câu thể hiện sự liên quan giữa hai câu với nhau. Một kỹ thuật thường được sử dụng để nối hai đỉnh đó là đo lường sự tương đồng giữa hai câu và nếu nó lớn hơn một ngưỡng nhất định thì chúng liên thông nhau. Đồ thị này thể hiện kết quả ở hai phần: thứ nhất, một phần đồ thị con được tạo bảo các chủ đề rời rạc trong văn bản; thứ hai, các câu được kết nối tới nhiều câu khác trong đồ thị là các câu quan trọng có thể lựa chọn đưa vào văn bản tóm tắt. Một phương pháp dựa trên đồ thị tiêu biểu đó là TextRank [24] .

Phương pháp dựa trên đồ thị không cần các kỹ thuật xử lý ngôn ngữ tự nhiên đặc thù cho từng ngôn ngữ ngoài việc tách câu và từ, nên nó có thể áp dụng cho nhiều ngôn ngữ khác nhau.

2.1.2.2. Kỹ thuật học máy cho tóm tắt văn bản

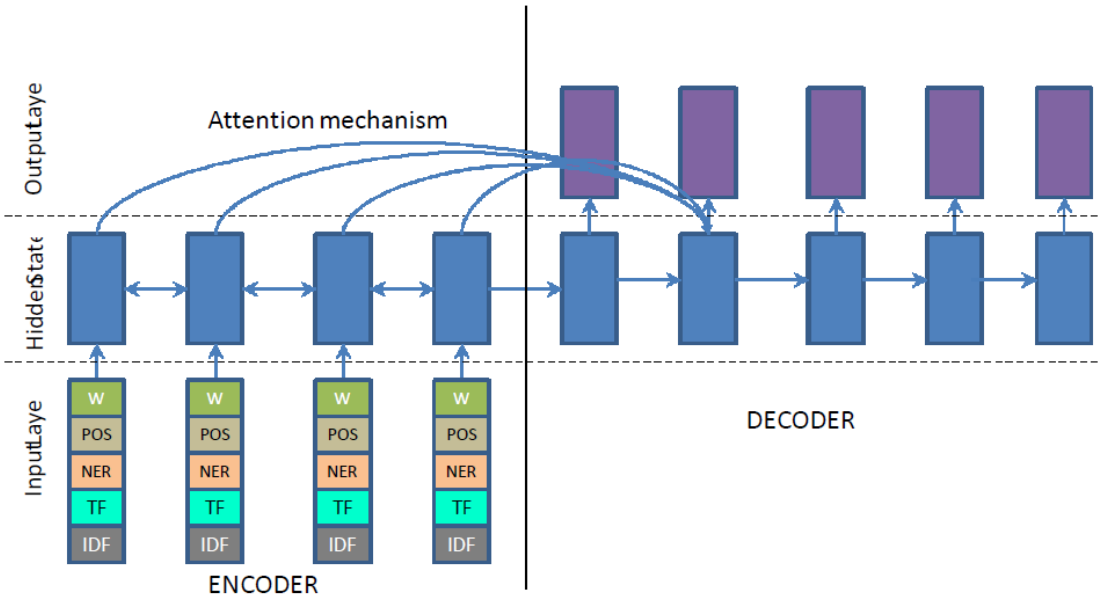
Phương pháp áp dụng học máy cho tóm tắt văn bản thực hiện giải bài toán phân loại nhị phân. Tư tưởng của chúng là phân loại các câu trong văn bản đầu vào thành hai tập là tập các câu tóm tắt và tập các câu không là tóm tắt dựa vào các đặc trưng mà chúng có. Tập dữ liệu huấn luyện gồm các văn bản và các bản tóm tắt trích chọn tương ứng.

Xác suất một câu được chọn vào văn bản tóm tắt là điểm số của câu. Việc lựa chọn các hàm phân loại đóng vai trò quan trọng trong việc tính điểm cho các câu. Một số đặc trưng phân loại thường được sử dụng trong tóm tắt văn bản gồm có vị trí của câu trong văn bản, độ dài của câu, tồn tại của các từ viết hoa, độ tương đồng của câu với tiêu đề của văn bản… Có nhiều kỹ thuật học máy được áp dụng trong tóm tắt văn bản, tiêu biểu là áp dụng của mô hình Markov ẩn (Hidden Markov Model) [14].

2.2. Tóm tắt văn bản theo hướng tóm lược

Những năm gần đây với sự phát triển của phần cứng máy tính, cùng với nhiều kỹ thuật tiên tiến dựa trên mạng nơ ron nhân tạo và kiến trúc mạng học sâu, một số nghiên cứu về tóm tắt văn bản bằng tóm lược đã được thực hiện với mục tiêu tạo được văn bản tóm tắt giống như cách con người thực hiện.

Nallapati và cộng sự [22] áp dụng mô hình chuỗi sang chuỗi (sequence-to-sequence) với cơ chế attention kết hợp với các đặc trưng ngôn ngữ (part-of-speech, name-entity và TF-IDF) để thực hiện tóm tắt văn bản theo hướng tóm lược (hình 2.1). Kết quả cho thấy mô hình có khả năng sinh ra các từ không có trong văn bản đầu vào, nhiều ví dụ cho thấy mô hình có thể sinh ra được đoạn tóm tắt gần giống với con người viết.



Hình 2.1. Mô hình sequence-to-sequence với cơ chế attention

Tác giả See và cộng sự trong [28] đề xuất cải tiến mạng pointer-generator trên mô hình chuỗi sang chuỗi cho phép thực hiện sao chép một (các từ) từ văn bản gốc vào văn bản tóm tắt trong trường hợp mô hình sinh ra một từ không có trong tập từ vựng (unknown word). Mô hình được thử nghiệm trên bộ dữ liệu tiếng anh các bài báo của CNN/DailyMail cho kết quả khá khả quan. Hình 2.2. minh họa ví dụ chạy thử nghiệm được tác giả công bố.

Article: smugglers lure arab and african migrants by offering discounts to get onto overcrowded ships if people bring more potential passengers, a cnn investigation has revealed. (...)
Summary: cnn investigation **uncovers** the **business inside** a **human smuggling ring**.

Article: eyewitness video showing white north charleston police officer michael slager shooting to death an unarmed black man has exposed discrepancies in the reports of the first officers on the scene. (...)
Summary: more **questions than answers emerge** in **controversial s.c.** police shooting.

Hình 2.2. Ví dụ văn bản tóm tắt được sinh bởi mô hình pointer-generator networks

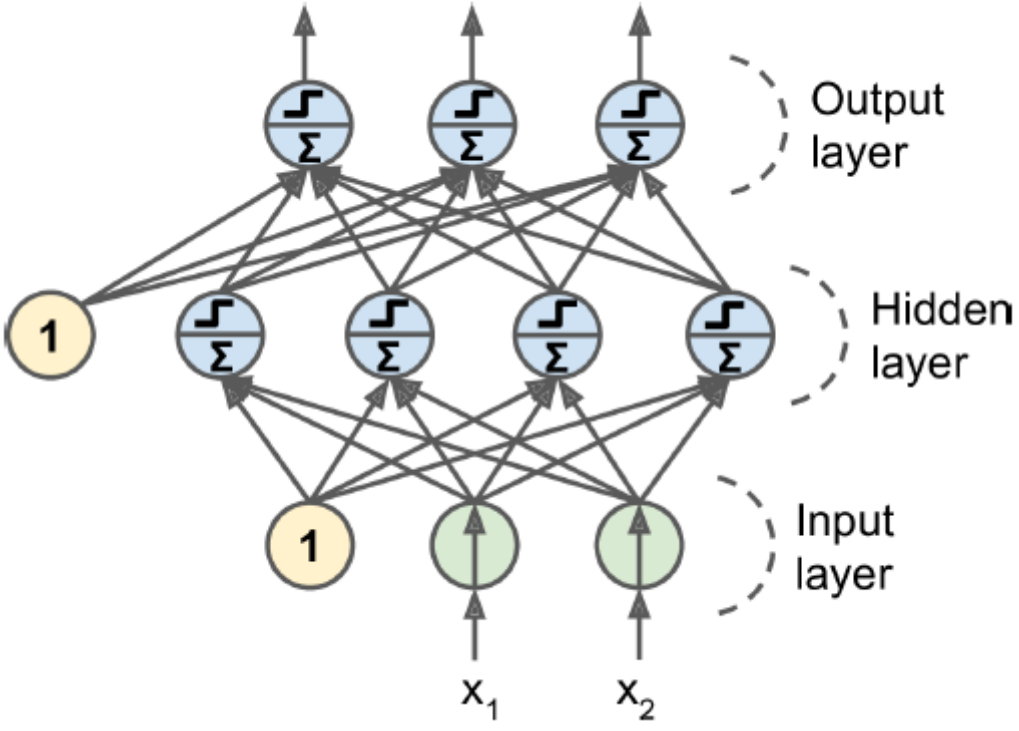
CHƯƠNG 3: MẠNG NƠ RON NHÂN TẠO

3.1. Mạng nơ ron nhân tạo ANN

Mạng nơ ron nhân tạo (ANN – Artificial Neural Network) là một mô phỏng xử lý thông tin, được nghiên cứu ra từ hệ thống thần kinh của con người, giống như bộ não để xử lý thông tin. Mạng ANN bao gồm số lượng lớn các mối gắn kết cấp cao để xử lý các thông tin trong mối liên hệ rõ ràng. Nó có khả năng học bởi kinh nghiệm từ huấn luyện, lưu những kinh nghiệm thành tri thức và áp dụng trong những dữ liệu mới trong tương lai.

3.1.1. Cấu trúc mạng nơ ron nhân tạo

Mỗi nơ ron (gọi là nút mạng) là yếu tố cơ bản nhất cấu tạo nên mạng nơ ron, tham gia vào xử lý thông tin trong mạng. Các nơ ron trong mạng liên kết với nhau, xử lý và chuyển tiếp thông tin dựa trên các trọng số liên kết và hàm kích hoạt.



Hình 3.1. Cấu trúc mạng nơ ron nhân tạo

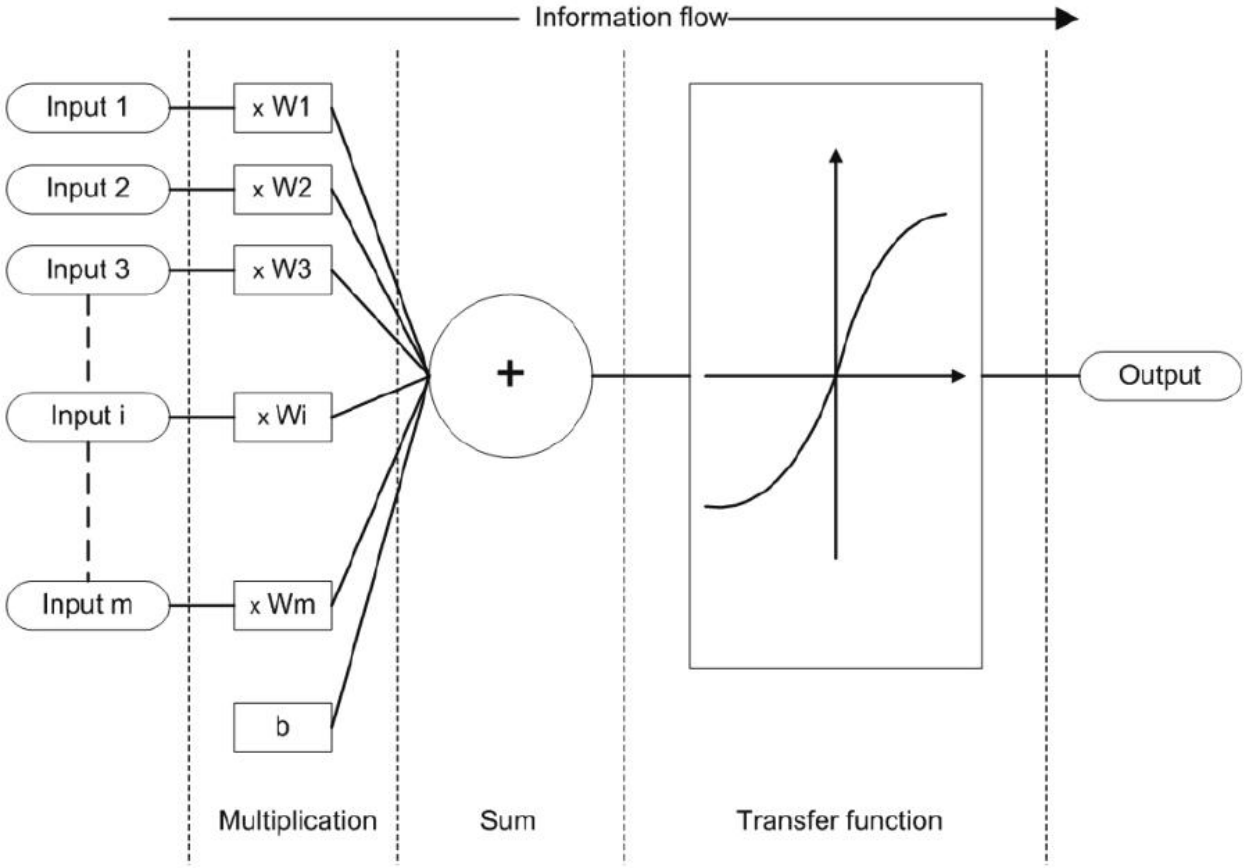
Cấu trúc mạng nơ ron nhân tạo về cơ bản gồm ba lớp: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer) được minh họa như hình 3.1. Khi một mạng ANN có nhiều hơn hai lớp ẩn thì được gọi là một mạng nơ ron sâu (deep neural network hay DNN) [8].

3.1.2. Hoạt động của mạng ANN

Đầu vào: dữ liệu vào của mạng ANN tùy thuộc vào ứng dụng mà mô hình cần xử lý. Ví dụ với bài toán kinh điển nhận dạng ký tự viết tay, đầu vào là các ảnh chụp các số viết tay từ 0 đến 9.

Đầu ra của mạng ANN là lời giải cho bài toán cần giải quyết, ví dụ với bài toán nhận dạng ký tự chữ viết tay thì đầu ra sẽ là dự đoán tương ứng cho ảnh đầu vào, ví dụ ảnh đầu vào là số 7 viết tay, thì đầu ra là kết quả đúng nếu dự đoán là số 7, và sai nếu trả kết quả là một số khác số 7 (ví dụ số 1 hay số 4).

Hoạt động của mạng ANN được minh họa trong hình 3.2 [15]. Thông tin tới một nơ ron được nhân với một trọng số (mỗi đầu vào có thể được nhân với một trọng số khác nhau), sau đó nơ ron sẽ tính tổng các đầu vào đã tính trọng số và tham số hiệu chỉnh (bias) và xử lý tổng này thông qua một hàm kích hoạt (activation function) hay còn gọi là chuyển đổi (transfer function).



Hình 3.2. Nguyên lý hoạt động của mạng ANN

Quá trình tính toán được thực hiện bằng công thức:

$$y(k) = F\left(\sum_{i=0}^{m} w_i\,(k) * x_i(k) + b\right) \quad (3.1)$$

Trong đó: $x_i(k)$ là giá trị đầu vào tại từng thời điểm k, $w_i(k)$ là giá trị trọng số của đầu vào i, b là tham số hiệu chỉnh (bias), F là một hàm kích hoạt và $y(k)$ là giá trị đầu ra tương ứng.

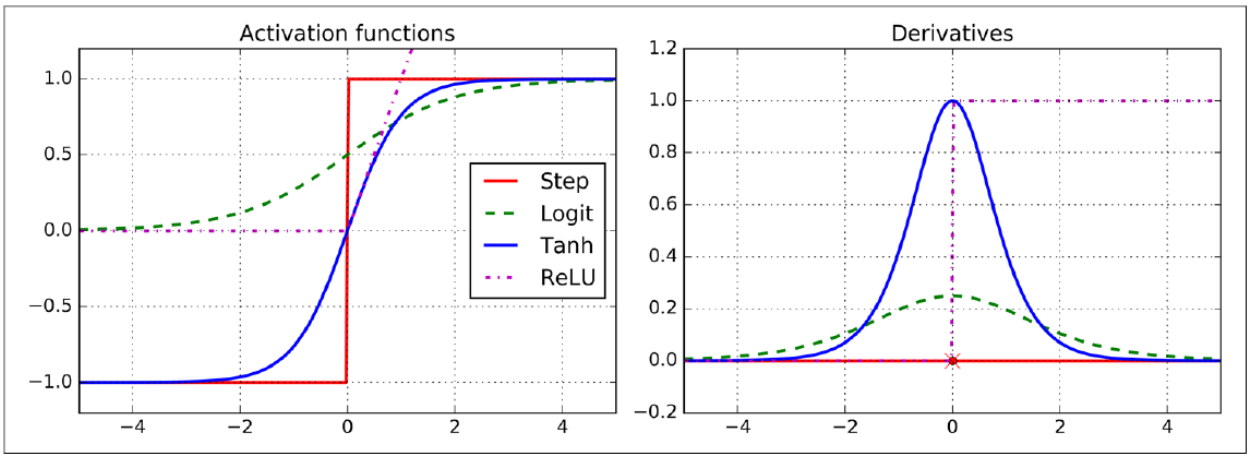
Một số hàm kích hoạt thường được sử dụng là hàm bước nhảy (step function), hàm logit (hay hàm sigmoid), hàm tanh và hàm Rectified Linear Unit (ReLU) [8].

$$Sigmoid(z) = \frac{1}{1 + e^{-z}} \quad (3.2)$$

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \quad (3.3)$$

$$ReLU(z) = \max(0, z) \quad (3.4)$$

Đồ thị của các hàm kích hoạt này và đạo hàm của nó được thể hiện trong hình 3.3. [8].

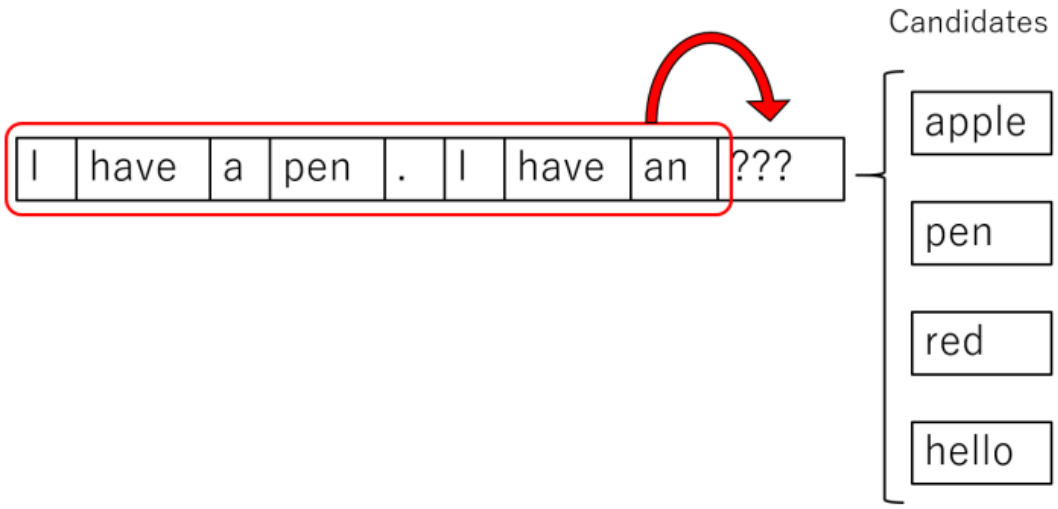


Hình 3.3. Đồ thị của các hàm kích hoạt phổ biến và đạo hàm của chúng.

3.2. Mạng nơ ron hồi quy RNN

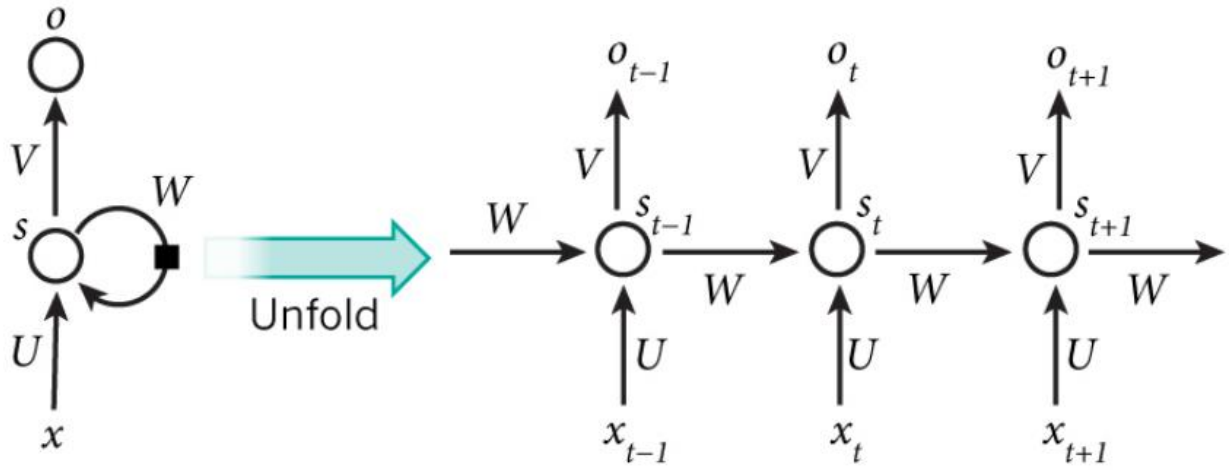
Trong một mạng nơ ron truyền thống, chúng ta giả sử rằng tất cả các dữ liệu đầu vào và dữ liệu đầu ra là độc lập với nhau, nhưng trong nhiều bài toán thực tế thì giả sử này hoàn toàn sai. Ví dụ trong bài toán dự đoán từ tiếp theo trong một câu như minh họa hình 3.4 [5], việc suy diễn sẽ có căn cứ và có xác suất đúng ngữ cảnh là cao hơn nếu biết các từ trước nó. Việc sử dụng thông tin có tính chuỗi tuần tự chính

là tư tưởng cho việc nghiên cứu và phát triển mạng nơ ron hồi quy RNN (Recurrent Neural Network) [6].



Hình 3.4. Ví dụ bài toán dự đoán từ.

Các mạng RNN được gọi là hồi quy (hay hồi tiếp) bởi vì chúng thực thi cùng một tác vụ cho mỗi thành phần của chuỗi với đầu ra phụ thuộc vào các kết quả tính toán trước đó. Có thể hiểu theo một cách khác đó là RNN có bộ nhớ mà đã lưu trữ các thông tin đã xử lý trước đó. Về lý thuyết thì mạng RNN có thể xử lý thông tin cho một chuỗi dài tùy ý, song trên thực tế thì khả năng này khá giới hạn trong chỉ vài bước [6]. Một mạng RNN tiêu biểu có cấu trúc như hình 3.5:



Hình 3.5. Cấu trúc mạng RNN tiêu biểu

Hình 3.5 minh họa một mạng RNN trải ra thành một mạng đầy đủ [6]. Điều này có nghĩa là ta có thể trải một mạng RNN để xử lý cho một chuỗi đầy đủ. Ví dụ, nếu một chuỗi là một câu gồm năm từ, thì mạng có thể trải ra thành năm lớp mạng nơ

ron, mỗi lớp xử lý một từ. Các công thức tính toán trong mạng RNN cụ thể như sau:

- x_t là đầu vào tại thời điểm t, ví dụ, x_1 có thể là một véc tơ one-hot tương ứng với từ thứ hai của một câu.

- s_t là trạng thái ẩn tại thời điểm t. Nó giống như là bộ nhớ của mạng, st được tính dựa vào trạng thái ẩn trước đó và đầu vào của bước hiện tại: $st = f(Ux_t + Ws_{t-1})$. Hàm f thường là một hàm phi tuyến như là hàm tanh hoặc hàm ReLU, s_{t-1} thường được khởi tạo là 0 khi tính toán trạng thái ẩn thứ nhất.

- O_t là đầu ra (output) tại bước t. Ví dụ với bài toán dự đoán từ tiếp theo trong câu thì O_t có thể là một véc tơ xác suất các từ trong từ điển: $O_t = \text{softmax}(Vs_t)$.

Không giống với mạng nơ ron thông thường với các tham số khác nhau tại mỗi lớp mạng (layer), mạng RNN sử dụng cùng một bộ tham số (U, V, W) trong tất cả các bước. Điều này ám chỉ rằng nó sẽ thực hiện cùng một tác vụ tại mỗi bước, nhưng với các đầu vào khác nhau. Chính đặc trưng này làm giảm đi đáng kể số lượng các tham số cần học trong mạng. Mạng RNN có thể có đầu ra tại mỗi bước, nhưng tùy theo bài toán cần xử lý mà các kết quả này có cần thiết hay không; tương tự với đầu vào, mạng RNN không nhất thiết cần có đầu vào tại mỗi thời điểm. Đặc trưng quan trọng nhất của RNN là trạng thái ẩn của nó, với khả năng nắm giữ thông tin về một chuỗi liên tiếp [6].

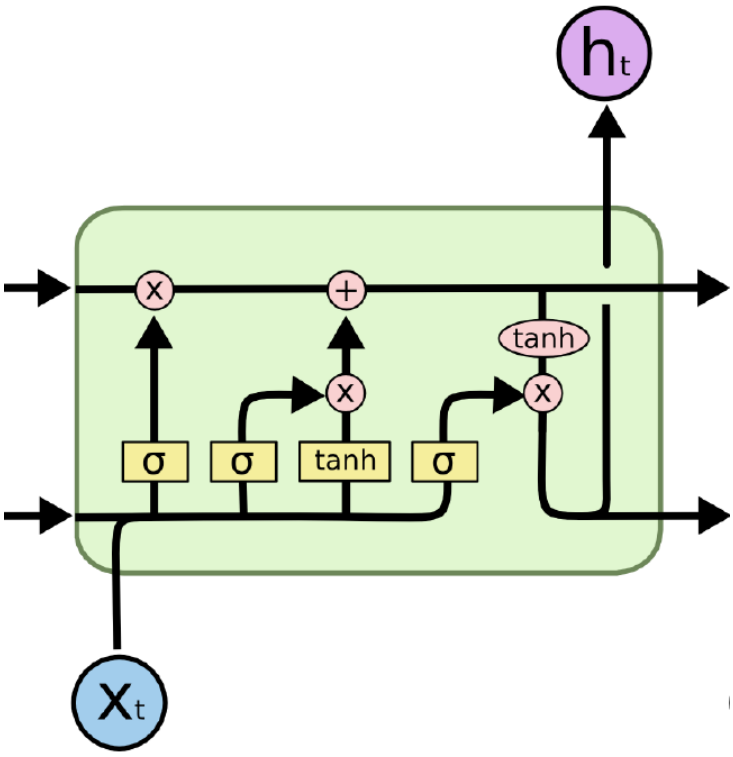
3.3. Mạng nơ ron có nhớ LSTM

Khi quan sát một mạng nơ ron, chức năng của nó giống như một chiếc hộp đen. Dữ liệu được đưa vào một chiều và kết quả được đưa ra ở một chiều khác, quá trình quyết định đưa ra kết quả chỉ phụ thuộc vào các đầu vào hiện tại. Nhìn chung, mạng nơ ron không hoàn toàn là không có khả năng nhớ, vì về cơ bản chúng học các tham số logic trong quá trình huấn luyện [29]. Tuy nhiên khả năng nhớ này là rất hạn chế và không phù hợp đối với các trường hợp khi cần sử dụng trạng thái nhớ trung gian để sử dụng sau này, ví dụ như tóm tắt nội dung chính của bài báo.

Cách cơ bản nhất để một mạng nơ ron chấp nhận dữ liệu theo thời gian (time series data) đó là kết nối vài mạng nơ ron lại với nhau, mỗi mạng nơ ron xử lý một bước theo thứ tự thời gian. Tức là thay vì đưa dữ liệu đầu vào rời rạc, dữ liệu được đưa theo một cửa sổ thời gian, hay một ngữ cảnh, vào mạng nơ ron.

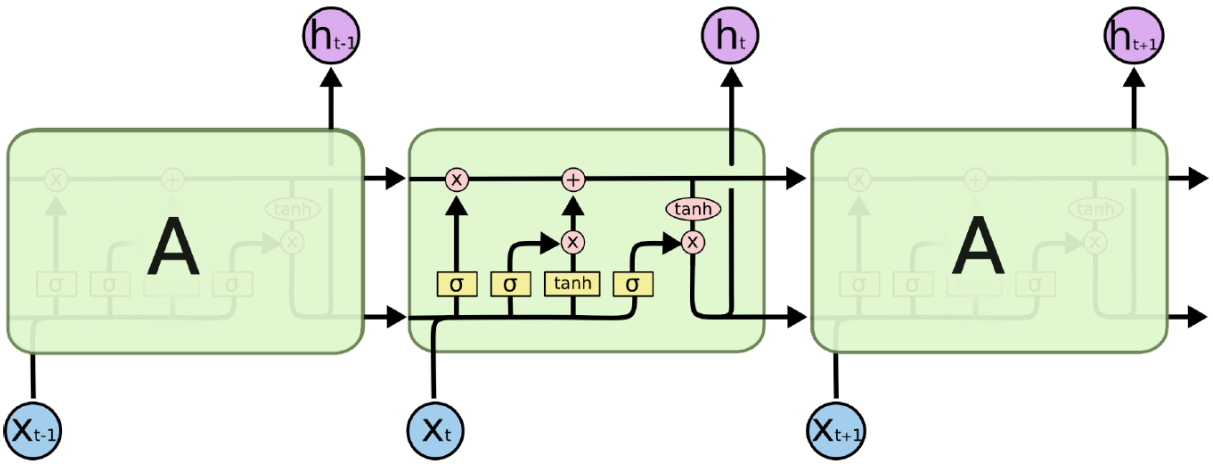
Về lý thuyết thì mạng nơ ron hồi tiếp (recurrent neural network) có thể hoạt động, nhưng thực tế trong nhiều nghiên cứu chỉ ra hạn chế của mạng RNN là sự hội tụ và phân kỳ gradient (vanishing gradient và exploding gradient) [29]. Hạn chế này khiến RNN không hiệu quả đối với các bài toán cần xử lý dữ liệu theo thời gian đòi hỏi trạng thái nhớ trung gian.

LSTM (Long short term memory) [12] ra đời để giải quyết hạn chế của RNN bằng việc đưa vào mạng một đơn vị nhớ được gọi là memory unit hay Cell.



Hình 3.6. Kiến trúc LSTM

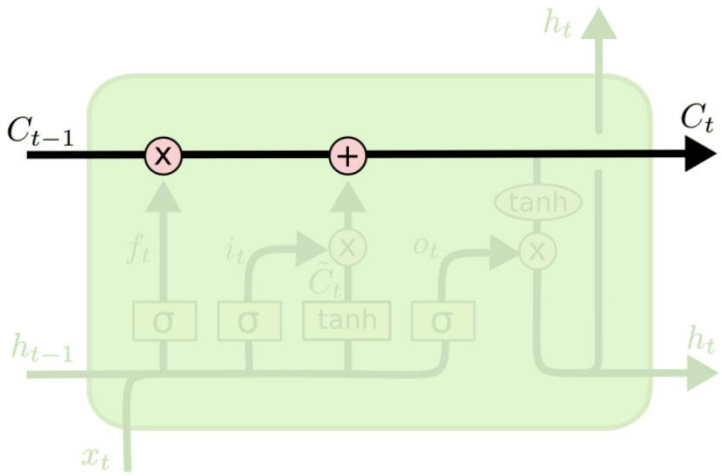
Kiến trúc một khối LSTM được thể hiện trên Hình 3.6 [4]. Đầu vào gồm ba thành phần. Xt là đầu vào tại bước hiện tại. h_{t-1} là đầu ra từ một khối LSTM trước và C_{t-1} là "nhớ" của khối trước, và đây cũng chính là điểm quan trọng nhất của LSTM. Đầu ra của nó gồm h_t là kết quả của khối LSTM hiện tại và C_t là nhớ của nó. Như vậy, một khối đơn LSTM đưa ra quyết định dựa vào việc xem xét đầu vào hiện tại, kết quả và nhớ của khối trước và nó sinh ra một đầu ra mới cũng như là nhớ của nó. Một mô hình mạng LSTM [4] được minh họa trong hình 3.7.



Hình 3.7. Kiến trúc mạng LSTM

Điểm quan trọng nhất của LSTM chính là trạng thái nhớ (cell state), thể hiện ở đường kẻ ngang trên cùng của Hình 3.8.

Véc tơ nhớ C_{t-1} được đưa vào một ống nhớ (memory pipe) qua một cổng gọi là cổng bỏ nhớ (forget gate), cổng bỏ nhớ thực chất là một toán hạng nhân ma trận (element-wise multiplication operation). C_{t-1} sẽ được nhân với một véc tơ, và nếu kết quả là gần 0, thì kết quả nhớ C_{t-1} sẽ bị loại bỏ, ngược lại nếu kết quả là 1 thì C_{t-1} sẽ được đi tiếp. Hình 3.8 minh họa hoạt động của ống nhớ trong khối LSTM [4].

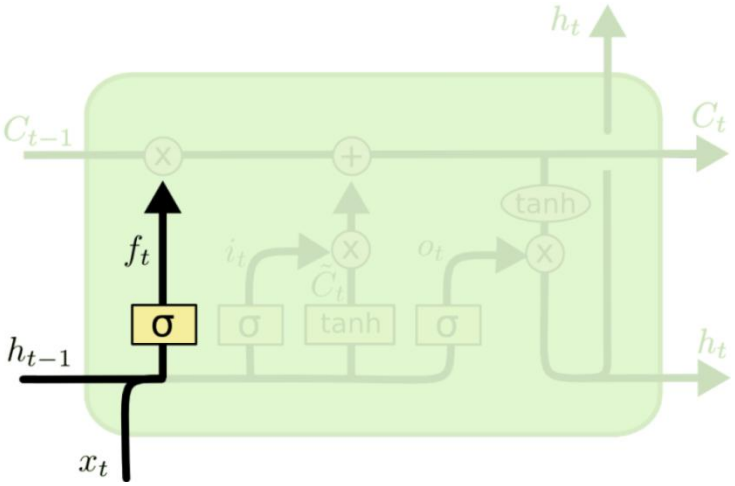


Hình 3.8. Ống nhớ trong khối LSTM

Cụ thể cách hoạt động của LSTM [4] như sau:

Đầu tiên khối LSTM là quyết định thông tin nào sẽ loại bỏ khỏi cell state. Quá trình quyết định này do một lớp sigmoid gọi là "forget gate layer" thực hiện. Cổng bỏ

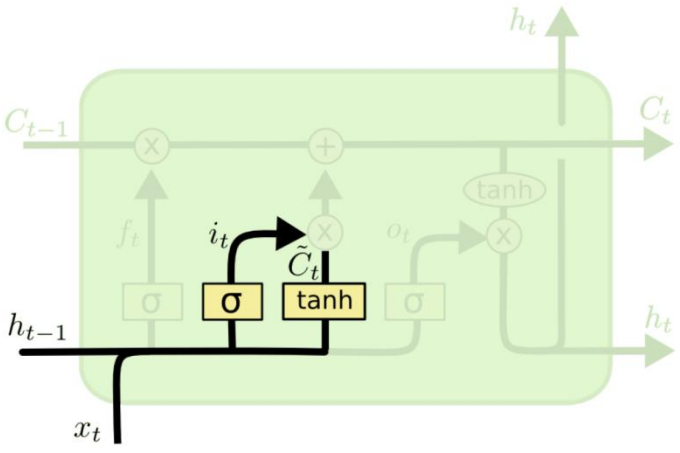
nhớ lấy đầu vào là h_{t-1} và x_t và cho đầu ra là một giá trị nằm trong khoảng [0, 1] cho cell state C_{t-1}. Nếu kết quả đầu ra là 1 thể hiện cho việc "giữ lại thông tin", và 0 thể hiện rằng "thông tin bị loại bỏ".



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right) \tag{3.5}$$

Hình 3.9. Cổng bỏ nhớ của LSTM

Tiếp theo LSTM quyết định thông tin mới sẽ được lưu lại tại cell state như thế nào. Việc này được gồm hai phần, một là lớp sigmoid gọi là "input gate layer" (lớp đầu vào) quyết định giá trị sẽ được cập nhật, và một lớp tanh tạo ra một véc tơ các giá trị mới, $\widetilde{C_t}$, mà có thể được thêm vào cell state.



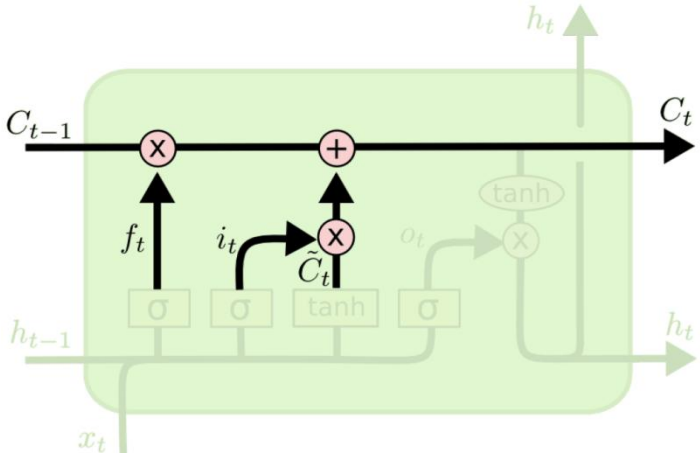
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C) \tag{3.6}$$

Hình 3.10. LSTM tính toán giá trị lưu tại cell state

Kế tiếp, trạng thái cell state cũ C_{t-1} được cập nhật tại trạng thái cell state mới C_t theo công thức:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.7}$$

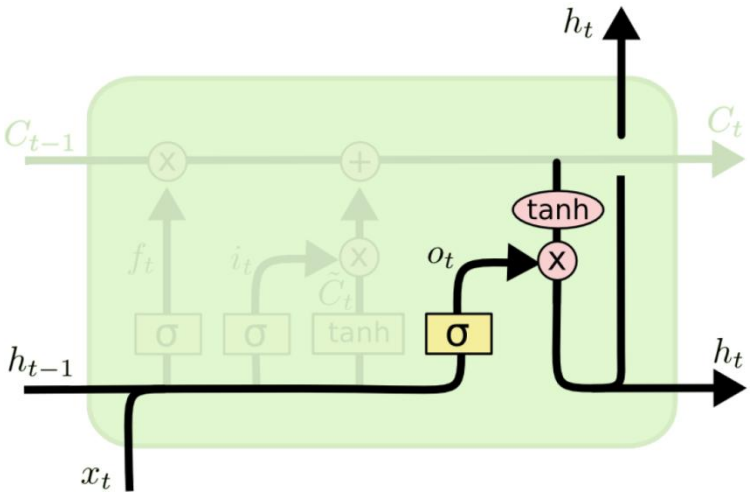
Trạng thái nhớ cũ C_{t-1} được nhân với giá trị kết quả của cổng bỏ nhớ f_t, thực hiện việc loại bỏ những gì đã được quyết định loại bỏ ở bước trước. Giá trị $i_t*\widetilde{C_t}$ thể hiện giá trị ứng viên mới cho cell state được quyết định bởi hệ số giãn nở i_t cụ thể cho việc cập nhật giá trị cho mỗi cell state. Hình 3.11 minh họa việc cập nhật giá trị cho cell state tại bước này.



Hình 3.11. Cập nhật giá trị Cell State

Bước cuối cùng, khối LSTM quyết định đầu ra của nó dựa trên cell state được minh họa trong hình 3.12. Lớp sigmoid được dùng để tính toán thành phần của cell state sẽ được xuất ra. Sau đó, giá trị cell state được đưa vào hàm tanh (kết quả sẽ thuộc khoảng [-1,1]) và nhân với kết quả đầu ra của cổng sigmoid, để quyết định cái gì sẽ được khối LSTM xuất ra. Công thức tính toán cho các thành phần của bước này như sau:

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right) \tag{3.8}$$



Hình 3.12. Đầu ra của khối LSTM

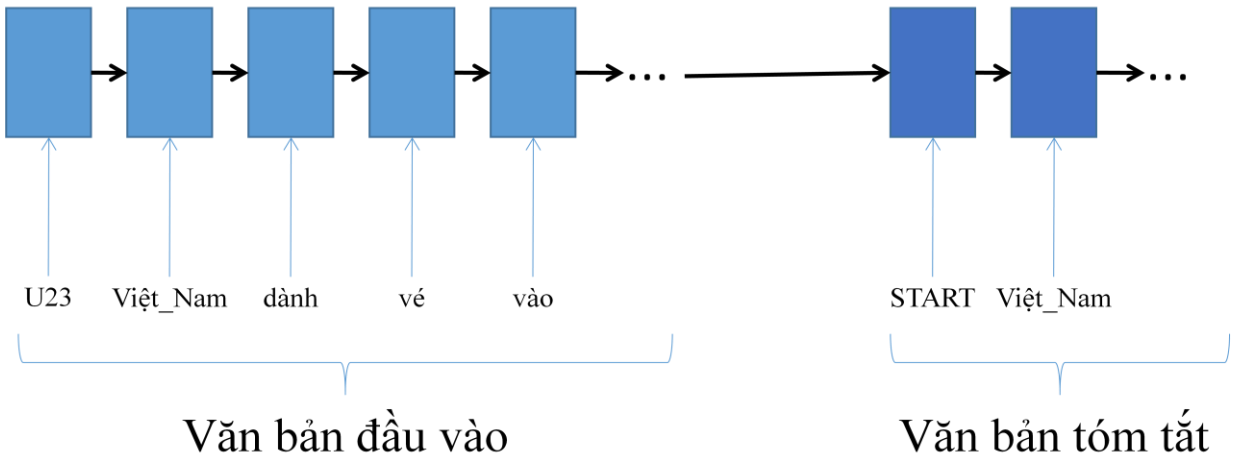
Mạng LSTM là kết hợp của các khối LSTM kết nối kế tiếp nhau qua theo chuỗi thời gian. Hoạt động của mỗi khối LSTM tại một thời điểm được đảm trách bởi các cổng: cổng bỏ nhớ f_t, cổng đầu vào i_t và cổng đầu ra o_t, trong đó cổng bỏ nhớ chính là điểm đáng chú ý nhất của LSTM, đem lại khả năng sử dụng thông tin tính toán từ các thời điểm trước đó.

CHƯƠNG 4: XÂY DỰNG HỆ THỐNG TÓM TẮT VĂN BẢN THEO HƯỚNG TÓM LƯỢC

Bài toán tóm tắt văn bản theo hướng tóm lược có thể được phát biểu như sau: đầu vào của bài toán là một văn bản **x** gồm M từ: x_1, x_2, …, x_m. Chúng ta sẽ ánh xạ chuỗi M từ này thành một chuỗi đầu ra **y** gồm N từ: y_1, y_2, …, y_n; trong đó N<M dựa trên một tập từ vựng có kích thước cố định V. Các từ thuộc N không nhất định phải thuộc M. Mục tiêu là tìm một chuỗi đầu ra y làm cực đại hóa xác suất có điều kiện của y theo chuỗi đầu vào x:

$$\text{argmax}_{y \in V} P(y|x) \tag{4.1}$$

Hình 4.1 minh họa mô hình bài toán tóm tắt văn bản tự động.



Hình 4.1. Mô hình bài toán tóm tắt văn bản

4.1. Quy trình tóm tắt theo hướng tóm lược sử dụng mạng LSTM

Để thực hiện tóm tắt văn bản theo hướng tóm lược sử dụng mạng LSTM, chúng tôi thực hiện các bước như sau:

- Thu thập dữ liệu phù hợp: dữ liệu phù hợp cho bài toán tóm tắt văn bản tiếng việt áp dụng LSTM là bộ dữ liệu gồm một cặp tương ứng: văn bản đầy đủ và văn bản tóm tắt mẫu (do con người thực hiện tóm tắt).

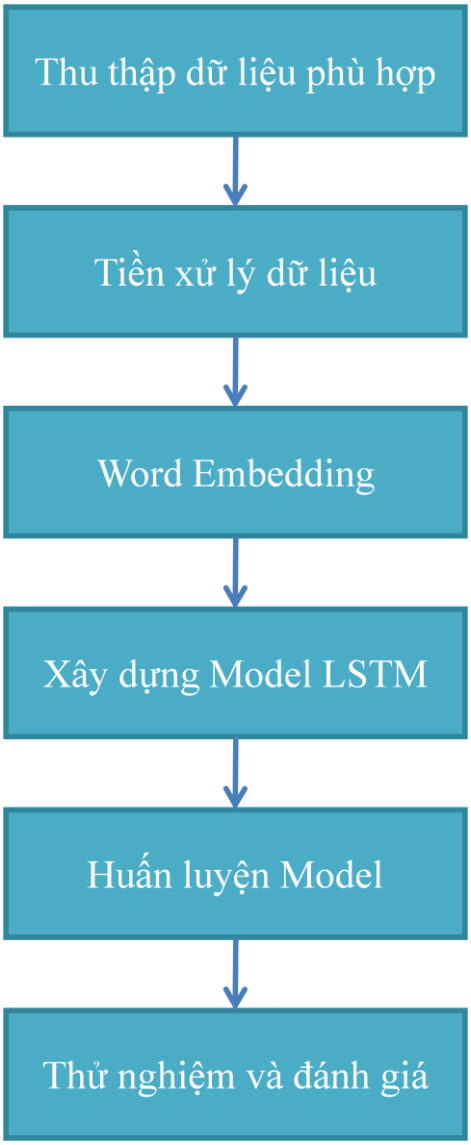
- Xử lý dữ liệu: làm sạch dữ liệu, loại bỏ các ký tự không cần thiết, các lỗi phân tách câu.

- Word embedding: véc tơ hóa dữ liệu về dạng số để đưa vào mô hình LSTM

- Xây dựng mô hình LSTM: xây dựng mô hình xử lý chuỗi văn bản đầu vào, mục tiêu là tạo ra chuỗi văn bản tóm tắt bằng cách áp dụng các khối LSTM.

- Huấn luyện và đánh giá mô hình sử dụng bộ dữ liệu đã được xử lý phía trên.

Các bước được tiến hành như thể hiện trong hình 4.2, chi tiết các bước được thể hiện trong các mục tiếp theo của luận văn.



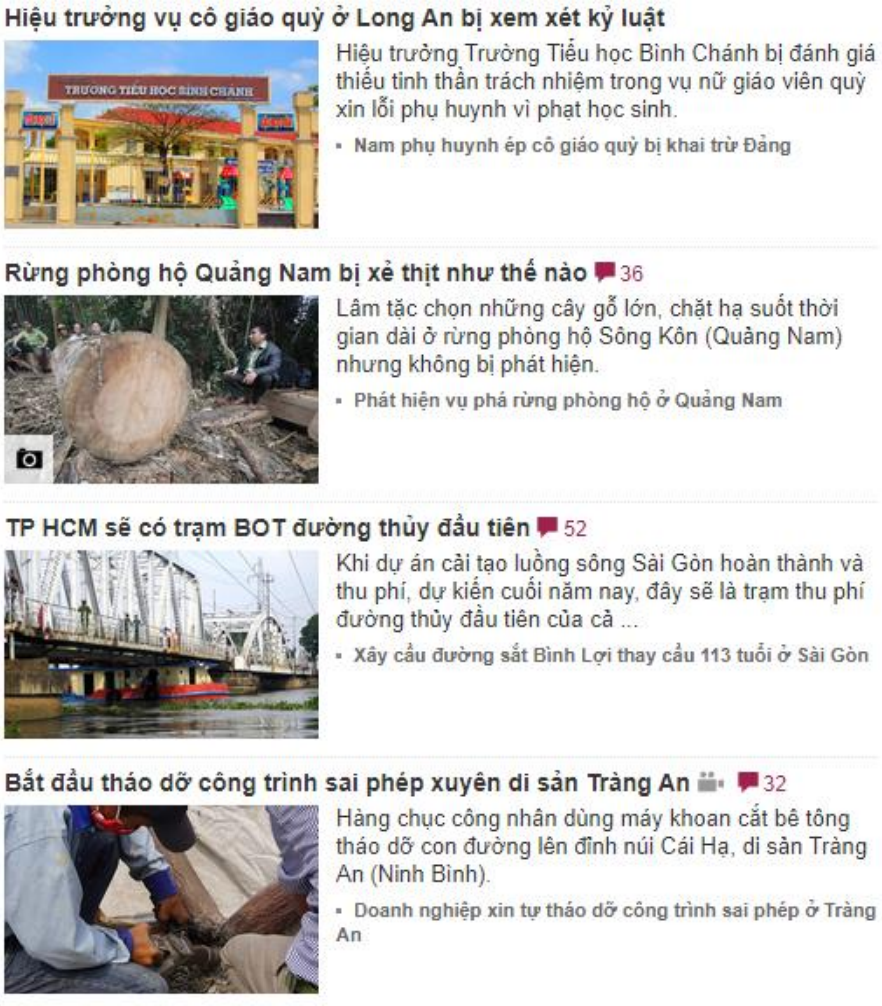
Hình 4.2. Quy trình thực hiện tóm tắt văn bản tiếng Việt với LSTM

4.2. Xây dựng bộ dữ liệu cho tóm tắt văn bản tiếng Việt

Bài toán tóm tắt văn được đã được rất nhiều tác giả nghiên cứu, đặc biệt là đối với tóm tắt văn bản tiếng Anh. Với tóm tắt văn bản tiếng Anh, bộ dữ liệu kinh điển được sử dụng là bộ dữ liệu Gigaword với khoảng bốn triệu bài báo (Graff và các

cộng sự, 2003 [9]), chi phí mua giấy phép sử dụng bộ dữ liệu này là 6,000 USD nên chỉ có những tổ chức lớn mới có khả năng tiếp cận kho dữ liệu này. Một kho dữ liệu khác thường được sử dụng cho tóm tắt văn bản tiếng Anh đó là bộ dữ liệu các bài báo của CNN/Daily Mail với hơn 90,000 bài báo CNN và hơn 200,000 bài báo Daily Mail [11].

Tuy nhiên, đối với tóm tắt văn bản tiếng Việt, hiện tại chưa có kho dữ liệu chính thức nào được công bố, đây là thách thức lớn đối với chúng tôi. Vì vậy, để chuẩn bị dữ liệu thực hiện bài toán tóm tắt văn bản tiếng Việt, chúng tôi tiến hành thu thập dữ liệu là các bài báo trên một số website tin tức của Việt Nam. Dữ liệu mà chúng tôi quan tâm đó là phần tóm tắt dưới tiêu đề của bài báo, và nội dung văn bản của bài báo. Hình 4.3 minh họa một website mà chúng tôi thực hiện thu thập dữ liệu.



Hình 4.3. Thu thập dữ liệu cho tóm tắt văn bản tiếng Việt

Với dữ liệu thu được từ các website tin tức trực tuyến của Việt Nam, chúng tôi tiến hành tiền xử lý để làm sạch dữ liệu và loại bỏ các ký tự nhiễu trong văn bản như sau:

- Loại bỏ các dấu gạch đầu dòng, các dấu gạch ngang trong văn bản.

- Loại bỏ các dấu hai chấm ":" trước mỗi danh sách liệt kê.

- Loại bỏ các dấu ba chấm, các dấu ngoặc đơn và phần chú thích thêm trong ngoặc đơn, các dấu nháy đơn, các dấu nháy kép.

- Thay thế các dấu chấm phẩy ";" phân tách ý thành dấu chấm ngắt câu "."

- Thêm dấu chấm kết thúc câu cho những chú thích dưới ảnh không có dấu kết thúc câu.

- Tách các câu trong phần tóm tắt của bài báo bằng phân tách các câu dựa trên kết thúc câu bởi dấu chấm, dấu chấm hỏi và dấu chấm than.

- Tách văn bản thành các token.

- Chuyển đổi các phần tóm tắt và bài báo từ dạng văn bản thông thường thành dạng nhị phân và ghi vào file.

4.3. Word Embedding

Word embedding là quá trình chuyển đổi văn bản thành các con số và có thể có nhiều đại diện dạng số khác nhau thể hiện cùng một văn bản. Word embedding là kỹ thuật để thể hiện các từ thành các véc tơ có kích thước cố định, sao cho các từ có nghĩa tương tự hoặc gần nghĩa được thể hiện bằng các véc tơ gần nhau (tính theo khoảng cách euclid) [13].

Nhiều thuật toán học máy và hầu hết tất cả các kiến trúc học sâu (deep learning) không thể xử lý trực tiếp các xâu hay các văn bản thông thường. Chúng yêu cầu đầu vào là các con số để thực thi các tác vụ của mình như phân loại văn bản, dịch. Word embedding về cơ bản sẽ thực hiện ánh xạ một từ trong một từ điển thành một véc tơ [27]. Chính vì vậy có thể hiểu word embedding là quá trình véc tơ hóa một từ, hay tổng quát là véc tơ hóa văn bản. Một véc tơ đại diện của một từ có thể là một véc tơ one-hot, véc tơ one-hot chỉ có một giá trị là 1, còn lại tất cả đều là 0, giá trị 1 thể hiện vị trí của từ trong từ điển.

Ví dụ, giả sử ta có hai câu: " Tôi thích chơi piano" và "Tôi thích chơi guitar".

Đầu tiên chúng ta tách chọn các từ phân biệt trong hai câu, tập các từ phân biệt thu được là tập V = { Tôi, thích, chơi, piano, guitar} gồm 5 từ. Các từ trong tập V có thể được véc tơ hóa dạng one-hot như sau:

Tôi = [1,0,0,0,0], thích = [0,1,0,0,0], chơi = [0,0,1,0,0], piano = [0,0,0,1,0] , guitar = [0,0,0,0,1].

Phần tiếp theo luận văn sẽ giới thiệu một số kỹ thuật word embedding.

4.3.1. Embedding dựa trên tần xuất xuất hiện của từ.

4.3.1.1. Count vector

Xem xét một corpus C của D tài liệu (d_1, d_2, …, d_N) và N token phân biệt được trích chọn từ tập từ vựng C [27] . N token sẽ thể hiện từ điển và kích thước của Count vector ma trận M sẽ được xác định bằng D x N. Mỗi dòng trong ma trận M gồm tần xuất xuất hiện của token trong một tài liệu D_i. Giả sử

D_1: He is a lazy boy. She is also lazy.

D_2: Tom is a lazy person.

Từ điển được tạo có thể là một danh sách các token phân biệt trong corpus= [‘He’, ‘She’, ‘lazy’, ‘boy’, ‘Tom’, ‘person’]

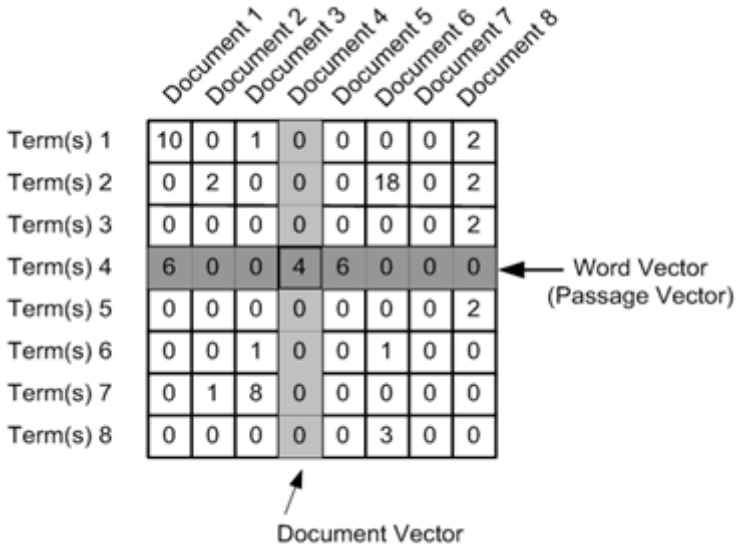
Ta có D = 2 và N = 6. Count matrix M có kích thước 2x6 được thể hiện trong bảng 4.1:

Bảng 4.1. Count matrix M có kích thước 2x6

	He	She	lazy	boy	Tom	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

Một cột có thể được hiểu như là một véc tơ từ cho một từ tương ứng trong ma trận M. Ví dụ, véc tơ cho từ ‘lazy’ trong ma trận trên là [2,1]. Và một dòng thể hiện tương ứng cho một tài liệu trong corpus và các cột tương ứng là các token trong từ điển.

Việc xây dựng ma trận M như trên phụ thuộc vào hai yếu tố: cách từ điển được xây dựng và cách đếm của các từ [27]. Thứ nhất, một corpus trong thực tế có thể gồm hàng triệu tài liệu, và với số lượng tài liệu lớn như vậy thì hàng trăm triệu từ phân biệt có thể được trích chọn ra. Do đó, ma trận M xây dựng như trên sẽ rất thưa và không hiệu quả cho việc tính toán. Vì lý do này, một từ điển thường được xây dựng từ khoảng 10000 từ dựa trên tần xuất xuất hiện của nó trong corpus. Thứ hai, cách đếm một từ có thể được tính bằng số lần từ đó xuất hiện trong tài liệu hoặc có mặt của từ đó trong tài liệu. Cách thức đếm tần xuất xuất hiện của từ trong tài liệu thường được dùng hơn, vì nó cũng tương đồng với cách xây dựng từ điển. Hình 4.4 dưới dây thể hiện hình ảnh của ma trận M:



Hình 4.4. Ma trận M được xây dựng theo phương pháp Count vector

4.3.1.2. Phương pháp vector hóa TF-IDF

TF-IDF là viết tắt của term frequency–inverse document frequency đây là phương pháp khác dựa trên tần xuất xuất hiện của từ nhưng có cách véc tơ hóa khác so với count vector, đó là nó không chỉ xem xét sự xuất hiện của một từ trong một tài liệu mà trong toàn bộ corpus [27].

Các từ phổ biến (common words) như “is”, “the”, “a”… thường có tần xuất xuất hiện cao hơn so với các từ quan trọng trong một tài liệu. Ví dụ, một tài liệu X về Ambert Einstein thì từ “Einstein” có tần xuất xuất hiện cao hơn trong các tài liệu khác, nhưng các từ phổ biến như “the” luôn có tần xuất cao hơn trong hầu hết các tài liệu.

Ý tưởng của phương pháp này là chúng ta sẽ đặt trọng số thấp cho các từ phổ biến xuất hiện trong hầu hết các tài liệu và đặt trọng số cao cho các từ mà chỉ xuất hiện trong một số tài liệu của tập tài liệu đang xét [27]. Xét hai tài liệu D_1 và D_2 với tần xuất của các token được cho ở Bảng 4.2 dưới đây:

Bảng 4.2. Minh họa phương pháp TF-IDF

Tài liệu D_1	
Token	Count
This	1
Is	1
About	2
Einstein	4

Tài liệu D_2	
Token	Count
This	1
Is	2
About	1
Me	1

TF thể hiện sự đóng góp của từ trong một tài liệu, tức là các từ liên quan tới tài liệu thì sẽ xuất hiện nhiều lần trong tài liệu. Công thức tính TF được tính như sau:

TF = (số lần xuất hiện của token T trong một tài liệu) / (tổng số token trong tài liệu đó).

Áp dụng công thức trên cho token "This" ta có TF(This.D_1) = 1/8 và TF(This.D_2) = 1/5.

IDF = log(N/n) trong đó N là tổng số tài liệu xem xét và n là tổng số tài liệu chứa token T. Ta có IDF(This) = log(2/2) = 0.

IDF(Einstein) = log(2/1) = 0.301

TF-IDF(This.D1) = (1/8)*(0) = 0

TF-IDF(This.D2) = (1/5)*(0) = 0

TF-IDF(Einstein.D1) = (4/8)*0.301 = 0.15

Phương pháp TF-IDF đánh giá một từ nếu xuất hiện trong tất cả các tài liệu thi khả năng từ đó không liên quan tới một tài liệu cụ thể, nhưng nếu một từ chỉ xuất hiện trong một vài tài liệu thì từ đó có khả năng là một từ quan trọng trong tài liệu chứa nó.

4.3.2. Word2Vec

Trong rất nhiều bài toán xử lý ngôn ngữ tự nhiên, các từ thường được đại diện bằng điểm TF-IDF. Mặc dù các điểm này mang lại ý tưởng về độ quan trọng tương ứng của các từ trong một văn bản, chúng không thể hiện được ngữ nghĩa của các từ. Word2vec [1] là một kỹ thuật trong đó áp dụng một lớp mạng nơ ron cùng với một tập dữ liệu huấn luyện không đánh nhãn, để tạo ra một véc tơ cho mỗi từ trong tập dữ liệu chứa cả những thông tin về ngữ nghĩa. Các véc tơ này hữu ích vì hai yếu tố quan trọng của chúng:

- Chúng ta có thể đo lường độ tương đồng ngữ nghĩa giữa hai từ bằng cách đo độ tương đồng cosine giữa hai véc tơ tương ứng.

- Chúng ta có thể sử dụng các véc tơ như là các đặc trưng cho các bài toán xử lý ngôn ngữ tự nhiên có giám sát như phân loại văn bản hay phân tích quan điểm.

Ví dụ, các từ đồng nghĩa thường có các véc tơ khá tương đồng dựa trên độ tương đồng cosine và các từ trái nghĩa thường là các véc tơ hoàn toàn không tương đồng. Hơn nữa, các véc tơ từ thường có xu hướng tuân theo các luật suy diễn, ví dụ: "Woman is to queen as man is to king" có thể suy ra:

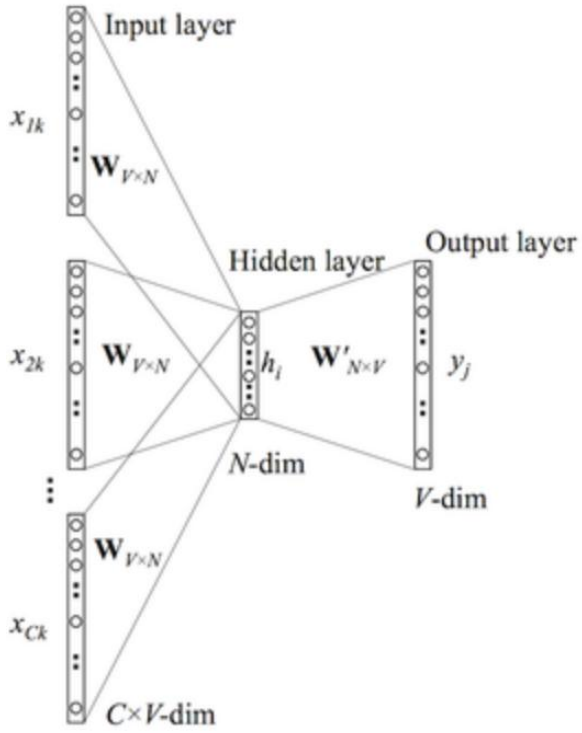
$$V_{queen} - V_{woman} + V_{man} \sim V_{king}$$

Trong đó V_{quen}, V_{woman}, V_{man}, và V_{king} là lần lượt là các véc tơ từ của queen, woman, man và king. Qua ví dụ này có thể thấy rằng các véc tơ từ có thể hàm chứa các thông tin ngữ nghĩa quan trọng của các từ mà chúng đại diện [1].

Word2Vec là phương pháp véc tơ hóa từ do Mikolov và cộng sự nghiên cứu và phát triển [21]. Đây là phương pháp dựa trên dự đoán từ, trong đó cơ sở của việc dự đoán dựa vào xác suất của các từ, độ tương tự và liên quan giữa các từ. Word2Vec kết hợp hai kỹ thuật là CBOW (Continuous bag of words) và mô hình Skip-gram (Skip-gram model). Ý tưởng của word2vec là việc đại diện các từ sử dụng các từ xung quanh từ đó. Điều này tương tự với việc con người biết nghĩa của một từ dựa trên các từ gần nó. Ví dụ xét câu "Tôi thích chơi X", với X là một từ chưa biết. Tuy nhiên, dù chưa biết nghĩa của từ X, nhưng ta có thể biết "X" là một thứ gì đó mà ta có thể "chơi" được và nó cũng tạo cảm giác "thích" [13].

4.3.2.1. CBOW (Continuous Bag of Word)

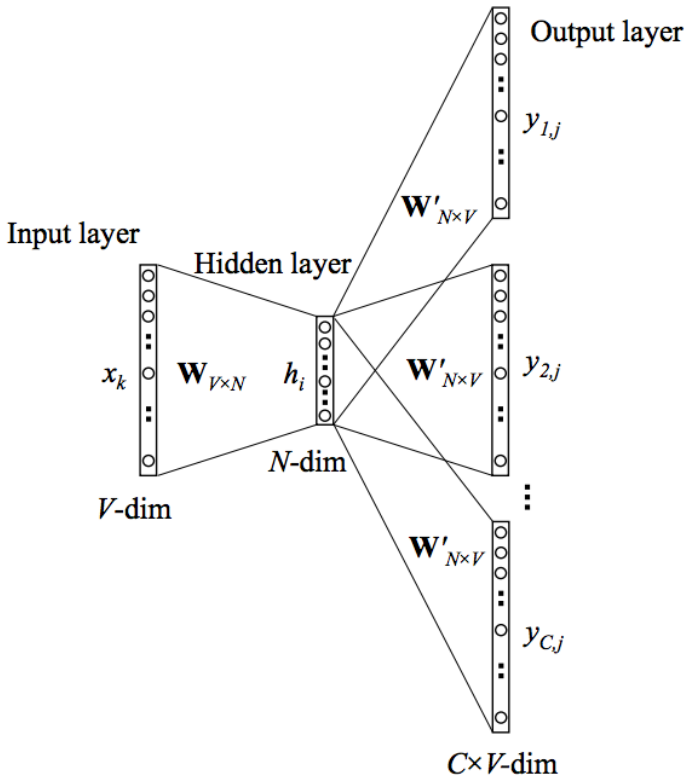
Cách hoạt động của CBOW đó là dự đoán xác suất của một từ được cho trong một ngữ cảnh (context) dựa trên các từ gần nó. Một ngữ cảnh có thể là một từ đơn hoặc một tập các từ.



Hình 4.5. Cách hoạt động của CBOW

CBOW là một mạng nơ ron nông (Shallow Neural Network) với chỉ 1 lớp ẩn hoạt động như một lớp chiếu (projection layer) của lớp đầu vào. Mục tiêu là để dự đoán được từ đích dựa trên các từ xung quanh nó. Đầu vào của CBOW là N từ, với N là kích thước của cửa sổ của ngữ cảnh được định nghĩa trước và đầu ra là từ dự đoán sử dụng lớp Softmax [13].

4.3.2.2. Mô hình Skip-gram

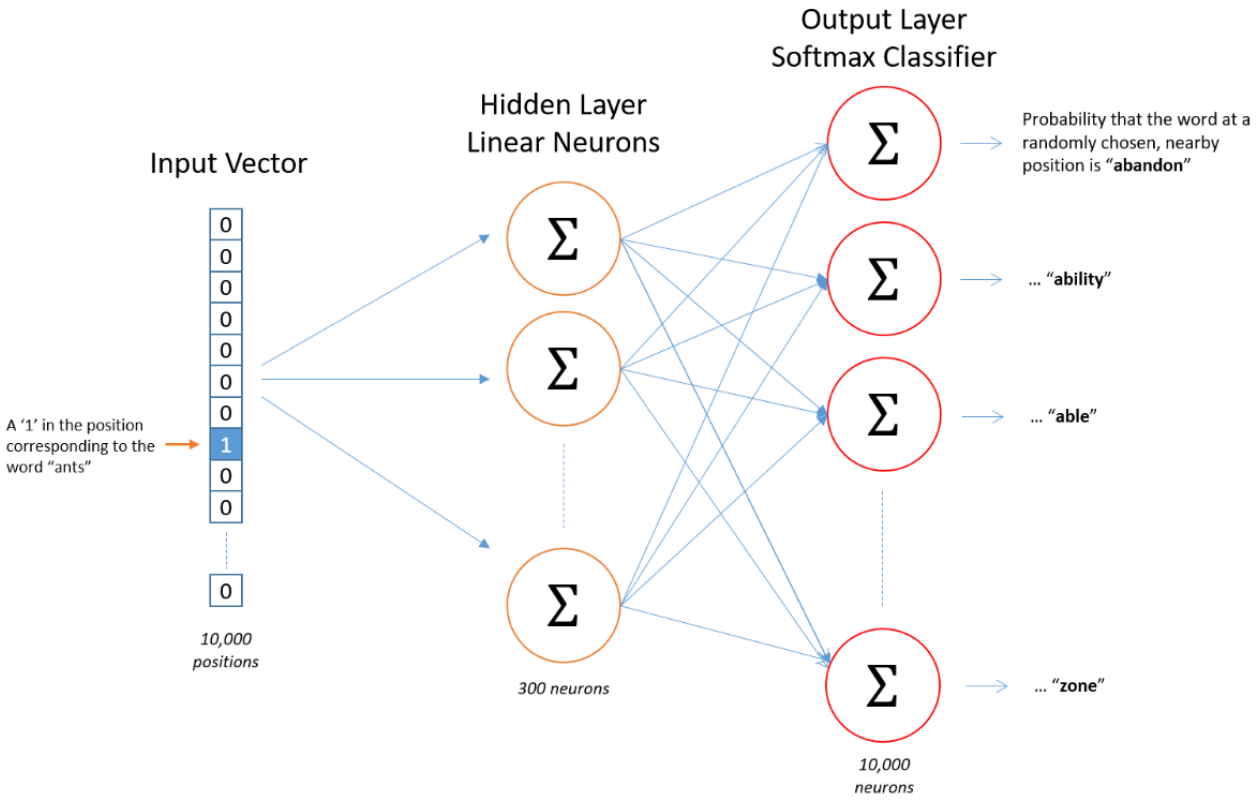


Hình 4.6. Mô hình Skip-gram

Skip-gram cũng là một mạng nơ ron chỉ gồm một lớp ẩn. Mục tiêu của mô hình này là dự đoán các từ gần với một từ đích. Đầu vào của mô hình là một véc tơ one-hot của từ đích, và đầu ra của nó là N từ với N là kích thước cửa sổ của ngữ cảnh được định nghĩa trước [13].

Trong các bài toán thực tế, mô hình skip-gram thường được áp dụng do nó đem lại độ chính xác cao hơn [21]. Chi tiết cách thực hiện word2vec với mô hình skip-gram [19] như sau.

Đầu tiên chúng ta cần xây dựng tập từ vựng từ các văn bản huấn luyện, ví dụ tập từ vựng gồm 10000 từ phân biệt. Để có thể đưa các từ vào mạng nơ ron huấn luyện, các từ cần được véc tơ hóa, mỗi từ sẽ được thể hiện bằng một véc tơ one-hot. Vector này sẽ có 10000 phần tử với mỗi phần tử thể hiện vị trí tương ứng của từ trong tập từ vựng. Ví dụ véc tơ one-hot cho từ “ants” sẽ có phần tử có giá trị bằng 1 tương ứng với vị trí của từ “ants” trong tập từ vựng, các vị trí khác có giá trị bằng 0. Kiến trúc mạng nơ ron được thể hiện trong hình 4.7.

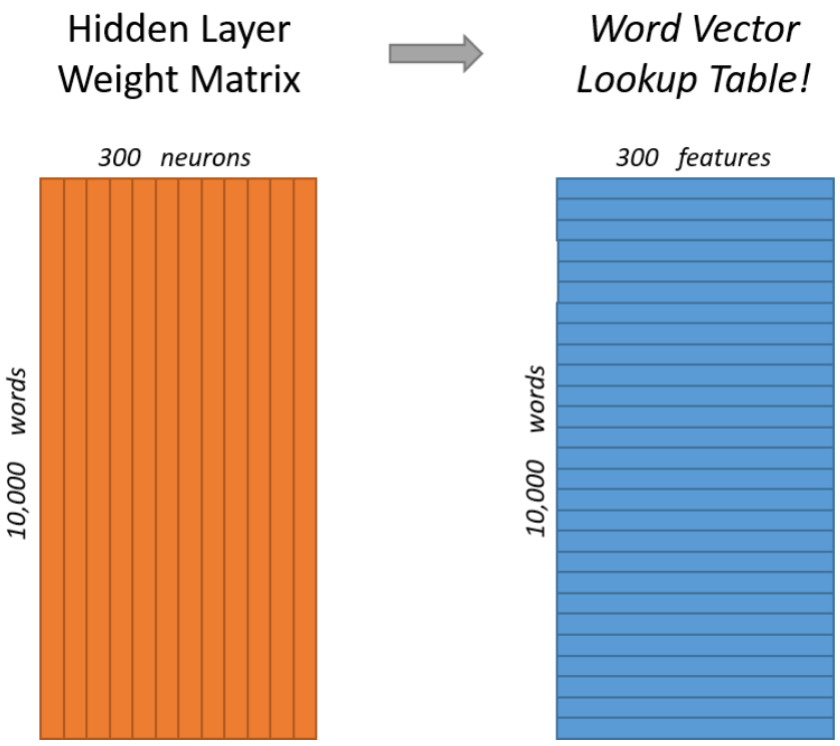


Hình 4.7. Kiến trúc mạng mô hình skip-gram

Lớp ẩn trong ví dụ hình 4.7 gồm 300 nơ ron không sử dụng một hàm kích hoạt nào nhưng đầu ra thì sử dụng một hàm softmax. Lớp ẩn được thể hiện bằng một ma trận trọng số gồm 10000 hàng (tương ứng với mỗi từ trong tập từ vựng) và 300 cột (tương ứng với mỗi nơ ron ẩn). Số nơ ron ẩn được gọi là số đặc trưng hay số chiều của word2vec là một siêu tham số có thể được tùy chỉnh tùy theo từng bài toán.

Các hàng của ma trận trọng số của lớp ẩn, thực chất chính là các véc tơ từ, đây chính là mục tiêu của word2vec. Với word2vec, chúng ta tiến hành huấn luyện một mạng nơ ron đơn giản với chỉ một lớp ẩn để tiến hành véc tơ hóa các từ trong tập từ vựng. Tuy nhiên, chúng ta không thực sự sử dụng kết quả đầu ra của mạng nơ ron sau khi huấn luyện, mà sẽ sử dụng trọng số của lớp ẩn. Ma trận trọng số của lớp ẩn giống như một bảng tìm kiếm các từ được thể hiện bằng các véc tơ từ tương ứng được minh họa như hình 4.8.

Với đầu vào là một từ được thể hiện bằng một véc tơ one-hot, việc đưa véc tơ này qua lớp ẩn về bản chất chính là việc tìm kiếm trên ma trận trọng số của lớp ẩn một véc tơ có số đặc trưng bằng số cột của ma trận trọng số.



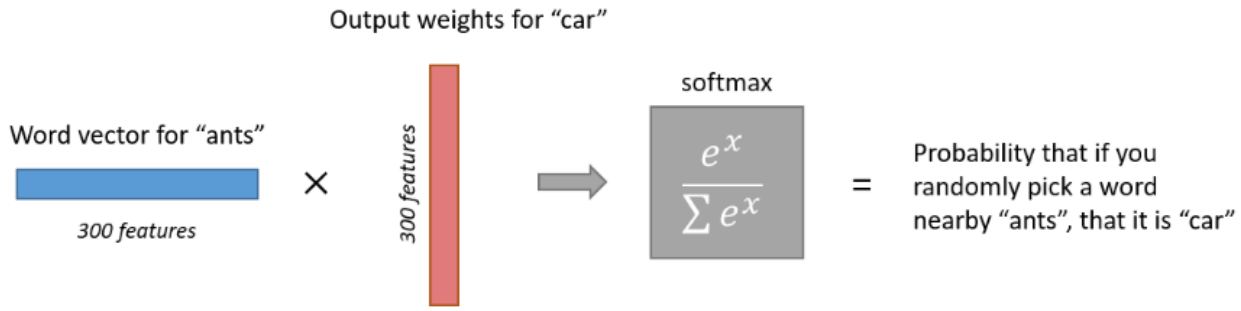
Hình 4.8. Ma trận trọng số lớp ẩn word2vec

Khi nhân một véc tơ one-hot 1x10000 với ma trận 10000x300 thì kết quả của phép nhân ma trận chỉ ảnh hưởng bởi hàng có phần tử 1 của véc tơ one-hot. Hình 4.9 minh họa kết quả nhân véc tơ one-hot với ma trận trọng số của lớp ẩn.

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Hình 4.9. Lớp ẩn hoạt động như một bảng tra cứu

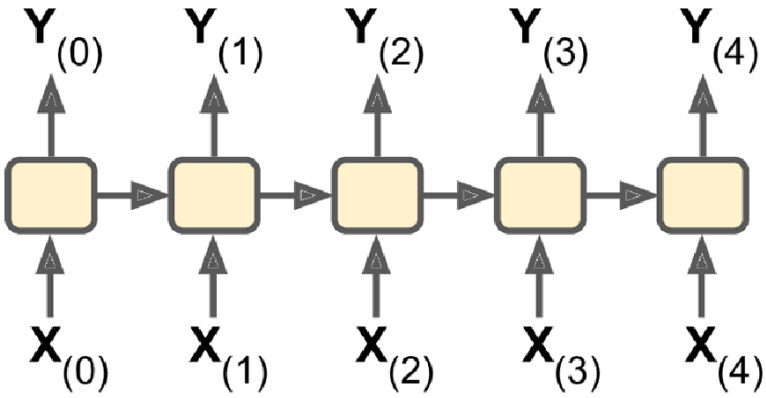
Đầu ra của word2vec là một bộ phân loại sử dụng hàm softmax. Hàm softmax cho kết quả là một giá trị thuộc khoảng 0 tới 1, chính là xác suất của mỗi đầu ra, tổng các giá trị này bằng 1. Hình 4.10 minh họa hoạt động của mô hình thể hiện xác suất từ “car” là từ lân cận từ “ants”.



Hình 4.10. Tương quan giữa hai từ thực hiện với word2vec

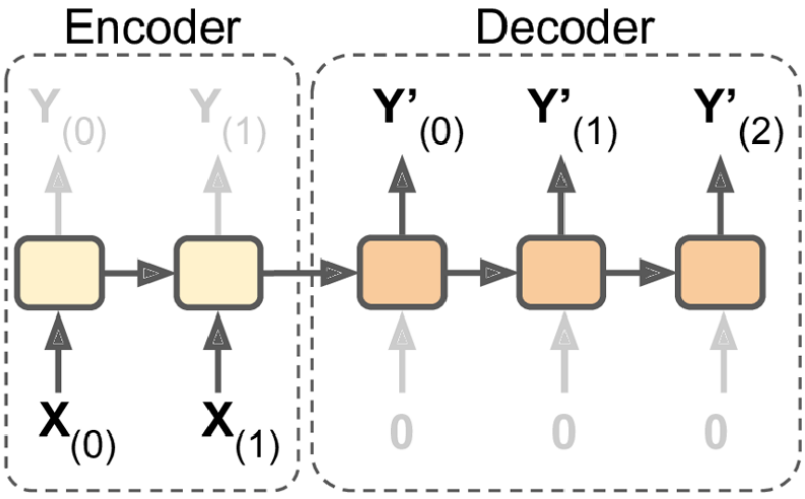
4.4. Xây dựng mô hình

Tư tưởng của bài toán tóm tắt văn bản theo hướng tóm lược là đưa ra văn bản tóm tắt là một chuỗi các từ (hay token) dựa trên chuỗi các từ của văn bản đầu vào, đây chính là mô hình chuỗi sang chuỗi (sequence-to-sequence). Mô hình chuỗi sang chuỗi được thể hiện như trong hình 4.11, trong đó các nút mạng RNN có thể lấy đầu vào là một chuỗi và sinh ra một chuỗi đầu ra [8].



Hình 4.11. Mô hình chuỗi sang chuỗi

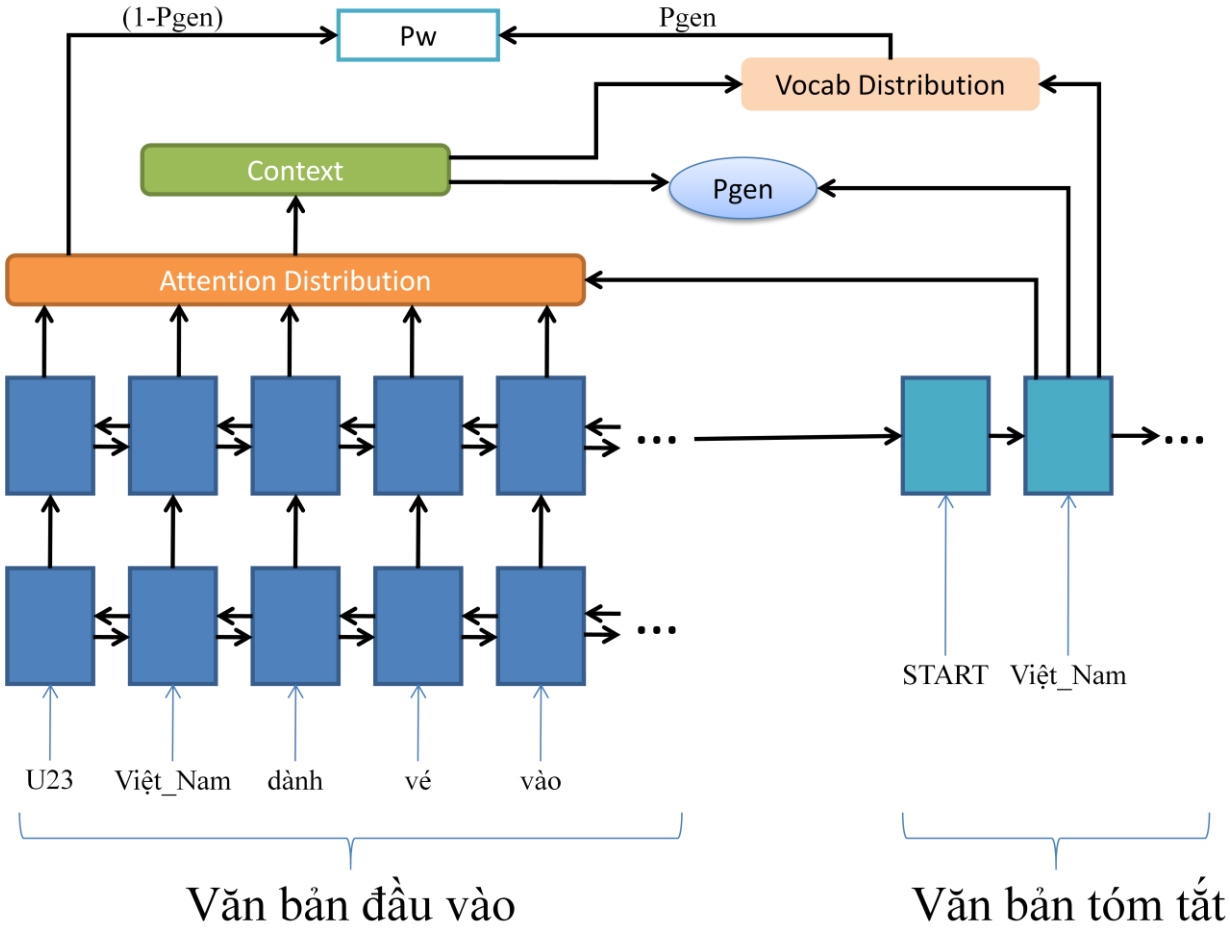
Mô hình chuỗi sang chuỗi có thể được xây dựng bằng kết hợp của hai mạng RNN, một là một mạng chuỗi sang véc tơ (sequence-to-vector) thường được gọi là bộ mã hóa (encoder), theo sau là một mạng véc tơ sang chuỗi (vector-to-sequence) thường được gọi là bộ giải mã (decoder). Hình 4.12. minh họa cho mô hình chuỗi sang chuỗi thực hiện bằng bộ mã hóa-giải mã [8]. Mô hình bộ mã hóa-giải mã được áp dụng thành công trong nhiều bài toán xử lý ngôn ngữ tự nhiên, trong đó đáng chú ý là các nghiên cứu về dịch máy [3 , [30].



Hình 4.12. Mô hình bộ mã hóa-giải mã

Nallapati và cộng sự [22] áp dụng mô hình bộ mã hóa cho bài toán tóm tắt văn bản tiếng Anh cho thấy kết quả khả thi của mô hình với bài toán tóm tắt văn bản tự động. Vì vậy, cách tiếp cận của chúng tôi để xây dựng mô hình cho bài toán tóm tắt văn bản tiếng Việt tự động là sử dụng mô hình chuỗi sang chuỗi thực hiện bằng bộ mã hóa-giải mã với các khối LSTM cho cả bộ mã hóa và bộ giải mã.

Bộ mã hóa được xây dựng từ 2 lớp mạng LSTM nạp chồng, mỗi nút mạng là một khối LSTM hai chiều (Bidirectional LSTM) kích thước 256. Bộ giải mã là một mạng LSTM với mỗi nút mạng là một khối LSTM một chiều (unidirectional LSTM). Kiến trúc mô hình chúng tôi xây dựng dựa trên kết quả nghiên cứu của tác giả See và cộng sự [28] và được thể hiện như hình 4.13.



Hình 4.13. Kiến trúc mô hình tóm tắt văn bản tiếng việt sử dụng LSTM

Các token của văn bản đầu vào được lần lượt đưa vào bộ mã hóa, sinh ra một chuỗi các trạng thái ẩn của bộ mã hóa. Word embedding được khởi tạo ngẫu nhiên theo phân phối chuẩn và được học để điều chỉnh các hệ số trong quá trình huấn luyện. Bộ giải mã nhận các word embedding của các từ ở thời điểm trước: trong quá trình huấn luyện chính là các từ của văn bản tóm tắt tham chiếu và trong quá trình chạy thì các từ ở thời điểm trước chính là các từ được sinh bởi bộ giải mã. Để bộ giải mã có thể học cách tự sinh các từ cho văn bản tóm tắt, chúng tôi sử dụng cơ chế chú ý (attention) giống như tác giả Bahdanau và cộng sự thực hiện [3]. Cơ chế attention dựa trên phân phối xác suất của các từ trong văn bản gốc, giúp bộ giải mã xác định được vị trí của từ sẽ được lựa chọn cho văn bản tóm tắt. Một véc tơ ngữ cảnh (context vector) được tính tại mỗi trạng thái bộ giải mã dựa trên các trạng thái ẩn của bộ mã hóa và trạng thái ẩn trước đó của bộ giải mã.

$$c_i = \sum_i a_i^t h_i \qquad (4.2)$$

Trong đó c_i là véc tơ ngữ cảnh, h_i là chuỗi trạng thái ẩn của bộ mã hóa, a_i^t là phân phối attention.

$$a_i^t = softmax(e^t) \quad (4.3)$$

Với $e_i^t = \tanh(W_h h_i + W_s s_t + bias)$

W_h, W_s, và bias là các tham số được điều chỉnh trong quá trình huấn luyện.

Véc tơ ngữ cảnh là một vector có kích thước cố định thể hiện những gì đã được đọc từ văn bản gốc, kết hợp với trạng thái ẩn của bộ giải mã để tính phân bố xác suất của một token trong tập từ vựng P_{vocab}.

Do thực tế việc sinh ra từ tiếp theo của văn bản tóm tăt có khả năng đối mặt với một từ không tìm thấy trong tập từ vựng (Out Of Vocabulary – OOV). Để xử lý vấn đề này, See và cộng sự [28] đề xuất cơ chế mạng con trỏ (pointer network) hoạt động giống như một bộ chuyển đổi cho phép bộ giải mã quyết định sinh một từ có trong tập từ vựng đưa vào văn bản tóm tắt hay là sao chép một từ từ văn bản đầu vào. Xác suất một từ được sinh trong văn bản tóm tắt được tính như sau:

$$p_{(w)} = p_{gen} * p_{vocab}(w) + (1 - p_{gen}) * \sum_{i:w_i=w} a_i^t \quad (4.4)$$

Trong đó:

$$p_{gen} = sigmoid(w_c c_t + w_s S_t + w_x x_t + b) \quad (4.5)$$

Với $p_{gen} \in [0,1]$ cho mỗi thời điểm t được tính từ véc tơ ngữ cảnh c_t, trạng thái của bộ giải mã S_t và đầu vào của bộ giải mã x_t; (w_c, w_s, w_x, b) là các tham số được học trong quá trình huấn luyện.

Trong công thức tính $p_{(w)}$, nếu một từ là từ không có trong tập từ vựng thì $p_{vocab}(w) = 0$, từ được lấy từ văn bản gốc đưa vào văn bản tóm tắt; và nếu từ đó không xuất hiện trong văn bản gốc thì $\sum_{i:w_i=w} a_i^t = 0$, từ được lấy từ tập từ vựng đưa vào văn bản tóm tắt.

CHƯƠNG 5: THỬ NGHIỆM VÀ ĐÁNH GIÁ

5.1. Môi trường thử nghiệm

Mô hình tóm tắt văn bản tiếng Việt tự động được xây dựng và thử nghiệm trên máy tính có cấu hình như sau:

- CPU: I7700 HQ @2.80 GHZ
- RAM: 16GB.
- GPU: NVIDIA GTX1050Ti, 4Gb Memory.
- Hệ điều hành Windows 10 Pro.
- Ngôn ngữ lập trình: Python trên trình biên dịch Python 3.6.1
- IDE: Spyder.

Các công cụ chính sử dụng:

- Framework: Google Tensorflow, phiên bản 1.4. Chức năng: Tensorflow cho phép xây dựng và thử nghiệm model học sâu một cách trực quan. Nó cung cấp các thư viện tích hợp cho phép cấu hình các tham số trong quá trình huấn luyện, áp dụng các công thức tính toán trên số học và ma trận, đồng thời hiển thị các kết quả bằng các biểu đồ, đồ thị.
- NLTK: NLTK là viết tắt của Natural Language Toolkit, đây là công cụ xử lý ngôn ngữ tự nhiên mạnh trên môi trường Python. Luận văn sử dụng NLTK để thực hiện tách từ đơn, phục vụ cho việc chuyển văn bản từ dạng thông thường (text) sang dạng nhị phân (binary).
- Newspaper3k: Thư viện mở có khả năng trích xuất văn bản từ website [17]. Luận văn sử dụng newspaper3k để xây dựng script thực hiện thu dữ liệu từ các trang tin tức trực tuyến Việt Nam.
- GetURL: Python script do tác giả thực hiện nhằm trích xuất các liên kết từ các trang tin tức trước khi sử dụng newspaper3k để trích xuất dữ liệu từ trang web.

- Pyvi: Thư viện Python để tách từ Tiếng Việt [31]. Luận văn sử dụng Pyvi để xây dựng tập từ điển và tách từ từ văn bản đầu vào.
- Strawberry-PERL: Công cụ đánh giá điểm ROUGE cho tóm tắt văn bản. Luận văn sử dụng strawberry-PERL kết hợp với thư viện pyrouge [10] để thực hiện đánh giá độ chính xác của văn bản tóm tắt sinh bởi mô hình.

5.2. Quá trình thử nghiệm

5.2.1. Huấn luyện

Trong quá trình huấn luyện, chúng tôi sử dụng phương pháp word2vec embedding [21] với số chiều (số đặc trưng) là 128, được khởi tạo ngẫu nhiên và được cập nhật trong quá trình huấn luyện. Bộ mã hóa và bộ giải mã được xây dựng từ các khối LSTM kích thước 256. Bộ mã hóa là một mạng hai lớp bidirectional LSTM nạp chồng và bộ giải mã là một mạng đơn unidirectional LSTM. Văn bản đầu vào được tách thành các token sử dụng công cụ Pyvi [31] và đưa vào bộ mã hóa. Đầu vào của bộ giải mã trong quá trình huấn luyện là kết hợp của trạng thái ẩn của bộ mã hóa và các token của văn bản tóm tắt tham chiếu. Chúng tôi sử dụng thuật toán tối ưu Adam [7] với learning rate là 0.001. Adam là viết tắt của adaptive moment estimation, đây là thuật toán thích nghi tốc độ học với khả năng tự điều chỉnh tốc độ học trong suốt quá trình huấn luyện. Nhờ khả năng này của thuật toán Adam, nó không cần thiết kết hợp thêm một phương thức điều chỉnh tốc độ học để tăng tốc độ hội tụ. Chính vì vậy, thuật toán tối ưu Adam được đánh giá là có hiệu quả tốt trong hầu hết các bài toán học sâu đặc biệt trong thị giác máy tính và xử lý ngôn ngữ tự nhiên [8].

Chúng tôi lựa chọn 20K từ phổ biến nhất trong tập dữ liệu làm tập từ vựng. Để giảm thời gian huấn luyện và sinh văn bản tóm tắt, văn bản đầu vào được giới hạn tối đa là 300 token và văn bản tóm tắt được giới hạn tối đa là 100 token. Quá trình huấn luyện và giải mã sử dụng TensorFlow phiên bản 1.4 có hỗ trợ GPU, trên GPU GTX1050Ti. Chúng tôi sử dụng batch size là 8. Quá trình sinh văn bản tóm tắt, chúng tôi áp dụng thuật toán beam search [26] với beam size là 5. Beam search là một thuật toán tham lam, được cải tiến từ thuật toán tìm kiếm theo chiều rộng. Tư tưởng của thuật toán beam search là xây dựng cây tìm kiếm như tìm kiếm theo chiều rộng, nhưng tại mỗi nút, nó thực hiện đánh giá để giữ lại một số ứng viên tốt nhất để tiếp tục quá trình tìm kiếm. Số ứng viên được giữ lại tại mỗi bước tìm kiếm của thuật toán beam search gọi là beam size.

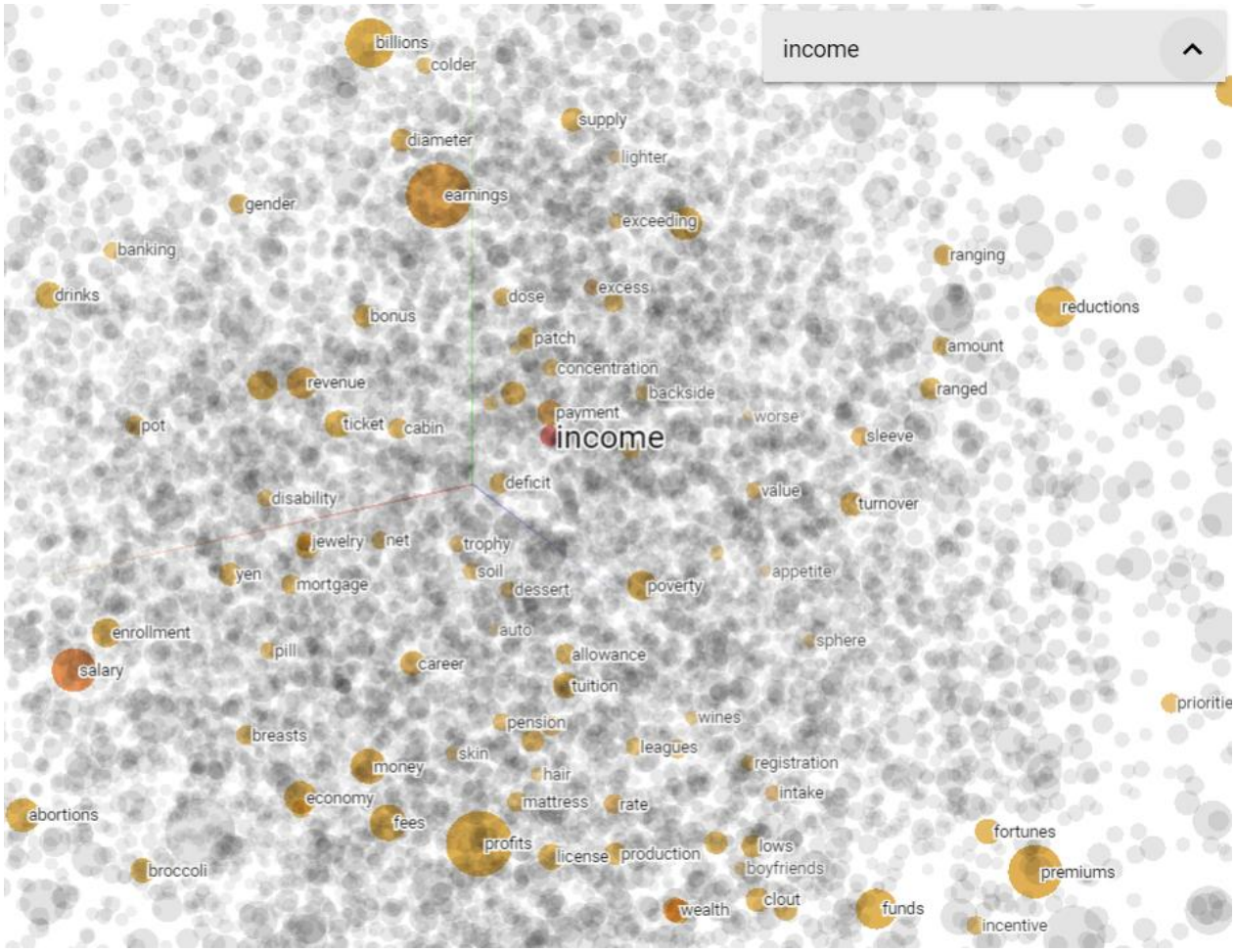
5.2.2. Thử nghiệm

5.2.2.1. Thử nghiệm 1.

Trong thử nghiệm thứ nhất, chúng tôi sử dụng tập dữ liệu là các bài báo CNN và Daily Mail [11]. Tập dữ liệu này gồm 287226 mẫu dành cho huấn luyện và 11490 mẫu dành cho kiểm thử mô hình huấn luyện. Các mẫu gồm hai thành phần: văn bản đầy đủ và văn bản tóm tắt tham chiếu.

Văn bản đầu vào được tách thành các token, giữ lại tối đa 300 token của văn bản gốc đưa vào bộ mã hóa và tối đa 100 token của văn bản tham chiếu đưa vào bộ giải mã. Các token được thực hiện véc tơ hóa bằng phương pháp word2vec [21] với số đặc trưng là 128.

Hình 5.1 thể hiện hình ảnh word embedding của tập dữ liệu được trực quan hóa bằng công cụ TensorBoard, một thư viện quan trọng của Tensorflow.

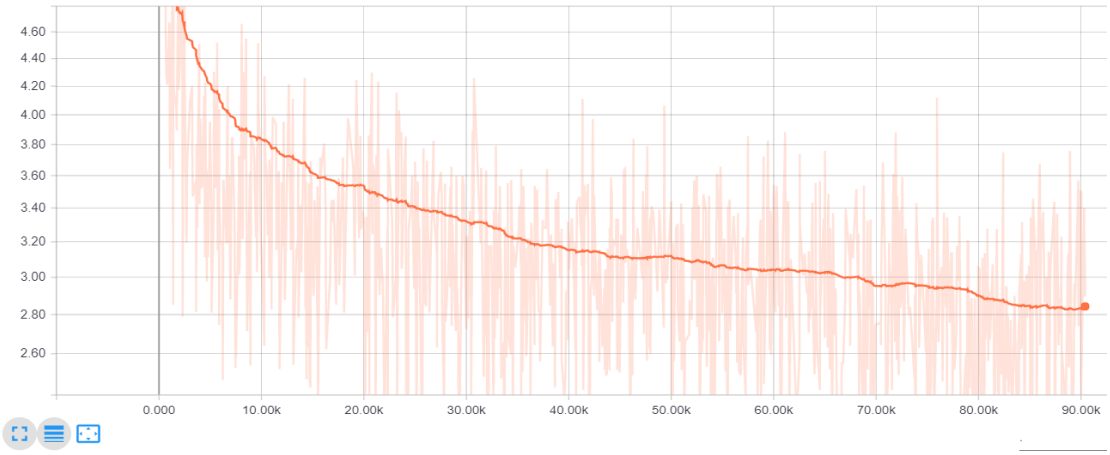


Hình 5.1. Minh họa kết quả word embedding

Word2vec cũng cho biết sự tương quan giữa các từ trong tập dữ liệu. Ví dụ trong hình 5.1, khi chọn một từ "income", word2vec hiển thị các từ tương quan với từ này. Hình 5.2 thể hiện các từ tương quan với từ "income" dựa trên khoảng cách cosine giữa các véc tơ đại diện của từ.

Hình 5.2. Tương quan giữa các từ với từ "income"

Quá trình huấn luyện mô hình với các tham số được mô tả trong mục 5.2.1, kết quả của hàm lỗi (loss) của quá trình huấn luyện mô hình được thể hiện trong hình 5.3.



Hình 5.3. Runing Average Loss

Để đánh giá độ chính xác của mô hình, chúng tôi tiến hành chạy mô hình với bộ dữ liệu test gồm 11490 mẫu, và sử dụng phương pháp ROUGE [16]. ROUGE viết tắt của Recall Oriented Understudy for Gist Evaluation, đây là phương pháp được coi là chuẩn mực và được sử dụng rộng rãi trong các nghiên cứu về tóm tắt văn bản. Điểm ROUGE-N được xác định như sau:

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{ReferenceSummary\ \}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummary\ \}} \sum_{gram_n \in S} Count(gram_n)}$$

Trong đó $Count_{match}(gram_n)$ là số lượng n-grams lớn nhất có trong văn bản tóm tắt sinh ra và văn bản tóm tắt tham chiếu

$Count(gram_n)$ là số lượng n-grams có trong văn bản tóm tắt tham chiếu.

Độ chính xác của mô hình với tập dữ liệu test được thể hiện trong bảng 5.1, chúng tôi tính toán điểm ROUGE sử dụng công cụ pyrouge [10].

Bảng 5.1. Đánh giá độ chính xác trên tập 11490 bài báo tiếng Anh

	ROUGE-1	ROUGE-2	ROUGE-L
Precision	37.38	16.02	33.99
Recall	36.76	15.62	33.39
F-score	35.90	15.30	32.62

ROUGE-1 và ROUGE-2 được đánh giá dựa trên số 1-gram và 2-gram cùng có trong văn bản tóm tắt do mô hình sinh ra và văn bản tóm tắt tham chiếu. Và ROUGE-L được đánh giá dựa trên chuỗi chung dài nhất có trong văn bản tóm tắt sinh ra và văn bản tóm tắt tham chiếu, đây là tham số quan trọng để đánh giá chất lượng của mô hình sinh tóm tắt. Điểm ROUGE-L F-score của mô hình trên tập dữ liệu CNN/DailyMail là 32.62. Bảng 5.2 thể hiện kết quả đối sánh giữa mô hình chúng tôi xây dựng và các mô hình đã công bố của tác giả Nallapati [22] và tác giả See [28] .

Bảng 5.2. So sánh một số mô hình học sâu cho tóm tắt văn bản tóm lược

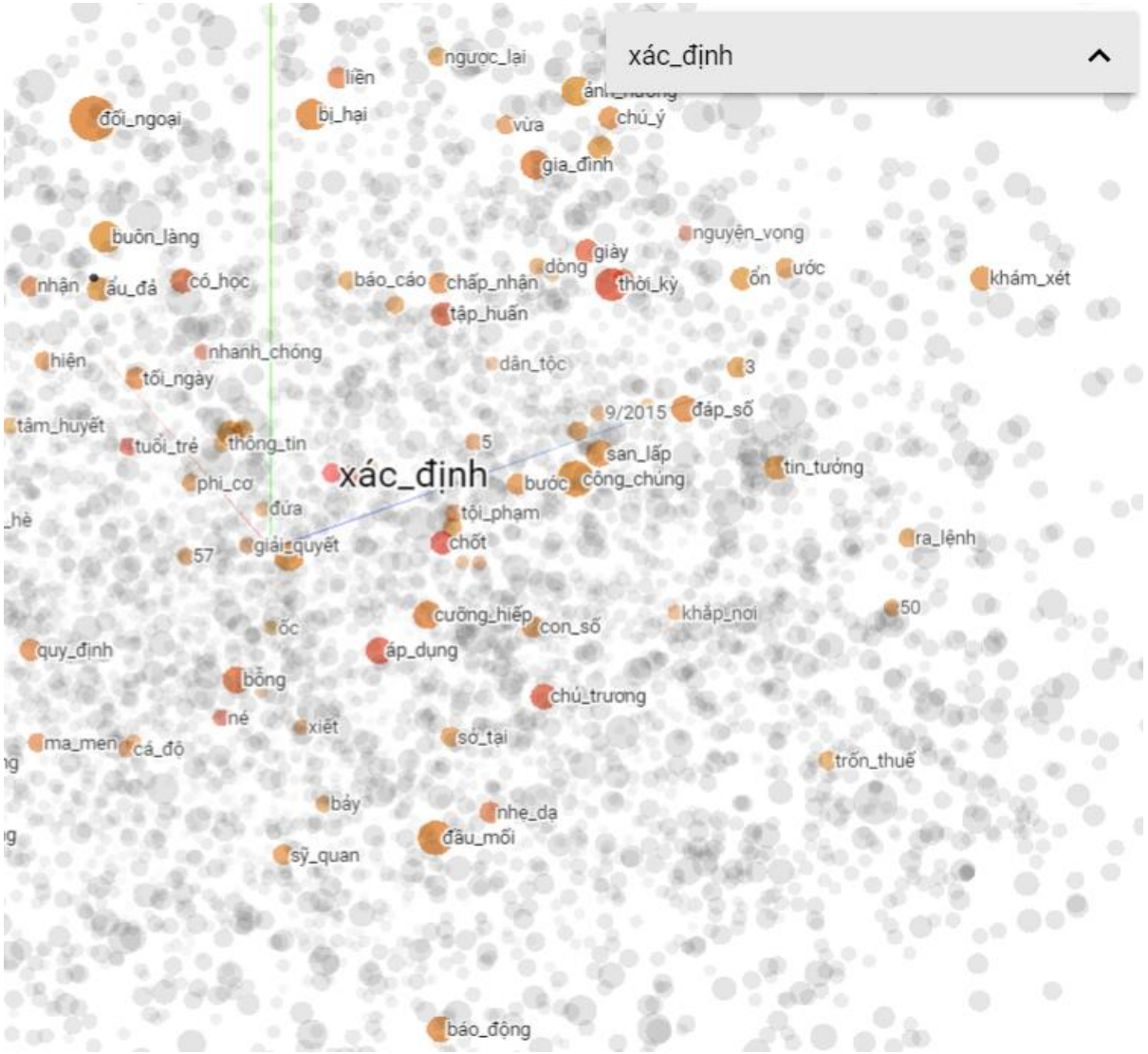
Tham số	Nallapati et al	See et al	Our Model
Mẫu huấn luyện	287226	287226	287226
Mẫu kiểm thử	11490	11490	11490
Số lượng từ vựng	150,000	50,000	20,000
Số đặc trưng word2vec	100	128	128
Số nơ ron ẩn	RNN 200	Single LSTM 256	2-Stacked LSTM 256
Thuật toán tối ưu	Adadelta	Adagrad	Adam
Tốc độ học	0.001	0.15	0.001
Huấn luyện	600K iterations ~ 7 days	230K iterations ~ 3days+4 hours	90.3K iterations ~ 1day+17 hours
GPU	Tesla K40	Tesla K40m	GTX 1050Ti
Beam size	5	4	5
ROUGE-L F-Score	**29.47**	**36.38**	**32.62**

Bảng 5.2 thể hiện kết quả của 3 mô hình thực hiện tóm tắt văn bản tự động theo hướng tóm lược được huấn luyện và đánh giá trên cùng bộ dữ liệu CNN/DailyMail. Dựa trên điểm ROUGE-L F-score, có thể nhận xét rằng mô hình của chúng tôi xây dựng cho kết quả tốt hơn mô hình của tác giả Nallapati trên bộ dữ liệu này. So với mô hình của tác giả See, mô hình của chúng tôi xây dựng cho điểm ROUGE-L F-score thấp hơn trên bộ dữ liệu CNN/Daily Mail, tuy nhiên, mô hình chúng tôi xây dựng được huấn luyện với số lượng từ vựng ít hơn (20,000 từ so với 50,000 từ) và trong thời gian ngắn hơn trên phần cứng cấu hình thấp hơn đáng kể so với tác giả See; do đó, nhìn chung độ chính xác của mô hình là chấp nhận được.

5.2.2.2. Thử nghiệm 2.

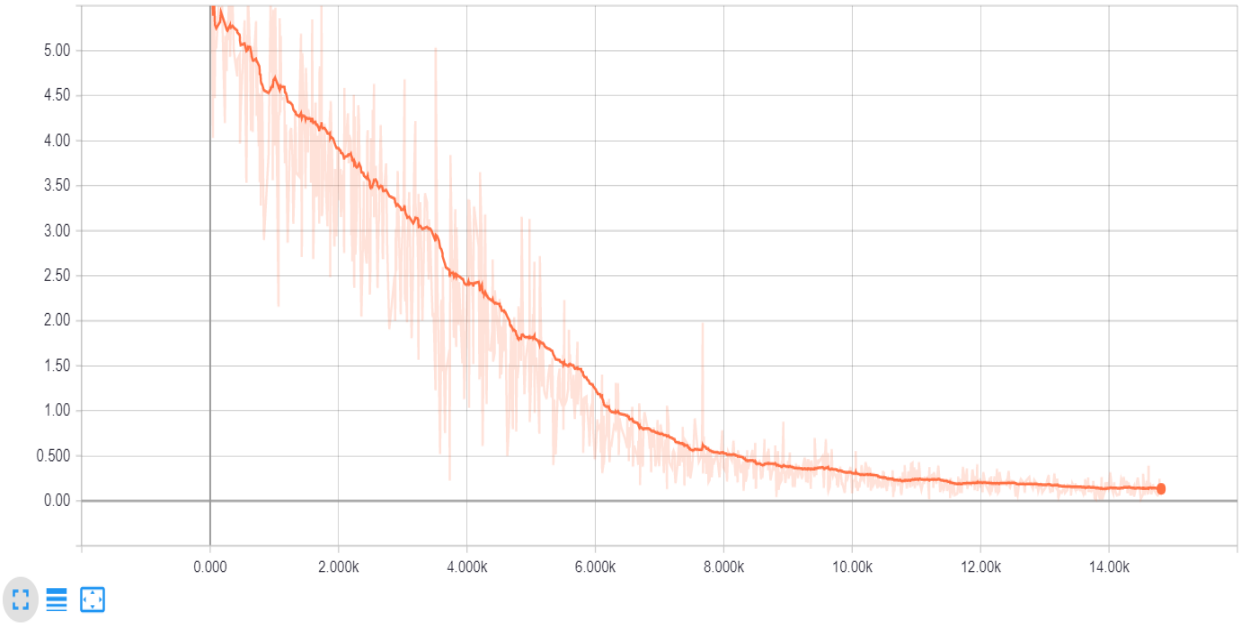
Trong thử nghiệm thứ 2, chúng tôi tiến hành thử nghiệm mô hình với tập dữ liệu tiếng việt. Hiện tại chưa có bộ dữ liệu cho tóm tắt văn bản tiếng Việt nào được công bố, nên chúng tôi tiến hành thu thập các mẫu là các bài báo trên các website tin tức trực tuyến. Sau khi có các mẫu dữ liệu, chúng tôi tiến hành tiền xử lý dữ liệu và chuyển về dạng nhị phân. Tập dữ liệu tiếng Việt thử nghiệm gồm 1120 bài báo cho huấn luyện và 316 bài báo cho kiểm thử.

Các văn bản đầu vào cũng được tách thành các token. Chúng tôi sử dụng thư viện Pyvi để thực hiện tokenize. Các token cũng được véc tơ hóa bằng phương pháp word2vec trước khi được đưa vào mô hình. Hình 5.4 minh họa kết quả word embedding cho tập dữ liệu trong thử nghiệm này.



Hình 5.4. Word2vec cho tập dữ liệu tiếng Việt

Kết quả hàm lỗi của quá trình huấn luyện với bộ dữ liệu 1120 bài báo tiếng việt được thể hiện trong hình 5.5.



Hình 5.5. Running Avarage Loss với bộ dữ liệu tiếng Việt

Độ chính xác của mô hình với tập dữ liệu gồm 316 bài báo tiếng việt cũng được thực hiện bằng phương pháp ROUGE và được thể hiện trong bảng 5.3.

Bảng 5.3. Đánh giá độ chính xác trên tập 316 bài báo tiếng Việt

	ROUGE-1	ROUGE-2	ROUGE-L
Precision	50.53	14.39	32.60
Recall	52.92	14.83	33.79
F-score	49.80	14.08	31.93

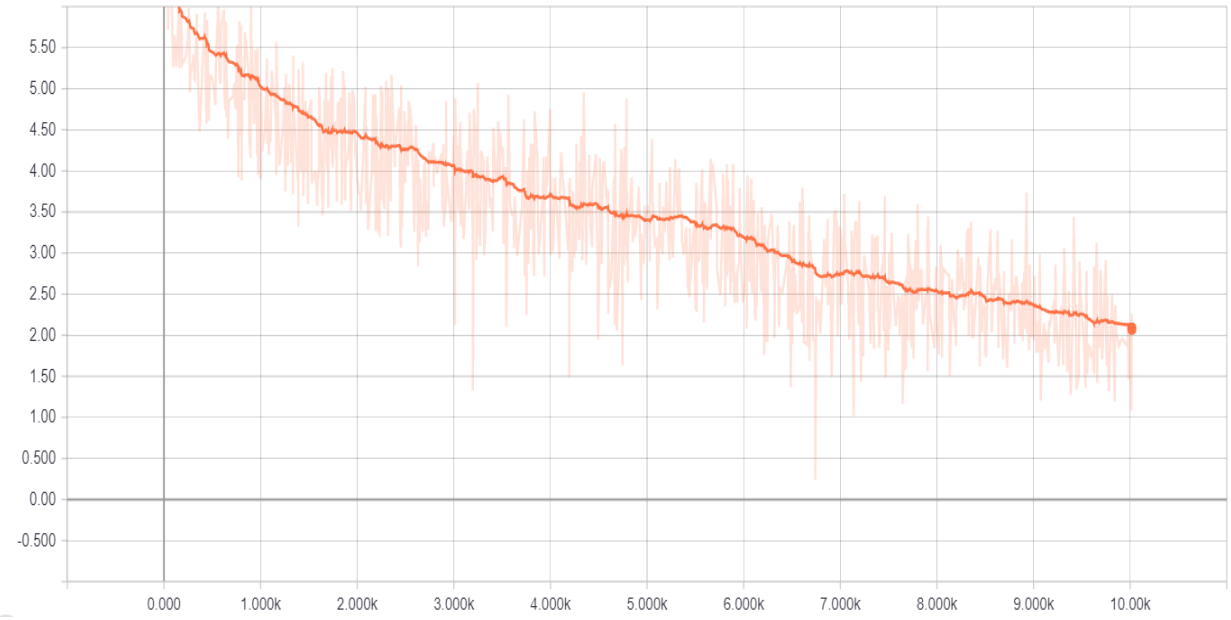
Trong thử nghiệm này, mô hình được huấn luyện với 1120 mẫu và kiểm thử với 316 mẫu, số lượng mẫu huấn luyện là ít. Tuy nhiên, tập dữ liệu chúng tôi sử dụng là các bài báo được thu thập ngẫu nhiên trên trang báo tin tức (báo Tiền Phong) không giới hạn chủ đề bài viết, văn bản tóm tắt do mô hình sinh chỉ giới hạn độ dài

trong khoảng từ 35 tới 100 token, không bị ràng buộc theo một định dạng cố định nào. Điểm ROUGE-L F-score thu được khá cao.

5.2.2.3. Thử nghiệm 3.

Trong thử nghiệm thứ 3, chúng tôi tiến hành thử nghiệm mô hình với tập dữ liệu tiếng Việt gồm 4000 bài báo cho huấn luyện và 500 bài báo cho kiểm thử. Các siêu tham số khác được thiết lập giống như trong hai thử nghiệm trước, tuy nhiên trong thử nghiệm này chúng tôi sử dụng 25000 từ cho tập từ vựng và kích thước batch size là 5.

Mô hình được huấn luyện qua 10000 bước lặp, kết quả hàm lỗi được thể hiện trong hình 5.6.



Hình 5.6. Running Avarage Loss với bộ dữ liệu 4000 bài báo tiếng Việt

Độ chính xác của mô hình đánh giá bằng tập thử nghiệm 500 bài báo tiếng Việt được thể hiện trong bảng 5.4.

Bảng 5.4. Đánh giá độ chính xác trên tập 500 bài báo tiếng Việt

	ROUGE-1	ROUGE-2	ROUGE-L
Precision	50.93	17.44	34.00
Recall	55.45	19.01	36.89
F-score	51.32	17.57	34.17

Từ kết quả tính toán điểm ROUGE-L F-score của mô hình trong bảng 5.4 có thể thấy rằng, chất lượng của mô hình được cải thiện khi được huấn luyện với nhiều mẫu hơn và sử dụng tập từ vựng với nhiều từ hơn.

5.2.2.4. Thử nghiệm 4.

Từ hai thử nghiệm trước với tiếng Việt, chúng tôi nhận thấy rằng mô hình cho kết quả tốt hơn ở thử nghiệm số 3 khi được huấn luyện với số lượng mẫu nhiều hơn và số lượng từ trong tập từ vựng nhiều hơn. Để kiểm chứng điều này, chúng tôi thử nghiệm mô hình với bốn tập dữ liệu có số lượng mẫu huấn luyện và số từ sử dụng trong tập từ vựng tăng dần như thể hiện trong bảng 5.5.

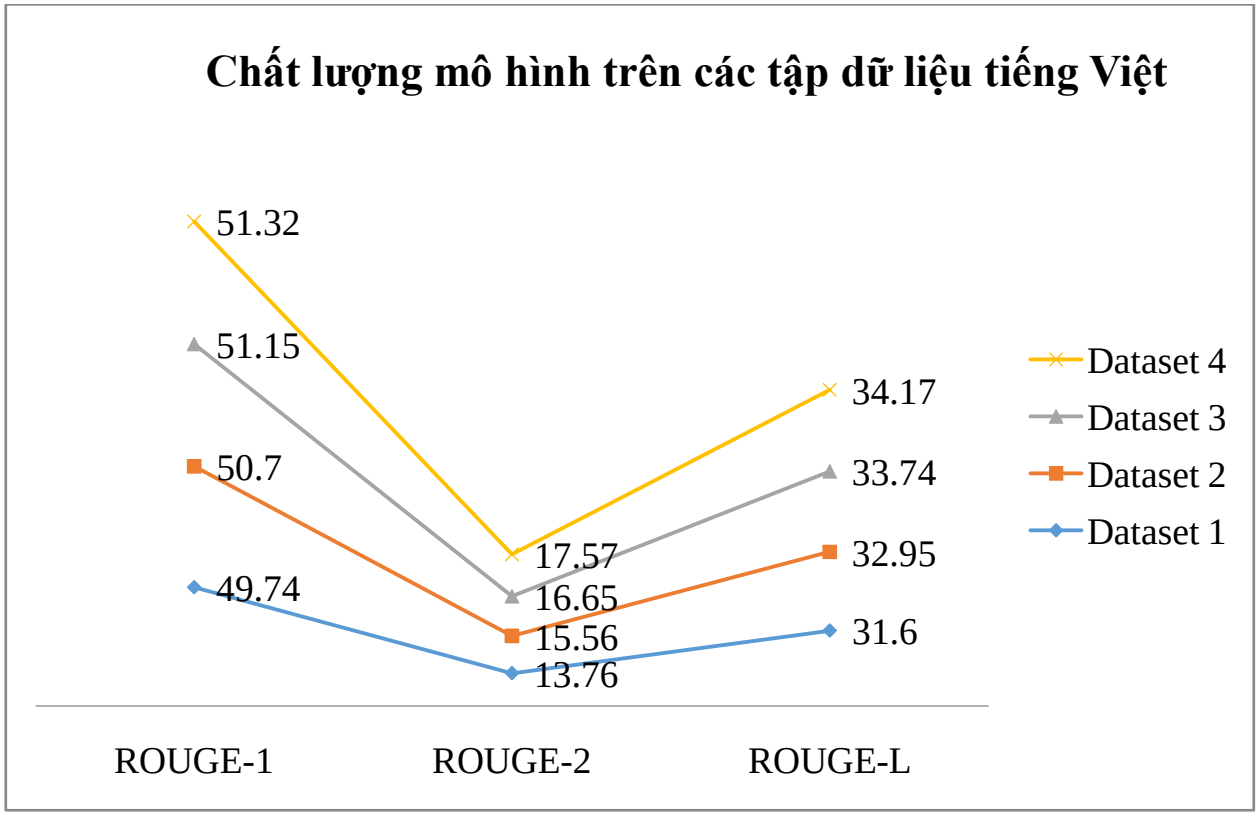
Bảng 5.5. Thử nghiệm chất lượng mô hình trên các tập dữ liệu tiếng Việt

Tham số	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Mẫu huấn luyện	1120	2000	3000	4000
Mẫu kiểm thử	500	500	500	500
Số lượng từ vựng	10000	15000	20000	25000
Số đặc trưng word2vec	128	128	128	128

Số nơ ron ẩn LSTM	256	256	256	256
Thuật toán tối ưu	Adam	Adam	Adam	Adam
Tốc độ học	0.001	0.001	0.001	0.001
Beam size	5	5	5	5

Tập dữ liệu kiểm thử là giống nhau trong cả bốn bộ dữ liệu dùng để so sánh chất lượng mô hình.

Hình 5.7 thể hiện điểm F-score được tính theo phương pháp ROUGE của mô hình trên các tập dữ liệu test gồm 500 bài báo tiếng Việt.



Hình 5.7. So sánh chất lượng mô hình trên các tập dữ liệu tiếng Việt

Từ kết quả thu được từ hình 5.7 có thể thấy rằng, chất lượng của mô hình được cải thiện khi được huấn luyện với nhiều mẫu hơn và sử dụng tập từ vựng phong phú

hơn. Tuy nhiên hiện tại do hạn chế về khả năng tính toán của môi trường phần cứng, chúng tôi đã không thể thực hiện thử nghiệm với tập từ vựng gồm nhiều từ hơn nữa, nhưng chúng tôi tin rằng, với việc đa dạng hóa tập dữ liệu huấn luyện và tăng chất lượng tập từ vựng, bài toán tóm tắt văn bản tự động theo hướng tóm lược sẽ cho kết quả rất khả quan.

Kết quả một số mẫu kiểm thử cho mô hình được thể hiện trong phần tiếp theo của luận văn, trong đó Model 1 là kết quả huấn luyện mô hình với Dataset 1, model 2 là là kết quả huấn luyện mô hình với Dataset 2, tương tự với model 3 và model 4.

Bài báo: Sáng nay 18/11 , tại buổi tiếp xúc cử tri trước kỳ họp HĐND TP Hà Nội , nhiều cử tri quận Cầu Giấy đã bày tỏ bức xúc và lo lắng trước tình trạng buông lỏng lỏng quản lý của hoạt động kinh doanh karaoke dẫn đến việc xảy ra các vụ cháy gây thiệt hại lớn . Cử tri Vũ Thị Liên cho rằng , trong thời gian vừa qua trên địa bàn thành phố xảy ra một số vụ cháy quán karaoke , có vụ lớn , vụ nhỏ . Song ít nhiều đã gây hoang mang , lo sợ , bất an cho nhân dân . Điển hình là vụ cháy quán karaoke số 68 Trần Thái Tông vừa qua làm 13 người thiệt mạng . Đây là vụ cháy kinh hoàng và đau xót , gây hậu quả rất lớn về người và của . Sự việc còn cho thấy các cấp lãnh đạo và các ngành chủ quan đã buông lỏng quản lý trong hoạt động kinh doanh này , cử tri Vũ Thị Liên bức xúc . Theo cử tri Vũ Thị Liên vụ cháy quán karaoke số 68 đường Trần Thái Tông còn phản ánh việc tuyên truyền , tập huấn cho các cơ sở karaoke và các ngành hàng khác trên địa bàn còn chủ quan , chưa bài bản , chế tài xử lý chưa nghiêm khắc , chỉ đến khi sự việc xảy ra gây hậu quả rồi mới vào cuộc quyết liệt . Từ vụ việc , cử tri đề nghị thành phố phải có kế hoạch rõ ràng trong việc quản lý karaoke trên địa bàn cũng như các hoạt động kinh doanh khác . Ông Dương Cao Thanh trả lời các ý kiến của cử tri . Trả lời các kiến nghị của cử tri , ông Dương Cao Thanh , Chủ tịch UBND quận Cầu Giấy cho rằng , đối với quận xảy ra vụ cháy ngày 1/11/2016 tại 68 Trần Thái Tông trong lịch sử 19 năm thành lập quận đến nay là một sự cố nặng nề nhất cũng như đối với thành phố , bởi chết rất nhiều người , thiệt hại rất nhiều tài sản . Ngay sau sự việc xảy ra , chúng tôi đã cùng lãnh đạo thành phố tập trung khắc phục sự cố cũng như hỗ trợ an táng các nạn nhân , ông Thanh nói . Theo ông Thanh , qua kiểm tra , cơ sở kinh doanh karaoke phải có 5 loại giấy tờ gồm đăng ký kinh doanh . thẩm định về phòng cháy chữa cháy . Giấy chứng nhận đủ điều kiện về phòng cháy chữa cháy . Sau đó Công an quận cấp giấy chứng nhận đảm bảo an ninh trật tự , khi có giấy này thì UBND quận mới cấp giấy phép kinh doanh . Tuy nhiên khi kiểm tra

cơ sở này thì mới có 2 loại giấy tờ là đăng ký kinh doanh và thẩm định thiết kế . còn 3 loại giấy tờ quan trọng khác thì chưa có . Quá trình kiểm tra trong vòng chưa đầy 20 ngày quận đã 3 lần kiểm tra , 1 lần bắt cam kết chỉ khi nào có đầy đủ giấy phép thì mới đưa vào hoạt động kinh doanh . Tuy nhiên đến 1/11 , vừa cho khách vào lại vừa sửa chữa ở tầng 2 cho nên phát cháy . Lãnh đạo quận Cầu Giấy cũng cho hay , qua kiểm tra 88 cơ sở kinh doanh karaoke có giấy phép trên địa bàn quận thì 3 cơ sở đã dừng , còn lại 85 cơ sở đều có vấn đề phòng cháy chữa cháy . Quan điểm của chúng tôi là cho tạm dừng tất cả các cơ sở vi phạm không đủ điều kiện về thoát nạn , cứu nạn rồi các vấn đề liên quan đến phòng cháy , chữa cháy , cũng như là vật liệu . Đây không phải mất bò mới lo làm chuồng như trước sự việc xảy ra thì thứ nhất phải ngăn chặn , thứ hai phải triển khai các biện pháp để đảm bảo an toàn tính mạng , cũng như tài sản của nhân dân . Đồng thời cũng là hồi chuông cảnh tỉnh với chúng ta về công tác phòng cháy chữa cháy , ông Thanh nói . Lãnh đạo quận Cầu Giấy cũng cho hay , hiện đang triển khai quyết liệt việc chấn chỉnh , yêu cầu các quán karaoke tháo dỡ biển quảng cáo sai phép , che chắn lối thoát hiểm , ảnh hưởng khả năng tiếp cận của lực lượng PCCC . Đồng thời , triển khai sang các nhà nghỉ , dỡ tất cả các biển hiệu quảng cáo trên 20m2 che khuất tầm nhìn của tất cả các cơ sở kinh doanh , kể cả ngân hàng hay các cơ sở massage . Đồng thời kiểm tra cả những cơ sở kinh doanh gas , những cơ sở nào nằm trong lòng khu dân cư thì phải di dời . Chúng tôi cũng đang triển khai các nội dung theo chỉ đạo của thành phố , của quận Uỷ trong việc kiểm điểm trách nhiệm tập thể , cá nhân liên quan đến vụ cháy , ông Thanh nhấn mạnh .
Mẫu tóm tắt: Ông Dương Cao Thanh , Chủ tịch UBND quận Cầu Giấy cho biết , sau vụ cháy quán karaoke khiến 13 người tử vong , cơ quan chức năng đã yêu cầu các quán karaoke tháo dỡ biển quảng cáo che chắn lối thoát hiểm , ảnh hưởng khả năng tiếp cận của lực lượng PCCC . Đồng thời , triển khai sang các nhà nghỉ , cơ sở massage hay các ngân hàng .
Model 1: [UNK] [UNK] quận Cầu [UNK] đang điều_tra đề_nghị xô_xát xảy ra nhằm biết , đề_nghị ngành karaoke karaoke . [UNK] là cho cử_tri đề_nghị người lãnh_đạo , tập_trung cử_tri đình_chỉ cơn địa_bàn , cử_tri 13 người quản_lý nhằm thúc_đẩy quản_lý các nạn_nhân m3 lớn .
Model 2: cử_tri đề_nghị có 3 ngành_hàng về quản_lý của hoạt_động kinh_doanh karaoke , cử_tri đề_nghị thành_phố phải tham_gia sự_cố gây thiệt_hại lớn cho một người và chết trên địa_bàn thành_phố xảy ra tối thành_phố quận Cầu Giấy đã

trao lỏng gây thiệt_hại .
Model 3: [UNK] cơ_sở quán karaoke và các cơ_sở karaoke và các ngành_hàng khác trên địa_bàn còn chủ_quan , chưa bài_bản , chế_tài xử_lý chưa nghiêm_khắc , chỉ đến khi sự_việc xảy ra các vụ cháy gây thiệt_hại .
Model 4: tập_huấn cho các cơ_sở karaoke và các ngành_hàng khác trên địa_bàn còn chủ_quan , chưa bài_bản , chế_tài xử_lý chưa nghiêm_khắc , chỉ đến khi sự_việc xảy ra các vụ cháy gây thiệt_hại lớn .

Bài báo: Mới đây , trên mạng xã hội xuất hiện đoạn clip ghi lại cảnh tai nạn giao thông nghiêm trọng , khiến người xem lạnh sống lưng . Theo người đăng tải clip , vụ tai nạn xảy ra vào tối 11/11 , trên địa bàn huyện Thống Nhất , Đồng Nai . Player Loading . xảy ra vào đêm . 11/11 , tại KM 12+350 trên QL20 đoạn qua xã Gia Tân 3 , huyện Thống Nhất , Đồng Nai . Người điều khiển xe máy sang đường là anh Vũ Quỳnh Như . Còn người điều khiển xe máy với tốc độ nhanh là ông Phạm Trí Đức . Hiện vụ việc đang được Công an huyện Thống Nhất điều tra xử lý theo thẩm quyền . Qua đoạn clip cho thấy , vụ tai nạn xảy ra trên đoạn đường khá vắng vẻ , thời điểm đó , 1 chiếc xe máy lao nhanh với tốc độ kinh hoàng đã đâm phải 1 chiếc xe máy khác do một thanh niên điều khiển đang sang đường . Sau cú đâm , người thanh niên sang đường ngã văng xuống đất , trong khi người điều khiển xe phóng với tốc độ cao cùng phương tiện thì trượt dài hàng chục mét trên mặt đường , tạo ra tia lửa bắn tung tóe . Vụ tai nạn đã làm cả 2 bị thương nặng , được người đi đường nhanh chóng đưa vào bệnh viện cấp cứu . Tuy nhiên do thương tích quá nặng , người đàn ông điều khiển xe phóng nhanh đã tử vong tại bệnh viện . Trong khi đó , chiều 12/11 , lãnh đạo Phòng CSGT đường bộ đường sắt Công an tỉnh Đồng Nai xác nhận với báo Dân Trí , trên địa bàn huyện Thống Nhất , tỉnh Đồng Nai đoạn qua QL20 xảy ra vụ tai nạn giao thông nghiêm trọng làm 2 người thương vong .
Mẫu tóm tắt: Nam thanh niên điều khiển xe máy qua đường thì bất ngờ bị chiếc xe máy khác chạy với tốc độ kinh hoàng đâm trúng . Sau cú đâm , người tông xe và phương tiện bị kéo lê hàng chục mét , toé lửa trong đêm .
Model 1: [UNK] tờ [UNK] , một nhóm chuyến nặng , gã đàn_ông điều_khiển bệnh_viện , bị lực_lượng lại cảnh tai_nạn giao_thông . Trong đó , trên [UNK] [UNK] [UNK] , lãnh_đạo Phòng tỉnh Đồng_Nai , tỉnh Đồng_Nai đã yêu_cầu , [UNK] bị_thương và đã xảy ra tại bệnh_viện .

Model 2: [UNK] chưa được đoạt xong của người thanh_niên sang đường sang đường phóng nhanh rồi 1 chiếc xe_máy rồi đường đường . [UNK] , sau nhiều người dân bị tai_nạn dã_man giữa đâm chết .
Model 3: Qua đoạn clip điều_khiển xe_máy sang đường là điều_khiển xe_máy với người điều_khiển xe phóng với tốc_độ cao cùng phương_tiện thì trượt dài hàng chục mét trên đất , trong khi người điều_khiển xe phóng với tốc_độ cao cùng phương_tiện – huyện Thống_Nhất , Đồng_Nai .
Model 4: người điều_khiển xe phóng với tốc_độ cao cùng phương_tiện thì trượt dài hàng chục mét trên mặt_đường qua xã Gia_Tân 3 , huyện Thống_Nhất , tỉnh Đồng_Nai , khiến nạn_nhân phải nhập_viện cấp_cứu .

Từ kết quả sinh tóm tắt của các model cho hai ví dụ trên có thể thấy rằng, model 4 có thể sinh văn bản tóm tắt tốt hơn, dễ hiểu hơn 3 model còn lại, văn bản tóm tắt sinh ra không phải là sao chép nguyên vẹn câu trong văn bản gốc mà có sự chọn lựa và ghép giữa các câu.

Đặc biệt trong ví dụ thứ 2, model 4 đã có khả năng sinh ra từ không có trong bài báo gốc đưa vào văn bản tóm tắt, đó là từ “nạn nhân” và từ “nhập viện”, kết quả cho thấy việc áp dụng mô hình LSTM cho bài toán tóm tắt văn bản tự động theo hướng tóm lược có thể cho kết quả khả quan, có khả năng tạo ra văn bản tóm tắt gần giống với cách con người thực hiện tóm tắt.

KẾT LUẬN

Những vấn đề đã được giải quyết trong luận văn

Luận văn đã tiến hành nghiên cứu giải quyết bài toán tóm tắt văn bản tự động, tập trung vào tóm tắt văn bản theo hướng tóm lược (abstractive summarization). Bài toán này được đánh giá có độ phức tạp cao và có thể làm cơ sở cho nhiều ứng dụng thực tế. Phương pháp giải quyết của luận văn tập trung vào xây dựng mô hình học sâu dựa trên mạng Long-Short Term Memory (LSTM).

Dựa trên các nghiên cứu về các mô hình mạng LSTM, các mô hình chuỗi sang chuỗi (sequence-to-sequence), các kỹ thuật vec tơ hóa từ và văn bản, luận văn đã xây dựng một kiến trúc mô hình học sâu sử dụng LSTM cho bài toán tóm tắt văn bản tự động với các tham số được tối ưu hóa cho việc huấn luyện và thử nghiệm trên máy tính cá nhân.

Luận văn cũng đã xây dựng tập dữ liệu cho tóm tắt văn bản tiếng Việt, sẵn sàng chia sẻ cho mục đích nghiên cứu và áp dụng trong tóm tắt văn bản tiếng Việt. Bộ dữ liệu gồm dữ liệu thô và dữ liệu đã được xử lý về dạng nhị phân.

Luận văn cũng đã thử nghiệm mô hình đã xây dựng với dữ liệu tiếng Anh và tiếng Việt và đánh giá bằng phương pháp ROUGE. Thử nghiệm với dữ liệu tiếng Việt về tin tức từ báo Tiền Phong và một số báo khác cho kết quả khả quan.

Định hướng nghiên cứu trong tương lai

Để tăng độ chính xác cho mô hình, một điều kiện quan trọng là xây dựng tập dữ liệu đầu vào word2vec chất lượng hơn, thể hiện chính xác hơn sự tương quan, mối liên hệ giữa các từ, các token. Do đó, việc xây dựng tập dữ liệu lớn và phong phú về chủ đề, đa dạng về mặt từ vựng là rất cần thiết cho mô hình tóm tắt văn bản tiếng Việt.

TÀI LIỆU THAM KHẢO

Tiếng Anh

[1]. Alex M. (2015), Word2Vec Tutorial Part I: The Skip-gram Model. *Retrieved from http://mccormickml.com/2016/04/27/word2vec-resources/#alex-minnaars-tutorials.*

[2]. Andrew T., Yohannes T., David H., and Hugh E.W. (2007), "Fast generation of result snippets in web search", *In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 127-134.

[3]. Bahdanau D., Cho K., Bengio Y. (2015), "Neural machine translation by jointly learning to align and translate". *In International Conference on Learning Representations (ICLR)*.

[4]. Christopher O. (2015), Understanding LSTM Networks. *Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

[5]. Corochann (2017), Recurrent Neural Network (RNN) introduction. *Retrieved from http://corochann.com/recurrent-neural-network-rnn-introduction-1286.html*

[6]. Denny B. (2015), Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs. *Retrieved from http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/*

[7]. Diederik P. K., Jimmy L.B. (2015), "Adam: A Method for Stochastic Optimization". *International Conference on Learning Representations*.

[8]. Géron A. (2017), *Hands-on Machine Learning with Scikit-Learn and Tensorflow – Concepts, Tools, and Techniques to Build Intelligent Systems*. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[9]. Graff D., Kong J., Chen K., and Maeda K. (2003). English gigaword. Linguistic Data Consortium, Philadelphia.

[10]. Heinzerling B., Johannsen A. (2016), A Python wrapper for the ROUGE summarization evaluation package. *Retrieved from https://pypi.org/project/pyrouge/*

[11]. Hermann K.M., Kocisky T., Grefenstette E., Espeholt L., Kay W., Suleyman M., Blunsom P. (2015). "Teaching machines to read and comprehend". *In Neural Information Processng Systems*.

[12]. Hochreiter S., Schmidhuber J. (1997), "LONG SHORT-TERM MEMORY". *Neural Computation 9(8)*, pp. 1735-1780.

[13]. Ibrahim A.H. (2017), Understanding Word2vec for Word Embedding I. *Retrieved from https://ahmedhanibrahim.wordpress.com/2017/04/25/thesis-tutorials-i-understanding-word2vec-for-word-embedding-i/*

[14] John M.C., Dianne P.O. (2001), "Text summarization via hidden markov models". *In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM*, pp. 406-407.

[15]. Krenker A., Bester J., Kos A. (2011), "Introduction to the Artificial Neural Networks". *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, ISBN: 978-953-307-243-2, InTech.

[16]. Lin C.Y. (2004). "Rouge: A package for automatic evaluation of summaries". *In Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL.*

[17]. Lucas O.Y. (2016). "Newspaper3K Article scraping library". *Retrieved from https://github.com/codelucas/newspaper.*

[18]. Lucy V., Hisami S., Chris B., and Ani N. (2007), "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion", *Information Processing & Management*, 43 (6), pp. 1606-1618.

[19]. McCormick C. (2016), Word2Vec Tutorial - The Skip-Gram Model. *Retrieved from http://www.mccormickml.com.*

[20]. Mehdi A., Seyedamin P., Mehdi A., Saeid S.,Elizabeth D. T., Juan B. G., Krys K. (2017), "Text Summarization Techniques: A Brief Survey", *arXiv.*

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

[22]. Nallapati R., Zhou B., Santos C.D., (2016), "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond", *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 280-290.

[23]. Nenkova A., McKeown K. (2012), "A survey of text summarization techniques", *In Mining Text Data. Springer*, pp. 43-76.

[24]. Rada M., Paul T. (2004), "TextRank: Bringing order into texts", *Association for Computational Linguistics.*

[25]. Radev D.R., Hovy E., and McKeown K. (2002), "Introduction to the special issue on summarization", *Computational linguistics*, 28(4), pp. 399-408.

[26]. Rush A.M., Chopra S., Weston J. (2015), "A Neural Attention Model for Sentence Summarization". *In Empirical Methods in Natural Language Processing*.

[27]. Sarwan N.S. (2017), An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec. *Retrieved from https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/*

[28]. See A., Peter J. L., Christopher D.M. (2017), "Get To The Point: Summarization with Pointer-Generator Networks", *arXiv*.

[29]. Shi Yan (2016), Understanding LSTM and its diagrams. *Retrieved from https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714*

[30]. Sutskever I., Vinyals O., Quoc V.L. (2014), "Sequence to Sequence Learning with Neural Networks", *arXiv*.

[31]. Trung V.T. (2017). "Python Vietnamese Toolkit". *Retrieved from https://pypi.python.org/pypi/pyvi*

[32]. Yogan J. K., Ong S. G., Halizah B., Ngo H. C. and Puspalata C. S. (2016), "A Review on Automatic Text Summarization Approaches", *Journal of Computer Science*, 12 (4), pp. 178-190.