

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Tóm tắt tóm lược văn bản
với học sâu và ứng dụng
đọc tin nhanh

Đặng Trung Anh

Hà Nội, 05/2019

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Tóm tắt tóm lược văn bản với học sâu và ứng dụng đọc tin nhanh

Sinh viên thực hiện Đặng Trung Anh

Người hướng dẫn TS. Nguyễn Thị Thu Trang

Hà Nội, 05/2019

Lời cam kết

Họ và tên sinh viên: Đặng Trung Anh

Điện thoại liên lạc: 0356198955 Email: dangtrunganh.hust@gmail.com

Lớp: CNTT-TT 2.04 – K59 Hệ đào tạo: Đại học chính quy

Tôi – *Đặng Trung Anh* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 25 tháng 5 năm 2019

Tác giả ĐATN

Đặng Trung Anh

Lời cảm ơn

Năm năm, quãng thời gian không quá ngắn cũng không quá dài trong cuộc đời mỗi con người, đối với em, năm năm tại đại học Bách Khoa Hà Nội có lẽ là quãng thời gian tuyệt vời nhất. Cảm ơn Bách Khoa đã cho em những người bạn đồng hành, những kiến thức sâu rộng, những kỷ niệm khó quên và những kỹ năng vô cùng quý giá, đó sẽ là những hàng trang vững chắc, giúp em tự tin bước trên con đường sự nghiệp sắp tới.

Em xin được gửi lời cảm ơn chân thành đến các thầy cô trong trường Đại học Bách Khoa Hà Nội cũng như các thầy cô trong viện Công nghệ thông tin và truyền thông đã truyền dạy cho em những kiến thức, kinh nghiệm quý báu trong suốt quá trình học tập và rèn luyện.

Em xin gửi lời cảm ơn sâu sắc đến cô giáo, TS. Nguyễn Thị Thu Trang. Cảm ơn cô đã giúp em định hướng đê tài cho đồ án tốt nghiệp, hướng dẫn và chỉ bảo tận tình cho em trong suốt quá trình nghiên cứu và thực hiện đồ án.

Em xin cảm ơn anh Tạ Công Sơn, cựu sinh viên viện Công nghệ thông tin và truyền thông, đã giúp đỡ em rất nhiều trong việc xây dựng API cho ứng dụng đọc tin nhanh.

Cuối cùng, xin cảm ơn gia đình và bạn bè đã luôn ở bên em, động viên, giúp đỡ và động viên em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Tóm tắt

Sự tiến bộ vượt bậc của khoa học và kỹ thuật đem lại một lượng thông tin vô cùng lớn cho con người, trong đó, văn bản là một trong những dạng biểu diễn thông tin phổ biến nhất. Tuy nhiên, việc có quá nhiều thông tin khiến chúng ta khó tổng hợp hơn, không tập trung được vào thông tin quan trọng. Do đó, cần tóm lược nội dung văn bản một cách ngắn gọn, súc tích mà không mất đi thông điệp muốn truyền tải, giúp tiết kiệm thời gian cho người đọc.

Hiện nay, tóm tắt văn bản được chia làm hai hướng chính là tóm tắt trích rút (Extractive Summarization) và tóm tắt tóm lược (Abstractive Summarization). Trong đó, tóm tắt trích rút thực hiện lấy một số câu quan trọng làm nội dung tóm tắt cho cả văn bản còn tóm tắt tóm lược tạo ra văn bản tóm tắt dựa trên việc hiểu nội dung văn bản gốc, sinh ra một văn bản mới trôi chảy, gần giống như con người tóm tắt. Gần đây, với sự phát triển của học sâu cùng các máy tính hiệu năng cao đã đem lại những hướng tiếp cận mới cho tóm tắt tóm lược, trong đó phổ biến nhất là phương pháp dựa trên mô hình Sequence to Sequence. Tuy nhiên, mô hình này có khả năng tạo các chi tiết nội dung không chính xác, không xử lý được các từ nằm ngoài tập từ điển và bị lặp lại các nội dung trong bản tóm tắt. Mô hình Sequence to Sequence với mạng Pointer-Generator kết hợp kỹ thuật Coverage được đề xuất bởi Abigail See vào năm 2017 [1] đã giải quyết được các vấn đề nêu trên. Đây cũng chính là mô hình cơ sở được em lựa chọn để giải quyết bài toán tóm tắt tóm lược cho tiếng Anh và tiếng Việt. Để thực hiện, em đã sử dụng bộ dữ liệu Daily Mail/CNN [2] cho tiếng Anh và bộ dữ liệu của Báo Mới để xử lý cho tiếng Việt. Dựa vào mô hình trên, em cũng đề xuất các cơ chế sử dụng Word embedding: Word2Vec và Fasttext cho đầu vào để cải thiện hệ thống. Các kết quả thực nghiệm cho thấy cách tiếp cận và cơ chế đề xuất rất hứa hẹn trong việc giải quyết nhiệm vụ tóm tắt tóm lược văn bản tiếng Anh cũng như tiếng Việt.

Bên cạnh đó, hiện nay, việc sử dụng smartphone để truy cập thông tin qua báo điện tử trở nên ngày càng phổ biến. Tuy nhiên, các báo hiện nay đa số đều khá dài, khiến người đọc mất nhiều thời gian trong việc nắm bắt thông tin trọng tâm. Do vậy, em đã chọn xây dựng ứng dụng đọc tin nhanh, tóm tắt các bài báo trên nền tảng hệ điều hành Android. Với sự trợ giúp của công nghệ tổng hợp văn bản thành tiếng nói Text-To-Speech (TTS), ứng dụng còn giúp cho người đọc không cần đọc báo trực tiếp, có thể vừa nghe để nắm bắt thông tin, vừa làm việc khác tại cùng thời điểm.

Mục lục

Lời cam kết.....	iii
Lời cảm ơn	iv
Tóm tắt	v
Mục lục	vii
Danh mục hình vẽ	xi
Danh mục bảng	xiii
Danh mục công thức	xiv
Danh mục các từ viết tắt.....	xvi
Danh mục thuật ngữ.....	xviii
Chương 1 Giới thiệu đề tài	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi đề tài	2
1.3 Định hướng giải pháp	4
1.4 Bố cục đồ án	5
Chương 2 Cơ sở lý thuyết.....	6
2.1 Mạng nơ-ron nhân tạo.....	6
2.1.1 Kiến trúc cơ bản	6
2.1.2 Mạng nơ-ron hồi quy.....	9

2.1.3 Mạng Long Short Term Memory.....	11
2.1.4 Mô hình Sequence to Sequence	12
2.2 Một số kỹ thuật nâng cao	13
2.2.1 Kỹ thuật attention trong mô hình Sequence to Sequence	13
2.2.2 Mạng Pointer.....	14
2.2.3 Kỹ thuật Coverage.....	14
2.3 Word embedding	15
2.3.1 Tổng quan.....	15
2.3.2 Word2Vec.....	16
2.3.3 Fasttext.....	18
Chương 3 Tình hình nghiên cứu hiện nay	20
3.1 Mô hình Sequence to Sequence kết hợp Attention	20
3.1.1 Mô hình.....	20
3.1.2 Đánh giá.....	22
3.2 Mô hình Pointer-Generator kết hợp Coverage	23
3.2.1 Mạng Pointer-Generator	23
3.2.2 Mạng Pointer-Generator kết hợp Coverage	25
3.2.3 Đánh giá.....	26
Chương 4 Đề xuất giải pháp và thực nghiệm	28
4.1 Giải pháp cải tiến đề xuất.....	28
4.2 Bộ dữ liệu	30
4.3 Thực nghiệm	32
4.3.1 Môi trường thực nghiệm.....	32

4.3.2 Phương pháp đánh giá	33
4.3.3 Kết quả thực nghiệm	35
4.4 Nhận xét	42
Chương 5 Phát triển ứng dụng đọc tin nhanh.....	44
5.1 Khảo sát và phân tích yêu cầu	44
5.1.1 Khảo sát hiện trạng.....	44
5.1.2 Tổng quan chức năng	45
5.1.3 Đặc tả chức năng	49
5.1.4 Yêu cầu phi chức năng	53
5.2 Thiết kế và xây dựng ứng dụng.....	53
5.2.1 Công nghệ sử dụng.....	53
5.2.2 Thiết kế kiến trúc	55
5.2.3 Thiết kế chi tiết	58
5.2.4 Xây dựng ứng dụng.....	62
5.2.5 Kiểm thử	67
5.2.6 Triển khai.....	69
Chương 6 Kết luận và hướng phát triển.....	70
6.1 Kết luận.....	70
6.2 Hướng phát triển	71
Tài liệu tham khảo	74
Phụ lục.....	A-1
A Cơ sở lý thuyết	A-1
A.1 Tính toán truyền tín hiệu trong mạng nơ-ron	A-1

A.2 Tính toán tại các cỗng trong mạng LSTM	A-2
A.3 Tính toán trong mô hình Sequence to Sequence	A-4
B Các ví dụ văn bản thực nghiệm.....	B-5
B.1 Văn bản ví dụ 1 được thực nghiệm trên tiếng Anh.....	B-5
B.2 Văn bản ví dụ 2 được thực nghiệm trên tiếng Anh.....	B-6
B.3 Văn bản ví dụ 1 được thực nghiệm trên tiếng Việt.....	B-7
B.4 Văn bản ví dụ 2 được thực nghiệm trên tiếng Việt.....	B-8

Danh mục hình vẽ

Hình 1 Mô hình mạng nơ-ron nhân tạo.....	7
Hình 2 Sử dụng Early Stopping tránh Overfitting trong học máy [5].	8
Hình 3 Kiến trúc mạng RNN [6].	9
Hình 4 Vấn đề phụ thuộc xa trong mạng RNN [7].....	10
Hình 5 Kiến trúc mạng LSTM [7].	11
Hình 6 Mô hình Sequence to Sequence [10].....	13
Hình 7 Mô hình Skip-gram trong Word2Vec [13].	17
Hình 8 Mô hình CBOW trong Word2Vec [14].....	18
Hình 9 Mô hình Sequence to Sequence kết hợp Attention [1].....	21
Hình 10 Mạng Pointer-Generator trong tóm tắt tóm lược văn bản [1].	24
Hình 11 Mô hình cải tiến sử dụng Word2Vec/Fasttext pre-trained.	29
Hình 12 Thống kê phân phối số lượng từ trong phần Sapo.	31
Hình 13 Thống kê phân phối số lượng từ trong phần Content.....	32
Hình 14 Một số biểu đồ giá trị mất mát khi huấn luyện cho tiếng Anh.	36
Hình 15 Một số biểu đồ giá trị mất mát khi huấn luyện cho tiếng Việt.	40
Hình 16 Biểu đồ use case tổng quan.....	45
Hình 17 Biểu đồ use case phân rã “Quản lý bài báo”.....	46
Hình 18 Biểu đồ use case phân rã “Nghe báo nói”.....	47

Hình 19 Biểu đồ use case phân rã “Quản lý sự kiện”.....	47
Hình 20 Biểu đồ use case phân rã “Quản lý dòng sự kiện”.....	48
Hình 21 Biểu đồ use case phân rã “Quản lý nguồn báo và thẻ loại báo”.....	49
Hình 22 Triển khai hệ thống FCM [21].	54
Hình 23 Kiến trúc tổng quan hệ thống.....	56
Hình 24 Biểu đồ thiết kế chi tiết gói.....	57
Hình 25 Biểu đồ dịch chuyển màn hình trong ứng dụng.....	59
Hình 26 Biểu đồ use case “Theo dõi sự kiện”.....	59
Hình 27 Biểu đồ use case “Tìm kiếm sự kiện”.....	60
Hình 28 Biểu đồ lớp cho use case “Theo dõi sự kiện”.....	60
Hình 29 Biểu đồ lớp cho use case “Tìm kiếm sự kiện”.....	61
Hình 30 Một số giao diện các chức năng chính của ứng dụng.....	66
Hình 31 Quá trình truyền tín hiệu giữa các nơ-ron thuộc hai tầng kế tiếp [25]... A-1	
Hình 32 Cổng lăng quên trong LSTM [7].....	A-2
Hình 33 Cổng đầu vào trong LSTM [7].....	A-3
Hình 34 Cổng đầu ra trong LSTM [7].....	A-3

Danh mục bảng

Bảng 1 Kết quả thống kê trên bộ dữ liệu Báo Mới	30
Bảng 2 Các kết quả thử nghiệm trên bộ Daily Mail/CNN	35
Bảng 3 Ví dụ số 1 về bản tóm tắt tiếng Anh sinh ra từ mô hình	37
Bảng 4 Ví dụ số 2 về bản tóm tắt tiếng Anh sinh ra từ mô hình	38
Bảng 5 Các kết quả thực nghiệm trên bộ Báo Mới	39
Bảng 6 Ví dụ số 1 về bản tóm tắt tiếng Việt sinh ra từ mô hình	41
Bảng 7 Ví dụ số 2 về bản tóm tắt tiếng Việt sinh ra từ mô hình	41
Bảng 8 Danh sách các use case của hệ thống	49
Bảng 9 Đặc tả use case “Nghe bài báo hiện tại”	50
Bảng 10 Đặc tả use case “Theo dõi sự kiện”	51
Bảng 11 Đặc tả của use case “Tìm kiếm sự kiện”	52
Bảng 12 Dữ liệu đầu vào khi tìm kiếm sự kiện	52
Bảng 13 Một số API quan trọng được sử dụng trong ứng dụng	62
Bảng 14 Các thư viện, ngôn ngữ lập trình và công cụ sử dụng	63
Bảng 15 Thông tin chi tiết của ứng dụng	64
Bảng 16 Một số trường hợp kiểm thử	67

Danh mục công thức

Công thức 1 Trạng thái ẩn tại bước thời gian t trong mạng RNN.	9
Công thức 2 Công thức tính đầu ra tại bước thời gian t trong mạng RNN.	10
Công thức 3 Công thức tính attention distribution.	21
Công thức 4 Công thức tính vector ngữ cảnh.	21
Công thức 5 Công thức tính phân phối từ vựng.	22
Công thức 6 Công thức tính phân phối dự đoán từ w	22
Công thức 7 Công thức tính giá trị mất mát tại bước thời gian t	22
Công thức 8 Công thức tính giá trị mất mát cho toàn bộ chuỗi.	22
Công thức 9 Công thức tính xác suất tạo Pgen.	23
Công thức 10 Công thức tính phân phối trên tập từ điển mở rộng.	24
Công thức 11 Công thức tính vector bao phủ.	25
Công thức 12 Công thức biến đổi scores.	25
Công thức 13 Công thức tính mất mát bao phủ.	26
Công thức 14 Công thức hàm mất mát tổng quát mới.	26
Công thức 15 Công thức tính ROUGE-N.	34
Công thức 16 Công thức tính đầu ra tại mỗi nơ-ron.	A-2
Công thức 17 Công thức trạng thái ẩn tại bước thời gian t của bộ mã hóa.	A-4
Công thức 18 Công thức tính vector sinh ra từ chuỗi trạng thái ẩn.	A-4

Công thức 19 Công thức tính xác suất cho mỗi từ trong bộ giải mã. A-4

Công thức 20 Công thức tính mỗi thành phần xác suất có điều kiện. A-5

Danh mục các từ viết tắt

RNN	Recurrent Neural Network Mạng nơ-ron hồi quy
OOV	Out-of-vocabulary Nǎm ngoài tập từ điển
LSTM	Long Short Term Memory Mô hình bộ nhớ gần-xa
DNNs	Deep Neural Networks Mạng nơ-ron sâu
CNN	Convolutional Neural Network Mạng nơ-ron tích chập
NMT	Neural Machine Translation Hệ dịch máy dựa trên mạng nơ-ron
CBOW	Continuous Bag of Words Túi từ liên tục
IDE	Integrated Development Environment Môi trường phát triển tích hợp

API	Application Programming Interface Giao diện lập trình ứng dụng
REST	Representational State Transfer Chuyển đổi trạng thái biểu diễn
XML	Extensible Markup Language Ngôn ngữ đánh dấu mở rộng
DATA	Đồ án tốt nghiệp

Danh mục thuật ngữ

Sapo	Đoạn tiêu đề, mở đầu của bài báo
Smartphone	Điện thoại thông minh
Server	Máy chủ
Client	Máy khách
Online	Trực tuyến
Offline	Ngoại tuyến
Tab	Trang giao diện ứng dụng

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Xã hội ngày càng phát triển đem lại sự tiến bộ vượt bậc của khoa học và kỹ thuật, bên cạnh việc giúp cho chất lượng cuộc sống con người trở nên tốt hơn, nó cũng đem lại một lượng thông tin vô cùng lớn. Bên cạnh việc tiếp nhận thông tin truyền thống, thông qua báo chí, sách vở, truyền hình, ngày nay, sự ra đời và phát triển của internet đã mang đến cho con người một cách tiếp nhận, thông qua nó, việc truy cập thông tin trở nên đơn giản hơn bao giờ hết. Mặt khác, lượng thông tin quá nhiều cũng khiến cho việc tổng hợp khó khăn hơn, tốn nhiều thời gian hơn để nắm được thông tin quan trọng. Hiện nay, đa phần lượng thông tin thường được biểu diễn dưới dạng các văn bản, có thể từ các trang báo, bài viết, tạp chí khoa học... Các văn bản này liên tục phát triển lên mỗi ngày, tạo ra một vấn đề trong việc tổng hợp, nắm bắt thông tin sao cho hiệu quả. Để giải quyết vấn đề này, chúng ta cần tóm lược nội dung thông tin trong văn bản một cách ngắn gọn, súc tích, tập trung vào những chi tiết nổi bật nhất mà không mất đi thông điệp muôn truyền tải của văn bản gốc. Tuy nhiên, với số lượng văn bản lớn, rất khó cho con người để có thể tóm tắt thủ công cho từng văn bản, công việc đòi hỏi rất nhiều thời gian và sức lực. Xuất phát từ thực trạng đó, nhu cầu xây dựng một hệ thống tóm tắt văn bản tự động trở nên cần thiết hơn bao giờ hết. Hơn nữa, kết quả đạt được có thể được ứng dụng cho rất nhiều lĩnh vực với các bài toán khác nhau như: (i) Bài toán tạo tiêu đề cho bài báo, (ii) Tạo đoạn sapo cho bài báo, (iii) Bài toán tóm tắt nội dung bài báo, tạp chí...

Tóm tắt tóm lược văn bản tự động là quá trình mà chương trình máy tính tạo ra bản tóm tắt sao cho vẫn có đủ nội dung chính, ý tưởng chính của văn bản gốc. Do vậy, để tóm tắt văn bản được chính xác, hệ thống phải hiểu được nội dung văn bản, bên cạnh đó cũng cần phải kiểm tra chất lượng tóm tắt dựa trên các thang đo tiêu chuẩn. Đối

với con người, việc tóm tắt có lẽ không quá khó khăn, bằng cách đọc hiểu văn bản gốc, sau đó chắt lọc ý nghĩa, viết lại dựa trên các chi tiết nổi bật, độ hiểu và vốn từ ngữ của người tóm tắt, tuy nhiên công việc này khá tốn thời gian với các văn bản dài và số lượng văn bản lớn. Với sự phức tạp của ngôn ngữ, tóm tắt văn bản tự động phải đổi mới với rất nhiều thách thức, mục tiêu của việc xây dựng hệ thống tóm tắt là tạo ra được các bản tóm tắt có kết quả tốt, trôi chảy, sát với những gì được viết bởi con người.

Nhiệm vụ tóm tắt đặc biệt có nhiều khó khăn đối với tóm tắt văn bản tiếng Việt. Bởi ngữ pháp tiếng Việt vô cùng đa dạng với vốn từ vựng rất lớn. Một từ tiếng Việt có thể là danh từ trong một ngữ cảnh cụ thể, nhưng cũng có thể là động từ hay tính từ trong các ngữ cảnh khác. Do đó, việc tóm tắt văn bản phải đổi mới với nhiều vấn đề cả ngữ pháp, ngữ nghĩa lẫn sự liên kết các câu trong văn bản.

Mặc khác, đa số các bài báo hiện nay thường viết khá dài, khiến người đọc tốn nhiều thời gian để nắm bắt được trong tâm. Song song với báo đọc, với sự ra đời và phát triển của công nghệ tổng hợp văn bản thành tiếng nói Text-To-Speech (TTS), người đọc sẽ không cần nhìn trực tiếp để đọc báo, vừa giúp bảo vệ mắt, vừa có thể làm việc khác trong khi đọc báo, thay vì phải nhìn trực tiếp vào thiết bị mà không làm được gì khác. Xuất phát từ nhu cầu thực tế, ứng dụng đọc tin nhanh, tóm tắt tin, gom nhóm tin tức sẽ giúp người đọc thuận tiện trong việc đọc báo, thay đổi thói quen, giúp việc nắm bắt thông tin trở lên dễ dàng hơn. Không chỉ giúp đọc tin nhanh và nghe báo nói, để không bị bỏ lỡ thông tin, cập nhật thông tin liên tục, em xây dựng tính năng gom nhóm sự kiện - tập hợp các bài báo tương đồng, dòng sự kiện - chuỗi các sự kiện theo dòng thời gian, gửi thông báo khi có thêm sự kiện mới, đảm bảo thông tin đưa đến người đọc nhanh nhất có thể.

1.2 Mục tiêu và phạm vi đề tài

Với những khó khăn trong việc tóm tắt tóm lược, nên phần lớn các phương pháp tóm tắt được sử dụng trong quá khứ sử dụng tóm tắt trích rút. Gần đây, với sự trợ giúp của các máy tính hiệu năng cao và sự phát triển của học sâu đã đem lại những hướng tiếp cận mới cho bài toán tóm tắt tóm lược văn bản. Một trong các phương pháp tiếp

cận phổ biến nhất hiện nay là dựa trên mô hình Sequence to Sequence kết hợp kỹ thuật Attention, một phương pháp được sử dụng nhiều trong các bài toán dịch máy. Mặc dù các mô hình đưa ra rất hứa hẹn, tuy nhiên chúng tồn tại các nhược điểm: (i) có khả năng tạo ra không chính xác các chi tiết thực tế, (ii) không có khả năng xử lý từ nằm ngoài tập từ điển (out-of-vocabulary - OOV), (iii) có xu hướng bị lặp lại các cụm từ. Phương pháp mới gần đây đã được ra đời để giải quyết vấn đề đó, được đề xuất trong “Get To The Point: Summarization with Pointer-Generator Networks” bởi Abigail See vào năm 2017 [1]. Phương pháp này bên cạnh việc sử dụng dựa trên mô hình cơ sở nêu trên, đã kết hợp thêm vào mạng Pointer-Generator, cùng với việc sử dụng kỹ thuật Coverage. Mô hình này đã được chứng minh có thể giải quyết hai vấn đề: (i) OOV và (ii) vấn đề các cụm từ bị lặp lại nằm trên của mô hình Sequence to Sequence kết hợp Attention. Tuy nhiên, vấn đề OOV vẫn còn tồn tại, chưa được giải quyết triệt để, do các từ đầu vào của các tầng mã hóa và giải mã có thể không có trong từ điển, bị biểu diễn sang ký tự “UNK” trước khi đưa vào mạng.

Trong đồ án tốt nghiệp của em hướng tới sử dụng mô hình được đưa ra bởi Abigail See [1] làm mô hình cơ sở, đồng thời đề xuất sử dụng các cơ chế Word embedding: Word2Vec và Fasttext để xử lý đầu vào, với hi vọng cải thiện hệ thống, sau đó áp dụng thực nghiệm cho cả tiếng Anh và tiếng Việt. Trong đó, Fasttext hứa hẹn sẽ cải thiện được vấn đề OOV.

Trong việc phát triển ứng dụng đọc tin, hiện nay đã có rất nhiều ứng dụng đọc báo cho người Việt như: (i) ứng dụng Báo Mới, (ii) Google tin tức, (iii) Vietnamnet, (iv) Vadi... Đa phần các ứng dụng này chỉ phục vụ được nhu cầu đọc báo trực tuyến, thay vì đọc trên báo giấy truyền thống, đặc biệt, ứng dụng Vadi đã phục vụ được nhu cầu nghe báo nói của độc giả. Tuy nhiên, các ứng dụng đó không giúp người đọc nắm bắt được thông tin nhanh, vẫn phải đọc hay nghe cả bài, tốn nhiều thời gian tổng hợp thông tin. Do đó, em hướng tới xây dựng một ứng dụng đọc tin, có thể tóm tắt lại các bài báo từ nhiều nguồn khác nhau, giúp người đọc nắm được nội dung thông tin quan trọng của bài báo một cách nhanh chóng, đồng thời gom nhóm các bài báo của cùng một vấn đề về chung một sự kiện, tập hợp các sự kiện liên quan vào một dòng sự kiện theo thời gian. Ứng dụng cho phép người dùng có thể nắm bắt thông tin nhanh nhất

có thể thông qua đọc bản tóm tắt các bài báo, có thể theo dõi một sự kiện, dòng sự kiện để nhận thông báo khi có sự kiện mới của dòng sự kiện đó xuất hiện, đảm bảo không bị lỡ mất thông tin quan trọng nào. Song song với đó, để thuận tiện hơn cho nhiều mục đích sử dụng, em cũng phát triển thêm tính năng báo nói, giúp đọc giả có thể vừa nắm bắt thông tin, vừa có thể làm được việc khác tại cùng thời điểm.

1.3 Định hướng giải pháp

Dựa trên mô hình đưa ra bởi Abigail See [1], em đề xuất sử dụng các cơ chế Word embedding: Word2Vec và Fasttext để xử lý đầu vào của mô hình, bỏ lớp embedding gốc của mô hình cơ sở đi, sau đó, so sánh các kết quả đạt được trong các thực nghiệm cho cả tiếng Anh và tiếng Việt. Để mô hình chạy được cho tiếng Việt, em cũng xây dựng một bộ xử lý dữ liệu cho tiếng Việt, dựa trên các đặc trưng của bộ dữ liệu, lọc ký tự, văn bản lỗi, tách từ, định dạng lại cho phù hợp với đầu vào của mô hình. Chi tiết về phương pháp thực hiện sẽ được trình bày trong Chương 4.

Sau khi hoàn thành các thực nghiệm, các kết quả thu được cho cả tiếng Anh và tiếng Việt đều khá tốt, cho thấy cơ chế đề xuất rất hứa hẹn trong việc giải quyết nhiệm vụ tóm tắt tóm lược văn bản.

Để xây dựng ứng dụng đọc tin nhanh, em lựa chọn sử dụng phát triển trên nền tảng Android, sử dụng ngôn ngữ lập trình Java. Khi xây dựng tính năng tóm tắt nội dung bài báo, em lựa chọn sử dụng mô hình tóm tắt đề xuất nêu trên, bên cạnh đó, để xây dựng tính năng báo nói, em sử dụng công nghệ tổng hợp văn bản thành tiếng nói thông qua API của Vbee [22]. Cũng ở phía Client, để giao tiếp, gọi API từ Server sẽ thông qua thư viện Retrofit. Hệ thống sử dụng Firebase cloud messaging để phục vụ cho việc nhận thông báo khi có sự kiện mới được tạo lên trong dòng sự kiện mà người dùng theo dõi. Để lưu trữ trong bộ nhớ máy, ứng dụng sử dụng các thư viện có sẵn trong Android, phục vụ cho việc lưu lại các bài báo trong tính năng “Lưu báo”, lưu lại các bài báo, sự kiện, dòng sự kiện chế độ sử dụng khi không có kết nối mạng.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về cơ sở lý thuyết của phương pháp tóm tắt đưa ra, trong đó phần đầu tiên sẽ trình bày các kiến thức nền tảng của mạng nơ-ron nhân tạo nói chung cũng như các mạng nơ-ron hồi quy, mạng Long Term Short Memory, mô hình Sequence to Sequence nói riêng. Tiếp theo, chương này sẽ trình bày các kỹ thuật nâng cao khác được áp dụng trong mô hình tóm tắt tóm lược văn bản, đó là kỹ thuật Attention cùng với mô hình Sequence to Sequence kết hợp attention, mạng Pointer-Generator và kỹ thuật Coverage.

Chương 3 trình bày về tình hình nghiên cứu hiện nay của bài toán tóm tắt tóm lược văn bản, xoay quanh mô hình Sequence to Sequence kết hợp kỹ thuật Attention, bên cạnh đó là mô hình mạng Pointer-generator cùng với kỹ thuật Coverage.

Trong Chương 4, em sẽ trình bày về các giải pháp đề xuất, phương pháp thực nghiệm và đánh giá kết quả thu được. Trong đó, các thực nghiệm đề xuất sẽ được thực hiện trên cả hai bộ dữ liệu tiếng Anh và tiếng Việt.

Chương 5 trình bày các nội dung về xây dựng ứng dụng đọc tin nhanh. Trong đó, phần đầu tiên sẽ trình bày về khảo sát và phân tích yêu cầu, nhu cầu sử dụng ứng dụng của người dùng. Phần tiếp theo sẽ trình bày về thiết kế và xây dựng ứng dụng, trong đó có các công nghệ sử dụng cũng như các công đoạn thiết kế kiến trúc, thiết kế chi tiết, xây dựng ứng dụng, kiểm thử và triển khai.

Chương 6 trình bày tổng kết các kết quả đạt được cũng như các hạn chế còn tồn tại, đồng thời định hướng phát triển thêm trong tương lai.

Chương 2 Cơ sở lý thuyết

Chương 1 đã giới thiệu về vấn đề, mục tiêu và phạm vi của đề tài, định hướng giải pháp và bối cảnh đồ án. Chương 2 này sẽ trình bày các kiến thức lý thuyết, làm cơ sở để hiểu các kết quả nghiên cứu liên quan trong Chương 3 và các giải pháp đưa ra ở Chương 4. Các nội dung sẽ trình bày trong chương này bao gồm: (i) Mạng nơ-ron nhân tạo, (iii) Các kỹ thuật nâng cao và (iv) Word embedding.

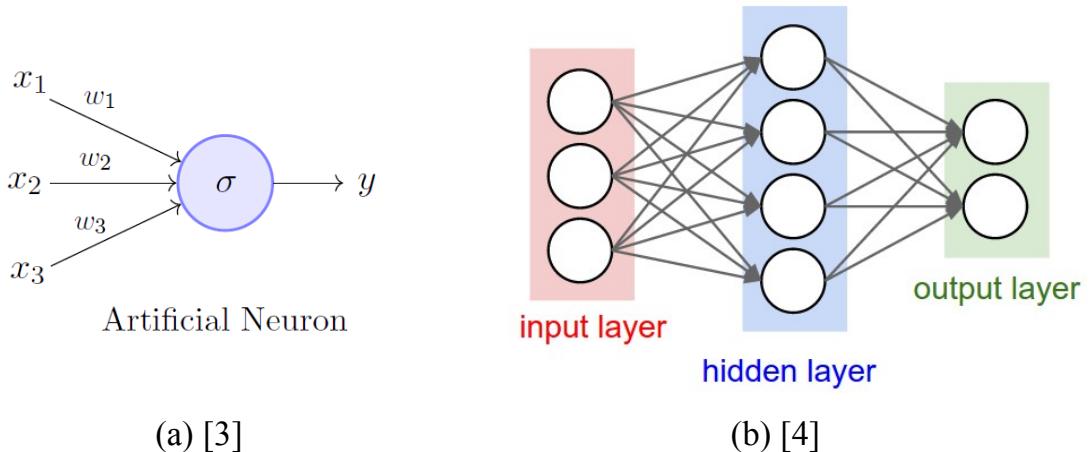
2.1 Mạng nơ-ron nhân tạo

Ngày nay, lĩnh vực học máy đang ngày càng phát triển không ngừng với những tiến bộ vượt bậc, ứng dụng trong nhiều bài toán khác nhau. Các thuật toán trong học máy cho phép máy tính được huấn luyện trên tập dữ liệu đầu vào, nhờ đó, các máy tính có thể xây dựng được mô hình để tự động đưa ra quyết định dựa trên tập dữ liệu đầu vào mới. Trong suốt quá trình phát triển, dù học máy có nhiều kiến trúc cũng như thuật toán khác nhau, nhưng phổ biến hơn cả vẫn là mạng nơ-ron nhân tạo.

2.1.1 Kiến trúc cơ bản

Mạng nơ-ron nhân tạo là một mô hình xử lý thông tin được lấy cảm hứng từ cách các hệ thống thần kinh sinh học, mô phỏng quá trình não xử lý thông tin. Mạng nơ-ron được cấu tạo bởi các nơ-ron đơn lẻ, được gọi là các perceptron, tương tự như các nơ-ron sinh học.

Trong mạng nơ-ron nhân tạo, mỗi nơ-ron đơn lẻ nhận dữ liệu đầu vào, đưa qua hàm kích hoạt để cho ra kết quả, được minh họa như trên Hình 1 (a). Hai nơ-ron nhân tạo kết nối với nhau thông qua cạnh nối tương tự như nơ-ron sinh học, quá trình tính hàm kích hoạt mô phỏng việc thay đổi trạng thái của tín hiệu lan truyền giữa các nơ-ron. Các nơ-ron nhân tạo được xếp thành các tầng liên tiếp tạo thành mạng nơ-ron nhân tạo, truyền tín hiệu đi giữa các tầng với nhau, được mô tả như trong Hình 1 (b).



Hình 1 Mô hình mạng nơ-ron nhân tạo.

Kiến trúc chung của một mạng nơ-ron nhân tạo bao gồm ba thành phần: (i) Tầng đầu vào (input layer), (ii) Các tầng ẩn (hidden layer), (iii) Tầng đầu ra (output layer). Số lượng tầng (layer) của một mạng nơ-ron được tính bằng số lượng tầng ẩn cộng với 1, tức là khi đếm số tầng, ta không tính tầng đầu vào, số lượng tầng này thường được ký hiệu là L . Hình 1 (b) mô tả một mạng nơ-ron đơn giản, một mạng nơ-ron feedforward (suy diễn tiến) với 1 tầng đầu vào, 1 tầng ẩn và 1 tầng đầu ra, số tầng của mạng nơ-ron này là $L = 3$. Quá trình tính toán truyền tín hiệu trong mạng nơ-ron sẽ được trình bày trong phụ lục A.1. Giá trị đầu ra tại mỗi nơ-ron được tính bằng cách áp dụng hàm kích hoạt lên đầu vào. Các hàm kích hoạt thường được sử dụng trong mạng nơ-ron nhân tạo bao gồm: (i) hàm sigmoid, (ii) hàm tanh, (iii) hàm ReLU...

Để tối thiểu hóa hàm mất mát, việc tối ưu hóa tham số cho mạng sẽ được thực hiện thông qua quá trình lan truyền ngược (backpropagation). Bên cạnh mạng nơ-ron nhân tạo cơ bản, trong thực tế, để phục vụ cho những bài toán cần kiến trúc phức tạp hơn, nhiều mạng khác đã ra đời, trong đó phải kể đến: (i) Convolutional Neural Network (CNN), (ii) Recurrent Neural Network (RNN), (iv) Long Short Term Memory (LSTM)... Trong đó, mạng RNN và LSTM sẽ được trình bày trong các phần sau.

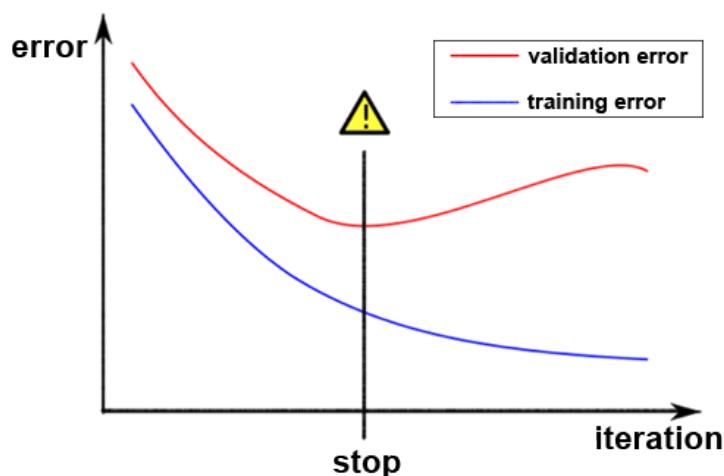
Vấn đề Overfitting

Overfitting trong học máy là hiện tượng mô hình thu được quá khớp (fit) với tập dữ liệu huấn luyện. Việc quá khớp này sẽ khiến cho khi thực hiện dự đoán, kết quả thu

được sẽ bị nhầm nhiều, chất lượng mô hình sẽ không còn tốt trên tập thực nghiệm nữa. Vấn đề này thường xảy ra khi độ phức tạp của mô hình quá cao để có thể mô phỏng dữ liệu huấn luyện với tập dữ liệu huấn luyện nhỏ. Có nhiều cách để khắc phục vấn đề overfitting trong học máy như: sử dụng tập đánh giá (validation set), dừng sớm (Early Stopping) , tắt ngẫu nhiên một vài nơ-ron (Dropout)...

Phần này trình bày về phương pháp sử dụng tập đánh giá kết hợp với Early Stopping, phương pháp sẽ được sử dụng trong quá trình thực nghiệm ở Chương 4. Trong phương pháp này, ngoài các tập dữ liệu huấn luyện và thực nghiệm (tập thực nghiệm không được sử dụng khi huấn luyện) thì để đánh giá được chất lượng mô hình với các dữ liệu mới chưa được gặp bao giờ, ta cần thêm một tập dữ liệu đánh giá (validation set), được cắt ra từ tập huấn luyện.

Để đánh giá chất lượng mô hình, ta cần một vài đại lượng, hay hàm mất mát để đánh giá, bao gồm: (i) Train error và (ii) Test error, với tập đánh giá được thêm vào, ta sẽ có thêm Validation error. Trong đó, train error là hàm mất mát được áp dụng trên tập dữ liệu huấn luyện, được tính dựa trên giá trị mất mát trung bình trên mỗi điểm dữ liệu. Tương tự, test error và validation error là hàm mất mát được áp dụng trên tập thực nghiệm và tập đánh giá.



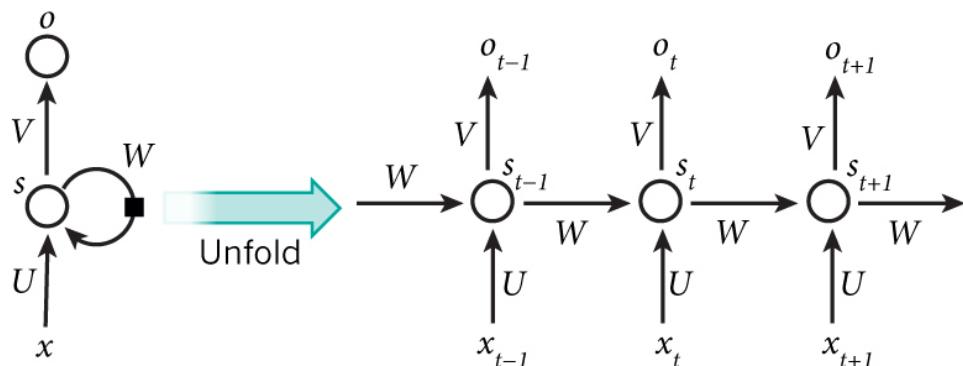
Hình 2 Sử dụng Early Stopping tránh Overfitting trong học máy [5].

Khi sử dụng các thuật toán lặp trong các bài toán học máy để tìm nghiệm, hàm mất mát thường sẽ giảm dần khi số vòng lặp tăng lên. Sử dụng Early Stopping, ta sẽ dừng

thuật toán trước khi hàm mất mát đạt giá trị quá nhỏ với số lượng các vòng lặp lớn để tránh trường hợp overfitting. Quá trình này được mô tả như trong Hình 2, trong đó, trục nằm ngang biểu thị số lượng vòng lặp, trục thẳng đứng biểu thị giá trị mất mát (error). Khi ở vòng lặp mà giá trị validation error có dấu hiệu tăng lên, trong khi training error lại giảm xuống, quá trình huấn luyện sẽ dừng lại để tránh bị overfitting.

2.1.2 Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy hay RNN lấy ý tưởng chính từ việc sử dụng chuỗi các thông tin có mối liên hệ với nhau. Ví dụ như trong xử lý ngôn ngữ tự nhiên, nếu muốn dự đoán từ tiếp theo trong câu, ta cần biết thêm các từ xuất hiện trước đó như thế nào. Được gọi là “hồi quy” bởi mạng RNN thực hiện cùng một tác vụ cho mỗi phần tử của chuỗi đầu vào, với kết quả đầu ra phụ thuộc vào tất cả các tính toán trước đó. Mô hình kiến trúc của một mạng RNN được minh họa như trên Hình 3.



Hình 3 Kiến trúc mạng RNN [6].

Hình 3 minh họa một mạng RNN với 8 tầng (layer) đã được duỗi thẳng thành một mạng đầy đủ, bằng cách viết lại mạng sang dạng một chuỗi các nơ-ron tuần tự. Trong mạng RNN, gọi x_t là đầu vào tại bước thời gian t , trạng thái ẩn s_t được tính qua Công thức 1 như sau:

$$s_t = f(Ux_t + Ws_{t-1})$$

Công thức 1 Trạng thái ẩn tại bước thời gian t trong mạng RNN.

Trong đó, hàm $f(\cdot)$ là một hàm kích hoạt phi tuyến tính như hàm tanh hay ReLU. Để tính toán trạng thái ẩn tại bước thời gian đầu, ta cần khởi tạo thêm s_{-1} , thường được khởi tạo bằng 0 hết.

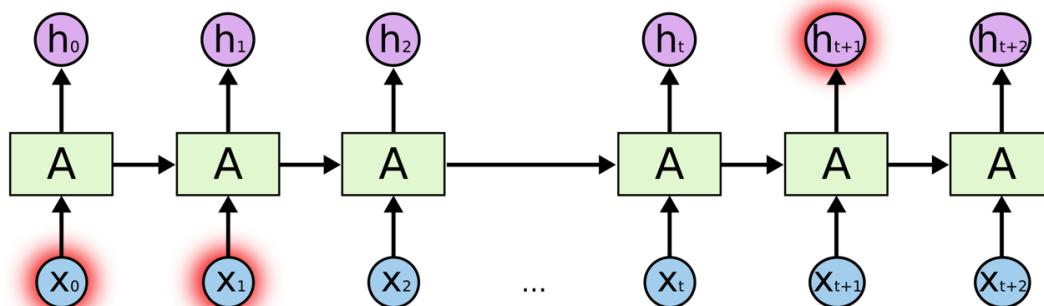
Kết quả đầu ra tại mỗi bước thời gian o_t được tính qua Công thức 2 sau:

$$o_t = \text{softmax}(Vs_t)$$

Công thức 2 Công thức tính đầu ra tại bước thời gian t trong mạng RNN.

Trong quá trình huấn luyện mạng RNN sẽ sử dụng thuật toán lan truyền ngược (backpropagation) để tối ưu các tham số của mạng, tương tự như mạng nơ-ron cơ bản trong phần 2.1.1. Trạng thái ẩn s_t của mạng RNN có thể được hình dung như “bộ nhớ” của mạng, lưu lại các thông tin trong quá khứ, được sử dụng để tính kết quả đầu ra o_t tại bước thời gian t .

Mặc dù về mặt lý thuyết, RNN có khả năng xử lý các từ phụ thuộc xa. Tuy nhiên, trong thực tế, khi khoảng cách thông tin lớn dần thì RNN không thể nhớ và học được nữa, được minh họa như trong Hình 4.

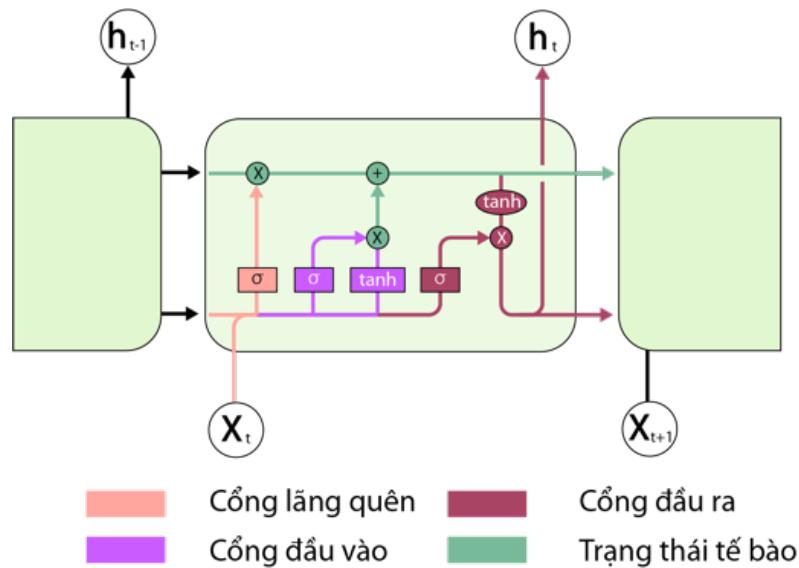


Hình 4 Vấn đề phụ thuộc xa trong mạng RNN [7].

Trong Hình 4, tại bước thời gian $t + 1$, trạng thái ẩn tương ứng là h_{t+1} không thể có được thông tin từ các bước thời gian 0 và 1 ban đầu (được bôi màu đỏ như trong hình). Mạng LSTM ra đời đã khắc phục được hạn chế này của RNN, sẽ được trình bày trong phần tiếp theo.

2.1.3 Mạng Long Short Term Memory

Mạng bộ nhớ dài ngắn hay mạng Long Short Term Memory (LSTM) là một dạng đặc biệt của RNN, khắc phục các hạn chế của RNN trong việc học các thông tin phụ thuộc xa. Với bản chất là một mạng hồi quy, LSTM có dạng là một chuỗi mô-đun (module) lặp lại của mạng nơ-ron. Tuy nhiên, các mô-đun của LSTM thay vì chỉ có một tầng mạng nơ-ron như trong RNN, nó có tới bốn tầng tương tác đặc biệt với nhau, được minh họa như trên Hình 5.



Hình 5 Kiến trúc mạng LSTM [7].

Trong LSTM, trạng thái tế bào (cell state) được bổ sung thêm, được mô tả như trong Hình 5 là đường chạy ngang trên màu xanh, nhằm ghi nhớ thông tin của toàn bộ chuỗi thời gian. Giá trị của trạng thái tế bào chạy xuyên suốt qua các bước thời gian, chỉ bị tác động bởi một vài phép toán, khiến nó có thể lưu trữ được thông tin bên trong mà ít bị biến đổi. LSTM có khả năng lọc bỏ đi hoặc thêm vào các thông tin cần thiết cho tế bào thông qua cơ chế cổng (gate). LSTM có 3 cổng để duy trì và kiểm soát trạng thái tế bào, trong đó, mỗi cổng là nơi sàng lọc thông tin, bao gồm một tầng mạng sigmoid và một phép nhân. Đầu tiên là cổng lãng quên (forget gate) thực hiện quyết định xem cần bỏ đi những thông tin nào trong trạng thái tế bào. Tiếp theo là cổng đầu vào (input gate) lựa chọn các thông tin phù hợp để lưu lại trong trạng thái tế bào. Cuối cùng là cổng đầu ra (output gate), dựa vào trạng thái tế bào mới cùng

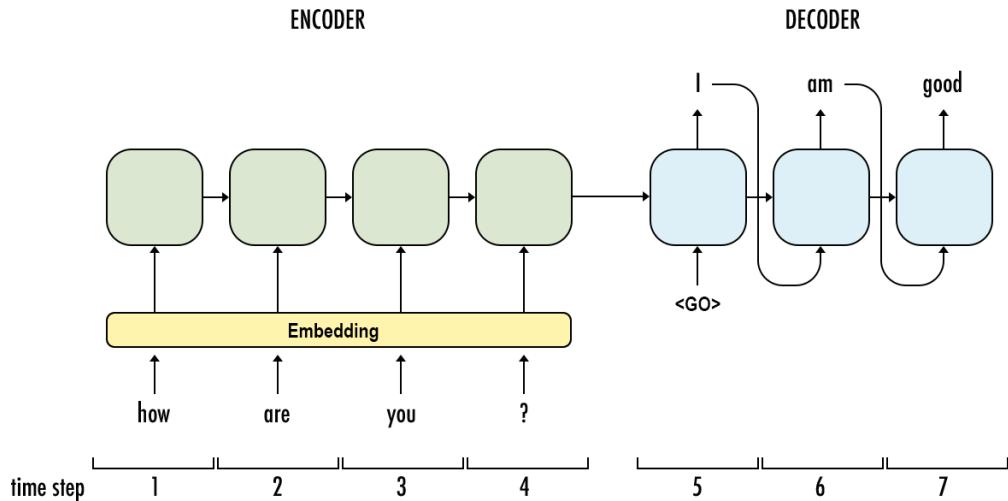
với đầu vào tại bước thời gian hiện tại để quyết định xem đầu ra là gì. Chi tiết các bước tính toán tại các cổng sẽ được trình bày trong phụ lục A.2.

Ngoài kiến trúc LSTM được trình bày ở trên, trong thực tế còn có nhiều biến thể khác, trong đó phải kể đến Bidirectional LSTM, được ra đời nhằm sử dụng thông tin theo cả hai chiều thời gian, được ứng dụng nhiều trong các hệ thống tóm tắt văn bản tóm lược.

2.1.4 Mô hình Sequence to Sequence

Mạng học sâu (DNNs) [8] là những mô hình học máy rất mạnh mẽ, đạt được hiệu suất tuyệt vời đối với các vấn đề khó khăn như nhận dạng giọng nói và nhận diện hình ảnh. Tuy nhiên, các mạng này chỉ có thể được áp dụng cho các bài toán mà đầu vào và mục tiêu được mã hóa thành các vector phù hợp có chiều cố định, do đó xử lý không tốt cho các bài toán với đầu vào là các chuỗi có độ dài không biết trước như bài toán nhận dạng giọng nói hay dịch máy. Để giải quyết vấn đề này, mô hình mạng Sequence to Sequence [9] ra đời với 2 tầng mã hóa (encoder) và giải mã (decoder). Tầng mã hóa có nhiệm vụ tại mỗi bước thời gian nhận chuỗi dữ liệu đầu vào và mã hóa chuỗi thành các vector có độ dài cố định gọi là vector ngữ cảnh. Sau đó, bộ giải mã sẽ lần lượt sinh từng từ trong chuỗi đầu vào dựa trên vector ngữ cảnh và những từ được dự đoán trước đó cho tới ký gấp từ kết thúc câu. Trong mô hình Sequence to Sequence, ta có thể sử dụng các kiến trúc mạng khác nhau cho tầng mã hóa và giải mã như mạng RNN hay CNN. Chi tiết các bước tính toán trong mô hình khi sử dụng mạng RNN cho tầng mã hóa và giải mã sẽ được trình bày trong phụ lục A.3.

Hình 6 mô tả một ví dụ về mô hình Sequence to Sequence trong hệ thống trả lời câu hỏi tự động bao gồm hai phần câu hỏi và câu trả lời, trong đó, mạng LSTM được sử dụng tại các lớp mã hóa và giải mã. Trong ví dụ này, ta đã có một chuỗi đầu vào là “How are you?”, sau khi được tách từ và đưa vào mô hình, lớp giải mã sẽ nhận trạng thái ẩn cuối cùng của lớp mã hóa, chừa thông tin cả câu hỏi đầu vào để tạo từng từ tại mỗi bước thời gian, cuối cùng chuỗi kết quả đầu ra được tạo là: “I am good”. Trong lớp giải mã, đầu vào đầu tiên sẽ là token <GO> để đánh dấu bắt đầu quá trình giải mã tại bước thời gian này.



Hình 6 Mô hình Sequence to Sequence [10].

2.2 Một số kỹ thuật nâng cao

2.2.1 Kỹ thuật attention trong mô hình Sequence to Sequence

Dựa trên cơ chế sự chú ý của con người, kỹ thuật Attention (chú ý) đã được đưa ra và áp dụng trong các mô hình mạng nơ-ron trong những nghiên cứu gần đây, đặc biệt là trong các bài toán dịch máy [9] và bài toán tóm tắt tóm lược văn bản [2], cho phép mô hình tập trung vào những thông tin quan trọng hơn là các thông tin không cần thiết khác.

Một vấn đề trong mô hình Sequence to Sequence đó là có hiệu năng kém với các chuỗi đầu vào và đầu ra dài bởi các biểu diễn bên trong có kích thước cố định trong lớp mã hóa. Hơn nữa, trạng thái ẩn cuối cùng của lớp mã hóa - trạng thái được sử dụng làm đầu vào của bộ giải mã, chứa phần lớn thông tin từ những phần tử cuối của lớp mã hóa, do đó nó có thể bị mất mát thông tin từ những phần tử ở đầu. Kỹ thuật Attention được thêm vào để giải quyết vấn đề hạn chế này. Attention hoạt động dựa trên việc cung cấp một vector ngữ cảnh mới giàu thông tin hơn từ lớp mã hóa đến lớp giải mã và cung cấp một cơ chế học trong đó, lớp giải mã sẽ học được những vị trí cần tập trung sự chú ý trong vector ngữ cảnh mới khi dự đoán từng từ trong chuỗi đầu ra tại mỗi bước thời gian. Trong bài toán tóm tắt tóm lược văn bản, việc sử dụng Attention giúp cho tại các bước thời gian của bộ giải mã, việc tính phân phối trên tập

từ điển sẽ được tập trung hơn vào các từ quan trọng trong văn bản nguồn có giá trị attention cao. Chi tiết các quá trình tính toán của kỹ thuật Attention trong mô hình Sequence to Sequence để giải quyết bài toán tóm tắt lược văn bản sẽ được trình bày trong phần 3.1.

2.2.2 Mạng Pointer

Mạng nơ-ron hồi quy (RNN) đã được sử dụng cho các bài toán cần học trên các chuỗi qua các bài toán cụ thể trong nhiều thập kỷ qua. Tuy nhiên, kiến trúc này giới hạn các cài đặt, trong đó yêu cầu kích thước của từ điển các từ sinh ra phải cố định trước. Điều này dẫn đến một vấn đề khi từ đầu ra không có trong tập từ điển (out-of-vocabulary - OOV). Do đó, mạng Pointer [11] (con trỏ) ra đời giúp giải quyết vấn đề này, đây là một cơ chế để sinh ra các từ không chỉ trong tập từ điển cố định mà còn trong cả chuỗi đầu vào. Để có thể biểu diễn từ điển kích thước thay đổi, mạng Pointer sử dụng một phân phối xác suất softmax như là một “con trỏ”. “Con trỏ” này sẽ được sử dụng như một công tắc mềm để chọn giữa việc tạo ra một từ trong từ điển hay sao chép lại một từ trong chuỗi đầu vào, đảm bảo hạn chế tối đa việc tạo ra các từ OOV.

Trong bài toán tóm tắt lược văn bản, mạng Pointer được đưa vào giúp mở rộng từ điển gốc, hạn chế được các trường hợp OOV, chi tiết sẽ được trình bày trong phần 3.2.1.

2.2.3 Kỹ thuật Coverage

Kỹ thuật Coverage được đề xuất ban đầu trong các bài toán của các hệ thống dịch máy thần kinh (Neural Machine Translation - NMT) [12]. Bằng việc học cách đóng từ và dịch, kỹ thuật Attention đã giúp tăng hiệu quả đáng kể của các hệ dịch máy thần kinh hiện đại. Tuy nhiên, các mô hình này có xu hướng bỏ qua các thông tin được đóng từ trước đó, dẫn đến việc bản dịch bị rườm rà, thừa nội dung (over-translation) hay dịch sót ý trong văn bản gốc (under-translation). Để giải quyết vấn đề này, kỹ thuật coverage duy trì một vector bao phủ (coverage vector) để lưu dấu các attention trong quá khứ. Vector bao phủ được đưa vào mô hình Attention để điều chỉnh

attention trong tương lai, cho phép hệ thống NMT có thể xem xét thêm về các từ nguồn chưa được dịch.

Trong hệ thống NMT, tại mỗi bước thời gian i của bộ giải mã, một vector bao phủ từ bước thời gian $i - 1$ được sử dụng như là một thành phần đầu vào bổ sung cho mô hình attention, điều này sẽ giúp cung cấp thông tin bổ sung về khả năng các từ nguồn đã được dịch trong quá khứ. Các thông tin bao phủ thêm vào này được kỳ vọng sẽ giúp mô hình tập trung hơn vào các từ nguồn chưa được dịch. Trong thực tế, mô hình bao phủ này thực hiện tốt như những gì được kỳ vọng.

Với bài toán tóm tắt lược văn bản, việc sử dụng Coverage giúp giảm thiểu được các trường hợp các từ bị lặp lại trong các bản tóm tắt sinh ra bởi mô hình Pointer-Generator. Trong kỹ thuật này, mô hình sẽ duy trì một vector bao phủ tương tự như trong các hệ thống NMT là tổng của các phân phối attention trên tất cả các bước thời gian trước đó của bộ giải mã. Vector này có thể được xem như là một phân phối không chuẩn hóa trên các từ của văn bản nguồn, biểu diễn độ bao phủ mà các từ đó nhận được từ cơ chế Attention cho đến bước thời gian hiện tại. Trong đó, một hàm mât mát bao phủ sẽ được bổ sung vào hàm mât mát tổng quát của mô hình, giúp phạt các trường hợp trùng lặp giữa phân phối attention và độ bao phủ đến thời điểm hiện tại, ngăn chặn sự chú ý lặp lại tại cùng một vị trí. Chi tiết về các bước thực hiện của kỹ thuật này trong mô hình Pointer-Generator sẽ được trình bày trong phần 3.2.2.

2.3 Word embedding

2.3.1 Tổng quan

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), để máy tính có thể hiểu được các từ, ta phải ánh xạ các từ thành các vector chứa các giá trị số. Trước đây, trong các phương pháp tiếp cận truyền thống, các từ sẽ được biểu diễn dưới dạng các one-hot vector, một vector mà chỉ có một phần tử bằng 1 còn lại có giá trị 0. Độ dài của vector này chính bằng kích thước của bộ từ điển được tạo ra từ dữ liệu, thông thường từ điển này đã được sắp xếp theo thứ tự bảng chữ cái, vị trí trong vector có giá trị bằng 1 chính là vị trí của từ cần biểu diễn. Kết quả là với mỗi câu sẽ tạo ra một ma trận biểu diễn

trong đó hầu hết các phần tử có giá trị 0. Cách tiếp cận này có nhiều hạn chế, đó là ma trận biểu diễn có số chiều rất lớn ($D \times V$) với D là số từ có trong văn bản, V là số từ có trong từ điển). Một hạn chế nữa là các từ được biểu diễn bình đẳng, không thể hiện được mối quan hệ với nhau. Để khắc phục các nhược điểm này, các phương pháp word embedding đã ra đời.

Word embedding là một loại ánh xạ cho phép các từ có ý nghĩa tương tự nhau sẽ được biểu diễn bằng các vector tương đồng với nhau. Có được điều này là do word embedding có khả năng nắm bắt được ngữ cảnh của một từ trong một tài liệu, ngữ pháp và ngữ nghĩa tương đồng, quan hệ với các từ khác... Trong các phương pháp word embedding thì hai trong số các phương pháp hiện đại nhất đó là Word2Vec và Fasttext.

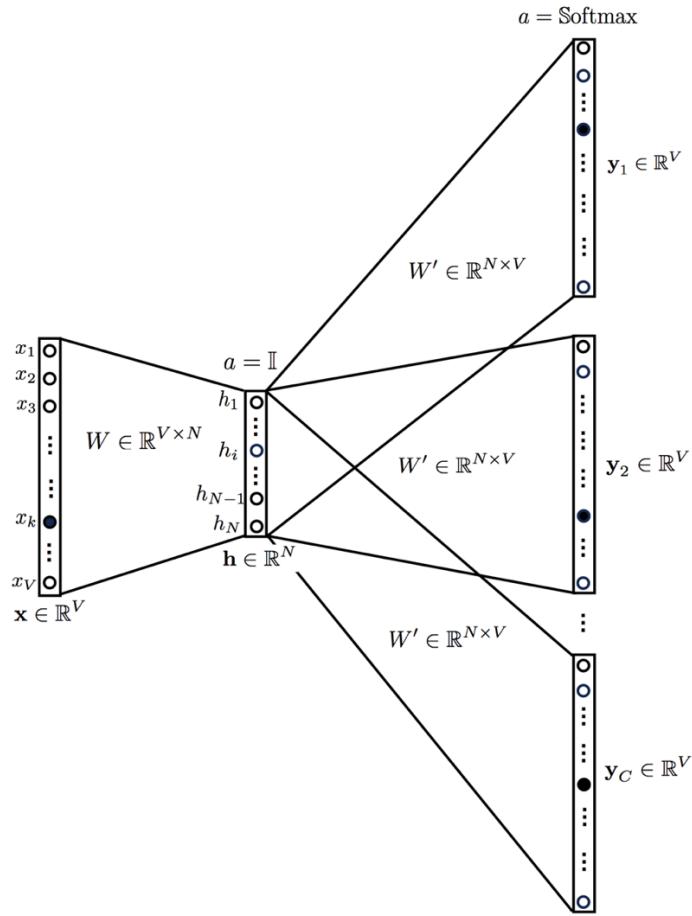
2.3.2 Word2Vec

Word2Vec được ra đời vào năm 2013 bởi một kỹ sư ở Google có tên là Tomas Mikolov. Về cơ bản, Word2Vec là một mô hình học không giám sát, được huấn luyện trên một tập dữ liệu lớn. Chiều của Word2Vec nhỏ hơn nhiều so với chiều của phương pháp mã hóa dạng one-hot vector, với số chiều là $D \times E$ với D là số từ có trong văn bản, E là số chiều của vector word embedding. Có 2 mô hình nổi tiếng của Word2Vec đó là Skip-gram và Continuous Bag of Words (CBOW).

Skip-gram

Trong mô hình skip-gram, đầu vào là một từ cho trước, thực hiện dự đoán đầu ra là các từ xung quanh từ đó. Ví dụ, trong câu “I have a pretty cat”, nếu sử dụng một cửa sổ tìm kiếm có kích thước 3, ta thu được: $\{(I, a), have\}$, $\{(have, pretty), a\}$, $\{(a, cat), pretty\}$. Với mỗi từ, ví dụ từ “have” thì kết quả thu được sẽ là các từ “I”, “a”.

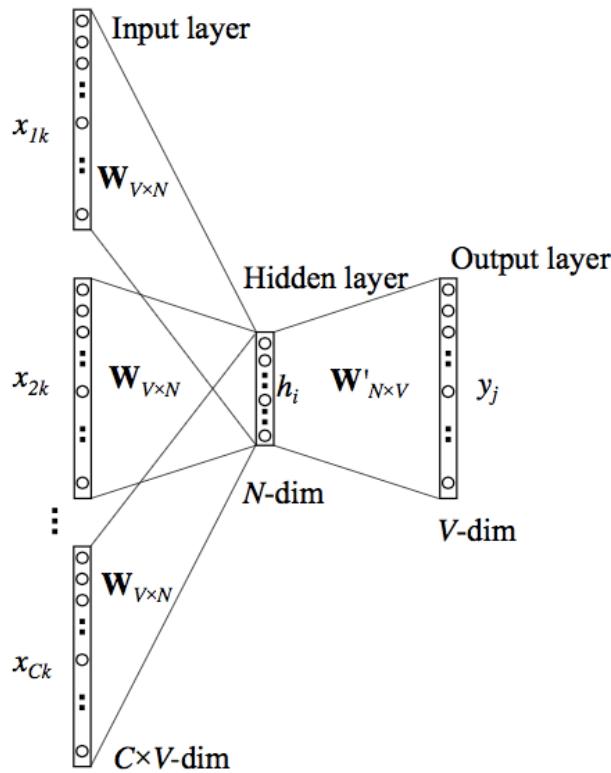
Mô hình Skip-gram được mô tả như trong Hình 7. Trong đó, mỗi đầu vào sẽ là một từ với one-hot vector tương ứng của nó, $x = (x_1, x_2, \dots, x_V)$, với V là kích thước của từ điển. Ma trận trọng số giữa tầng đầu vào và tầng ẩn là W , có kích thước là $V \times N$, ma trận trọng số giữa tầng ẩn và tầng đầu ra là W' , có kích thước là $N \times V$. Hàm kích hoạt (activation function) của đầu ra là hàm softmax.



Hình 7 Mô hình Skip-gram trong Word2Vec [13].

CBOW

Gần tương tự với Skip-gram, trong mô hình CBOW, khi cho một ngữ cảnh gồm các từ đầu vào, mô hình sẽ có gắng dự đoán ra từ phù hợp với ngữ cảnh đó. Điểm khác biệt lớn nhất giữa Skip-gram và CBOW đó là cách mà các vector biểu diễn từ được tạo ra. Trong CBOW, tất cả các từ trong ngữ cảnh đầu vào được đưa vào mạng, sau đó tính trung bình của các trạng thái ẩn được sinh ra trong tầng ẩn. Ví dụ, giả sử ta có hai câu: “Anh ấy là một người tốt”, “Cô ấy là một giáo viên”. Để tính toán biểu diễn của từ “một”, ta cần đưa vào mạng nơ-ron hai ví dụ: “Anh ấy là người tốt”, “Cô ấy là giáo viên”, sau đó tính trung bình giá trị của các trạng thái ẩn trong tầng ẩn. Trong khi ở mô hình Skip-gram, đầu vào chỉ là một từ được biểu diễn dưới dạng one-hot vector. Mô hình CBOW được mô tả như trong Hình 8 như sau:



Hình 8 Mô hình CBOW trong Word2Vec [14].

Cả hai mô hình Skip-gram và CBOW đều có những điểm mạnh và điểm yếu riêng. Khi thực nghiệm, Skip-gram cho thấy nó hoạt động tốt trên các bộ dữ liệu nhỏ và được sử dụng để biểu diễn các từ hiếm tốt. Mặc khác, CBOW nhanh hơn và có biểu diễn tốt hơn cho các từ thường gặp.

2.3.3 Fasttext

Fasttext là một mở rộng của Word2Vec, được đề xuất bởi Facebook vào năm 2016. Thay vì đưa các từ đơn lẻ vào mạng nơ-ron, Fasttext chia các từ thành nhiều phần n-grams (sub-words). Ví dụ, với tri-grams cho từ “school” sẽ là {“sch”, “cho”, “hoo”, “ool”} (bỏ qua các ký tự bắt đầu và kết thúc của từ). Vector embedding cho từ “school” sẽ là tổng của các thành phần n-grams này. Sau khi huấn luyện mạng nơ-ron sẽ thu được word embedding cho tất cả các n-grams thu được từ tập huấn luyện. Nếu như trong Word2vec, các từ không có trong từ điển sẽ không có vector biểu diễn, thì ở trong Fasttext, các từ hiếm có thể được biểu diễn chính xác vì rất có khả năng

các n-grams của nó cũng xuất hiện trong các từ khác. Nhờ đó, Fasttext được ứng dụng trong rất nhiều bài toán khác nhau trong xử lý ngôn ngữ tự nhiên.

Tương tự như Word2Vec, Fasttext cũng được huấn luyện bằng mô hình Skip-gram hoặc CBOW.

Kết chương

Chương này đã trình bày các kiến thức cơ sở lý thuyết liên quan được sử dụng, trong đó đã giới thiệu về mạng nơ-ron nhân tạo, sau đó mạng RNN cũng như LSTM, mô hình Sequence to Sequence. Bên cạnh đó, chương này cũng trình bày thêm về các kỹ thuật nâng cao được sử dụng, trong đó có kỹ thuật Attention giúp mô hình Sequence to Sequence tập trung vào những thông tin quan trọng trong xâu đầu vào, mạng Pointer giúp giải quyết các bài toán sinh ra các từ không chỉ trong tập từ điển cố định mà trong cả chuỗi đầu vào. Tiếp theo là kỹ thuật Coverage, duy trì một vector bao phủ để điều chỉnh attention trong tương lai. Cuối cùng, chương trình này trình bày về các phương pháp Word embedding biểu diễn từ hiện đại nhất hiện nay, bao gồm Word2Vec và Fasttext. Ở chương tiếp theo, Chương 3 sẽ trình bày về các kết quả nghiên cứu tương tự, hiện đại nhất (state-of-the-art) của bài toán tóm tắt lược văn bản.

Chương 3 Tình hình nghiên cứu hiện nay

Chương 2 đã trình bày về cơ sở lý thuyết, các kiến trúc liên quan được sử dụng trong mô hình tóm tắt tóm lược văn bản. Chương 3 này trình bày về ngữ cảnh của bài toán và các kết quả nghiên cứu tương tự. Với mục tiêu và phạm vi đề tài, các nội dung được trình bày bao gồm: (i) Mô hình Sequence to Sequence kết hợp kỹ thuật Attention trong phần 3.1 và (ii) Mạng Pointer-Generator kết hợp Coverage trong phần 3.2, được đề xuất bởi Abigail See trong “Get To The Point: Summarization with Pointer-Generator Networks” năm 2017 [1].

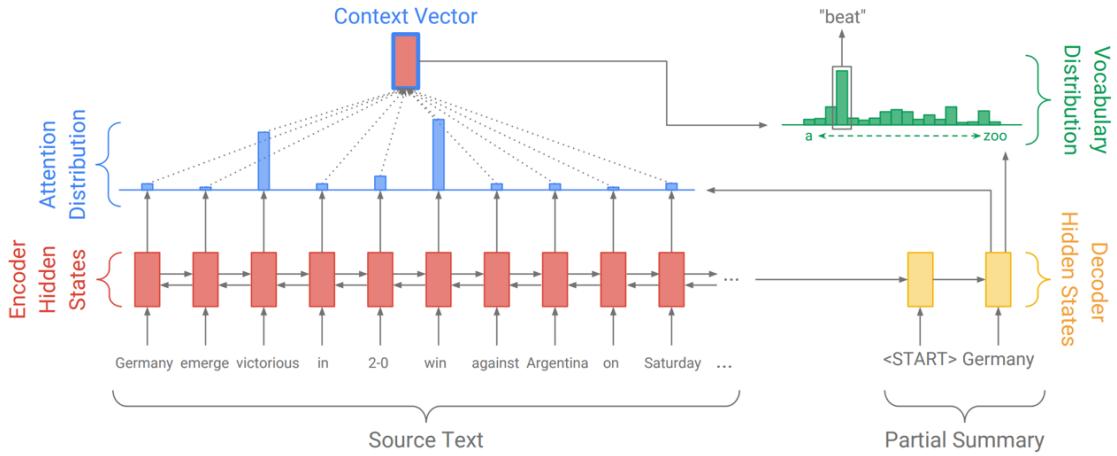
3.1 Mô hình Sequence to Sequence kết hợp Attention

Mô hình Sequence to Sequence kết hợp Attention này tương tự với mô hình được đề xuất bởi Nallapati và cộng sự (2016) [2]. Mô hình này đã giải quyết vấn đề về độ dài chuỗi động trong tóm tắt tự động, rất phổ biến khi áp dụng trong các mô hình DNNs (Deep Neural Network). Trong suốt quá trình huấn luyện và thực nghiệm, họ đã cắt ngắn bài báo thành 400 token và giới hạn độ dài của bản tóm tắt là 100 token trong quá trình huấn luyện, 120 token khi thực nghiệm. Tập dữ liệu được sử dụng là tập CNN/Daily Mail đã được sử dụng trong [2], bao gồm các bài báo (có độ dài trung bình là 39 câu) được ghép cặp với bản tóm tắt đa câu.

3.1.1 Mô hình

Mô hình Sequence to Sequence kết hợp Attention được minh họa như trong Hình 9. Mô hình này dựa trên mô hình Encoder-Decoder (Mã hóa-Giải mã) và kỹ thuật Attention.

Ở đây, tầng mã hóa (encoder) là một tầng LSTM hai chiều, tầng giải mã (decoder) là một tầng LSTM một chiều. Mỗi token của bài báo w_i được đưa vào tầng mã hóa tạo ra một chuỗi các trạng thái ẩn của tầng mã hóa h_i .



Hình 9 Mô hình Sequence to Sequence kết hợp Attention [1].

Tại mỗi bước thời gian t , tầng giải mã nhận word embedding của từ trước đó (trong huấn luyện, đây là từ trong bản tóm tắt tham chiếu (reference summary); tại quá trình thực nghiệm, đây là từ được sinh ra từ bước thời gian trước được tạo ra bởi tầng giải mã) làm đầu vào cho tầng giải mã, và có trạng thái ẩn của tầng giải mã tương ứng s_t . Phân phối attention (attention distribution) a^t được tính qua Công thức 3 như sau:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn})$$

$$a^t = \text{softmax}(e^t)$$

Công thức 3 Công thức tính attention distribution.

Trong đó, v , W_h , W_s và b_{attn} là các tham số học được.

Giá trị e_i^t trong Công thức 3 còn được gọi là scores (điểm), cho thấy mức độ khớp nhau của đầu vào xung quanh vị trí i và đầu ra ở vị trí t . Phân phối attention trên có thể được xem là phân phối xác suất trên các từ trong văn bản nguồn. Tiếp theo, phân phối attention được sử dụng để tạo ra vector tổng có trọng số của các trạng thái ẩn của tầng mã hóa, được gọi là vector ngữ cảnh h_t^* , được tính như trong Công thức 4:

$$h_t^* = \sum_i a^t_i h_i$$

Công thức 4 Công thức tính vector ngữ cảnh.

Vector ngữ cảnh h_t^* có thể được xem là vector biểu diễn kích thước cố định của những gì đã được đọc từ văn bản nguồn, được ghép (concatenate) với trạng thái s_t của bộ mã hóa, và được đưa qua hai lớp tuyến tính để tạo ra phân phối từ điển (vocabulary distribution), được tính qua Công thức 5:

$$P_{vocab} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b')$$

Công thức 5 Công thức tính phân phối từ vựng.

Trong đó, V , V' , b và b' là các tham số học được. P_{vocab} là phân phối xác suất trên tất cả các từ trong bộ từ điển, và nó cung cấp phân phối cuối cùng để dự đoán từ w như trong Công thức 6:

$$P(w) = P_{vocab}(w)$$

Công thức 6 Công thức tính phân phối dự đoán từ w .

Trong suốt quá trình huấn luyện, giá trị mất mát (loss) cho mỗi bước thời gian t được tính qua hàm negative log likelihood của từ đích w_t^* tại bước thời gian đó, được thể hiện Công thức 7:

$$loss_t = -\log P(w_t^*)$$

Công thức 7 Công thức tính giá trị mất mát tại bước thời gian t .

Và giá trị loss chung cho toàn bộ chuỗi được tính qua Công thức 8 như sau:

$$loss = \frac{1}{T} \sum_{t=0}^T loss_t$$

Công thức 8 Công thức tính giá trị mất mát cho toàn bộ chuỗi.

3.1.2 Đánh giá

Mặc dù mô hình triển rất có triển vọng, nhưng nó cũng gặp phải những vấn đề không mong muốn. Đầu tiên đó là khả năng xử lý các từ nằm ngoài tập từ điển (out-of-vocabulary - OOV). Do từ điển có kích thước cố định, mô hình trên không thể biểu diễn các từ mà không có trong từ điển. Vấn đề này sẽ được giải quyết bằng cách sử

dụng cơ chế Pointer-Generator sẽ được trình bày trong phần tiếp theo. Vấn đề gặp phải tiếp theo đó là bản tóm tắt bị lặp lại các nội dung trong nó. Trong phần tiếp theo, một kỹ thuật tên là Coverage sẽ được tích hợp vào mạng để giúp điều chỉnh attention và cải thiện đáng kể sự liên kết tổng thể giữa câu nguồn và câu đích, giải quyết vấn đề lặp lại nêu trên.

3.2 Mô hình Pointer-Generator kết hợp Coverage

Mạng Pointer-Generator sử dụng mô hình Sequence to Sequence kết hợp Attention trong phần 3.1 làm mô hình cơ sở. Mô hình cơ sở được kết hợp với kỹ thuật Pointer-Generator để giải quyết vấn đề của các từ OOV, bên cạnh đó sử dụng cơ chế coverage để tránh lặp lại các từ. Trong quá trình huấn luyện và thực nghiệm, tương tự như trong mô hình Sequence to Sequence kết hợp Attention trong phần 3.1, họ đã cắt ngắn mỗi bài báo thành 400 token và giới hạn độ dài của bản tóm tắt là 100 token trong quá trình huấn luyện, 120 token khi thực nghiệm. Tập dữ liệu được sử dụng là tập CNN/Daily Mail đã được sử dụng trong [2], bao gồm các bài báo (có độ dài trung bình là 39 câu) được ghép cặp với bản tóm tắt đa câu.

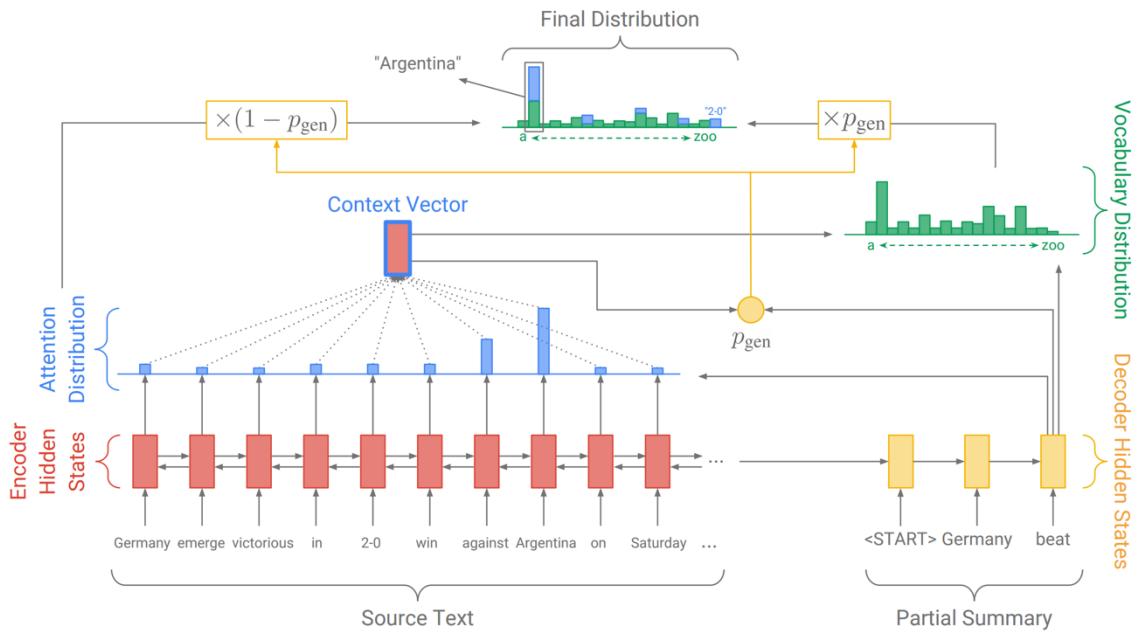
3.2.1 Mạng Pointer-Generator

Mạng Pointer-Generator là một mạng lai giữa mô hình cơ sở trong phần 3.1 và mạng Pointer [11], nó vừa cho phép sao chép từ thông qua con trỏ, vừa cho phép tạo ra từ từ một từ điển cố định. Mô hình Pointer-Generator được mô tả như trong Hình 10.

Trong đó, phân phối attention a^t và vector ngữ cảnh h_t^* được tính như trong Công thức 4. Ngoài ra, một xác suất tạo $p_{gen} \in [0, 1]$ cho mỗi bước thời gian t được tính dựa trên vector ngữ cảnh h_t^* , trạng thái s_t của bộ giải mã và đầu vào của bộ giải mã x_t , thể hiện qua Công thức 9 như sau:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr})$$

Công thức 9 Công thức tính xác suất tạo p_{gen} .



Hình 10 Mạng Pointer-Generator trong tóm tắt tóm lược văn bản [1].

Trong Công thức 9, w_{h^*} , w_s , w_x và đại lượng vô hướng b_{ptr} là các tham số học được, σ là hàm sigmoid. Tiếp theo, p_{gen} được sử dụng như một công tắc mềm để chọn giữa việc tạo ra một từ từ điển bằng cách lấy mẫu từ P_{vocab} , hoặc sao chép một từ từ chuỗi đầu vào bằng cách lấy mẫu từ phân phối attention a^t . Với mỗi tài liệu, xác định một từ điển mở rộng là hợp của từ điển và tất cả các từ xuất hiện trong tài liệu nguồn. Một phân phối trên tập từ điển mở rộng được tính qua Công thức 10 như sau:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a^t_i$$

Công thức 10 Công thức tính phân phối trên tập từ điển mở rộng.

Chú ý rằng với w là một từ không nằm trong tập từ điển (OOV), thì $P_{vocab}(w)$ là 0, tương tự nếu w không xuất hiện trong tài liệu nguồn thì $\sum_{i:w_i=w} a^t_i$ có giá trị 0. Khả năng tạo ra các từ không nằm trong tập từ điển là một trong những lợi thế chính của các mô hình Pointer-Generator, bởi các mô hình như mô hình cơ sở bị giới hạn trong tập từ điển được thiết lập sẵn.

Hàm mất mát (loss function) được mô tả như trong Công thức 7 và Công thức 8, nhưng $P(w)$ ở đây là $P(w)$ đã được thay đổi như trong Công thức 10.

Minh họa trong Hình 10, tại mỗi bước thời gian của decoder, một xác suất tạo $p_{gen} \in [0, 1]$ được tính, thực hiện tính xác suất của việc tạo ra từ trong từ điển, so với sao chép từ từ văn bản nguồn. Phân phối từ điển và phân phối attention được tính trọng số, sau đó tính tổng lại để thu được phân phối cuối cùng, từ đó mô hình đưa ra dự đoán của mình.

3.2.2 Mạng Pointer-Generator kết hợp Coverage

Sự lặp lại là một vấn đề phổ biến trong các mô hình Sequence to Sequence, đặc biệt là việc tạo ra các bản tóm tắt đa câu. Để giải quyết vấn đề này, mô hình Coverage được đưa ra, nó duy trì một vector bao phủ c^t là tổng của các phân phối attention trên tất cả các bước thời gian trước đó của bộ giải mã, được tính như trong Công thức 11:

$$c^t = \sum_{t'=0}^{t-1} a^{t'}$$

Công thức 11 Công thức tính vector bao phủ.

Một cách trực quan, c^t là một phân phối (không chuẩn hóa) trên các từ trong văn bản nguồn, biểu diễn độ bao phủ mà các từ đó đã nhận được từ cơ chế attention cho đến nay. Lưu ý rằng c^0 là một vector không, bởi vì ở bước thời gian đầu tiên, không có tài liệu nguồn nào được bao phủ. Vector bao phủ được sử dụng như là một đầu vào mở rộng trong kỹ thuật attention, biến đổi e_i^t trong Công thức 3 thành:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + w_c c^t + b_{attn})$$

Công thức 12 Công thức biến đổi scores.

Trong Công thức 12, w_c là tham số vector học được, có cùng độ dài với v . Điều này đảm bảo rằng quyết định hiện tại của cơ chế attention (Chọn vị trí từ tiếp theo) được thông báo bởi một lời nhắc nhớ về các quyết định trước đó (được tóm tắt trong c^t). Điều này sẽ giúp cho cơ chế attention dễ dàng hơn trong việc tránh việc bị lặp lại nhiều lần tại cùng một vị trí, và do đó tránh được việc tạo ra văn bản bị lặp lại.

Một giá trị mất mát bao phủ (coverage loss) được thêm vào để phạt những trường hợp xuất hiện nhiều lần tại cùng một vị trí, được mô tả như trong Công thức 13:

$$covloss_t = \sum_i \min(a_i^t, c_i^t)$$

Công thức 13 Công thức tính mất mát bao phủ.

Chú ý rằng, giá trị mất mát bao phủ bị chặn, đặc biệt $covloss_t \leq \sum_i a_i^t = 1$. Công thức 13 khác với mất mát bao phủ trong dịch máy. Trong dịch máy, ta giả sử cần có một tỷ lệ dịch một-một, theo đó, vector bao phủ cuối cùng sẽ bị phạt nếu nó bé hoặc lớn hơn 1.

Hàm mất mát này rất linh hoạt: Vì việc tóm tắt không yêu cầu bao phủ thống nhất, nên chỉ thực hiện xử phạt sự trùng lặp giữa phân phối attention và độ bao phủ cho đến nay – ngăn chặn sự chú ý lặp lại, tức là lặp lại tại vị trí attention đó(attention). Cuối cùng, giá trị mất mát bao phủ được nhân với một hằng số λ , được cộng vào hàm mất mát chính để tạo ra hàm mất mát tổng hợp mới, được tính như trong Công thức 14:

$$loss_t = -\log P(w^* | t) + \lambda \sum_i (a_i^t, c_i^t)$$

Công thức 14 Công thức hàm mất mát tổng quát mới.

3.2.3 Đánh giá

Về cơ bản, mạng này đã giải quyết các vấn đề về từ nằm ngoài tập từ điển và sự lặp lại của các từ, điều mà mô hình cơ sở Sequence to Sequence kết hợp Attention gặp phải. Mặc dù mạng này có thể tốt trong việc tạo ra các từ nằm ngoài từ điển và cũng tránh được các cụm từ bị lặp lại, nhưng nó không hoàn toàn giải quyết được vấn đề biểu diễn từ nằm ngoài từ điển. Những từ mới đều được biểu diễn bởi token “UNK” trước khi nó được đưa vào các lớp tiếp theo, điều này có thể gây ra việc mất mát thông tin. Bên cạnh đó, mạng hiện tại đang là mạng end-to-end, các vector biểu diễn mỗi từ được cập nhật qua mỗi bước, nên nếu ở một số lượng vòng lặp lớn nhất định, các vector biểu diễn này có thể bị overfitting (vấn đề này đã được trình bày trong

2.1.1) với tập dữ liệu huấn luyện. Một trong những giải pháp được đề xuất sẽ được trình bày chi tiết trong Chương 4. Đầu tiên đó là đề xuất đưa mỗi từ qua một mô hình Word2Vec đã được huấn luyện trước đó (pre-trained) để thu được biểu diễn của từ đó, mô hình Word2Vec này được huấn luyện trên một tập dữ liệu khác, lớn hơn tập dữ liệu huấn luyện hiện tại, do đó có thể tránh được trường hợp overfitting. Bên cạnh đó, với tập từ điển khác với tập từ điển sử dụng trong mạng, lớn hơn rất nhiều, nên có thể giảm thiểu được trường hợp một từ mới không có trong từ điển, được biểu diễn bởi token “UNK” trước khi đưa vào mạng, giảm thiểu việc mất mát thông tin. Đồng thời, việc biểu diễn vector của mỗi từ được xác định từ đầu trước khi đưa vào mạng, không thực hiện cập nhật lại biểu diễn biểu diễn vector đó tại mỗi bước hứa hẹn sẽ mang lại thời gian huấn luyện được rút ngắn. Đề xuất thứ hai đó là sử dụng một cơ chế biểu diễn từ mới trong mạng, sử dụng Fasttext. Mạng mới với Fasttext embedding bên cạnh các ưu điểm như trong giải pháp sử dụng Word2Vec, sẽ đảm bảo mỗi từ đều có biểu diễn của nó, kể cả khi nó không tồn tại trong tập từ điển, vấn đề này sẽ gặp phải khi từ đầu vào không có trong từ điển của Word2Vec.

Kết chương

Chương này đã trình bày về các kết quả nghiên cứu liên quan về bài toán tóm tắt lược văn bản, chỉ ra đặc điểm, tính chất chi tiết của mô hình Sequence to Sequence kết hợp Attention cùng các hạn chế của nó. Sau đó, trình bày về về mạng Pointer-Generator kết hợp Coverage, giải quyết các hạn chế của mô hình Sequence to Sequence, đây cũng chính là mô hình cơ sở được sử dụng cho các giải pháp đưa ra trong Chương 4. Ở chương tiếp theo, Chương 4 sẽ trình bày về các giải pháp đề xuất để cải tiến chất lượng tóm tắt văn bản lược, thực hiện thực nghiệm và đánh giá kết quả trên hai bộ dữ liệu tiếng Anh và tiếng Việt.

Chương 4 Đề xuất giải pháp và thực nghiệm

Chương 3 đã thảo luận về các kết quả nghiên cứu liên quan hiện tại của bài toán tóm tắt tóm lược văn bản. Chương 4 này sẽ trình bày các đề xuất để cải tiến chất lượng tóm tắt văn bản, chạy các thực nghiệm và đánh giá kết quả thu được.

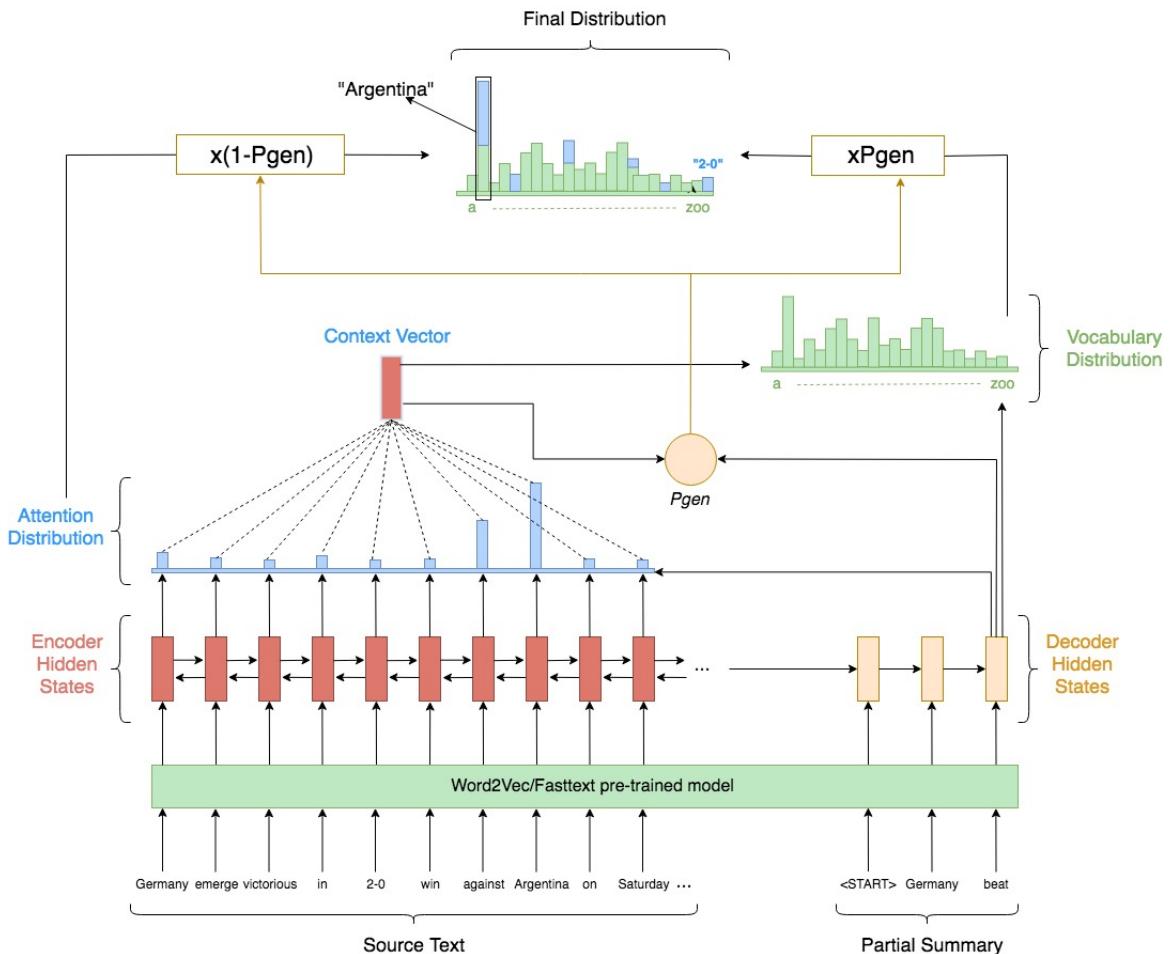
4.1 Giải pháp cải tiến đề xuất

Trong các cải tiến đề xuất, em sử dụng mô hình cơ sở là mô hình tóm tắt tóm lược văn bản sử dụng mạng Pointer-Generator kết hợp Coverage như đã trình bày trong phần 3.2.2. Trong mô hình này, mỗi từ trước khi được đưa vào mạng đều được đưa qua một lớp embedding, khởi tạo một vector embedding biểu diễn từ ngẫu nhiên, vì là mạng end-to-end nên sau đó, vector embedding này sẽ được cập nhật qua mỗi bước. Do đó, mô hình sẽ gặp phải hai vấn đề, thứ nhất đó là những từ mới đầu vào ở cả tầng giải mã và mã hóa, không có trong từ điển mở rộng sẽ được biểu diễn bởi token “UNK” trước khi nó được đưa vào lớp embedding của mạng, điều này có thể dẫn đến mất mát thông tin. Vấn đề thứ hai là do là mạng end-to-end, lớp embedding của mỗi từ sẽ được học thông qua quá trình huấn luyện cùng với mô hình, nên có thể sẽ bị overfitting với tập dữ liệu huấn luyện.

Trong phần này, em đề xuất giải pháp sử dụng các phương pháp word embedding gồm Word2Vec và Fasttext trong xử lý dữ liệu đầu vào để cải tiến mô hình. Cụ thể, trong giải pháp này, mỗi từ trước khi được đưa vào mạng, sẽ được được biểu diễn sang vector embedding tương ứng trong tập pre-trained (tập các vector embedding của các từ đã học được từ các mô hình Word2Vec, Fasttext trong bộ dữ liệu khác lớn hơn), đồng thời sẽ loại bỏ lớp embedding trong mô hình cơ sở. Do tập pre-trained được huấn luyện trên một bộ dữ liệu khác lớn hơn, với từ điển kích thước lớn hơn rất nhiều nên sẽ giảm thiểu được nhiều trường hợp một từ mới trong đầu vào của cả hai tầng giải mã và mã hóa không có trong từ điển mở rộng sẽ bị chuyển thành token

“UNK” trước khi đưa vào mạng (vấn đề OOV), giảm thiểu bị mất mát thông tin. Bên cạnh đó, lớp embedding trong mô hình cơ sở được loại bỏ, embedding của mỗi từ sẽ không được cập nhật nữa nên sẽ tránh được trường hợp overfitting với tập dữ liệu huấn luyện, đồng thời cũng vì quá trình học embedding này không còn nên sẽ rút ngắn được thời gian huấn luyện xuống

Mô hình cải tiến thu được mô tả qua Hình 11.



Hình 11 Mô hình cải tiến sử dụng Word2Vec/Fasttext pre-trained.

Thực hiện giải pháp trên, có hai hướng tiếp cận đó là sử dụng Word2Vec và Fasttext embedding. Với việc sử dụng Word2Vec embedding, embedding sẽ được biểu diễn ở mức từ, nên có thể có những trường hợp từ đầu vào không có trong từ điển của bộ pre-trained này, nó sẽ được nhận một vector khởi tạo ngẫu nhiên, dẫn đến khả năng mà nó biểu diễn các từ OOV phụ thuộc phần lớn vào từ điển của nó. Khắc phục hạn chế này của Word2Vec, với giải pháp khi sử dụng Fasttext, embedding sẽ được biểu

diễn không ở mức từ, thay đó, mỗi từ sẽ được chia làm nhiều thành phần n-grams như đã trình bày trong phần 2.3.3. Vector embedding biểu diễn cho mỗi từ sẽ là tổng của các vector tương ứng với từng thành phần n-grams trong nó. Do đó, trong Fasttext, các từ hiếm mà trong từ điển của Word2Vec trên không có, vẫn có thể được biểu diễn chính xác vì khả năng các n-grams của nó cũng xuất hiện trong các từ khác trong bộ dữ liệu huấn luyện của Fasttext.

4.2 Bộ dữ liệu

Với tiếng Anh, em sử dụng bộ dữ liệu Daily Mail/CNN được giới thiệu và sử dụng gần đây trong [2]. Bộ dữ liệu này gồm trung bình 781 từ trong mỗi bài báo, được ghép cặp với một bản tóm tắt tham chiếu đa câu, bản tóm tắt này có trung bình 3,75 câu hay 56 từ. Sau khi được xử lý như trong [2], bộ dữ liệu được chia thành: (i) 287.226 cặp dữ liệu cho quá trình huấn luyện, (ii) 13.368 cặp dữ liệu cho quá trình đánh giá (validation), (iii) 11.490 cặp dữ liệu cho quá trình thực nghiệm. Trong đó, độ dài tối đa của phần nội dung bài báo là 400 từ, phần tóm tắt tham chiếu có độ dài tối đa là 100 từ, độ dài nhỏ nhất là 35 từ, từ điển sử dụng có 50.000 từ.

Với tiếng Việt, dựa trên bộ dữ liệu gồm 1.172.931 bài báo, mỗi bài báo gồm 3 phần: (i) Headline (Tiêu đề), (ii) Sapo và (iii) Content (Nội dung bài báo). Vì sử dụng cho bài toán tóm tắt các bài báo nên em chỉ sử dụng phần sapo cho tóm tắt tham chiếu và content cho văn bản đầu vào.

Trước khi thực hiện làm dữ liệu cho mô hình, em đã thống kê trên bộ dữ liệu này theo số từ trong từng phần, thu được kết quả như trong Bảng 1.

Bảng 1 Kết quả thống kê trên bộ dữ liệu Báo Mới

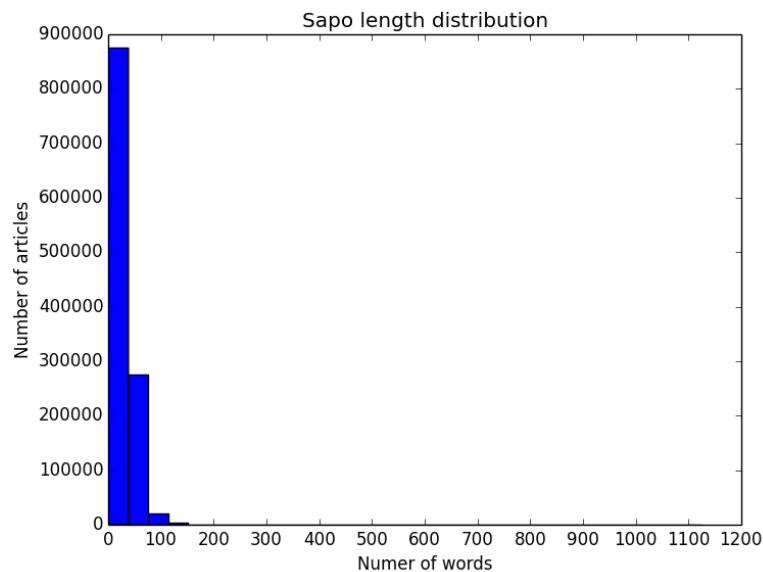
Chú thích: Đơn vị: Số từ/văn bản; Min: Số từ trong phần (Headline/Sapo/Content) của bài báo tương ứng nhỏ nhất trong cả bộ dữ liệu; Max: Số từ trong phần (Headline/Sapo/Content) của bài báo tương ứng lớn nhất trong cả bộ dữ liệu; Average: Số từ trung bình của phần tương ứng (Headline/Sapo/Content) trên cả bộ dữ liệu

Headline			Sapo			Content		
Min	Max	Average	Min	Max	Average	Min	Max	Average
21	116	8	1	1125	33	1	23929	467

Sau đó, thực hiện thống kê theo độ dài chuỗi Sapo và Content để đề xuất ngưỡng xử lý dữ liệu, được thể hiện qua các biểu đồ trong Hình 12 và Hình 13.

Từ Hình 12 và Hình 13, em đề xuất các ngưỡng xử lý dữ liệu như sau: Với phần Sapo, văn bản được lấy sẽ có độ dài tối đa là 40 từ, tối thiểu là 6 từ; Với phần Content, độ dài tối đa của văn bản được lấy là 800 từ, tối thiểu là 15 từ. Tập từ điển được xây dựng có 50.000 từ. Quá trình xử lý dữ liệu chia làm 3 bước chính.

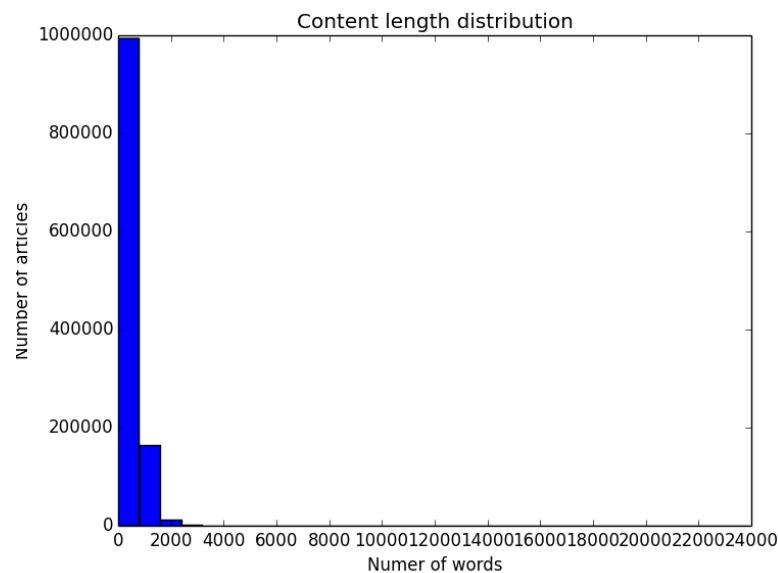
Bước 1 là lọc dữ liệu, loại bỏ các bài báo không có đủ 3 phần, loại bỏ các bài báo bị lỗi phần tóm tắt cũng như nội dung, quá ngắn hoặc quá dài, nằm ngoài ngưỡng đã đề xuất bên trên.



Hình 12 Thống kê phân phối số lượng từ trong phần Sapo.

Bước 2 thực hiện chuyển các từ sang viết thường (lowercase) toàn bộ, tách từ sử dụng bộ tách từ ViTokenizer thuộc gói pyvi [15]. Cuối cùng, sử dụng mã nguồn xử lý dữ liệu cho tiếng Anh trong [1] làm cơ sở, chỉnh sửa lại cho phù hợp với tiếng Việt. Sau quá trình xử lý dữ liệu trên, trong kết quả thu được, tập dữ liệu huấn luyện có 602.104

cặp dữ liệu, tập đánh giá có 75.263 cặp dữ liệu, tập thực nghiệm có 75.263 cặp dữ liệu với tập từ điển 50.000 từ.



Hình 13 Thống kê phân phối số lượng từ trong phần Content.

4.3 Thực nghiệm

4.3.1 Môi trường thực nghiệm

Mô hình được xây dựng và thực nghiệm trên máy tính cá nhân có cấu hình với các hệ điều hành, phần mềm như sau:

- Hệ điều hành: Ubuntu 16.04 LTS 64-bit.
- Vi xử lý: Intel Core i7 - 7700.
- RAM: 32GB.
- IDE: PyCharm.
- Ngôn ngữ sử dụng: Python 2.7.12
- GPU: Nvidia GeForce GTX 1080 Ti - 11GB VRAM.
- Bộ công cụ tách từ tiếng Việt: pyvi của tác giả Trần Việt Trung [15].
- Bộ thư viện sử dụng trong huấn luyện mô hình: Pytorch.
- Bộ thư viện sử dụng để làm việc với mô hình pre-trained Fasttext: Pyfasttext.
- Bộ thư viện sử dụng để làm việc với mô hình pre-trained Word2Vec: Gensim.

Ngoài các thư viện phần mềm trên, quá trình thực nghiệm đã sử dụng mô hình pre-trained Word2Vec cho tiếng Việt, nằm trong bộ công cụ xử lý ngôn ngữ tự nhiên cho tiếng Việt VnCoreNLP, trong [16]. Mô hình này được huấn luyện với số chiều của embedding là 300, kích thước của số từ là 2, trên bộ dữ liệu với 1.675.819 từ (đã được tách từ) khác nhau từ một tập 97.440 văn bản. Đối với tiếng Anh, em sử dụng mô hình pre-trained Word2Vec của Google [17] với 3 triệu từ, số chiều của embedding là 300, được huấn luyện trên bộ dữ liệu Google News.

Mô hình pre-trained Fasttext được sử dụng cho cả tiếng Anh và tiếng Việt đều được tải về từ trang chủ của Fasttext [18], đều với số chiều của embedding là 300.

Thư viện Pytorch

Pytorch¹ là một thư viện học máy mã nguồn mở cho Python, được xây dựng trên ngôn ngữ lập trình Lua bởi Facebook. Pytorch có hai chức năng chính: tính toán hiệu năng cao với sự tăng tốc của GPU, là một nền tảng học sâu phục vụ cho nghiên cứu, xây dựng và huấn luyện các mạng nơ-ron, mang lại sự linh hoạt và tốc độ.

So với Tensorflow², Pytorch mang lại khác nhiều ưu điểm. Đầu tiên, Pytorch mang lại khả năng debug (sửa lỗi) dễ dàng hơn theo hướng trực quan. Bên cạnh đó, nếu như ở Tensorflow, trước tiên ta cần xác định toàn bộ biểu đồ tính toán trước khi có thể chạy mô hình thì với Pytorch, nó cho phép ta xác định một biểu đồ tính toán động. Cuối cùng, Pytorch cũng hỗ trợ cả API cấp cao và API cấp thấp, với việc sử dụng tương đối dễ dàng.

4.3.2 Phương pháp đánh giá

Để đánh giá chất lượng các hệ thống văn bản, cách làm hiệu quả nhất đó là nhờ các ý kiến đánh giá từ các chuyên gia ngôn ngữ, tuy nhiên làm như vậy sẽ tốn rất nhiều thời gian và chi phí. Do đó đòi hỏi cần có một phương pháp đánh giá tự động,

¹ <https://pytorch.org/>, lần truy cập cuối: 15/05/2019.

² <https://www.tensorflow.org/>, lần truy cập cuối: 15/05/2019.

giúp dễ dàng hơn trong việc đánh giá chất lượng. Trong số đó, nổi bật là phương pháp sử dụng độ đo ROUGE. Trong các thực nghiệm đề xuất, độ đo này được em sử dụng để đánh giá kết quả của mô hình.

Recall Oriented Understudy hay ROUGE [19] là một phương pháp do Lin và Hovy đưa ra vào năm 2003 để đánh giá chất lượng tóm tắt văn bản cũng như các hệ thống dịch máy. Phương pháp này hoạt động dựa trên việc sử dụng n-grams để so sánh sự tương quan giữa bản tóm tắt do hệ thống sinh ra và bản tóm tắt tham chiếu (có thể do con người làm). Cho đến nay, phương pháp này đã được sử dụng rộng rãi trong các bài toán tóm tắt văn bản, cho kết quả đánh giá rất khả quan.

Trong ROUGE-N có thể được coi là thực hiện việc so sánh các chi tiết giữa bản tóm tắt tự động sinh ra bởi hệ thống và bản tóm tắt tham chiếu. Các n-grams có thể là unigram, bigram, trigram... Ví dụ, ROUGE-1 đề cập đến sự lặp lại của các từ đơn (unigrams) giữa bản tóm tắt tự động và bản tóm tắt tham chiếu, ROUGE-2 đề cập đến sự lặp lại các bigrams giữa bản tóm tắt tự động và bản tóm tắt tham chiếu. Cụ thể, với ROUGE-N được tính như trong Công thức 15 sau:

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

Công thức 15 Công thức tính ROUGE-N.

Trong đó, n là chiều dài của n-gram, $Count_{match}(gram_n)$ đếm số lượng tối đa các n-grams xuất hiện cả trong bản tóm tắt tham chiếu và bản tóm tắt tự động. Trong công thức trên, dưới mẫu là tổng số n-grams xuất hiện trong bản tóm tắt tham chiếu, nên có thể coi ROUGE-N như là độ bao phủ (recall).

Bên cạnh ROUGE-N, ROUGE-L đo số chuỗi “khớp” dài nhất có trong cả bản tóm tắt tham chiếu và bản tóm tắt tự động, sử dụng thuật toán trong LCS (Longest Common Subsequence - Bài toán tìm xâu con chung dài nhất). Vì đã đếm các chuỗi n-grams chung dài nhất nên không việc xác định độ dài n-grams là không cần thiết. Mặt khác, ROUGE-S đếm bất kỳ cặp từ trong một câu theo thứ tự, cho phép các khoảng trống bất kỳ, nó còn được gọi là skip-gram co-occurrence. Ví dụ, skip-bigram

đo sự lặp lại giữa các cặp từ có tối đa hai khoảng trống giữa các từ. Ví dụ câu: “tôi là sinh viên”, skip-bigrams sẽ là “tôi là, tôi sinh, tôi viên, là sinh, là viên, sinh viên”.

4.3.3 Kết quả thực nghiệm

4.3.3.1 Thực nghiệm cho tiếng Anh

Các thực nghiệm được thực hiện trên bộ dữ liệu tiếng Anh trong cả hai trường hợp có và không sử dụng Coverage, với mỗi trường hợp đều thực nghiệm kết hợp mô hình pre-trained Word2Vec/Fasttext cho đầu vào của mô hình, loại bỏ lớp embedding trong mô hình cơ sở, thu được kết quả như trong Bảng 2. Trong đó, mã nguồn của các mô hình kết hợp giải pháp cải tiến được xây dựng dựa trên mã nguồn của Atul Kumar [20], sử dụng thư viện Pytorch.

Bảng 2 Các kết quả thử nghiệm trên bộ Daily Mail/CNN

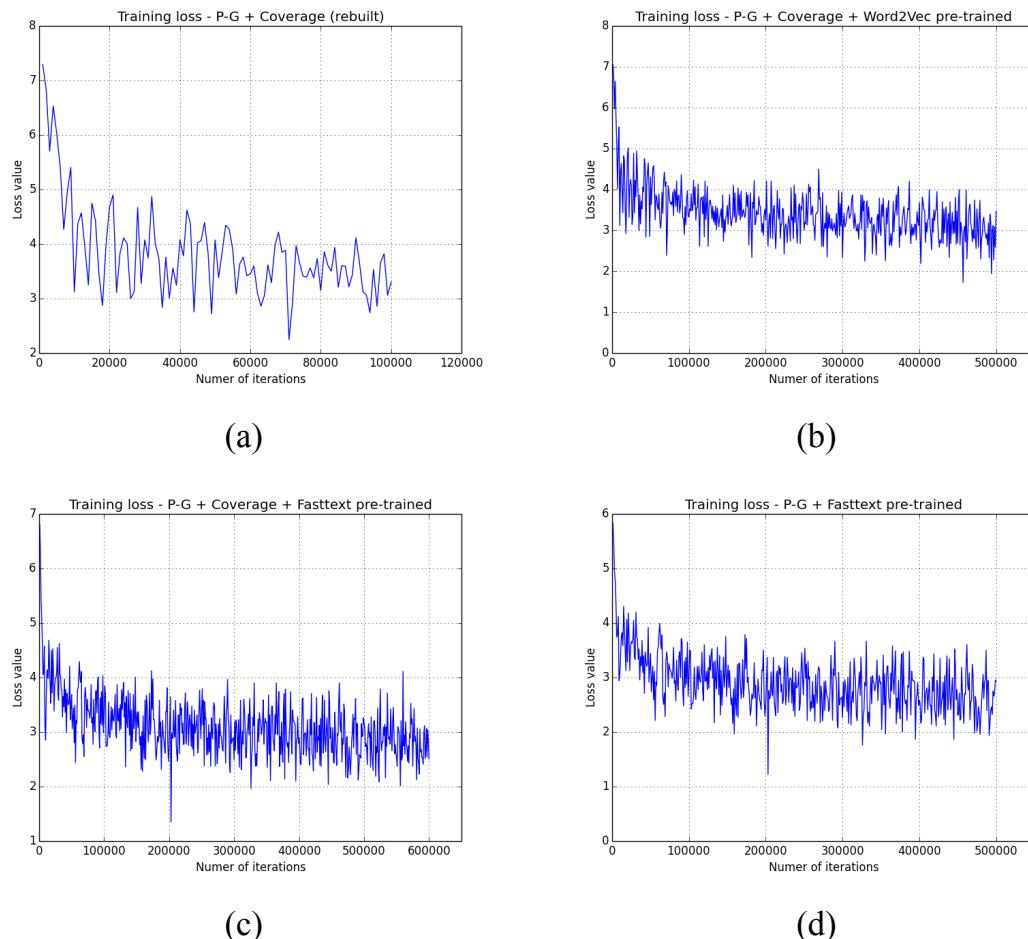
Thực nghiệm	ROUGE		
	1	2	L
Pointer-Generator (Chạy lại)	35,77	15,22	32,60
Pointer-Generator + Word2Vec pre-trained (Đè xuất)	36,14	15,31	33,06
Pointer-Generator + Fasttext pre-trained (Đè xuất)	36,28	15,61	33,10
Pointer-Generator + Coverage (Chạy lại)	38,56	16,75	35,10
Pointer-Generator + Coverage + Word2Vec pre-trained (Đè xuất)	38,33	16,55	35,16
Pointer-Generator + Coverage + Fasttext pre-trained (Đè xuất)	39,06	17,05	35,85

Kết quả thực nghiệm trong Bảng 2 cho thấy, trong cả hai trường hợp có và không sử dụng Coverage, khi thêm Word2Vec pre-trained và Fasttext pre-trained vào đều giúp cải thiện kết quả, duy chỉ có trường hợp sử dụng Coverage, kết quả mô hình có thêm Word2Vec đem lại kết quả thấp hơn mô hình gốc ở điểm ROUGE-1 và ROUGE-2. Đặc biệt với khi sử dụng Fasttext pre-trained, trong trường hợp không sử dụng

Coverage, điểm ROUGE-1 tăng 0,51 điểm, điểm ROUGE-2 tăng 0,39 điểm và điểm ROUGE-L tăng 0,5 điểm so với mô hình cơ sở.

Một số biểu đồ giá trị mất mát trong quá trình huấn luyện được thể hiện qua Hình 14.

Chú thích Hình 14: P-G: Pointer-Generator; Loss value: giá trị mất mát; rebuilt: chạy lại mô hình; Number of iterations: số vòng lặp.



Hình 14 Một số biểu đồ giá trị mất mát khi huấn luyện cho tiếng Anh.

Trong Hình 14, hình (a) là biểu đồ giá trị mất mát khi huấn luyện mô hình P-G với Coverage (chạy lại); hình (b) là biểu đồ trong huấn luyện mô hình P-G với Coverage và Word2Vec pre-trained. Tiếp theo, hình (c) là biểu đồ giá trị mất mát khi huấn luyện mô hình P-G với Coverage và Fasttext pre-trained; hình (d) là biểu đồ trong huấn luyện mô hình P-G với Fasttext pre-trained.

Trong Bảng 3 và Bảng 4 sẽ trình bày một số bản tóm tắt thu được khi sử dụng kết hợp Coverage trong: (i) mô hình cơ sở, (ii) mô hình cơ sở thêm Word2Vec pre-trained và (iii) mô hình cơ sở thêm Fasttext pre-trained.

Bảng 3 Ví dụ số 1 về bản tóm tắt tiếng Anh sinh ra từ mô hình

<p>Nội dung bài báo: Được trình bày trong phụ lục B.1.</p>
<p>Sapo (Tóm tắt tham chiếu): student is no longer on duke university campus and will face disciplinary review . school officials identified student during investigation and the person admitted to hanging the noose , duke says . the noose , made of rope , was discovered on campus about 2 a.m.</p>
<p>Kết quả tóm tắt sử dụng Pointer-Generator + Coverage: a duke student has admitted to hanging a noose made of rope from a tree near a student union . the prestigious private school did n't identify the student , citing federal privacy laws . officials are still trying to determine if other people were involved .</p>
<p>Kết quả tóm tắt sử dụng Pointer-Generator + Coverage+ Word2Vec pre-trained: the prestigious private school did n't identify the student , citing federal privacy laws . in a news release , it said the student was no longer on campus and will face student conduct review .</p>
<p>Kết quả tóm tắt sử dụng Pointer-Generator + Coverage + Fasttext pre-trained: a duke student has admitted to hanging a noose made of rope from a tree near a student union . the prestigious private school did n't identify the student , citing federal privacy laws . in a news release , it said the student was no longer on campus and will face student conduct review .</p>

Bảng 4 Ví dụ số 2 về bản tóm tắt tiếng Anh sinh ra từ mô hình

Nội dung bài báo:

Được trình bày phụ lục B.2.

Sapo (Tóm tắt tham chiếu):

abdirahman sheik mohamud pleads not guilty to charges of providing material support to terrorists and lying to the fbi .

the columbus , ohio , resident became a u.s. citizen in february 2014 .

in april 2014 , he went to syria for terrorism training , prosecutors say in a news release .

Kết quả tóm tắt sử dụng Pointer-Generator + Coverage:

abdirahman sheik mohamud , 23 , of columbus , allegedly traveled to syria for training and wanted to return home to kill americans -- particularly u.s. soldiers , execution style , the u.s. department of justice said thursday .

`` i am confident in the system working fairly and -lrb- in -rrb- our client getting a vigorous and aggressive defense , " said his lawyer , sam shamansky .

Kết quả tóm tắt sử dụng Pointer-Generator + Coverage + Word2Vec pre-trained:

new : mohamud i am confident in the system working fairly and " u.s. department of justice says .

the fbi .

abdirahman sheik mohamud , columbus , allegedly traveled to syria for training .

wanted to return home to kill americans -- particularly u.s. soldiers , execution style , mohamud allegedly said he was happy that his brother , aden , died fighting for al-nusra front , al qaeda 's largest affiliate in syria .

Kết quả tóm tắt sử dụng Pointer-Generator + Coverage + Fasttext pre-trained:

abdirahman sheik mohamud , 23 , of columbus , allegedly traveled to syria for training and wanted to return home to kill americans .

mohamud allegedly said he was happy that his brother , aden , died fighting for al-nusra front , al qaeda 's largest affiliate in syria .

mohamud told someone he planned to join aden in death soon , the indictment says .

4.3.3.2 Thực nghiệm cho tiếng Việt

Vì đặc trưng của bộ dữ liệu, hơn nữa do các bản tóm tắt tương đối ngắn (từ 6 đến 40 từ) nên khi quan sát, em thấy trường hợp bị lặp lại gần như không xảy ra, bên cạnh đó, do thời gian huấn luyện lâu và bị tối ưu hóa cục bộ nên các thực nghiệm trên bộ dữ liệu tiếng Việt sẽ không sử dụng Coverage. Trong đó, tương tự như ở phần 4.3.3.1, mã nguồn của các mô hình kết hợp giải pháp cải tiến được xây dựng dựa trên mã nguồn của Atulkum sử dụng thư viện Pytorch [20]. Chi tiết các kết quả thực nghiệm được thể hiện qua Bảng 5 như sau:

Bảng 5 Các kết quả thực nghiệm trên bộ Báo Mới

Thực nghiệm	ROUGE		
	1	2	L
Pointer-Generator	52,95	22,27	36,73
Pointer-Generator + Word2Vec pre-trained (Đề xuất)	51,43	19,63	35,24
Pointer-Generator + Fasttext pre-trained (Đề xuất)	53,44	21,53	36,49

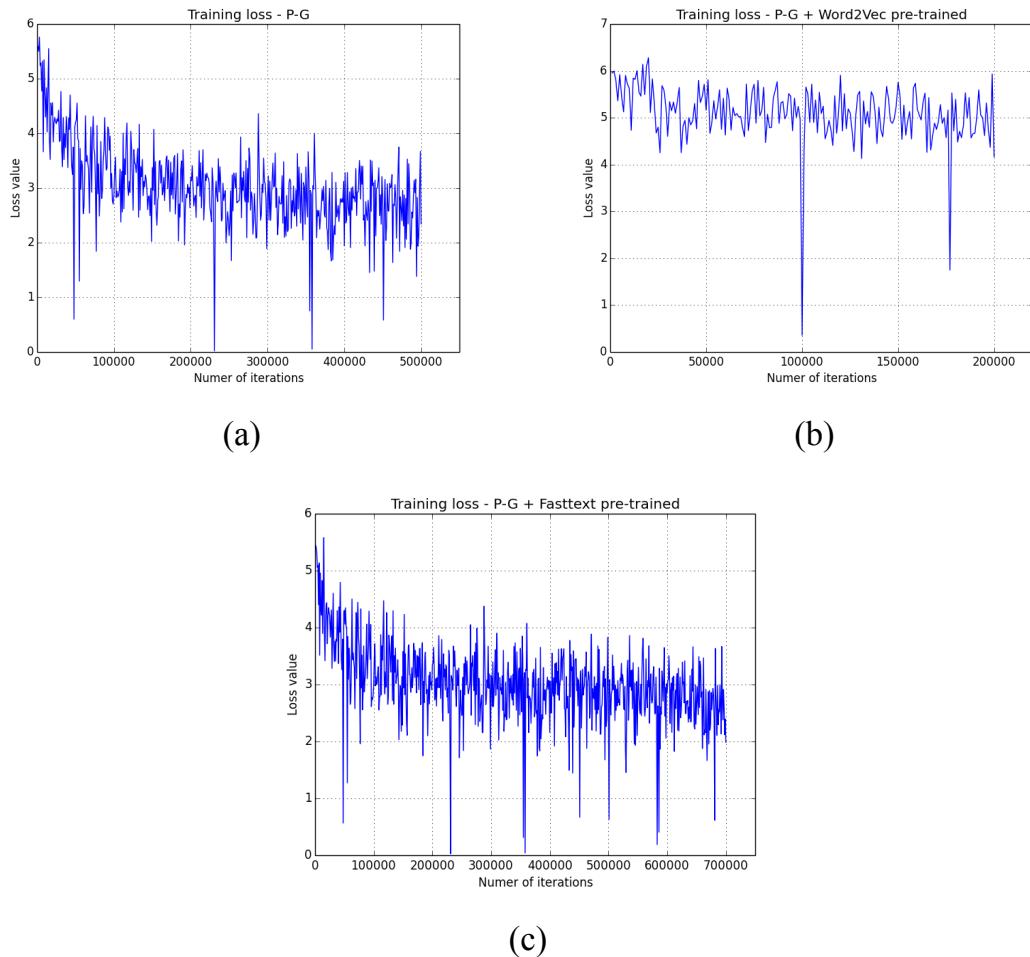
Với bộ tiếng Việt, các kết quả ROUGE thu được tương đối cao. Khi sử dụng mô hình cơ sở có thêm Word2Vec pre-trained, kết quả thấp hơn so với mô hình cơ sở, thấp hơn lần lượt là 1,52; 2,64 và 1,49 điểm cho các điểm ROUGE-1, ROUGE-2, ROUGE-L. Với khi thêm Fasttext pre-trained, điểm ROUGE-1 tăng lên được 0,49 điểm, các điểm ROUGE-2 và ROUGE-L lần lượt thấp hơn so với mô hình cơ sở là 0,74 điểm và 0,24 điểm.

Một số biểu đồ giá trị mất mát trong quá trình huấn luyện được thể hiện qua **Hình 15**.

Chú thích Hình 15: P-G: Pointer-Generator; Loss value: giá trị mất mát; rebuilt: chạy lại mô hình; Number of iterations: số vòng lặp.

Trong Hình 15, hình (a) là biểu đồ giá trị mất mát khi huấn luyện mô hình P-G; hình (b) là biểu đồ trong huấn luyện mô hình P-G với Word2Vec pre-trained. Tiếp theo,

hình (c) là biểu đồ giá trị mất mát khi huấn luyện mô hình P-G với Fasttext pre-trained.



Hình 15 Một số biểu đồ giá trị mất mát khi huấn luyện cho tiếng Việt.

Trong Bảng 6 và Bảng 7 sẽ là một số bản tóm tắt tiếng Việt thu được khi chạy: (i) mô hình cơ sở, (ii) mô hình cơ sở thêm Word2Vec pre-trained và (iii) mô hình cơ sở thêm Fasttext pre-trained. Quan sát ví dụ trong hai bảng này, do đặc trưng dữ liệu, các bản tóm tắt tham chiếu còn khá ngắn, và hầu như không tóm lại được nhiều nội dung chính của bài báo nên các bản tóm tắt sinh ra từ mô hình cũng ngắn, với nội dung mang lại không nhiều. Tuy nhiên, các bản tóm tắt được sinh ra này đã khá gần với bản tóm tắt tham chiếu, nên kỳ vọng trong tương lai, khi dữ liệu đủ tốt, bản tóm tắt tham chiếu được con người làm thì bản tóm tắt sinh ra bởi mô hình cũng tốt theo.

Bảng 6 Ví dụ số 1 về bản tóm tắt tiếng Việt sinh ra từ mô hình

Nội dung bài báo: Được trình bày trong phụ lục B.3.
Sapo (Tóm tắt tham chiếu): hai người _đẹp của showbiz việt và hoa _ngữ đều mê _mẩn thiết _kế nằm _trong bộ sưu _tập xuân hè 2014 của thương _hiệu versace , có _giá khoảng 120 triệu đồng .
Kết quả tóm tắt sử dụng Pointer-Generator: chiếc váy bạch _kim được cả ba người _đẹp yêu _thích là một thiết _kế mới của thương _hiệu versace .
Kết quả tóm tắt sử dụng Pointer-Generator + Word2Vec pre-trained: trên tạp _chí figaro số mới _nhất , lý _băng _băng khoe trọn thân _hình mảnh _mai , vòng eo con _kiến trong một thiết _kế ôm sát có

Bảng 7 Ví dụ số 2 về bản tóm tắt tiếng Việt sinh ra từ mô hình

Nội dung bài báo: Được trình bày trong phụ lục B.4.
Sapo (Tóm tắt tham chiếu): ngọn _lửa bùng lên tại khu _vực nhà _kho của nhà _máy giấy thành _đạt trong khu _công _nghiệp phong _khê (bắc _ninh) , thiêu rụi hàng trăm cuộn giấy thành _phẩm . đến 23h30 đám _cháy vẫn chưa được ché _ngự .
Kết quả tóm tắt sử dụng Pointer-Generator: ngọn _lửa bùng _cháy dữ _dội tại khu sản _xuất của nhà _máy giấy thành _đạt , nằm _trong cụm công _nghiệp phong _khê (thành _phố bắc _ninh) .
Kết quả tóm tắt sử dụng Pointer-Generator + Word2Vec pre-trained: hơn 19h tối 4 .

5 , ngọn_lửa bùng_cháy dữ_dội tại khu sản_xuất của nhà_máy giấy thành_đạt ,
nằm_trong cụm công_nghiệp phong_khê (thành_phố bắc_ninh) .

Kết quả tóm tắt sử dụng Pointer-Generator + Fasttext pre-trained:

sáng nay (4 .

5) , tại khu sản_xuất giấy thành_đạt , nằm_trong cụm công_nghiệp phong_khê (thành_phố bắc_ninh) bất_ngò bốc_cháy dữ_dội .

4.4 Nhận xét

Từ thực nghiệm cho thấy, kết quả các điểm ROUGE cho tiếng Anh khi sử dụng mô hình cơ sở, bỏ lớp embedding gốc, thêm vào các pre-trained: Word2Vec và Fasttext đều cải tiến hơn so với mô hình cơ sở, đặc biệt là Fasttext. Với bộ dữ liệu tiếng Việt, các điểm ROUGE khi thêm vào các pre-trained không được cải thiện, duy chỉ có với Fasttext pre-trained, kết quả thu được mới cao hơn được mô hình cơ sở ở điểm ROUGE-1. Điều này có thể là do bộ dữ liệu cho tiếng Việt cũng như các bộ pre-trained Word2Vec, Fasttext cho tiếng Việt đều chưa đủ tốt. Khi sử dụng hai bộ pre-trained này, vẫn có thể có những trường hợp từ mới không có trong từ điển gốc, cũng không có trong hai bộ pre-trained trên, nên sẽ bị biểu diễn thành token “UNK” với một vector embedding khởi tạo ngẫu nhiên rồi đưa vào mạng, dẫn đến kết quả không cao. Bên cạnh đó, việc sử dụng đoạn sapo của mỗi bài báo làm bản tóm tắt không thực sự chính xác, vì nhiều khi trong đa số các bài báo, đoạn này chỉ là đoạn mở đầu hay giới thiệu, đặc biệt là với các bài báo viết theo kiểu diễn dịch, thì đoạn này gần như không mang lại nhiều thông tin trong bài báo. Tuy nhiên, từ kết quả cải thiện trên bộ dữ liệu tiếng Anh cho ta thấy, nếu bộ dữ liệu tiếng Việt cũng như các mô hình pre-trained Word2Vec, Fasttext đủ tốt hứa hẹn sẽ mang lại kết quả tốt hơn cho tiếng Việt.

Mặc khác, việc sử dụng độ đo ROUGE trong bài toán tóm tắt tóm lược văn bản cũng chưa được hợp lý, bởi vì độ đo ROUGE thực hiện so khớp ký tự của bản tóm tắt tham chiếu và bản tóm tắt sinh ra bởi hệ thống, mà đặc trưng của phương pháp tóm tắt văn bản tóm lược không phải là trích rút từ những câu có sẵn trong bản bản vào, nên có kết quả không cao với độ đo này.

Kết chương

Chương này đã trình bày về các giải pháp đưa ra để cải tiến chất lượng tóm tắt tóm lược văn bản, dựa trên mô hình cơ sở của Abigail See [1], bằng việc bỏ lớp embedding trong mô hình gốc, thêm vào các mô hình pre-trained Word2Vec/Fasttext, sau đó tiến thành thực nghiệm trên hai bộ dữ liệu CNN/Daily Mail cho tiếng Anh và Báo Mới cho tiếng Việt. Các kết quả thu được đều rất khả quan, đặc biệt là cho tiếng Anh, hứa hẹn sẽ mang lại kết quả tốt tương tự cho tiếng Việt khi bộ dữ liệu cũng như mô hình pre-trained đủ tốt. Ở chương tiếp theo, Chương 5 sẽ trình bày về việc xây dựng và phát triển ứng dụng đọc tin nhanh trên nền tảng Android.

Chương 5 Phát triển ứng dụng đọc tin nhanh

Chương 4 đã trình bày về các giải pháp cũng như đưa ra thực nghiệm, đánh giá kết quả thu được từ các cải tiến đề xuất dựa trên mô hình cơ sở cho bài toán tóm tắt lược văn bản. Chương 5 này sẽ trình bày về xây dựng và phát triển ứng dụng đọc tin nhanh trên nền tảng Android. Với yêu cầu và phạm vi đề tài, chương này sẽ bao gồm các phần: (i) Khảo sát và phân tích yêu cầu và (iii) Thiết kế và xây dựng ứng dụng.

5.1 Khảo sát và phân tích yêu cầu

5.1.1 Khảo sát hiện trạng

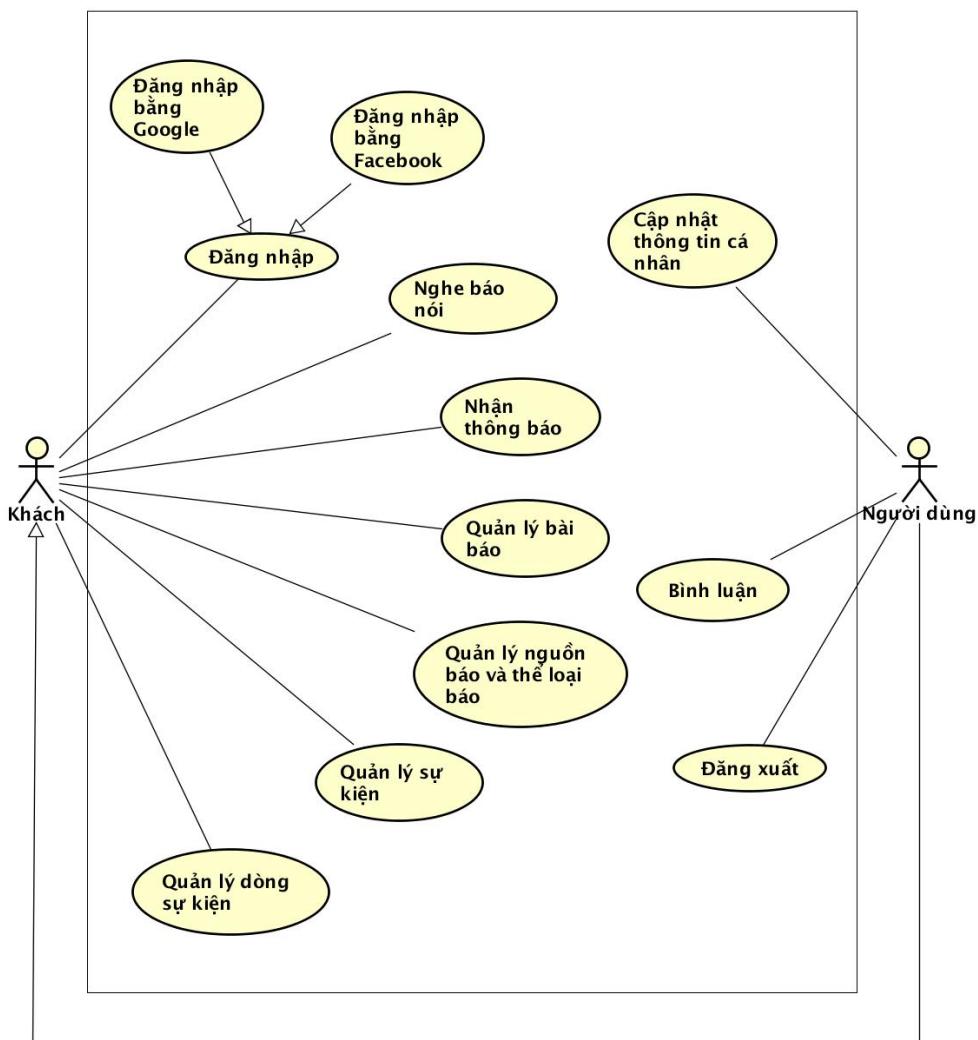
Hiện nay, việc sử dụng smartphone đã trở lên vô cùng phổ biến, thay thế gần như hoàn toàn các điện thoại di động truyền thống, nhu cầu của người dùng sử dụng trong việc sử dụng được nhiều tiện ích trên smartphone cũng được tăng lên. Trong đó, nhu cầu đọc báo điện tử đang rất được chú trọng bởi sự tiện dụng của nó, độc giả có thể đọc được mọi lúc mọi nơi trên smartphone. Trên thực tế, các ứng dụng có triển khai báo nói đang khá ít, chưa mang lại hiệu quả sử dụng cao, các ứng dụng nổi bật về đọc báo điện tử cũng như báo nói đã được em trình bày trong phần 1.2. Khi đọc báo trên smartphone, đa phần người dùng đều mong muốn nắm bắt được thông tin nhanh nhất có thể, tuy nhiên, các báo hiện nay đều có nội dung khá dài, khiến người đọc khó tập trung được vào thông tin quan trọng. Với sự phát triển của công nghệ tóm tắt văn bản tóm lược, việc tóm tắt các bài báo lại đã trở nên hoàn toàn hoàn toàn khả thi. Hơn nữa, một thực trạng là các bài báo hiện nay đều khá rời rạc, hoặc chỉ có thể được phân theo các thể loại, khiến độc giả rất khó để biết được các diễn biến của một sự kiện xảy ra hay các bài báo liên quan. Nắm bắt được nhu cầu đó của người dùng, ứng dụng Dora News đã được xây dựng với các tính năng báo nói, tóm tắt bài báo, gom nhóm

các bài báo về chung một sự kiện, gom các sự kiện thành một dòng sự kiện, giúp người đọc có thể nắm được các thông tin tổng quan trong dòng chảy thời gian của một sự kiện nhanh nhất có thể. Trong Dora News, em đã sử dụng hệ thống tóm tắt văn bản tóm lược đã được đề xuất trong Chương 4 để tóm tắt các bài báo, phần nào giúp đưa kết quả nghiên cứu gần hơn với các ứng dụng thực tế.

5.1.2 Tổng quan chức năng

5.1.2.1 Biểu đồ use case tổng quan

Dựa trên nhu cầu của người đọc như trong phần 5.1.1, các chức năng được xây dựng trong hệ thống được mô tả như trong biểu đồ use case tổng quan ở Hình 16.

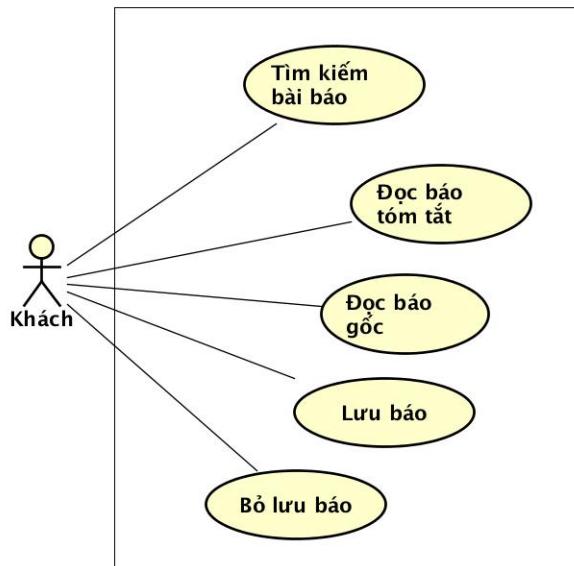


Hình 16 Biểu đồ use case tổng quan.

Như được mô tả trên Hình 16, hệ thống có 2 tác nhân chính là: Khách và người dùng đã đăng nhập. Khi chưa đăng nhập, khách có thể sử dụng ứng dụng bình thường với gần đủ các chức năng chính của ứng dụng như: (i) Quản lý bài báo, (ii) Quản lý sự kiện, (iii) Quản lý dòng sự kiện, (iv) Nghe báo nói, (v) Quản lý nguồn báo và thể loại báo. Sau khi đăng nhập, khách sẽ trở thành người dùng, bên cạnh các chức năng như của khách, người dùng sẽ có thêm các chức năng: (i) Cập nhật thông tin cá nhân, (ii) Bình luận các bài báo.

Các use case chính của hệ thống bao gồm: (i) Quản lý bài báo, (ii) Nghe báo nói, (iii) Quản lý sự kiện, (iv) Quản lý dòng sự kiện, (v) Quản lý nguồn báo và thể loại báo, (vi) Nhận thông báo, (vii) Bình luận báo.

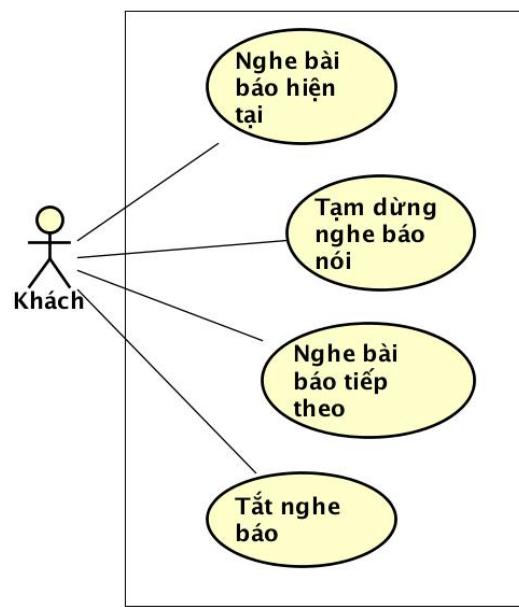
5.1.2.2 Biểu đồ use case phân rã “Quản lý bài báo”



Hình 17 Biểu đồ use case phân rã “Quản lý bài báo”.

Biểu đồ use case phân rã “Đọc báo” được minh họa trong Hình 17. Khách có thể lựa chọn đọc bài báo gốc hoặc đọc bản tóm tắt của bài báo đó. Với chức năng đọc báo tóm tắt, khách chỉ có thể đọc báo tóm tắt trực tuyến, bên cạnh đó cũng có thể đọc báo tóm tắt ngoại tuyến khi thiết bị không có kết nối mạng. Khách có thể tìm kiếm bài báo dựa trên từ khóa, lưu lại các bài báo xuống bộ nhớ máy hoặc bỏ lưu các bài báo.

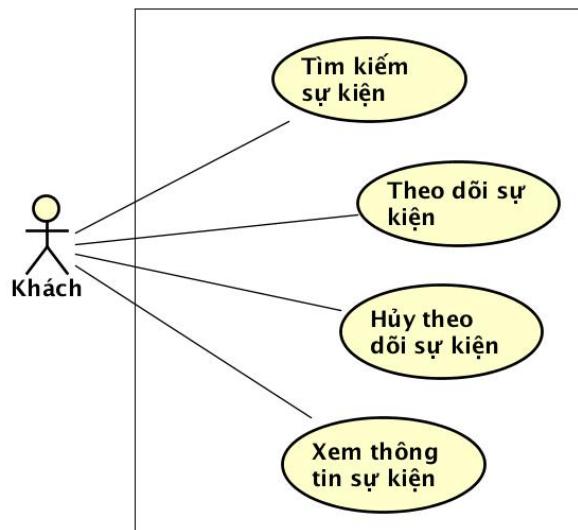
5.1.2.3 Biểu đồ use case phân rã “Nghe báo nói”



Hình 18 Biểu đồ use case phân rã “Nghe báo nói”.

Biểu đồ use case phân rã “Nghe báo nói” được minh họa trong Hình 18. Khách ngoài việc có thể nghe bài báo hiện tại, còn có thể tạm dừng, nghe bài tiếp theo hoặc tắt nghe báo.

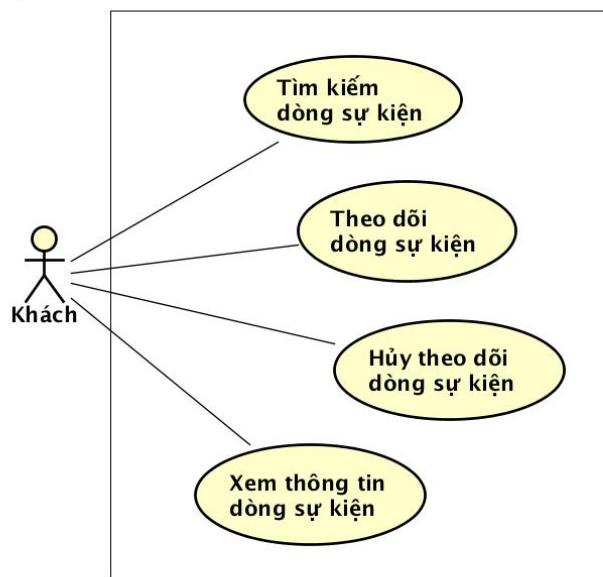
5.1.2.4 Biểu đồ use case phân rã “Quản lý sự kiện”



Hình 19 Biểu đồ use case phân rã “Quản lý sự kiện”.

Biểu đồ use case phân rã “Quản lý sự kiện” được minh họa trong Hình 19. Khách có thể tìm kiếm sự kiện dựa trên từ khóa, xem thông tin chi tiết sự kiện, theo dõi cũng như hủy theo dõi sự kiện.

5.1.2.5 Biểu đồ use case phân rã “Quản lý dòng sự kiện”

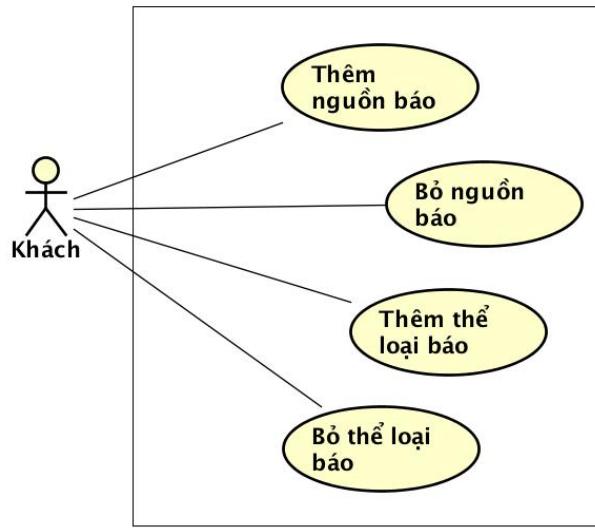


Hình 20 Biểu đồ use case phân rã “Quản lý dòng sự kiện”.

Biểu đồ use case phân rã “Quản lý dòng sự kiện” được minh họa trong Hình 20. Khách có thể tìm kiếm dòng sự kiện dựa trên từ khóa, xem thông tin chi tiết dòng sự kiện, theo dõi dòng sự kiện để nhận thông báo khi có sự kiện mới cũng như hủy theo dõi dòng sự kiện để ngừng nhận thông báo.

5.1.2.6 Biểu đồ use case phân rã “Quản lý nguồn báo và thể loại báo”

Biểu đồ use case phân rã “Quản lý nguồn báo và thể loại báo” được minh họa trong Hình 21. Khách có thể thêm nguồn báo ưu tiên muốn đọc hoặc bỏ nguồn báo nếu không muốn đọc ưu tiên từ danh sách tất cả các nguồn báo. Mặc định ban đầu, tất cả các nguồn báo đều có độ ưu tiên ngang nhau, nên nếu muốn điều chỉnh lại độ ưu tiên, khách sẽ sử dụng chức năng này. Bên cạnh đó, với thể loại báo, khách cũng có thể thêm thể loại báo muốn đọc hoặc bỏ thể loại báo nếu không muốn đọc từ danh sách tất cả các thể loại báo.



Hình 21 Biểu đồ use case phân rã “Quản lý nguồn báo và thẻ loại báo”.

5.1.3 Đặc tả chức năng

Danh sách các use case của hệ thống được thể hiện như trong Bảng 8:

Bảng 8 Danh sách các use case của hệ thống

Mã use case	Tên use case	Mã use case	Tên use case
UC01	Đăng nhập bằng Facebook	UC15	Hủy theo dõi sự kiện
UC02	Đăng nhập bằng Google	UC16	Theo dõi dòng sự kiện
UC03	Nghe bài báo hiện tại	UC17	Hủy theo dõi dòng sự kiện
UC04	Tạm dừng nghe báo nói	UC18	Lưu báo yêu thích
UC05	Nghe bài báo tiếp theo	UC19	Bỏ lưu báo yêu thích
UC06	Tắt nghe báo	UC20	Tìm kiếm bài báo
UC07	Nhận thông báo	UC21	Tìm kiếm sự kiện
UC08	Đọc báo gốc	UC22	Tìm kiếm dòng sự kiện
UC09	Đọc báo tóm tắt	UC23	Xem thông tin sự kiện
UC10	Thêm thẻ loại báo	UC24	Xem thông tin dòng sự kiện
UC11	Bỏ thẻ loại báo	UC25	Cập nhật thông tin cá nhân

Mã use case	Tên use case	Mã use case	Tên use case
UC12	Thêm nguồn báo	UC26	Bình luận
UC13	Bỏ nguồn báo	UC27	Đăng xuất
UC14	Theo dõi sự kiện		

Do giới hạn, phạm vi của báo cáo nên phần này chỉ đặc tả các use case quan trọng của hệ thống. Các use case được đặc tả bao gồm: (i) use case “Nghe bài báo hiện tại”, (ii) use case “Theo dõi sự kiện” và (iii) use case “Tìm kiếm sự kiện”.

5.1.3.1 Đặc tả use case “Nghe bài báo hiện tại”

Đặc tả của use case “Nghe bài báo hiện tại” được trình bày trong Bảng 9.

Bảng 9 Đặc tả use case “Nghe bài báo hiện tại”

Mã use case	UC03		
Tên use case	Nghe bài báo hiện tại		
Tác nhân	Khách		
Tiền điều kiện	Khách đã vào giao diện đọc bản tóm tắt của bài báo		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Khách	Ấn vào nút “play” (chạy) ở giữa hình ảnh của bài báo tương ứng
	2	Hệ thống	Lấy đường dẫn URL âm thanh của bài báo
	3	Hệ thống	Phát âm thanh bài báo
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Hệ thống	Tự động chuyển sang chạy âm thanh bài báo tiếp theo khi không lấy được âm thanh của bài báo hiện tại
Hậu điều kiện	Không		

5.1.3.2 Đặc tả use case “Theo dõi sự kiện”

Đặc tả của use case “Theo dõi sự kiện” được trình bày trong Bảng 10.

Bảng 10 Đặc tả use case “Theo dõi sự kiện”

Mã use case	UC03		
Tên use case	Theo dõi sự kiện		
Tác nhân	Khách		
Tiền điều kiện	Khách đã vào giao diện xem chi tiết một sự kiện		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Khách	Ấn vào nút “Theo dõi” sự kiện
	2	Hệ thống	Gửi yêu cầu theo dõi sự kiện lên server
	3	Hệ thống	Cập nhật lại trạng thái nút “Theo dõi” thành “Bỏ theo dõi”
	4	Hệ thống	Thêm sự kiện tương ứng được theo dõi thành công vào mục “Đang theo dõi” của trang chức năng “Yêu thích”
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Khách	Server trả về kết quả theo dõi thất bại, không thực hiện cập nhật trạng thái nút “Theo dõi”
	4a	Hệ thống	Server trả về kết quả theo dõi thất bại, không thực hiện thêm sự kiện tương ứng vào mục “Đang theo dõi” của trang chức năng “Yêu thích”
Hậu điều kiện	Không		

5.1.3.3 ĐẶC TẢ USE CASE “TÌM KIẾM SỰ KIỆN”

Đặc tả của use case “Tìm kiếm sự kiện” được trình bày trong Bảng 11.

Bảng 11 Đặc tả của use case “Tìm kiếm sự kiện”

Mã use case	UC03		
Tên use case	Tìm kiếm sự kiện		
Tác nhân	Khách		
Tiền điều kiện	Khách đã vào trang chức năng tìm kiếm		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Khách	Ấn vào thanh tìm kiếm
	2	Khách	Nhập từ khóa muốn tìm kiếm
	3	Khách	Ấn nút tìm kiếm trên bàn phím
	4	Hệ thống	Hiển thị kết quả tìm kiếm trên màn hình theo các trang mục
Luồng sự kiện thay thế	5	Khách	Ấn vào mục “Sự kiện” để xem các kết quả sự kiện tìm kiếm được
	4a	Hệ thống	Hiển thị thông báo “Không tìm thấy tin tức nào” khi không tìm thấy bất kỳ tin tức nào, đồng thời hiển thị danh sách rỗng
Hậu điều kiện	Không		

Dữ liệu đầu vào khi tìm kiếm được mô tả như trong Bảng 12:

Bảng 12 Dữ liệu đầu vào khi tìm kiếm sự kiện

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Từ khóa	Xâu ký tự	Có	Khác rỗng	Hoa Bình

5.1.4 Yêu cầu phi chức năng

Tính khả thi: các chức năng, thao tác của hệ thống dễ sử dụng, dễ thực hiện với cả những người sử dụng lần đầu.

Hiệu năng: Thời gian đáp ứng của hệ thống với các thao tác của người sử dụng không quá 2 giây, với chức năng nghe báo nói, hệ thống phản hồi không quá 5 giây.

Độ tin cậy: Tính an toàn và bảo mật cao, thông tin truyền tải chính xác, cập nhật liên tục.

5.2 Thiết kế và xây dựng ứng dụng

5.2.1 Công nghệ sử dụng

5.2.1.1 Nền tảng Android

Được phát triển bởi Google, Android là một hệ điều hành mã nguồn mở ra mắt vào năm 2007. Song song với iOS, Android là một trong hai hệ điều hành với số thiết bị của nó chiếm gần như toàn bộ thị phần smartphone trên toàn thế giới. Với đặc tính mở của mình, Android có thể chạy trên rất nhiều thiết bị công nghệ khác nhau như: (i) smartphone, (ii) smartwatch, (iii) smartTV, (iv) các thiết bị chơi game cầm tay... Chính bởi đặc tính này đã khiến Android trở thành sự lựa chọn hàng đầu của các hãng sản xuất các thiết bị công nghệ khi lựa chọn hệ điều hành cho các sản phẩm của mình. Để phát triển, triển khai các ứng dụng Android, có hai hướng chính là xây dựng ứng dụng native (gốc) hoặc hybrid (lai). Ngôn ngữ lập trình được sử dụng khi xây dựng ứng dụng native là Java hoặc Kotlin. Trong khi đó, ứng dụng hybrid là ứng dụng được phát triển dựa trên nền tảng web như HTML, CSS kết hợp với các thành phần native khác. Ưu điểm của ứng dụng hybrid là có thể phát triển cùng lúc cho nhiều hệ điều hành khác nhau, tuy nhiên nó lại gặp hạn chế ở hiệu năng kém, không ổn định, mang lại trải nghiệm người dùng không được như ứng dụng native. Do đó, trong đồ án này, em lựa chọn sử dụng ngôn ngữ Java, phát triển ứng dụng native trên hệ điều hành Android để mang lại hiệu năng tốt nhất có thể.

5.2.1.2 Thư viện Retrofit

Retrofit là thư viện để kết nối ứng dụng Android với server, dựa trên việc chuyển đổi các API thành Java interface. Là một type-safe HTTP client cho Java và Android, retrofit được tạo ra để giúp cho việc kết nối giữa client và server trở nên dễ dàng và thuận tiện hơn. Trong đó, type-safe có nghĩa là, Retrofit sẽ kiểm tra các kiểu dữ liệu xem có đúng không, nếu có lỗi sẽ trả về ngay. Khi được so sánh với thư viện Volley của Google với chức năng tương tự, theo một số kết quả sử dụng thực tế cho thấy thời gian thực hiện của Retrofit cao hơn rất nhiều so với Volley. Tuy nhiên, Retrofit không được tích hợp bộ chuyển đổi từ đối tượng kiểu Json sang đối tượng trong Java, do đó, ứng dụng sử dụng thêm thư viện Gson để giải quyết vấn đề này.

5.2.1.3 Firebase cloud messaging

Trong sự kiện Google I/O được tổ chức vào tháng 5 năm 2016, Google đã giới thiệu Firebase - nền tảng đám mây với nhiều tính năng mạng mẽ, hỗ trợ tốt cho các nhà phát triển ứng dụng di động, trong đó phải kể đến: (i) Cơ sở dữ liệu thời gian thực (Realtime Database), (ii) Bộ API cung cấp các tính năng dựa trên học máy cho ứng dụng (ML Kit), (iii) Hệ thống xác thực (Firebase Authentication) và (iv) Hệ thống gửi thông điệp (Firebase cloud messaging - FCM). Trong đó, ứng dụng đọc tin nhanh Dora News sử dụng hệ thống gửi thông điệp để có thể gửi thông báo một cách nhanh chóng, an toàn tới tất cả các thiết bị client.



Hình 22 Triển khai hệ thống FCM [21].

Việc triển khai hệ thống FCM bao gồm hai thành phần chính, phục vụ cho việc gửi và nhận thông điệp, được minh họa như trên Hình 22. Thành phần đầu tiên là một môi trường tin cậy (trusted environment), ví dụ như server cho ứng dụng, để xác định đối tượng, nội dung cần gửi và tiến hành gửi thông điệp cho các thiết bị client. Thành phần còn lại là client, ở trong hệ thống đọc tin nhanh, đây chính là phía ứng dụng Android. Trong Hình 22, Notifications Console GUI là một giao diện quản lý các thông điệp đã gửi, được cung cấp sẵn bởi Firebase, hỗ trợ cho nhà phát triển có thể quản lý tốt, trực quan hơn các thông điệp đã gửi trong hệ thống của mình.

5.2.2 Thiết kế kiến trúc

5.2.2.1 Lựa chọn kiến trúc phần mềm

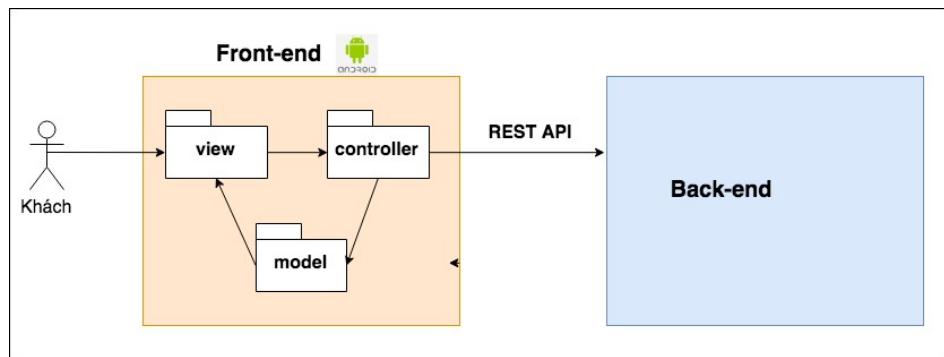
Kiến trúc phần mềm được lựa chọn để thiết kế ứng dụng là kiến trúc dựa trên mô hình MVC (Model - View - Controller). Tuy nhiên để phù hợp với việc xây dựng ứng dụng Android native, các gói này có chút thay đổi để phù hợp hơn. Trong đó, View là gói giao diện, chức các Activity và Fragment (tệp XML) - các giao diện người dùng làm nhiệm vụ hiển thị dữ liệu, Model là gói chứa các lớp đại diện cho dữ liệu trong ứng dụng, Controller là gói chứa các lớp có nhiệm vụ xử lý những yêu cầu gửi đến, tại Model, các lớp Controller sẽ xử lý dữ liệu và trả về kết quả cho View hiển thị. Để thay đổi cho phù hợp với kiến trúc Android, Controller sẽ bao gồm các lớp java Activity, Fragment tương ứng với các tệp XML Activity, Fragment, bên cạnh đó, Controller sẽ bao gồm thêm các lớp Adapter, đây là lớp đặc biệt trong Android, làm nhiệm vụ ánh xạ, bắt sự kiện cho các view thành phần dạng danh sách.

5.2.2.2 Thiết kế tổng quan

Kiến trúc tổng quan của hệ thống được thể hiện trên hình Hình 23.

Trong đó, phía client sẽ gọi các REST API từ server thông qua các lớp thuộc gói controller, sau đó, các lớp này sẽ trả dữ liệu kết quả cho các lớp thuộc model để thực hiện hiển thị lên giao diện của view. Với REST API (Representational State Transfer Application Programming Interface) là một giao diện ứng dụng chuyển đổi cấu trúc

dữ liệu, ở đó có các phương thức để thực hiện kết nối với các thư viện và ứng dụng khác. Thiết kế chi tiết các gói ở ứng dụng sẽ được trình bày trong phần 5.2.2.3.



Hình 23 Kiến trúc tổng quan hệ thống.

5.2.2.3 Thiết kế chi tiết gói

Thiết kế chi tiết gói trong ứng dụng được thể hiện trên Hình 24, trong hình chỉ trình bày một số mô-đun chính thuộc mỗi gói. Trong đó, ứng dụng được chia làm ba gói chính là: (i) view, (ii) controller và (iii) model.

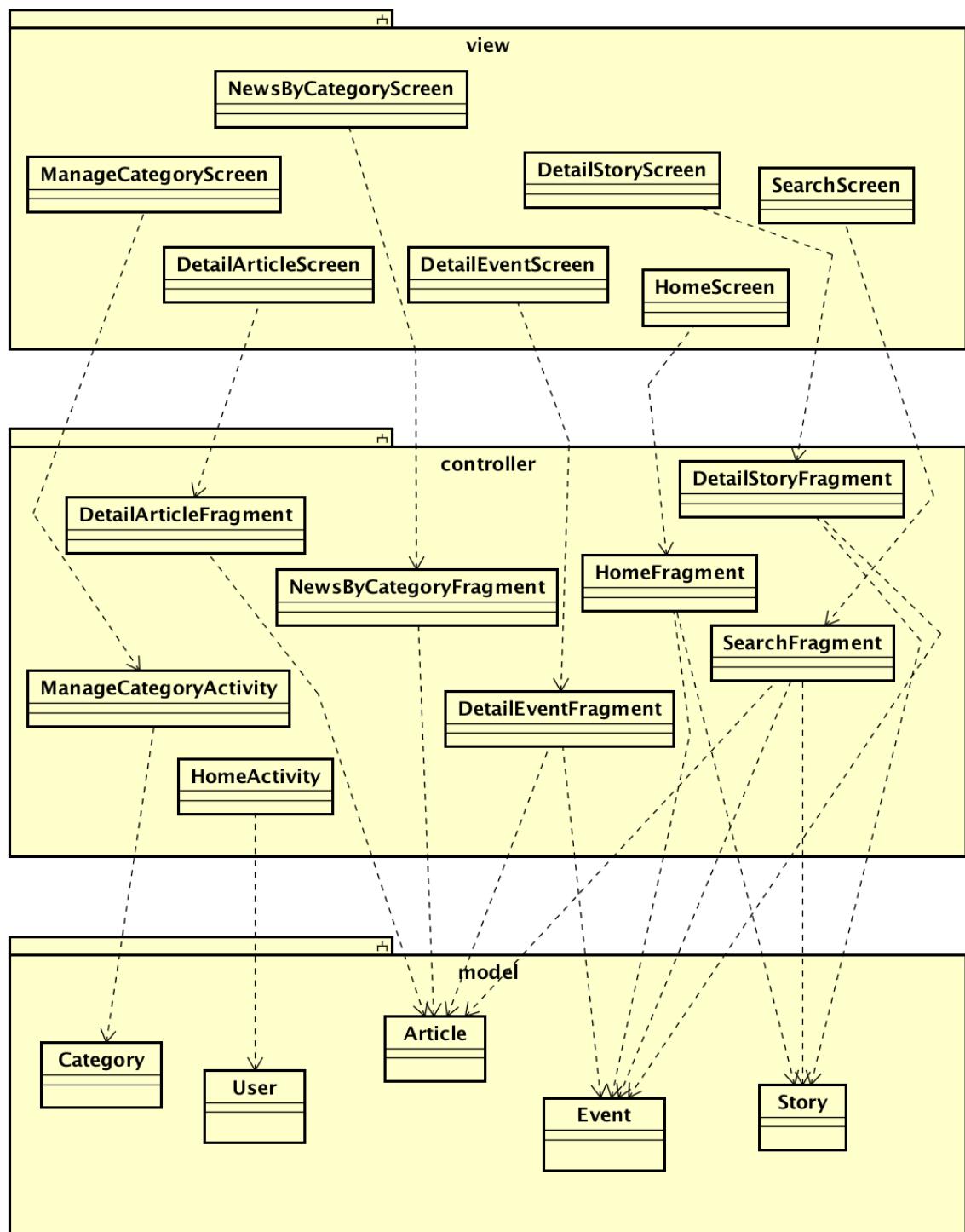
Gói view bao gồm các mô-đun: (i) DetailEventScreen, (ii) DetailArticleScreen, (iii) ManageNewsSourceScreen, (iv) DetailStoryScreen, (v) NotificationScreen, (vi) SettingScreen, (vii) ManageCategoryScreen, (viii) HomeScreen, (ix) NewsByCategoryScreen, (x) FavoriteScreen, (xi) SearchScreen.

Có tổng số 11 mô-đun trong gói view, thực hiện hiển thị dữ liệu lên màn hình ứng dụng tương ứng, đồng thời nhận sự kiện tương tác, logic cho các mô-đun trong gói controller.

Gói controller bao gồm các mô-đun: (i) HomeFragment, (ii) NewsByCategoryFragment, (iii) FavoriteFragment, (iv) SearchFragment, (v) NotificationFragment, (vi) SettingActivity, (vii) ManageCategoryActivity, (viii) DetailArticleFragment, (ix) DetailStoryFragment, (x) DetailEventFragment, (xi) HomeActivity, (xii) ManageNewsSourceActivity.

Có 12 mô-đun trong gói controller, nhận các sự kiện gửi từ view, thực hiện xử lý logic, lấy dữ liệu và trả về cho các mô-đun trong view hiển thị.

Gói model bao gồm các mô-đun: (i) Article, (ii) Story, (iii) Category, (iv) User, (v) Event, (vi) NewsSource. Có 6 lớp trong gói này, tương ứng là các lớp đại diện cho dữ liệu trong ứng dụng.



Hình 24 Biểu đồ thiết kế chi tiết gói.

5.2.3 Thiết kế chi tiết

5.2.3.1 Thiết kế giao diện

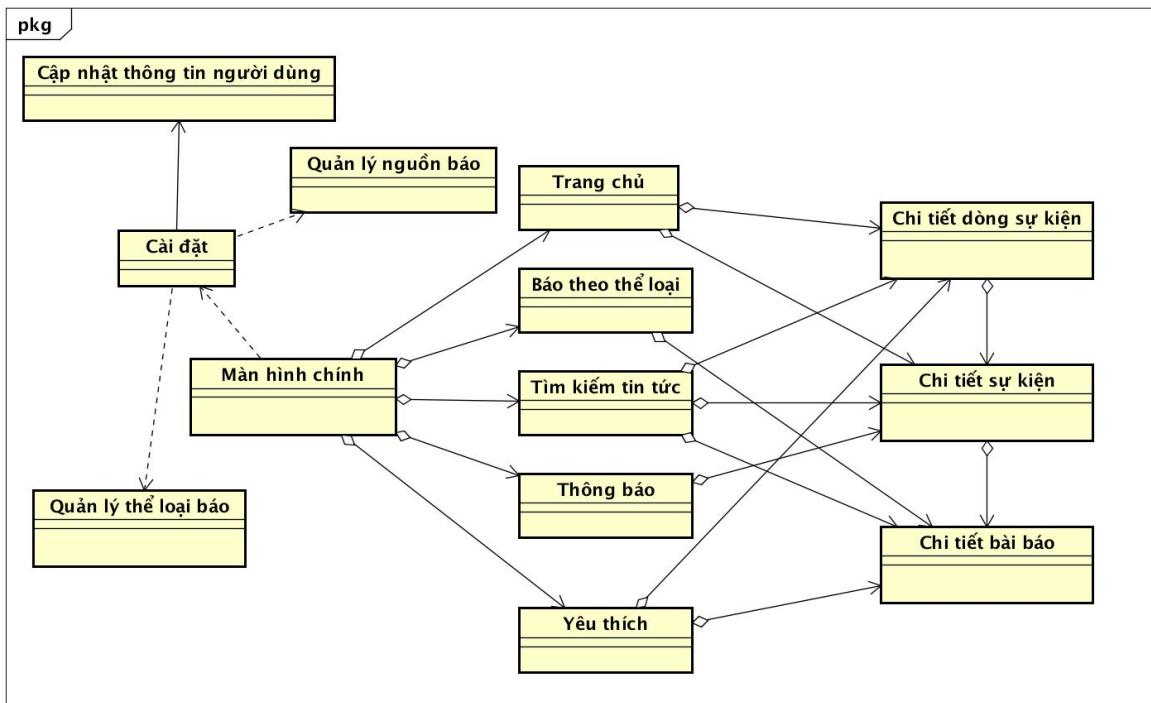
Thông tin màn hình của ứng dụng đọc tin nhanh Dora News:

- Độ phân giải màn hình: Hỗ trợ tốt nhất với các màn hình có độ phân giải là 2560x1440, 1920x1080 và 1280x720.
- Kích thước màn hình: Hỗ trợ tốt nhất với các màn hình có kích thước từ 5 inch đến 7 inch.
- Số lượng màu sắc hỗ trợ: Ba màu chính của ứng dụng là: #008577, #00574B và #007AFF.

Khi thiết kế giao diện, các thông điệp sẽ bao gồm Hộp thoại cảnh báo (Alert Dialog) và Toast. Trong đó, hộp thoại cảnh báo sẽ xuất hiện ở giữa màn hình, hiển thị nội dung yêu cầu người dùng phải thực hiện một thao tác xác nhận tương ứng mới có thể đóng được hộp thoại. Toast là một thành phần trong Android, được sử dụng để hiển thị các thông báo tức thời trong thời gian ngắn, tự động tắt đi mà không cần xác nhận của người dùng, Toast được hiển thị ở phía dưới màn hình.

Thanh điều khiển báo nói được đặt ngay trên thanh điều hướng ở cuối màn hình với các nút nghe báo, tạm dừng, chuyển bài báo tiếp theo, dừng hẳn nghe báo được thiết kế ở bên phải của thanh điều khiển, giúp thuận tiện cho người sử dụng, phù hợp với thói quen sử dụng trên các ứng dụng quản lý âm thanh khác.

Biểu đồ dịch chuyển màn hình được thể hiện như trên Hình 25. Ứng dụng Dora News được chia thành 5 tab (trang) chính: (i) Trang chủ, (ii) Báo theo thể loại, (iii) Tìm kiếm tin tức, (iv) Thông báo và (v) Yêu thích. Từ các tab này, có thể thực hiện điều hướng sang các màn hình con khác hoặc điều hướng sang một màn hình độc lập, khi điều hướng sang màn hình con, màn hình con mới này sẽ được lưu vào ngăn xếp của tab đó. Việc lưu lại các ngăn xếp màn hình tại mỗi tab giúp cho người sử dụng có thể truy cập nhanh vào các chức năng chính của ứng dụng mọi lúc, dù đang ở bất cứ tab hay giao diện chức năng khác.

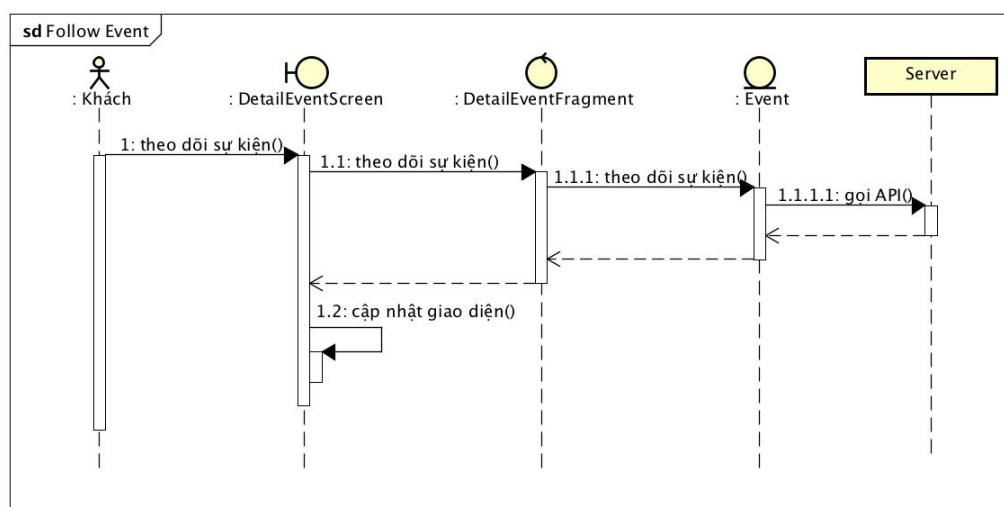


Hình 25 Biểu đồ dịch chuyển màn hình trong ứng dụng.

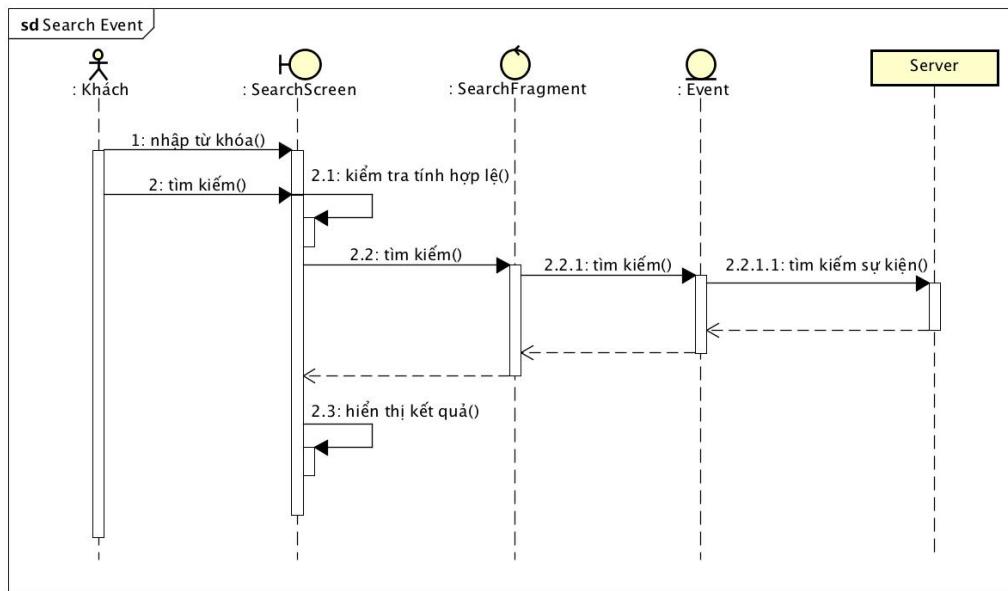
5.2.3.2 Thiết kế lớp

Phần này sẽ trình bày một số biểu đồ trình tự, thể hiện các luồng truyền thông điệp trong các use case “Theo dõi sự kiện” và “Tìm kiếm sự kiện”.

Biểu đồ trình tự của use case “Theo dõi sự kiện” và “Tìm kiếm sự kiện” lần lượt được mô tả tương ứng trong các hình Hình 26 và Hình 27.

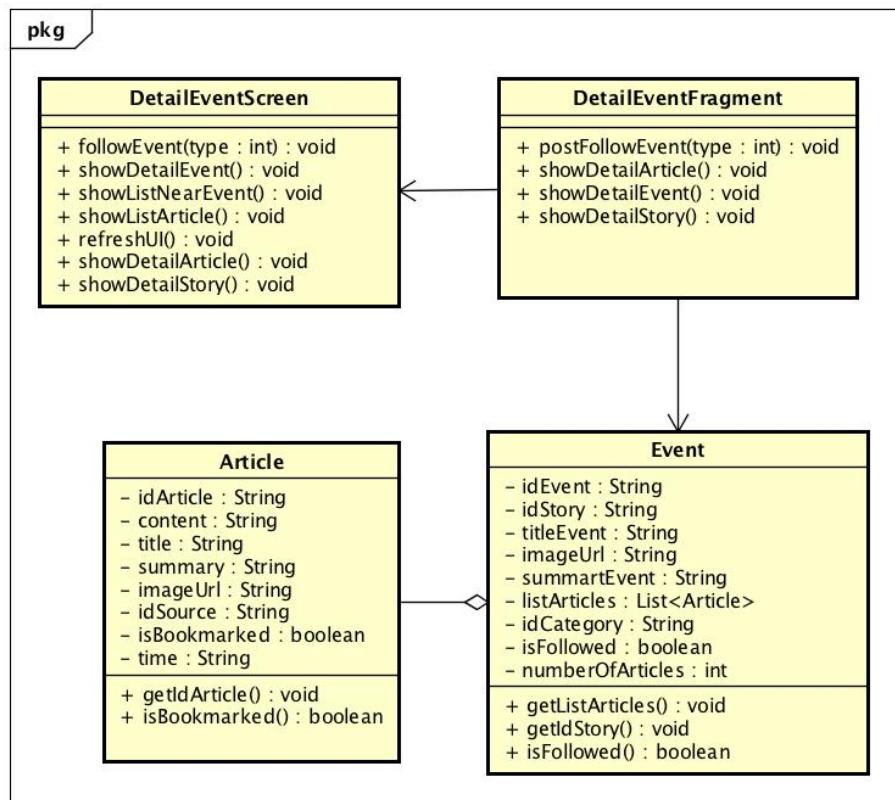


Hình 26 Biểu đồ use case “Theo dõi sự kiện”.

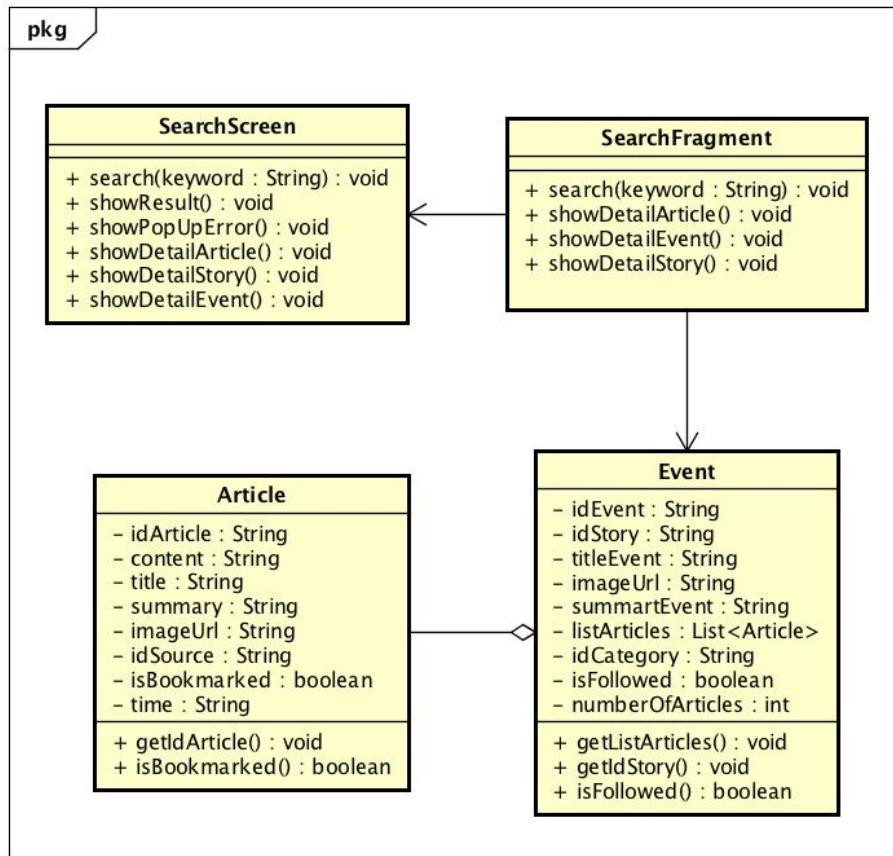


Hình 27 Biểu đồ use case “Tìm kiếm sự kiện”.

Biểu đồ lớp cho các use case “Theo dõi sự kiện” và “Tìm kiếm sự kiện” được thể hiện trong Hình 28 và Hình 29.



Hình 28 Biểu đồ lớp cho use case “Theo dõi sự kiện”.



Hình 29 Biểu đồ lớp cho use case “Tìm kiếm sự kiện”.

Trong use case “Theo dõi sự kiện”, khi khách đã ở màn hình xem chi tiết của sự kiện và sau đó ấn nút “Theo dõi” ở sự kiện tương ứng, yêu cầu sẽ được thực hiện ở phương thức postFollowEvent(int type) của lớp DetailEventFragment, kết quả trả về từ server sẽ được chuyển đổi sang đối tượng thuộc lớp Event, trong đó có trường isFollowed được cập nhật sang trạng thái tương ứng khi yêu cầu theo dõi sự kiện được thực hiện thành công. Dựa vào trạng thái trả về này, lớp DetailEventScreen thực hiện cập nhật giao diện tương ứng hoặc thông báo cho khách nếu như yêu cầu theo dõi thất bại.

Trong use case “Tìm kiếm sự kiện”, khi khách đã ở màn hình tìm kiếm, bước đầu tiên sẽ ấn vào thanh tìm kiếm. Sau đó, khách thực hiện nhập từ khóa rồi ấn tìm kiếm, yêu cầu tìm kiếm sẽ được thực hiện trong phương thức search(String keyword) của lớp SearchFragment, kết quả tìm kiếm trả về từ server sẽ được chuyển đổi sang đối tượng thuộc lớp Event. Nếu kết quả trả về khác rỗng, nó sẽ được lớp SearchScreen

hiển thị lên giao diện, nếu kết quả trả về rỗng, SearchScreen sẽ hiển thị lên một thông báo cho khách thông qua hàm showPopUpError().

5.2.4 Xây dựng ứng dụng

5.2.4.1 Thư viện và công cụ sử dụng

Trong quá trình phát triển ứng dụng Dora News, em đã sử dụng hệ thống API do anh Tạ Công Sơn - cựu sinh viên viện Công nghệ thông tin và truyền thông xây dựng. Danh sách một số các API quan trọng được trình bày như trong Bảng 13.

Bảng 13 Một số API quan trọng được sử dụng trong ứng dụng

Mục đích	Phương thức	Địa chỉ URL
Tự động đăng nhập với tài khoản ảo cho khách mới sử dụng.	POST	http://topica.ai:6968/api/v1/user/active
Lấy danh sách tin tức nóng trong ngày	GET	http://topica.ai:6968/api/v1/news/hot?reload=\${reload}&deviceid=\${deviceid}&uid=\${uid}
Lấy danh sách tin tức theo thể loại	GET	http://topica.ai:6968/api/v1/news/categories/newsfeed?reload=\${reload}&deviceid=\${deviceid}&catid=\${catid}&uid=\${uid}
Lấy danh sách thể loại báo	GET	http://topica.ai:6968/api/v1/news/categories
Tìm kiếm tin tức	POST	http://topica.ai:6968/api/v1/news/search
Lấy thông tin chi tiết sự kiện	GET	http://topica.ai:6968/api/v1/news/event/detail?eventid=\${eventid}&uid=\${uid}
Lấy thông tin chi tiết dòng sự kiện	GET	http://topica.ai:6968/api/v1/news/story/detail?storyid=\${storyid}&uid=\${uid}
Theo dõi/Hủy theo dõi sự kiện	GET	http://topica.ai:6968/api/v1/user/follow?uid=\${uid}&eid=\${eid}&type=\${type}

Mục đích	Phương thức	Địa chỉ URL
Theo dõi/Hủy theo dõi dòng sự kiện	GET	http://topica.ai:6968/api/v1/user/follow?uid=\${uid}&storyid=\${storyid}&type=\${type}
Lấy danh sách các sự kiện/dòng sự kiện đã theo dõi	GET	http://topica.ai:6968/api/v1/user/follow/stories?uid=\${uid}

Bên cạnh đó, để xây dựng chức năng phát báo nói, em đã sử dụng API của Vbee [22].

Một số công cụ, ngôn ngữ lập trình, IDE, thư viện được sử dụng như trong bảng Bảng 14.

Bảng 14 Các thư viện, ngôn ngữ lập trình và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Android Studio 3.2.1	https://developer.android.com/studio/
Ngôn ngữ lập trình	Java 8	https://www.java.com/
Kiểm tra kết quả API	Postman 6.7.4	https://www.getpostman.com/
Gửi thông báo từ server tới các client	Firebase Cloud Messaging	https://firebase.google.com/
Thư viện chuyển đổi đối tượng Json sang đối tượng Java	Gson 2.8.5	https://github.com/google/gson/
Thư viện gọi API	Retrofit 2.3.0	http://square.github.io/retrofit/
Thư viện hiển thị ảnh	Picasso 2.71828	https://square.github.io/picasso/

5.2.4.2 Kết quả đạt được

Kết quả thu được, ứng dụng Dora News được xây dựng, sản phẩm được đóng gói dưới dạng tệp cài đặt có đuôi “.apk”, cho phép người dùng cài đặt trực tiếp trên thiết bị sau khi đã tải tệp này về. Ứng dụng cho phép người dùng có thể xem chi tiết sự

kiện, dòng sự kiện, đọc báo tóm tắt cũng như báo gốc, phát báo nói, theo dõi sự kiện, dòng sự kiện cũng như lưu lại báo yêu thích. Tuy nhiên, do thời gian có hạn nên ứng dụng vẫn chưa xây dựng được chức năng “Đăng nhập bằng Facebook”, “Đăng nhập bằng Google”, song song với đó, chức năng “Cập nhật thông tin cá nhân” và “Bình luận báo” cũng chưa triển khai được.

Thông tin chi tiết về ứng dụng được mô tả như trong Bảng 15.

Bảng 15 Thông tin chi tiết của ứng dụng

Mô tả	Thông tin chi tiết
Số dòng code	20.000 dòng
Dung lượng toàn bộ mã nguồn	142,1 MB
Dung lượng sản phẩm	14,7 MB
Môi trường lập trình	macOS 10.13.6

5.2.4.3 Minh họa các chức năng chính

Phần này trình bày một số giao diện cho các chức năng chính của ứng dụng sau khi đã được xây dựng và đưa vào sử dụng, phù hợp với các màn hình đã được thiết kế. Các giao diện quan trọng được đưa ra trong Hình 30 bao gồm: (i) giao diện trang chủ, (ii) giao diện xem chi tiết sự kiện, (iii) giao diện xem chi tiết dòng sự kiện, (iv) giao diện đọc báo tóm tắt. Trong các phần mô tả dưới đây, quy ước người dùng là người sử dụng chung, không phải là tác nhân “Người dùng”.

Hình 30 (a) là giao diện của trang chủ ứng dụng, chứa danh sách các sự kiện/dòng sự kiện mới nhất trong ngày. Dưới cùng của màn hình là thanh điều hướng chính của ứng dụng, từ đây người dùng có thể truy cập nhanh vào các chức năng của ứng dụng khi đang ở bất cứ đâu. Trên thanh điều hướng này là thanh điều khiển báo nói. Ở thanh điều khiển báo nói này, người dùng có thể thực hiện dừng báo nói, chuyển bài báo tiếp theo hoặc tắt hẳn báo nói. Thanh điều khiển báo nói này chỉ xuất hiện khi người dùng đã chọn phát báo nói lần đầu ở giao diện xem chi tiết các bài báo. Khi ấn

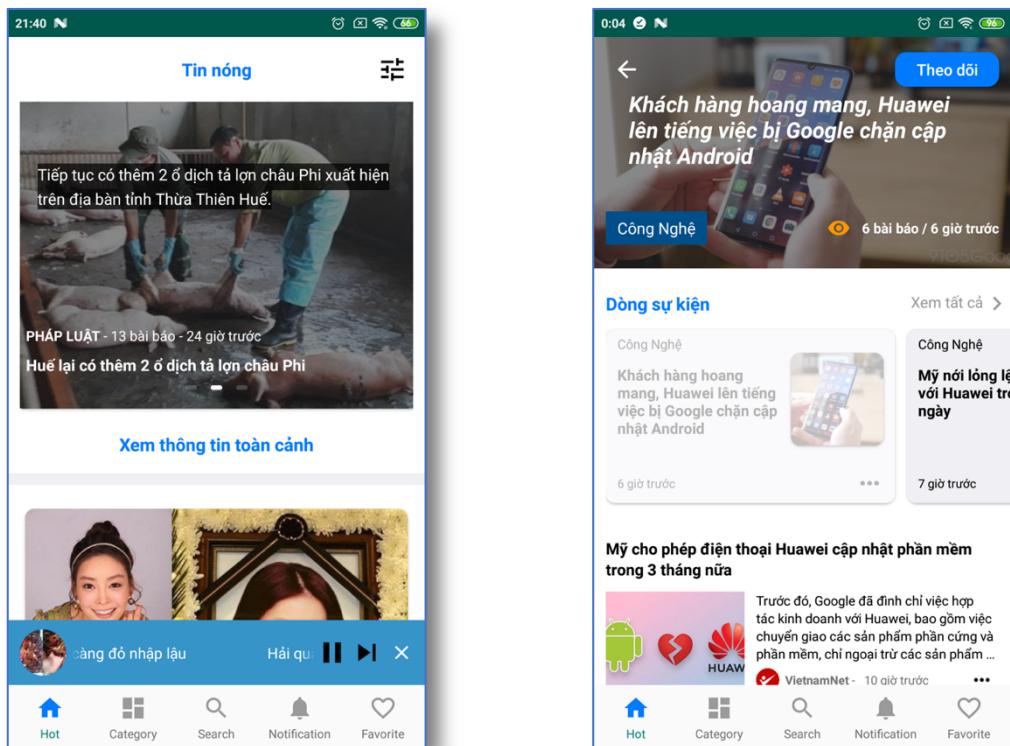
vào thanh điều khiển này, giao diện chi tiết danh sách báo đang phát sẽ được hiển thị ngay trên màn hình tương ứng hiện tại.

Hình 30 (b) là giao diện xem chi tiết sự kiện, trong màn hình này, nút “Theo dõi” màu xanh được đặt trên cùng bên phải, để người dùng có thể thực hiện theo dõi hoặc hủy theo dõi sự kiện này. Phía dưới là các sự kiện liên quan trong dòng sự kiện cũng như danh sách các bài báo trong sự kiện này. Với các sự kiện đơn lẻ, không thuộc một dòng sự kiện nào thì sẽ chỉ có thông tin chi tiết sự kiện đó và các bài báo của nó. Với các sự kiện nằm trong dòng sự kiện, trong màn hình xem chi tiết này sẽ có thêm danh sách nằm ngang các sự kiện thuộc cùng dòng sự kiện, người dùng có thể ấn vào để xem nhanh các sự kiện này, hoặc chọn “Xem tất cả” để xem chi tiết dòng sự kiện.

Hình 30 (c) là giao diện xem chi tiết dòng sự kiện, gồm các sự kiện được sắp xếp và lọc theo trình tự thời gian. Tương tự như giao diện xem chi tiết sự kiện, trên cùng bên phải của màn hình cũng là nút “Theo dõi” để người dùng có thể theo dõi hay hủy theo dõi dòng sự kiện đó.

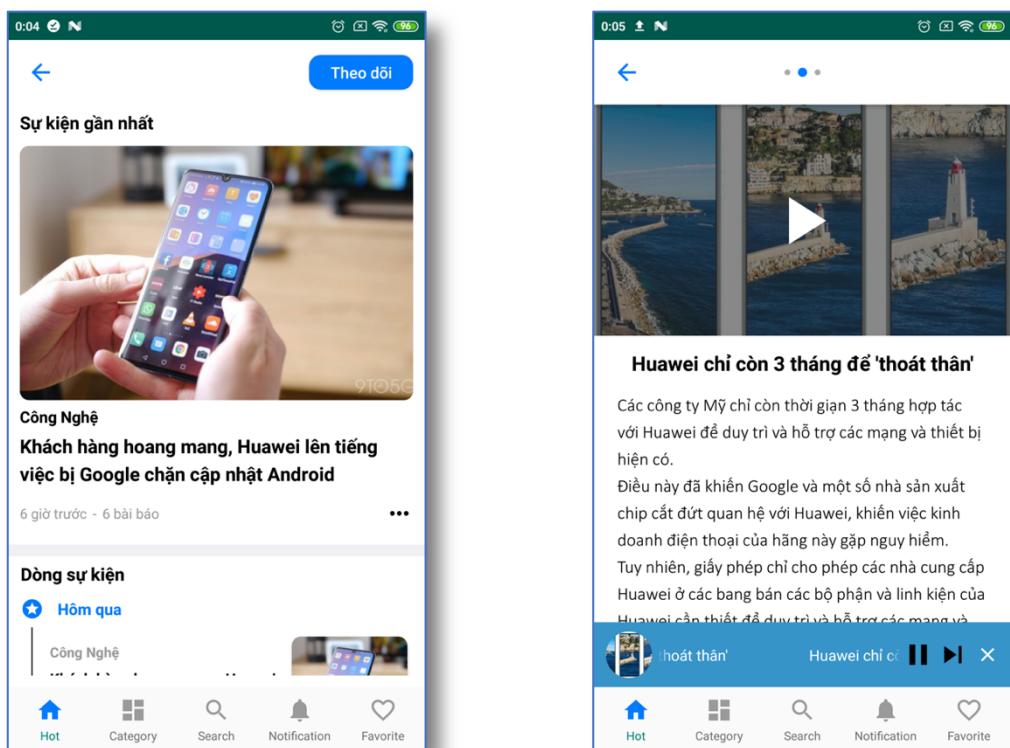
Hình 30 (d) là giao diện đọc báo tóm tắt, gồm danh sách các bài báo được trình bày theo dạng nằm ngang, ở giữa mỗi màn hình tương ứng của từng bài báo là nút “Nghe báo”, tiêu đề và nội dung tóm tắt của bài báo đó. Khi người dùng thực hiện ấn vào nút “Nghe báo”, bài báo sẽ được phát, thanh điều khiển báo nói hiện lên ở bên dưới, người dùng có thể điều khiển báo nói thông qua thanh điều khiển này. Bên cạnh các giao diện này, ứng dụng còn nhiều giao diện khác. Trong đó có giao diện quản lý nguồn báo cũng như giao diện quản lý thể loại báo, giúp cho người sử dụng có thể điều chỉnh được danh sách các thể loại báo muốn đọc, cũng như thay đổi thứ tự ưu tiên muốn đọc từ các nguồn báo. Ở giao diện trang chủ của ứng dụng sẽ chỉ có những sự kiện đơn lẻ cũng như các dòng sự kiện mà không có các bài báo, nhằm mục đích giúp người đọc tập trung nhanh vào các sự kiện, dòng sự kiện nổi bật, nơi các bài báo liên quan đã được gom nhóm lại. Với các bài báo đơn lẻ, không thuộc sự kiện nào sẽ được hiển thị trong giao diện đọc báo theo thể loại.

Ngoài ra, ứng dụng còn các giao diện ứng với các chức năng phụ khác như: (i) giao diện cài đặt, (ii) giao diện màn hình thông báo, (iii) giao diện màn hình yêu thích...



(a)

(b)



(c)

(d)

Hình 30 Một số giao diện các chức năng chính của ứng dụng.

5.2.5 Kiểm thử

Bảng 16 minh họa một vài trường hợp kiểm thử, trong đó sử dụng kỹ thuật kiểm thử hộp đen để kiểm thử ứng dụng.

Bảng 16 Một số trường hợp kiểm thử

Chú thích: STT: số thứ tự; B: bước

STT	Kịch bản	Thủ tục kiểm thử		Kết quả thực tế	Kết luận
		Bước thực hiện	Kết quả kỳ vọng		
1	Xem chi tiết danh sách các bài báo đang phát	B1: Mở phát báo nói từ màn hình xem chi tiết bài báo từ trang chủ ứng dụng B2: Thoát khỏi màn hình xem chi tiết bài báo B3: Chuyển sang trang danh sách các bài báo theo thể loại B4: Án vào thanh điều khiển báo nói	Hiển thị màn hình chi tiết danh sách các bài báo đang phát ở ngay trang danh sách các bài báo theo thể loại	Giống như kỳ vọng	Đạt
2	Xem danh sách các sự kiện, dòng sự kiện khi kết nối mạng bị mất độ xuất	B1: Mở ứng dụng B2: Chờ khi ứng dụng tải được trang chủ xong, ngắt kết nối mạng thiết bị	Hiển thị một thông báo với dòng chữ “Không có kết nối”	Giống như kỳ vọng	Đạt

STT	Kịch bản	Thủ tục kiểm thử		Kết quả thực tế	Kết luận
		Bước thực hiện	Kết quả kỳ vọng		
3	Không nhập từ khóa khi thực hiện tìm kiếm tin tức	B1: Vào trang chức năng tìm kiếm B2: Án vào thanh tìm kiếm trên cùng B3: Án nút tìm kiếm trên bàn phím	Hệ thống không thực hiện tìm kiếm, con trỏ gõ vẫn bản vẫn hiện thị trên thanh tìm kiếm	Giống như kỳ vọng	Đạt
4	Đồng bộ trạng thái theo dõi ở các sự kiện giống nhau thuộc các trang chức năng khác nhau	B1: Vào trang chủ của ứng dụng B2: Án vào xem chi tiết một sự kiện bất kì mà chưa được theo dõi trước đó B3: Án theo dõi sự kiện đó B4: Vào trang chức năng yêu thích, chọn sự kiện vừa theo dõi xuất hiện trong danh sách B5: Quay lại trang chủ của ứng dụng, án bỏ theo dõi sự kiện đang xuất hiện trên màn hình B6: Quay lại trang chức năng yêu thích	Sau khi án theo dõi sự kiện ở trang chủ, ngay lập tức ở trang chức năng sẽ xuất hiện sự kiện tương ứng. Khi thực hiện bỏ theo dõi sự kiện này từ trang chủ, ngay lập tức màn hình chi tiết của sự kiện đó ở trang chức năng yêu thích cũng được cập nhật giao diện, đổi trạng thái nút theo dõi, khi thực hiện án vào nút quay lại, sự kiện tương ứng trong trang chức năng yêu thích sẽ biến mất.	Giống như kỳ vọng	Đạt

Khi kiểm thử ứng dụng, có khoảng 20 test case đã được tạo ra, trong đó có 19 test case đạt, 1 test case còn lại không đạt do test case được thực hiện với tốc độ mạng rất chậm nên ảnh hưởng đến kết quả cuối cùng không đạt.

5.2.6 Triển khai

Ứng dụng đọc tin nhanh Dora News đã hoàn thành các chức năng cơ bản, duy chỉ có các chức năng liên quan đến đăng nhập như: (i) cập nhật thông tin cá nhân và (ii) bình luận là chưa có. Dora News đã được triển khai đến một số lượng người dùng nhất định (khoảng 30 người), với phiên bản Android thấp nhất là Android 6.0. Sau một thời gian, ứng dụng đã thu được những phản hồi khá tích cực từ phía những người sử dụng.

Kết chương

Chương này đã trình bày về từng bước trong quá trình xây dựng và phát triển ứng dụng đọc tin nhanh Dora News. Ứng dụng thu được về cơ bản đã có đầy đủ các chức năng quan trọng, chỉ còn thiếu một số chức năng sẽ được đề cập trong Chương 6. Khi được triển khai đến một số lượng người dùng nhỏ đã mang lại những phản hồi khá tích cực, trong chương tiếp theo, Chương 6 sẽ trình bày về kết luận kết quả thực hiện đề tài và các hướng phát triển trong tương lai.

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Với giải pháp sử dụng mô hình Word2Vec, Fasttext pre-trained để xử lý đầu vào của mô hình cơ sở, bỏ lớp embedding của mô hình cơ sở đi, kết quả thực nghiệm cho thấy giải pháp khá thành công trên bộ dữ liệu Daily Mail/CNN cho tiếng Anh, giúp tăng các điểm ROUGE lên tương đối. Đặc biệt khi đưa thêm mô hình Fasttext pre-trained vào mô hình cơ sở có Coverage, các điểm ROUGE-1, ROUGE-2, ROUGE-L đã tăng lên lần lượt là 0,5, 0,39 và 0,5 điểm so với mô hình cơ sở. Tuy nhiên, với bộ dữ liệu tiếng Việt, mặc dù các điểm ROUGE đều khá cao sao với tiếng Anh nhưng khi thêm các mô hình pre-trained Word2Vec và Fasttext vào thì không mang lại kết quả tốt hơn mô hình cơ sở, duy chỉ có điểm ROUGE-1 tăng được 0.49 điểm với việc sử dụng Fasttext pre-trained, còn các điểm còn lại đều thấp hơn. Điều này có thể do bộ dữ liệu tiếng Việt và bộ pre-trained Word2Vec cũng như Fasttext cho tiếng Việt chưa thực sự tốt. Tuy nhiên, với việc các điểm ROUGE tăng tương đối trên tiếng Anh, việc đưa mô hình Word2Vec/Fasttext pre-trained hứa hẹn cũng sẽ mang lại kết quả tốt tương tự cho tiếng Việt khi có bộ dữ liệu và bộ pre-trained thực sự đủ tốt.

Trong giải pháp cải tiến đưa ra, mặc dù nó giúp đem lại nhiều ưu điểm, kết quả thu được khác tốt, tuy nhiên vẫn có những trường hợp các từ không có trong từ điển của Word2Vec, cũng như các thành phần n-grams của nó không có trong Fasttext thì nó vẫn sẽ được biểu diễn sang dạng một vector embedding với các giá trị khởi tạo ngẫu nhiên trước khi đưa vào mạng, dẫn đến mất mát thông tin. Tuy nhiên các trường hợp này sẽ không nhiều, bởi với việc biểu diễn từ không ở đơn vị từ mà ở đơn vị n-grams, Fasttext sẽ có khả năng tạo ra số lượng từ rất lớn nếu bộ pre-trained đủ tốt. Bên cạnh đó, có những trường hợp trong quá trình huấn luyện, từ đầu vào của lớp giải mã (từ nằm trong bản tóm tắt tham chiếu) có thể sẽ không có cả trong từ điển mở rộng (từ

diễn bao gồm từ điển gốc và những từ mới có trong bài báo gốc) thì các từ này vẫn sẽ được chuyển sang token “UNK”, do đó khi tính giá trị mất mát theo hàm negative log likelihood, ví trí đúng (true index) của từ này từ điển mở rộng sẽ là vị trí của token “UNK”, dẫn đến việc tính toán không chính xác. Ngoài ra, trên thực tế khi quan sát kết quả các bản tóm tắt sinh ra, mặc dù khá “mượt” nhưng nội dung có thể không có mối liên kết, hay bản dịch không mang lại nhiều thông tin như bản tóm tắt tham chiếu.

Với ứng dụng đọc tin nhanh Dora News, so với các ứng dụng đọc báo cũng như đọc báo nói khác, các tính năng mới đã mang lại những tiện ích nhất định cho người dùng, bao gồm: (i) gom nhóm các bài báo về chung một sự kiện, (ii) gom các sự kiện về thành một dòng sự kiện, (iii) cho phép theo dõi và nhận thông báo khi có sự kiện mới tiếp theo của chuỗi sự kiện hiện tại, (iv) cho phép đọc các bản tóm tắt của các bài báo, rút ngắn thời gian nắm bắt thông tin, (v) cho phép quản lý điều khiển báo nói dễ dàng. Với sự phản hồi tích cực từ phía những người dùng đầu tiên, hứa hẹn sẽ trở thành một ứng dụng hữu ích, phổ biến, đáp ứng được nhu cầu đọc tin nhanh của đông đảo mọi tầng lớp người dùng. Tuy nhiên, do thời gian hạn chế nên ứng dụng vẫn còn nhiều khuyết điểm, khi dữ liệu trong một màn hình lớn, thao tác chuyển sang màn hình tiếp theo còn khá giật. Bên cạnh đó, chức năng “Đăng nhập” cùng các chức năng liên quan như “Bình luận” và “Cập nhật thông tin cá nhân” vẫn chưa được xây dựng.

6.2 Hướng phát triển

Với các hạn chế như đã nêu trong phần 6.1, để giải quyết được các hạn chế này, trong tương lai, em vẫn sẽ sử dụng mô hình cơ sở cũ, với giải pháp đưa thêm vào lớp Fasttext pre-trained cả về mặt lý thuyết lẫn kết quả thực nghiệm đều tốt hơn so với việc sử dụng Word2Vec, nên các giải pháp tiếp theo sẽ chỉ sử dụng Fasttext pre-trained. Bên cạnh mô hình pre-trained này, em đề xuất từ điển mở rộng thêm cả các từ mới xuất hiện trong bản tóm tắt tham chiếu cho quá trình huấn luyện mô hình. Việc này đảm bảo không bỏ sót các từ mới trong bản tóm tắt tham chiếu mà không có trong từ điển gốc cũng như trong văn bản gốc, để các từ này không bị biểu diễn thành token “UNK”, giúp quá trình tính toán và tối ưu hàm mất được chính xác. Tuy nhiên khi đó, việc tính phân phối cuối cùng sẽ cần loại ra những vị trí trong từ điển mở rộng

của các từ mới trong bản tóm tắt tham chiếu này vì từ tại các vị trí này không có attention.

Việc sử dụng Word2Vec cũng như Fasttext để biểu diễn các từ chỉ thể được mối quan hệ giữa các từ với nhau thông qua ngữ cảnh chung của chúng, do đó, các mô hình này có nhược điểm là không thể hiện được sự hiện diện của ngữ cảnh cụ thể hay trong từng lĩnh vực hay văn cảnh cụ thể. Ví dụ, các từ như “con chuột” có ngữ nghĩa khác nhau ở các ngữ cảnh khác nhau như: “Con chuột máy tính này đẹp quá” và “Con chuột này to thật”. Trong khi các mô hình Word2Vec/Fasttext này sẽ tìm ra một vector đại diện duy nhất cho mỗi từ dựa trên một tập dữ liệu lớn nên không thể hiện được sự đa dạng của ngữ cảnh. Do đó, việc tạo ra biểu diễn của mỗi từ dựa trên các từ khác trong câu sẽ mang lại kết quả ý nghĩa hơn rất nhiều. Trong ví dụ trên, ý nghĩa của từ “con chuột” sẽ được biểu diễn dựa trên các từ trước và sau nó ở trong câu, việc được xây dựng dựa trên những ngữ cảnh như vậy sẽ giúp cho vector biểu diễn cho từ “con chuột” được tốt hơn. Trong các phương pháp biểu diễn từ ở mức ngữ cảnh này thì phổ biến hơn cả là ELMo (Embeddings from Language Models) [23] và Google BERT (Bidirectional Encoder Representations from Transformers) [24]. Trong tương lai, em dự kiến sẽ dụng chúng để giải quyết các vấn đề trong bài toán tóm tắt lược văn bản.

Trong thực nghiệm cho thấy, các bản tóm tắt được sinh ra bởi hệ thống khá “mượt”, tuy nhiên gặp hiện tượng một số trường hợp các từ rời rạc về mặt ý nghĩa, đồng thời nội dung bản tóm tắt không tập trung được vào các thông tin như kỳ vọng. Do đó, trong tương lai, em sẽ thực hiện cải tiến thêm mô hình, giúp học được các từ quan trọng trong bài báo gốc, cẩn tập trung vào khi tạo ra bản tóm tắt. Khi các bản tóm tắt tham chiếu tốt, các từ trong bản tóm tắt này đều nằm trong văn bản gốc, thì những từ quan trọng có thể sẽ là những từ trong từng cụm lặp lại ở cả hai bản. Ý tưởng này đã được đề cập đến trong [27], dựa trên mô hình cơ sở [1] cho tiếng Anh và được gọi là “Bottom-Up”. Để áp dụng ý tưởng cải tiến này cho tiếng Việt, cần phải có bộ dữ liệu với cả bản tóm tắt tham chiếu đủ tốt, tóm lược súc tích được nội dung văn bản, nếu không, các từ trong bản tóm tắt tham chiếu sẽ không mang lại nhiều ý nghĩa cho quá

trình học. Với ý tưởng như vậy, em kỳ vọng mô hình mới sẽ học được tốt hơn, tóm tắt lại được nội dung trọng tâm hơn, mang lại kết quả tốt hơn so với hiện tại.

Cuối cùng, việc xử lý liên quan đến dữ liệu là vô cùng cần thiết, đặc biệt là đối với với tóm tắt bản tiếng Việt. Em sẽ thực hiện huấn luyện một bộ mô hình Fasttext riêng với bộ dữ liệu lớn hơn, bên cạnh đó, bộ dữ liệu sử dụng sẽ cần được chỉnh lại, không lấy đoạn sapo làm bản tóm tắt tham chiếu nữa, thay vào đó là tóm tắt thủ công, tuy nhiên công việc này khá tốn thời gian và công sức, nhưng nếu làm được, đây sẽ là một đóng góp lớn cho bài toán tóm tắt lược văn bản tiếng Việt. Với bộ dữ liệu mới này, em sẽ đưa thêm kỹ thuật Coverage vào, tối ưu lại mã nguồn tránh tối ưu cục bộ để cải thiện mô hình hiện tại, hứa hẹn sẽ mang lại kết quả tốt hơn trong việc tránh lặp lại các từ trong bản tóm tắt.

Với ứng dụng đọc tin nhanh Dora News, trong tương lai, em sẽ xây dựng tiếp chức năng “Đăng nhập”, “Cập nhật thông tin cá nhân” và “Bình luận”, nhằm mục đích tạo ra một mạng xã hội nhỏ của những người sử dụng ứng dụng. Song song với đó, ứng dụng sẽ cần phát triển một hệ thống quản lý bình luận, đồng thời có thêm tính năng báo cáo các trường hợp những bình luận thiếu đi sự tôn trọng người khác, những bình luận mang hàm ý, nội dung xấu, không lành mạnh... Để người dùng có được những trải nghiệm tốt nhất, em sẽ phát triển thêm hệ thống gợi ý, dựa trên thói quen đọc báo của người dùng, để người đọc có thể nhận được những thông tin cần quan tâm nhanh nhất có thể. Tính năng báo nói hiện vẫn chưa ổn định, nên trong tương lai, em sẽ sử dụng bộ API khác để cho kết quả tốt hơn. Nhằm mục đích tiếp cận với một lượng người dùng lớn hơn, ứng dụng sẽ được triển khai trên kho ứng dụng “Google play” để người dùng có thể tải và sử dụng thuận tiện hơn, dễ dàng hơn trong việc chia sẻ đến bạn bè, người thân.

Tài liệu tham khảo

- [1] Abigail See, Peter J. Liu and Christopher D. Manning, Get To The Point: Summarization with Pointer-Generator Networks, arXiv preprint arXiv:1704.04368, 2017.
- [2] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre and Bing Xiang, Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond, In The SIGNLL Conference on Computational Natural Language Learning, Berlin, 2016.
- [3] Perceptron: The Artificial Neuron, mc.ai, <https://mc.ai/perceptron-the-artificial-neuron>, last visited May 2019.
- [4] Neural Networks, CS231n Convolutional Neural Networks for Visual Recognition, <https://cs231n.github.io/neural-networks-1/>, last visited May 2019.
- [5] Vu Huu Tiep, Overfitting, <https://machinelearningcoban.com/2017/03/04/overfitting/>, last visited May 2019.
- [6] Britz D, Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs, WildML, <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>, last visited May 2019.
- [7] Understanding LSTM Networks, colah's blog, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, last visited May 2019.

- [8] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, Sequence to Sequence Learning with Neural Networks, arXiv:1409.3215, 2014.
- [9] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, In International Conference on Learning Representations, San Diego, 2015.
- [10] Chablani M., Sequence to sequence model: Introduction and concepts, Towards Data Science, <https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d>, last visited May 2019.
- [11] Oriol Vinyals, Meire Fortunato and Navdeep Jaitly, Pointer Networks, arXiv preprint arXiv:1506.03134v2, 2017.
- [12] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, Hang Li, Modeling Coverage for Neural Machine Translation, arXiv preprint arXiv:1601.04811, 2016.
- [13] Karani D., Introduction to Word Embedding and Word2Vec, Towards Data Science, <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa>, last visited May 2019.
- [14] Understanding Word Vectors and Word2Vec, Stokastik, <http://www.stokastik.in/understanding-word-vectors-and-word2vec/>, last visited May 2019.
- [15] Tran Viet Trung, Python Vietnamese Core NLP Toolkit, <https://github.com/trungtv/pyvi>, last visited May 2019.
- [16] Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras and Mark Johnson, VnCoreNLP: A Vietnamese Natural Language Processing Toolkit, In Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, 2018.
- [17] Google Code Archive - Long-term storage for Google Code Project Hosting, <https://code.google.com/archive/p/word2vec/>, last visited May 2019.

- [18] Word vectors for 157 languages, fastText, <https://fasttext.cc/docs/en/crawl-vectors.html>, last visited May 2019.
- [19] Kavita Ganesan, An intro to ROUGE, and how to use it to evaluate summaries, <https://medium.freecodecamp.org/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840>, last visited May 2019.
- [20] Atul Kumar, Pytorch implementation of “Get To The Point: Summarization with Pointer-Generator Networks”,
https://github.com/atulkum/pointer_summarizer/, last visited May 2019.
- [21] Firebase Cloud Messaging, Firebase, <https://firebase.google.com/docs/cloud-messaging>, last visited May 2019.
- [22] Vbee TTS, Giải pháp chuyển văn bản thành giọng nói tiếng Việt,
<https://vbee.vn>, last visited May 2019.
- [23] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee and Luke Zettlemoyer, Deep contextualized word representations, In Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, 2018.
- [24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018.
- [25] Vu Huu Tiep, Multi-layer Perceptron và Backpropagation,
<https://machinelearningcoban.com/2017/02/24/mlp/>, last visited May 2019.
- [26] Nal Kalchbrenner and Phil Blunsom, Recurrent Continuous Translation Models, In ACL Conference on Empirical Methods in Natural Language Processing, Washington, 2013.
- [27] Sebastian Gehrmann, Yuntian Deng, Alexander M. Rush, Bottom-Up Abstractive Summarization, arXiv preprint arXiv:1808.10792, 2018.

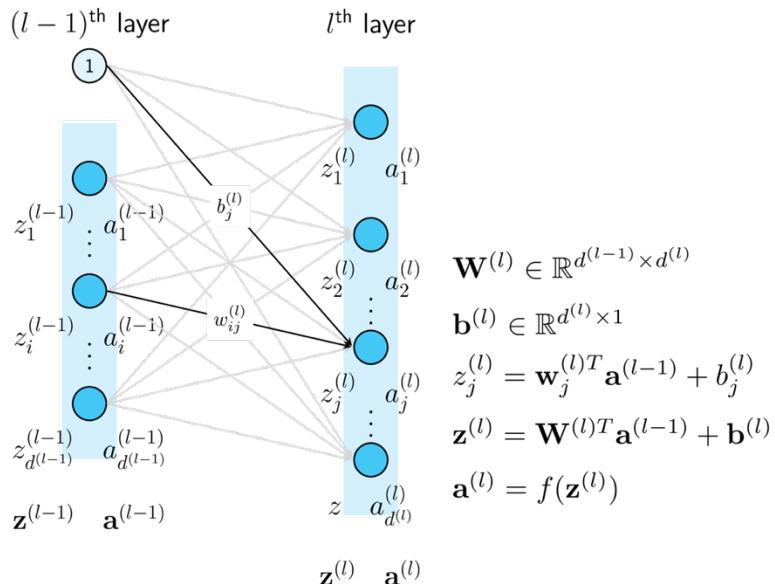
Phụ lục

A Cơ sở lý thuyết

A.1 Tính toán truyền tín hiệu trong mạng nơ-ron

Như trong phần 2.1.1, Số lượng tầng (layer) của một mạng nơ-ron được tính bằng số lượng tầng ẩn cộng với 1, được kí hiệu là L .

Giả sử gọi số lượng nơ-ron tại tầng (l) là $d^{(l)}$ (không tính nơ-ron bias), trong đó $l = 0$ với tầng đầu vào. Tại mỗi nơ-ron của mỗi tầng l , quy ước đầu vào là $\mathbf{z}^{(l)} = \{z_1^{(l)}, z_2^{(l)}, \dots, z_{d^{(l)}}^{(l)}\}$, đầu ra là $\mathbf{a}^{(l)} = \{a_1^{(l)}, a_2^{(l)}, \dots, a_{d^{(l)}}^{(l)}\}$. Quá trình truyền tín hiệu giữa hai nơ-ron thuộc hai lớp kế tiếp nhau được thể hiện như trong Hình 31:



Hình 31 Quá trình truyền tín hiệu giữa các nơ-ron thuộc hai tầng kế tiếp [25].

Trong Hình 31, $\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l)}}$ là thành phần bias của tầng thứ (l). Kích thước, số chiều của mỗi thành phần trong mạng nơ-ron được mô tả như trong hình trên. Trong một mạng nơ-ron có L tầng, có L ma trận trọng số tương ứng, được ký hiệu là $\mathbf{W}^{(l)} \in$

$R^{d^{(l-1)} \times d^{(l)}}$, với $l = 1, 2, \dots, L$. Trong đó, $\mathbf{W}^{(l)}$ thể hiện các kết nối từ tầng $(l-1)$ đến tầng l , mỗi phần tử $w_{ij}^{(l)}$ thể hiện kết nối từ nơ-ron thứ i của tầng $(l-1)$ đến nơ-ron thứ j của tầng (l) . Khi cần tối ưu một mạng nơ-ron, ta cần đi tìm, tối ưu các tham số \mathbf{W} và \mathbf{b} này.

Tại mỗi nơ-ron, trừ nơ-ron thuộc tầng đầu vào, được tính theo Công thức 16 như sau:

$$a_j^{(l)} = f(z_j^{(l)}) = f(\mathbf{w}_j^{(l)T} \mathbf{a}^{(l-1)} + b_j^{(l)})$$

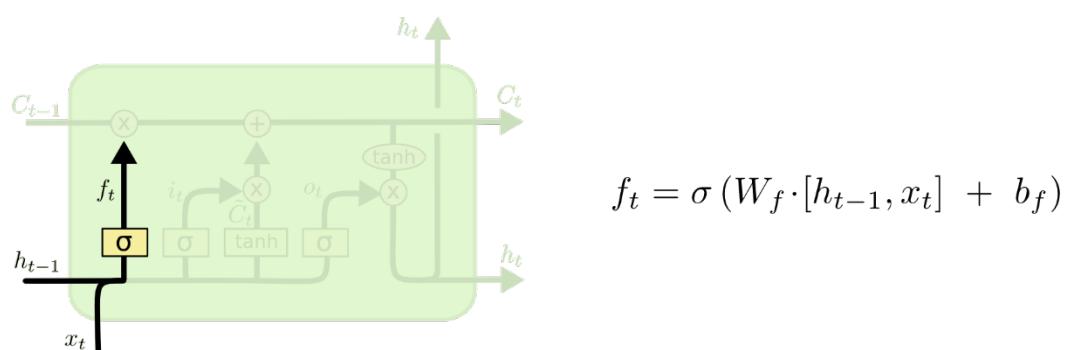
Công thức 16 Công thức tính đầu ra tại mỗi nơ-ron.

Trong đó, $f(\cdot)$ được gọi là hàm kích hoạt. Hàm kích hoạt được sử dụng để phi tuyến hóa các đầu vào, giúp cho khi sử dụng nhiều tầng sẽ có ý nghĩa cho quá trình học. Các hàm kích hoạt thường được sử dụng trong mạng nơ-ron nhân tạo bao gồm: (i) hàm sigmoid, (ii) hàm tanh, (iii) hàm ReLU...

A.2 Tính toán tại các cổng trong mạng LSTM

Cổng lăng quên

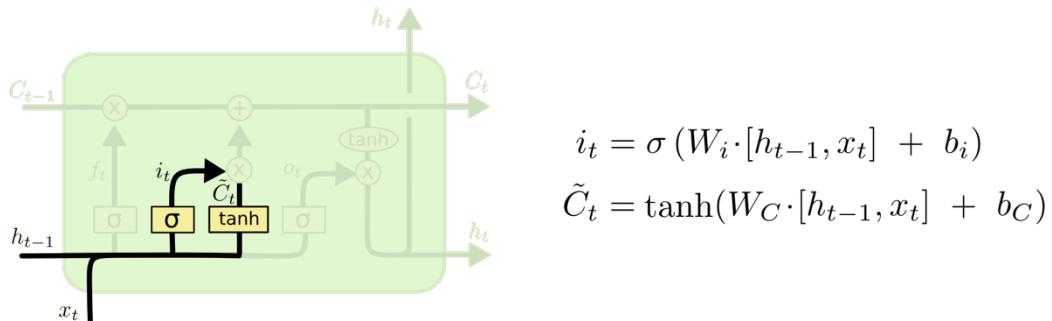
Bước đầu tiên trong LSTM là quyết định xem những thông tin nào cần được bỏ đi trong trạng thái tế bào. Quá trình này được thực hiện thông qua một tầng sigmoid - tầng cổng quên. Với đầu vào là x_t và h_{t-1} , đầu ra f_t là một giá trị nằm trong khoảng $[0, 1]$ cho mỗi số trong trạng thái tế bào C_{t-1} , thể hiện mỗi chiều của C_{t-1} sẽ giữ lại hay mất đi bao nhiêu phần, trong đó, 1 biểu thị là giữ lại toàn bộ thông tin, còn 0 là toàn bộ thông tin sẽ bỏ đi. Chi tiết công thức được biểu diễn như trong Hình 32:



Hình 32 Cổng lăng quên trong LSTM [7].

Cổng đầu vào

Bước thời gian tiếp theo là quyết định xem sẽ lưu lại thông tin mới nào vào trạng thái tế bào, được mô tả như trong Hình 33.

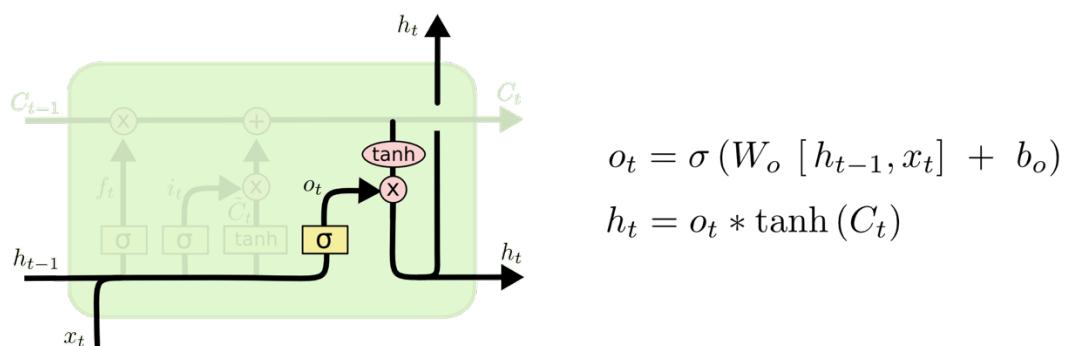


Hình 33 Cổng đầu vào trong LSTM [7].

Quá trình này gồm 2 bước. Bước đầu tiên là sử dụng một tầng sigmoid – tầng đầu vào với đầu vào là i_t , quyết định xem thông tin nào sẽ được cập nhật vào trạng thái tế bào. Ở bước thứ hai, tầng tanh sẽ tạo ra một vector \tilde{C}_t mới dựa trên những thông tin mới có được từ đầu vào. Sau đó, ta sẽ cập nhật trạng thái tế bào cũ C_{t-1} thành trạng thái mới C_t . Ta thực hiện nhân trạng thái cũ C_{t-1} với f_t để bỏ đi những thông tin không cần thiết và cập nhật trạng thái tế bào bằng cách cộng với $i_t * \tilde{C}_t$.

Cổng đầu ra

Cuối cùng, dựa vào trạng thái tế bào mới cùng với đầu vào tại bước thời gian hiện tại, ta sẽ quyết định xem đầu ra là gì. Mặc dù đầu ra dựa trên trạng thái tế bào, nhưng nó vẫn sẽ được tiếp tục sàng lọc.



Hình 34 Cổng đầu ra trong LSTM [7].

Đầu tiên, cần chạy một tầng sigmoid - tầng đầu ra để tạo ra o_t , quyết định phần nào của trạng thái tế bào sẽ là đầu ra tại bước thời gian này. Tiếp theo, ta đưa trạng thái tế bào qua một tầng tanh để giá trị của nó về giữa -1 và 1, kết quả đầu ra cuối cùng thu được bằng cách nhân tầng giá trị thu được qua tầng tanh này với o_t . Chi tiết quá trình thực hiện này được minh họa như trên Hình 34:

A.3 Tính toán trong mô hình Sequence to Sequence

Trong mô hình Sequence to Sequence, lớp mã hóa đọc chuỗi đầu vào được biểu diễn dưới dạng chuỗi các vector $\mathbf{x} = (x_1, \dots, x_{T_x})$ thành một vector c , với T_x là độ dài cố định chuỗi đầu vào của lớp mã hóa, với cách tiếp cận sử dụng mạng RNN được biểu diễn qua Công thức 17 và Công thức 18 như sau:

$$h_t = f(x_t, h_{t-1})$$

Công thức 17 Công thức trạng thái ẩn tại bước thời gian t của bộ mã hóa.

$$c = q(\{h_1, \dots, h_{T_x}\})$$

Công thức 18 Công thức tính vector sinh ra từ chuỗi trạng thái ẩn.

Trong đó, $h_t \in \mathbb{R}^n$ là trạng thái ẩn tại bước thời gian t , c là một vector được sinh ra từ chuỗi các trạng thái ẩn, f và q là các hàm kích hoạt không tuyến tính (non-linear activation function), đơn giản có thể là một hàm sigmoid, phức tạp hơn có thể là một đơn vị LSTM. Với f là một đơn vị LSTM thì $q(\{h_1, \dots, h_T\}) = h_T$.

Lớp giải mã thường được huấn luyện để dự đoán từ tiếp theo y_t khi đã có vector ngữ cảnh c và tất cả các từ đã dự đoán trước đó $\{y_1, \dots, y_{t-1}\}$. Nói cách khác, bộ giải mã xác định xác suất trên cả chuỗi đích bằng cách phân tách xác suất chung thành các xác suất điều kiện theo thứ tự, được thể hiện qua Công thức 19 như sau:

$$p(\mathbf{y}) = \prod_{t=1}^T p(y_t | \{y_1, \dots, y_{t-1}\}, c)$$

Công thức 19 Công thức tính xác suất cho mỗi từ trong bộ giải mã.

Trong đó, $\mathbf{y} = (y_1, \dots, y_{T_y})$, với T_y là độ dài cố định chuỗi đầu vào của lớp giải hóa,. Với mạng RNN, mỗi xác suất có điều kiện trong Công thức 19 được tính qua Công thức 20 như sau:

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c)$$

Công thức 20 Công thức tính mỗi thành phần xác suất có điều kiện.

Với g là một hàm phi tuyến tính, có khả năng chứa nhiều lớp bên trong, đưa ra xác suất của y_t và s_t là trạng thái ẩn của RNN. Có một chú ý là có một số kiến trúc khác như mô hình lai giữa RNN và Deconvolutional Neural Network (mạng thần kinh khử tích chập) có thể được sử dụng [26].

B Các ví dụ văn bản thực nghiệm

B.1 Văn bản ví dụ 1 được thực nghiệm trên tiếng Anh

-lrb- cnn -rrb- a duke student has admitted to hanging a noose made of rope from a tree near a student union , university officials said thursday . the prestigious private school did n't identify the student , citing federal privacy laws . in a news release , it said the student was no longer on campus and will face student conduct review . the student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early wednesday , the university said . officials are still trying to determine if other people were involved . criminal investigations into the incident are ongoing as well . students and faculty members marched wednesday afternoon chanting `` we are not afraid . we stand together , " after pictures of the noose were passed around on social media . at a forum held on the steps of duke chapel , close to where the noose was discovered at 2 a.m. , hundreds of people gathered . `` you came here for the reason that you want to say with me , ' this is no duke we will accept . this is no duke we want . this is not the duke we 're here to experience . and this is not the duke we 're here to create , ' " duke president richard brodhead told the crowd . the incident is one of several recent racist events to affect college students . last month a fraternity

at the university of oklahoma had its charter removed after a video surfaced showing members using the n-word and referring to lynching in a chant . two students were expelled . in february , a noose was hung around the neck of a statue of a famous civil rights figure at the university of mississippi . a statement issued by duke said there was a previous report of hate speech directed at students on campus . in the news release , the vice president for student affairs called the noose incident a `` cowardly act . " `` to whomever committed this hateful and stupid act , i just want to say that if your intent was to create fear , it will have the opposite effect , " larry moneta said wednesday . duke university is a private college with about 15,000 students in durham , north carolina . cnn 's dave alsup contributed to this report .

B.2 Văn bản ví dụ 2 được thực nghiệm trên tiếng Anh

-lrb- cnn -rrb- a naturalized u.s. citizen pleaded not guilty in ohio friday to federal charges of providing material support to terrorists and lying to the fbi . abdirahman sheik mohamud , 23 , of columbus , allegedly traveled to syria for training and wanted to return home to kill americans -- particularly u.s. soldiers , execution style , the u.s. department of justice said thursday . mohamud was remanded into custody on friday . `` i am confident in the system working fairly and -lrb- in -rrb- our client getting a vigorous and aggressive defense , " said his lawyer , sam shamansky . mohamud told someone that he wanted to target u.s. armed forces , police officers or other people in uniform , the indictment alleges , adding that `` mohamud 's plan was to attack a military facility , and his backup plan was to attack a prison . " `` mohamud talked about doing something big in the united states . he wanted to go to a military base in texas and kill three or four american soldiers execution style , " it says . mohamud allegedly said he was happy that his brother , aden , died fighting for al-nusra front , al qaeda 's largest affiliate in syria . mohamud told someone he planned to join aden in death soon , the indictment says . he became a u.s. citizen in february 2014 and submitted a u.s. passport application days later , according to the indictment . mohamud traveled to syria in april 2014 `` for the

purpose of training and fighting with terrorists , " prosecutors said in a news release . to get there , mohamud bought a one-way ticket to greece with a layover in istanbul , turkey , the department of justice said . he skipped the connecting flight `` and instead completed pre-arranged plans to travel to syria . " once there , he trained in shooting weapons , breaking into homes , using explosives and hand-to-hand combat , prosecutors said . mohamud `` also stated that , after completing this training , he was instructed by a cleric in the organization to return to the united states and commit an act of terrorism . " cnn 's john newsome contributed to this story .

B.3 Văn bản ví dụ 1 được thực nghiệm trên tiếng Việt

trên tạp chí figaro số mới nhất , lý băng băng khoe trọn thân hình mảnh mai , vòng eo con kiến trong một thiết kế ôm sát có gam màu pastel tím nữ tính và dịu dàng . phần tóc bạch kim được vuốt ngược cầu kỳ kết hợp tới gương mặt được make-up sắc nét giúp mỹ nhân họ lý vừa xinh đẹp vừa cá tính . tuy nhiên , hà hò không phải là mỹ nhân đầu tiên gây ấn tượng với chiếc váy được cắt khoét táo bạo này . trước đó , lady gaga từng diện thiết kế này trong poster quảng cáo thương hiệu . mẫu váy được cả ba người đẹp yêu thích là một thiết kế mới của thương hiệu versace . sản phẩm này nằm trong bộ sưu tập xuân hè 2014 đang được bán ngoài thị trường với giá khoảng 120 triệu đồng . những hình ảnh khác trong bộ ảnh mới được thực hiện của nữ diễn viên hàng đầu cbiz . sự độc đáo trong mỗi bức hình khiến người hâm mộ liên tưởng đến nhân vật " ma nữ tóc trắng " của lý băng băng trong vua kungfu . đây là tác phẩm đã đưa tên tuổi của cô ra tầm thế giới . không những thế , lý băng băng còn khiến các fan trầm trồ ngưỡng mộ bởi nhan sắc không tuổi và vóc dáng minh hạc xương mai đáng mơ ước . sinh năm 1973 song tới nay , mỹ nhân họ lý vẫn chưa chịu yên bề gia thất . cô được liệt kê vào danh sách những gái é đất giá nhất showbiz hoa ngữ ở thời điểm hiện tại . dù đường tình duyên lận đận song sự nghiệp của mỹ nhân họ lý ngày một thăng hoa . không chỉ là một trong những ngôi sao hàng đầu hoa ngữ , băng băng còn được các

nhà_sản_xuất phim hollywood chọn_mặt_gửi_vàng trong các dự_án phim định_dám . theo tri_thúc trẻ . tweet .

B.4 Văn bản ví dụ 2 được thực nghiệm trên tiếng Việt

hơn 19h tối 4 . 5 , ngọn_lửa bùng_cháy dữ_dội tại khu sản_xuất của nhà_máy giấy thành_đạt , nằm_trong cụm công_nghiệp phong_khê (thành_phố bắc_ninh) . theo ghi_nhận của pv , ngọn_lửa nhanh_chóng lan khắp khu nhà_kho rộng cả nghìn m2 , bốc cao hàng chục mét , sáng_rực cả một vùng trong đêm_tối . đến 23h30 , cảnh_sát pccc vẫn chưa khống_ché được ngọn_lửa . khu nhà_xưởng được thiết_kế bằng mái_tôn bị ngọn_lửa nung_nóng đã sập_đổ . đến 22h30 ngọn_lửa vẫn chưa được khống_ché . gần 20 xe_cứu_hỏa và cả hàng trăm cảnh_sát có_mặt tại hiện_trường để khống_ché ngọn_lửa từ nhiều hướng , tuy_nhiên ngọn_lửa vẫn bốc lên nghi_ngút . đại_tá đoàn_việt_mạnh , cục_trưởng cảnh_sát pccc (bộ_công_an) cho_biết , đã huy_động 9 xe_cứu_hỏa của bắc_ninh , 4 xe của hà_nội và cả xe của cục tham_gia chữa_cháy . khu_nhà_sản_xuất và kho_chứa rộng khoảng 6000m2 bị lửa thiêu_hủy . diện_tích nhà_xưởng bị cháy ước khoảng 4000m2 phần_lớn là nơi sản_xuất và lưu_giữ giấy thành_phẩm của công_ty thành_đạt . anh trịnh , nhân_viên bảo_vệ cho_biết , khoảng 19h , lực_lượng bảo_vệ phát_hiện đám cháy_bùng lên từ khu_vực nhà điều_hành giáp nhà_kho . ngay_sau_đó , ảnh_hướng của gió do không_khí lạnh kèm mưa nhỏ tràn về khiến chỉ khoảng một phút sau ngọn_lửa lan nhanh sang khu kho và xưởng sản_xuất và bùng lên dữ_dội . theo anh trịnh , toàn_bộ khu_nhà điều_hành và xưởng sản_xuất đã được niêm_phong trước dịp nghỉ_lễ , nhà_máy nghỉ sản_xuất nên không_có người trong khu_vực xáy_ra hỏa_hoạn . lực_lượng chức_năng đã phải huy_động xe_ủi đến phá tường rào ở một_số nơi để đưa thiết_bị chữa_cháy vào khống_ché ngọn_lửa . theo vnexpress .