HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tiến Đạt

PHÁT HIỆN Ý ĐỊNH NGƯỜI DÙNG TRONG HỆ THỐNG HỎI ĐÁP SỬ DỤNG MẠNG NƠRON

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

HÀ NỘI - NĂM 2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tiến Đạt

PHÁT HIỆN Ý ĐỊNH NGƯỜI DÙNG TRONG HỆ THỐNG HỎI ĐÁP SỬ DỤNG MẠNG NƠRON

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. Ngô Xuân Bách

HÀ NỘI - NĂM 2019

LỜI CAM ĐOAN

là Nguyễn Tiến Đạt, học viên lớp M18CQIS01 xin cam đoan báo cáo luận văn này được viết bởi dưới sự hướng dẫn của thầy giáo, tiến sĩ Ngô Xuân Bách. Trong toàn bộ nội dung của luận văn, những điều được trình bày là kết quả của cá nhân hoặc là được kế thừa, tổng hợp từ nhiều nguồn tài liệu khác được liệt kê trong danh mục tài liệu tham khảo rõ ràng.

Hà Nội, ngày..... tháng..… năm 2019

Học viên

Nguyễn Tiến Đạt

LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy cô tại trường Học viện Công nghệ Bưu chính Viễn thông, đặc biệt các thầy cô khoa Hệ thống thông tin, đã tận tình dạy dỗ, giúp đỡ và tạo mọi điều kiện tốt nhất cho em trong suốt quãng thời gian em theo học tại trường, để em có thể hoàn thành được luận văn này.

Em cũng xin gửi lời cảm ơn tới thầy hướng dẫn TS. Ngô Xuân Bách, thầy đã tận tình hướng dẫn khoa học và giúp đỡ, chỉnh sửa và chỉ bảo em trong suốt quá trình nghiên cứu và hoàn thành luận văn.

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

Luận văn được hỗ trợ bởi Đại học Quốc gia Hà Nội, thông qua đề tài mã số QG.19.59.

Em xin trân trọng cảm ơn.

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
ACC	Accuracy	Độ chính xác accuracy
CBOW	Continuous Bag of Words	Túi từ liên tiếp
CNN	Convolutional Neural network	Mạng nơron tích chập
IDF	Inverse Document Frequency	Tần số nghịch của 1 từ trong tập văn bản
LSTM	Long short-term memory	Mạng nơron cải tiến giải quyết vấn đề phụ thuộc từ quá dài
N-Gram	N-Gram	Tần suất xuất hiện của n kí tự liên tiếp
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
RNN	Recurrent Neural Network	Mạng nơron hồi quy
SVM	Support Vector Machine	Máy vector hỗ trợ
TF	Term Frequency	Tần số xuất hiện 1 từ trong 1 văn bản

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Nghiên cứu về hệ thống hỏi đáp tự động (Q&A) đã được quan tâm từ rất lâu trên thế giới. Ngay từ những năm 1960, các hệ thống hỏi đáp đầu tiên sử dụng cơ sở dữ liệu đã được ra đời. Với mục đích hệ thống được xây dựng để thực hiện việc tìm kiếm tự động câu trả lời từ một tập lớn các tài liệu cho câu hỏi đầu vào một cách chính xác.

Hiện nay, số lượng hệ thống hỏi đáp ngày càng tăng, số lượng câu hỏi gửi về các hệ thống mỗi ngày ngày càng nhiều và việc phát hiện được ý định câu hỏi của người dùng là một trong những bước đầu tiên để lựa chọn được câu trả lời đúng với mong muốn người dùng quan tâm.

Ở các trường Đại học, hệ thống hỏi đáp đang được áp dụng phổ biến và từng bước phát triển, điều này giúp các học sinh THPT muốn tiếp cận, tìm hiểu thông tin cũng như bản thân các sinh viên trong trường muốn biết rõ hơn về các khóa học, lợi ích mà trường Đại học đang có một cách thuận tiện, nhanh chóng. Tuy nhiên, để giải quyết số lượng câu hỏi lớn trong một thời gian thì việc xây dựng đề xuất giải pháp phát hiện thông tin người dùng muốn hỏi trong hệ thống hỏi đáp là tiền đề để xác định và tìm kiếm được câu trả lời phù hợp với ý định người dùng.

Vì những lý do trên nên quyết định lựa chọn đề tài ***"Phát hiện ý định người dùng trong hệ thống hỏi đáp sử dụng mạng nơron"*** để nghiên cứu và đưa ra một giải pháp sử dụng học máy để phát hiện ý định người dùng trong hệ thống hỏi đáp. Từ đó các hệ thống hỏi đáp sẽ tiết kiệm được thời gian, giải quyết được các câu hỏi nhanh chóng và đúng vấn đề mà các học sinh THPT hay Đại học đang có nhu cầu muốn hỏi. Cùng với đó, những nghiên cứu trong khóa luận có thể coi là tiền đề cho các nghiên cứu tiếp theo để đưa ra các câu trả lời và phân loại câu hỏi theo ý định người dùng cho một hệ thống hỏi đáp ngày một hoàn thiện.

Luận văn được tổ chức gồm ba chương gồm:

Chương 1: Giới thiệu tổng quan về bài toán xử lý ngôn ngữ tự nhiên. Tìm hiểu bài toán phân loại văn bản và giới thiệu bài toán phát hiện ý định người dùng trong hệ thống hỏi đáp.

Chương 2: Trình bày phương pháp giải quyết bài toán và các phương pháp biểu diễn đặc trưng cho văn bản cùng phương pháp học máy mà đề tài lựa chọn: sử dụng mạng nơron và so sánh với Support Vector Machine (SVM).

Chương 3: Trình bày về kịch bản thực nghiệm cho các trường hợp xác định ý định người dùng trên bộ dữ liệu thực nghiệm được thu thập từ: *Kênh thông tin trực tuyến, Khoa Quốc tế, Đại học quốc gia Hà Nội.*

CHƯƠNG 1: TỔNG QUAN BÀI TOÁN PHÁT HIỆN Ý ĐỊNH NGƯỜI DÙNG

1.1 Xử lý ngôn ngữ tự nhiên

Ngôn ngữ [5] là hệ thống để giao tiếp hay suy luận dùng một cách biểu diễn phép ẩn dụ và một loại ngữ pháp theo logic, mỗi cái đó bao hàm một tiêu chuẩn hay sự thật thuộc lịch sử và siêu việt. Nhiều ngôn ngữ sử dụng điệu bộ, âm thanh, ký hiệu, hay chữ viết, và cố gắng truyền khái niệm, ý nghĩa, và ý nghĩ, nhưng mà nhiều khi những khía cạnh này nằm sát quá, cho nên khó phân biệt nó.

Xử lý ngôn ngữ chính là xử lý thông tin khi đầu vào là "dữ liệu ngôn ngữ" (dữ liệu cần biến đổi), tức dữ liệu "văn bản" hay "tiếng nói". Đặc điểm chính của các kiểu dữ liệu này là không có cấu trúc hoặc nửa cấu trúc và chúng không thể lưu trữ trong các khuôn dạng cố định như các bảng biểu.

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ - công cụ hoàn hảo nhất của tư duy và giao tiếp.

Xử lý ngôn ngữ tự nhiên là lĩnh vực đã được nghiên cứu từ nhiều năm nay và đạt được nhiều bước tiến quan trọng trong những năm gần đây với các ứng dụng về bài toán trong thực tế như:

- Nhận dạng chữ viết (bao gồm chữ in và chữ viết tay),
- Nhận dạng tiếng nói,
- Dịch tự động,
- Tìm kiếm thông tin,
- Tóm tắt văn bản,
- Khai phá dữ liệu,
- Phát hiện tri thức, v.v.

1.2 Bài toán phát hiện ý định người dùng trong hệ thống hỏi đáp

1.2.1 Phân loại văn bản

Phân loại văn bản là quá trình phân lớp một đối tượng dữ liệu vào một hay nhiều lớp cho trước nhờ một mô hình phân lớp mà mô hình này được xây dựng dựa trên một tập hợp các đối tượng dữ liệu đã được gán nhãn từ trước gọi là tập dữ liệu học (tập huấn luyện).

Quá trình phân lớp còn được gọi là quá trình gán nhãn cho các đối tượng dữ liệu. Các bài toán phân loại văn bản thường thấy là:

- Phân cụm văn bản,
- Tóm tắt văn bản,
- Xác định quan điểm,
- Phát hiện ý định,
- Phân tích cảm xúc, hành vi của người dùng, v.v.

Trong nội dung luận văn này sẽ tập trung vào bài toán phát hiện ý định của người dùng trong hệ thống hỏi đáp của trường Đại học.

1.2.2 Phát biểu bài toán

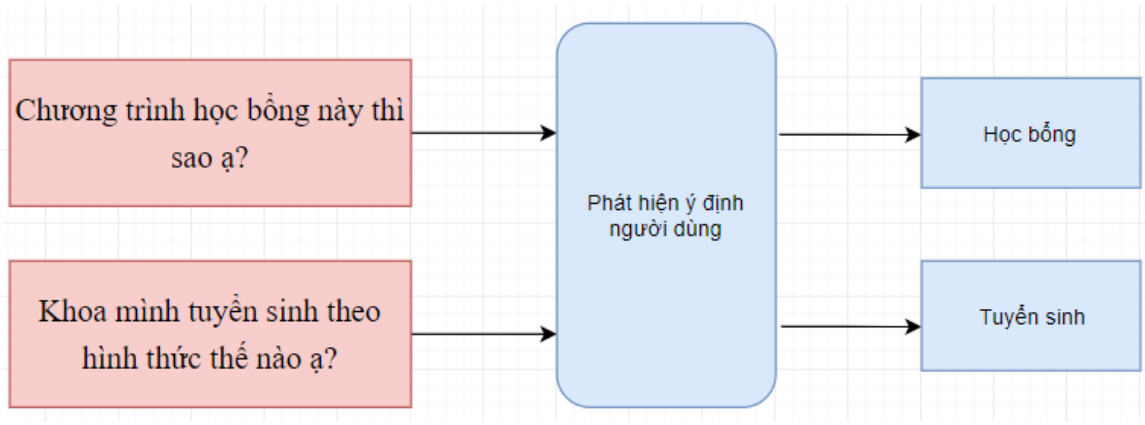
Nghiên cứu về hệ thống hỏi đáp tự động hiện đang thu hút sự quan tâm của rất nhiều các nhà nghiên cứu từ các trường đại học, các viện nghiên cứu và cả các doanh nghiệp lớn trong ngành công nghệ thông tin, có ý nghĩa khoa học lẫn ý nghĩa thực tế. Rất nhiều các hội nghị thường niên về khai phá dữ liệu, trích chọn thông tin dành một chủ đề riêng cho các nghiên cứu về hệ thống hỏi đáp như *Text REtrieval Conference* (TREC), *The Cross-Language Evaluation Forum* (CLEF)…

Bài toán xây dựng hệ thống hỏi đáp là một bài toán khó thuộc lĩnh vực xử lý ngôn ngữ tự nhiên. Ngôn ngữ tự nhiên vốn nhập nhằng, đa nghĩa, việc xác định được ngữ nghĩa của câu hỏi cũng như phát hiện ra câu trả lời là một thách thức không nhỏ. Không những vậy, các câu hỏi có thể mang theo các thói quen, phong cách gõ chữ của cá nhân người hỏi như “em muốn hỏi mã dăng ký cảu htttt ạ” (Em muốn hỏi mã đăng ký của HTTT ạ?), “Mã ngahf kh quản lí là bao nhiêu ak” (Mã ngành quản lí là bao nhiêu ạ?).

Ngoài ra, theo Bratman (1987) "Ý định người dùng còn có thể ở trạng thái *rõ ràng – explicitly hoặc tiềm ẩn/không rõ ràng – implicitly*, trực tiếp hoặc gián tiếp. Ý định rõ ràng là một tuyên bố rõ ràng và trực tiếp của người dùng về những gì người đó có kế hoạch làm" [9]. Theo Zhiyuan Chen, Bing Liu cùng cộng sự [16],[17] ý định có hai loại là ý định ẩn và ý định rõ ràng. Ý định rõ ràng tức là mong muốn của người dùng được thể hiện rõ ràng không cần kết hợp. Những trường hợp ý định kết hợp được xếp vào ý định ẩn. Ví dụ, một người dùng viết, " đang tìm kiếm một thương hiệu xe mới để thay thế cũ Ford Focus của " - "I am looking for a brand new car to replace my old Ford Focus"

Ý tưởng của luận văn là sẽ đi sâu vào giải quyết bài toán xác định ý định người dùng (học sinh, sinh viên) với:

- Đầu vào: Một câu hỏi của người dùng(học sinh, sinh viên)
- Đầu ra: Ý định của người dùng(thông tin mà học sinh, sinh viên muốn hỏi)



Hình 1.1 Bài toán phát hiện ý định người dùng

Chẳng hạn như ví dụ tại hình 1.1, với đầu vào câu hỏi trong hệ thống hỏi đáp là "Chương trình học bổng này thì sao ạ?" hệ thống sẽ đưa ra được ý định của người dùng là muốn hỏi về học bổng, hay với câu hỏi "Khoa mình tuyển sinh theo hình thức thế nào ạ?" thì hệ thống sẽ phát hiện được ý định của người dùng là muốn hỏi về vấn đề tuyển sinh.

1.2.3 Ý nghĩa bài toán

Hệ thống hỏi đáp ngày càng lớn mạnh, cùng với đó, dữ liệu câu hỏi được gửi về các hệ thống hỏi đáp ngày một lớn. Một số ứng dụng thông dụng về hệ thống hỏi đáp thường được mọi người sử dụng như:

- Siri (Apple),
- Cortana (Microsoft),
- Google Assistant (Google), …

Là những ứng dụng lớn về hệ thống hỏi đáp đang được phát triển mạnh mẽ. Việc đặt câu hỏi và nhận được câu trả lời từ các hệ thống có độ chính xác khá cao, đem lại nhiều trải nghiệm mới cũng như sự tiện ích cũng như sự hài lòng của người dùng với mục đích họ mong muốn. Việc phát hiện ý định người dùng trong hệ thống hỏi đáp có nhiều ý nghĩa, giúp ta thấy được ý định của người hỏi một cách nhanh chóng.

Ý định là một khái niệm quan trọng, được coi như chìa khóa để xây dựng các hệ thống hỏi đáp hiện nay. Luận văn mong muốn sẽ đưa ra được ý định người dùng dựa trên các ý định cho trước làm tiền đề cho các hệ thống gợi ý, giới thiệu,… vấn đề mà người dùng đang quan tâm.

Ví dụ: người dùng đặt câu hỏi *"Ngành quản lí thì cơ hội nghề nghiệp ntn ạ?"*; hệ thống sẽ đưa ra được ý định của người dùng là: *cơ hội nghề nghiệp*; từ đó làm tiền đề cho các hệ thống gợi ý, giới thiệu, đưa ra các lời mời về cơ hội việc làm liên quan đến thông tin nghề nghiệp người dùng muốn hỏi.

1.3 Các nghiên cứu liên quan

Trong những năm gần đây, đã có nhiều đề tài về phát hiện ý định người dùng với các phương pháp khác nhau được áp dụng ví dụ như đề tài *"Identifying Intention Posts in Discussion Forums"*[17] về xác định ý định người dùng dựa trên các bài viết đăng trong các diễn đàn thảo luận. Zhiyuan Chen, Bing Liu cùng cộng sự đã nghiên cứu một vấn đề không những mới lạ mà còn có giá trị lớn, cụ thể là xác định các bài viết thảo luận bày tỏ ý định của người dùng trên các diễn đàn thảo luận trực tuyến. Công trình tập trung vào việc xác định những bài đăng (post) của

người dùng với ý định rõ ràng. “Rõ ràng” nghĩa là ý định được nêu rõ ràng trong các văn bản, không cần phải suy luận. Tác giả thực hiện giải quyết vấn đề đặt ra như giải một bài toán phân loại 2 lớp lớp tích cực (bài viết chứa ý định) và lớp tiêu cực (bài viết không có ý định).

Ngoài ra, tác giả Ahmed Husseini Orabi cùng cộng sự đã thực hiện một đề tài rất thiết thực và có ý nghĩa về việc sử dụng học sâu để phát hiện trầm cảm của người dùng Twitter: *“Deep Learning for Depression Detection of Twitter Users”* [6]. Công trình trình bày việc xử lý ngôn ngữ tự nhiên trên mạng xã hội twitter, thực hiện đánh giá và so sánh trên một số mô hình học sâu, cụ thể là 3 mô hình CNN và 1 mô hình RNN và đưa ra kết quả về vấn đề rối loạn tâm thần và làm tiền đề cho hệ thống phát hiện các hành vi, cảm xúc tiêu cực của người dùng cá nhân trên mạng xã hội.

Không chỉ có vậy, đề tài *“Supervised Clustering of Questions into Intents for Dialog System Applications”* [12], của Iryna Haponchyk và cộng sự đề cập đến việc phân cụm các câu hỏi của các hệ thống hỏi đáp thành các ý định khác nhau. Cụ thể, công trình tập trung vào các ý định của người dùng hệ thống hỏi đáp thông dụng về các phân cụm như: thời tiết, giảm cân, địa điểm,… Công trình đã một phần nào đó chứng minh được “ý định” là chìa khóa quan trọng để xây dựng hệ thống hỏi đáp thông minh, xác định nhanh mục đích trong mỗi ngữ cảnh. Trong công trình này, nhóm tác giả cũng đã đề xuất một mô hình để tự động phân cụm các câu hỏi thành các mục đích của người dùng với độ chính xác của phân cụm khá cao (khoảng 80%), có thể giúp thiết kế các hệ thống hỏi đáp sau này.

Bên cạnh đó, với sức hút và sự phát triển nhanh chóng của lĩnh vực xử lý ngôn ngữ tự nhiên trong những năm gần đây, đã có rất nhiều công trình nghiên cứu của các tác giả [7], [8], [13], [14], [15] liên quan đến việc khai phá quan điểm, phân tích ý định từ nhiều nguồn dữ liệu với các phương pháp khác nhau như sử dụng phương pháp SVM, sử dụng mô hình mạng nơron hồi quy, mô hình mạng nơron tích chập,… với kết quả rất khả quan và hứa hẹn sẽ phát triển và bùng nổ trong những năm tới.

Qua việc nghiên cứu, khảo sát các đề tài liên quan đến vấn đề phát hiện ý định người dùng trong hệ thống hỏi đáp của trường Đại học còn hạn chế và chưa có nhiều. Bên cạnh đó, luận văn nhận thấy nhu cầu xử lý và phát hiện ý định người dùng trong hệ thống hỏi đáp dành cho học sinh, sinh viên mỗi kỳ tuyển dụng của trường Đại học ngày một lớn nên việc học hỏi, tiếp thu các đề tài phát hiện ý định người dùng để áp dụng với hệ thống hỏi đáp của trường Đại học là cần thiết.

Luận văn sẽ tham khảo, tìm hiểu và giới thiệu về các phương pháp phổ biến, sau đó sẽ áp dụng và đưa ra kết quả đánh giá cũng như đề xuất giải pháp để xây dựng phát triển hệ thống hỏi đáp cho các trường Đại học. Những đóng góp ban đầu của luận văn như: xử lý tiền dữ liệu, phân lớp dữ liệu trên các phương pháp khác nhau sẽ làm cơ sở ban đầu trong việc đánh giá và lựa chọn các phương pháp, mô hình học máy sao cho phù hợp với hệ thống hỏi đáp trong trường Đại học, làm tiền đề cho các ứng dụng tự động, phân tích sử dụng dữ liệu từ hệ thống hỏi đáp sau này.

1.4 Kết luận chương

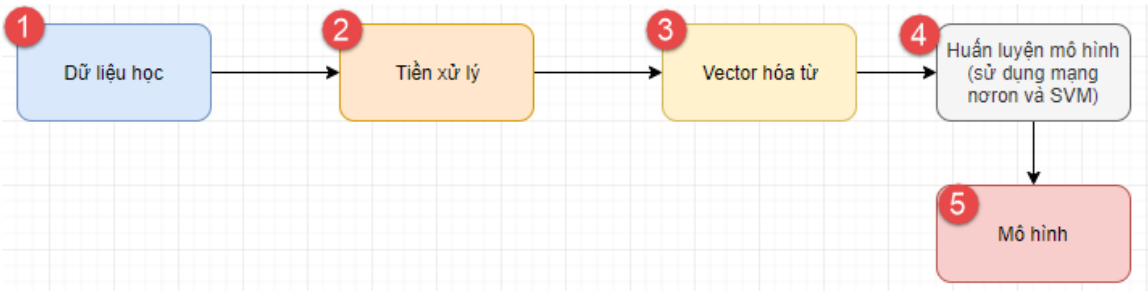
Chương 1 đã giới thiệu tổng quan về bài toán xử lý ngôn ngữ tự nhiên. Tìm hiểu bài toán phân loại văn bản và giới thiệu bài toán phát hiện ý định người dùng trong hệ thống hỏi đáp dành cho học sinh, sinh viên của trường Đại học, từ đó đưa ra những vấn đề cần làm rõ và giải quyết trong luận văn.

Trong chương 2, luận văn sẽ trình bày về hướng giải quyết cho bài toán phát hiện ý định người dùng, và đi sâu hơn trình bày về các phương pháp sẽ áp dụng để giải quyết bài toán.

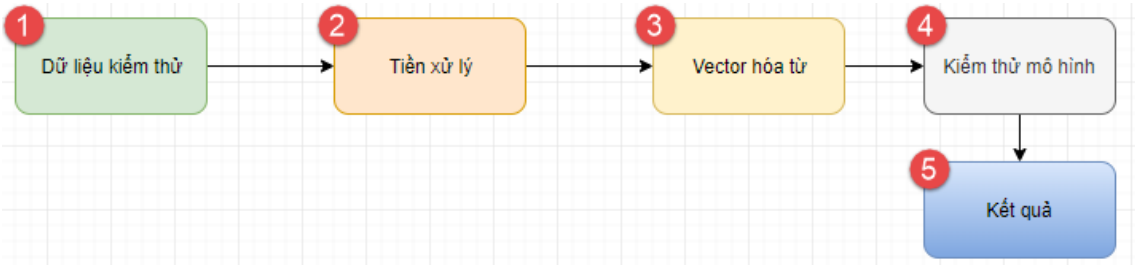
CHƯƠNG 2: PHƯƠNG PHÁP PHÁT HIỆN Ý ĐỊNH NGƯỜI DÙNG SỬ DỤNG HỌC MÁY

2.1 Phương pháp giải quyết bài toán

Để giải quyết bài toán phát hiện ý định người dùng trong hệ thống hỏi đáp của trường Đại học, từ những câu hỏi được tổng hợp từ hệ thống hỏi đáp ví dụ như: *"các chủ đề NCKH năm nay là như thế nào ạ?", "thủ tục đăng kí NCKH ?"*; ta sẽ phân lớp và đưa được về nhóm *"Nghiên cứu khoa học"*. Luận văn đã tham khảo và tìm hiểu sau đó đưa ra được các bước thực hiện để xây dựng phương pháp giải quyết cho bài toán xác định ý định người dùng được chia làm 2 giai đoạn: huấn luyện và kiểm thử. Hai giai đoạn được mô tả như trong hình 2.1 và 2.2 dưới đây:



Hình 2.1 Giai đoạn huấn luyện mô hình



Hình 2.2 Giai đoạn kiểm thử mô hình

Áp dụng phương pháp chia làm 2 giai đoạn như đã trình bày ở trên, bài toán phát hiện ý định người dùng trong hệ thống hỏi đáp, luận văn sẽ thực hiện các bước sau:

1. Chia dữ liệu thành 2 phần: dữ liệu học và dữ liệu kiểm thử
2. Tiền xử lý dữ liệu đầu vào: Loại bỏ các ký tự đặc biệt, các tiền tố dư thừa, các từ stopwords
3. Vector hóa từ cho tập dữ liệu

4. Áp dụng mô hình học máy để giải quyết bài toán, bao gồm mô hình mạng nơron và so sánh với phương pháp SVM
5. Đưa ra mô hình huấn luyện và kết quả kiểm thử.

Tại bước 1, luận văn sẽ áp dụng phương pháp K-fold cross validation và chia dữ liệu thành 3 phần bằng nhau. Cụ thể về phương pháp K-fold cross validation sẽ được luận văn trình bày tại mục 3.2 về thiết lập thực nghiệm.

Trong bước 2, tiền xử lý dữ liệu, chẳng hạn với dữ liệu đầu vào mẫu như trên, ta phải loại bỏ các tiền tố dư thừa của việc đánh số thứ tự như *"1767.", "1768."* và các *khoảng trắng* cùng với các stopwords: *"ạ", "gì", "thì", …*

Các phần tiếp theo của chương 2 sẽ trình bày chi tiết hơn về các phương pháp, mô hình và đưa ra đề xuất lựa chọn và áp dụng vào việc phát hiện ý định của người dùng trong hệ thống hỏi đáp.

2.2 Các phương pháp biểu diễn đặc trưng của văn bản

2.2.1 Phương pháp N-Gram

Mô hình ngôn ngữ là một phân bố xác suất trên các tập văn bản. Nói đơn giản, mô hình ngôn ngữ có thể cho biết xác suất một câu (hoặc cụm từ) thuộc một ngôn ngữ là bao nhiêu [2].

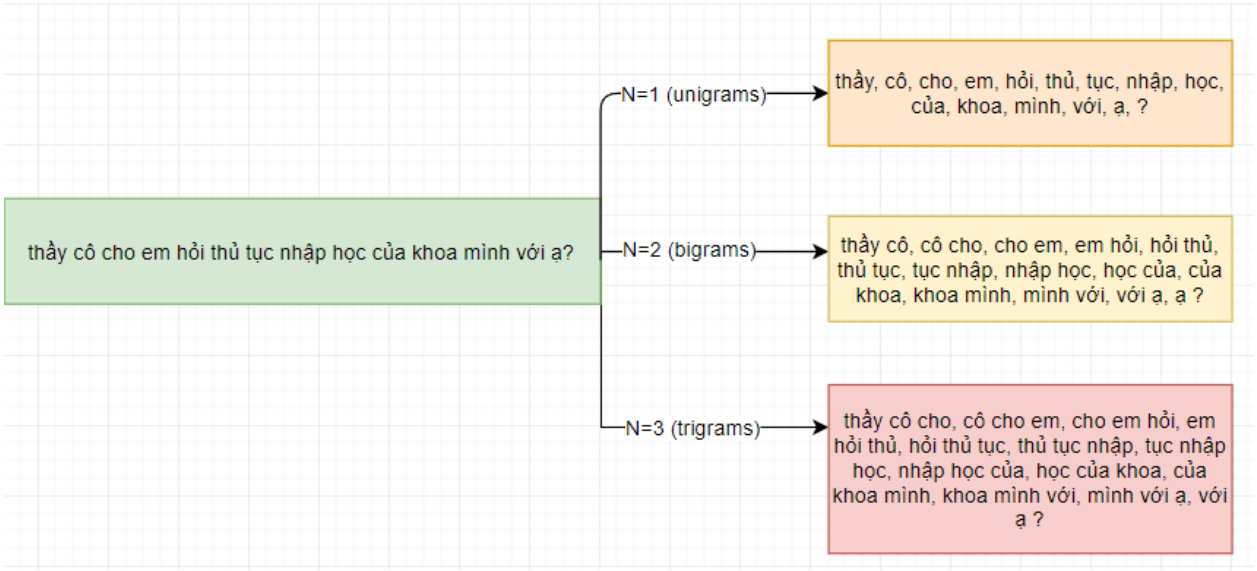
Ví dụ: khi áp dụng mô hình ngôn ngữ cho tiếng Việt:

P[“hôm qua là thứ năm”] = 0.001

P[“năm thứ hôm là qua”] = 0

Mô hình ngôn ngữ được áp dụng trong rất nhiều lĩnh vực của xử lý ngôn ngữ tự nhiên như: kiểm lỗi chính tả, dịch máy hay phân đoạn từ... Chính vì vậy, nghiên cứu mô hình ngôn ngữ chính là tiền đề để nghiên cứu các vấn đề, bài viết tiếp theo trong xử lý ngôn ngữ tự nhiên. Mô hình ngôn ngữ có nhiều hướng tiếp cận, nhưng chủ yếu được xây dựng theo mô hình N-Gram.

N-Gram là mô hình ngôn ngữ thống kê cho phép gán (ước lượng) xác suất cho một chuỗi m phần tử (thường là từ) P(w1w2…wm) tức là cho phép dự đoán khả năng một chuỗi từ xuất hiện trong ngôn ngữ đó.



Hình 2.3 Ví dụ về N-Gram

Hay ta có thể hiểu N-Gram là tần suất xuất hiện của n kí tự (hoặc từ) liên tiếp nhau có trong dữ liệu của kho ngữ liệu. Với n lần lượt bằng 1, 2, 3 ta có unigrams, bigrams, trigrams.

Bigram được sử dụng nhiều trong việc phân tích hình thái (từ, cụm từ, từ loại) cho các ngôn ngữ khó phân tích như tiếng Việt, tiếng Nhật, tiếng Trung, … Dựa vào tần suất xuất hiện cạnh nhau của các từ, người ta sẽ tính cách chia 1 câu thành các từ sao cho tổng bigram là cao nhất có thể. Với thuật giải phân tích hình thái dựa vào trọng số nhỏ nhất, người ta sử dụng n = 1 để xác định tần suất xuất hiện của các từ và tính trọng số. Do đó, để đảm bảo tính thống kê chính xác đòi hỏi các dữ liệu của kho ngữ liệu phải lớn và có tính đại diện cao.

2.2.2 Phương pháp TF-IDF

TF-TDF là thuật ngữ viết tắt của Term Frequency – Inverse Document Frequency. TF-IDF là trọng số của một từ trong văn bản thu được thông qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản. Mô hình TF-IDF là một cách để làm nổi bật các từ chỉ xuất hiện ở một vài văn bản. Bên cạnh đó là các từ xuất hiện càng nhiều ở các văn bản thì càng giảm giá trị của các từ này.

Các từ hiếm, quan trọng thường có đặc điểm sau:

- Xuất hiện nhiều trong một văn bản

- Xuất hiện ít trong cả tập ngữ liệu

$$\mathrm{TF(t,d)} = \frac{\text{Số lần từ t xuất hiện trong văn bản d}}{\text{Tổng số từ trong văn bản d}}$$

Công thức (2.1) Tính TF

$$\mathrm{IDF(t,D)} = \log \frac{\text{Tổng số văn bản trong tập mẫu D}}{\text{Số văn bản có chứa từ t}}$$

Công thức (2.2) Tính IDF

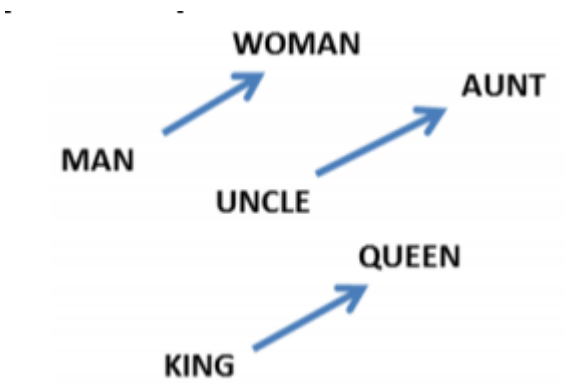
$$\mathrm{TF\_IDF(t,d,D)} = \mathrm{TF(t,d)} * \mathrm{IDF(t,D)}$$

Công thức (2.3) Tính TF_IDF

Nhận thấy hàm IDF(t,D) đảm bảo tính chất nêu trên của từ quan trọng. Một từ mà xuất hiện ở nhiều văn bản thì mẫu của hàm log lớn dẫn đến log tiến về 0 tương đương với từ này kém giá trị. Và ngược lại, số từ sử dụng trong các văn bản càng ít thì log sẽ tiến về giá trị lớn hơn. Sử dụng phương pháp TF-IDF sẽ mô tả được vector của tập ngữ liệu kích thước bằng số lượng văn bản x số lượng từ trong ngữ liệu. Mô hình TF-IDF nhấn mạnh được các từ quan trọng.

2.2.3 Phương pháp Word Vectors

Trong khi TF-IDF vẫn đặc trưng cho kiểu mô hình sử dụng phép đếm và xác suất thì Word2vec được ra đời với nhiều cải tiến đáng kể. Word2vec là phương pháp biểu diễn một từ dưới dạng một phân bố quan hệ với các từ còn lại. Mỗi từ được biểu diễn bằng một vector có các phần tử mang giá trị là phân bố quan hệ của từ này đối với các từ khác trong từ điển. Năm 2013, Google đã khởi dựng dự án word2vec của riêng mình với dữ liệu được sử dụng từ Google News [10]. Bộ dữ liệu được coi là đồ sộ nhất cho tới bây giờ với 100 tỷ từ.



Hình 2.4 Phân bố quan hệ giữa các từ trong word2vec [10]

Ví dụ bài toán kinh điển King + Man – Woman = ?. Việc nhúng các từ trong không gian vector cho thấy sự tương tự giữa các từ. Giả sử như tại hình 3.1 là một sự khác biệt về mặt giới tính giữa các cặp từ (“man”,”woman”), (“uncle”,”aunt”), (“king”,”queen”)

W(“woman”) – W(“man”) ≈ W(“aunt”) – W(“uncle”)

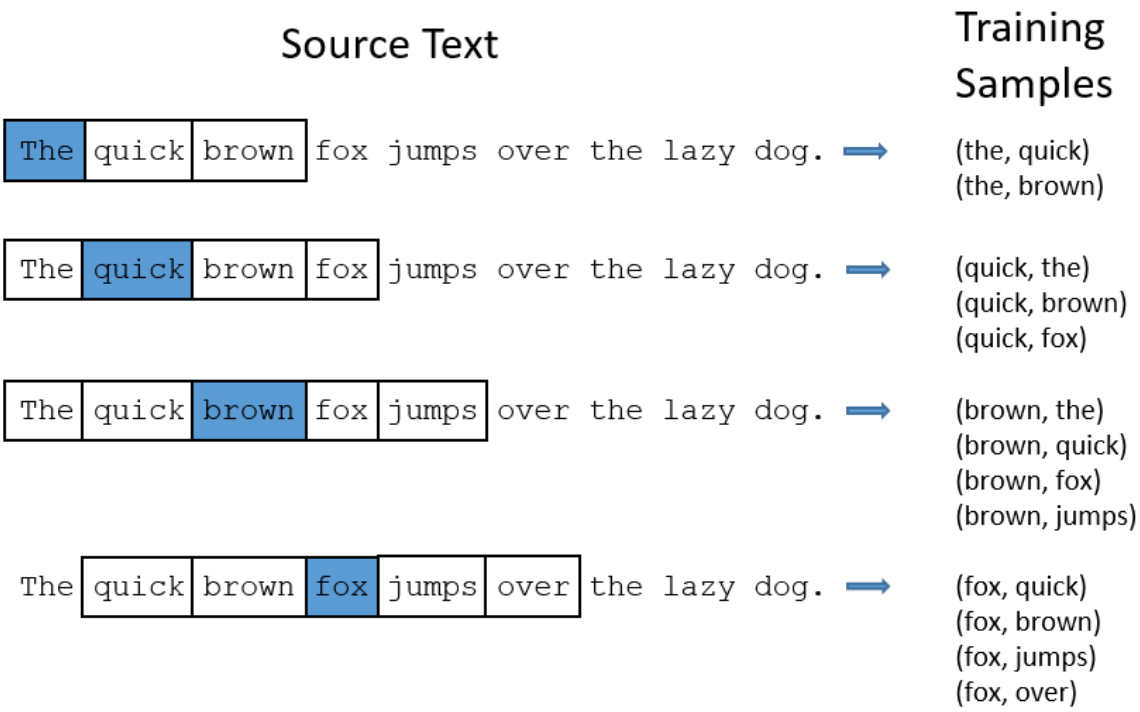
W(“woman”) – W(“man”) ≈ W(“queen”) – W(“king”)

Từ đó, kết quả của King + Man – Woman = Queen.

Để xây dựng được vector mô tả phân bố quan hệ với tập từ điển, bản chất mô hình Word2vec sử dụng một mạng nơron đơn giản với một lớp ẩn. Sau khi được huấn luyện trên toàn bộ tập văn bản, toàn bộ lớp ẩn sẽ có giá trị mô hình hóa quan hệ của từ trong tập văn bản được huấn luyện ở mức trừu tượng. Trong ngữ cảnh, từ sẽ được huấn luyện việc sử dụng thuật toán Continuous Bag of Words (CBOW) và skip gram. Bản chất của CBOW là sử dụng ngữ cảnh để đoán từ và bản chất của skip gram là dùng từ để dự đoán ngữ cảnh. Một trong hai cách sẽ được áp dụng để huấn luyện cho mô hình word2vec, trong đó cách sử dụng mô hình skip gram thường được sử dụng do việc đáp ứng tốt với tập dữ liệu lớn.

Khi sử dụng mô hình skip gram thì đầu vào là một từ trong câu, thuật toán sẽ nhìn vào những từ xung quanh nó. Giá trị số từ xung quanh nó được xét gọi là “window size”. Một window size bằng 5 có nghĩa sẽ xét 5 từ trước nó và 5 từ sau nó. Xác suất đầu ra sẽ liên quan tới khả năng tìm thấy các từ xung quanh từ hiện tại

đang xét. Xét câu “The quick brown fox jumps over the lazy dog” với window size bằng 2. Từ được bôi đậm là từ đầu vào.

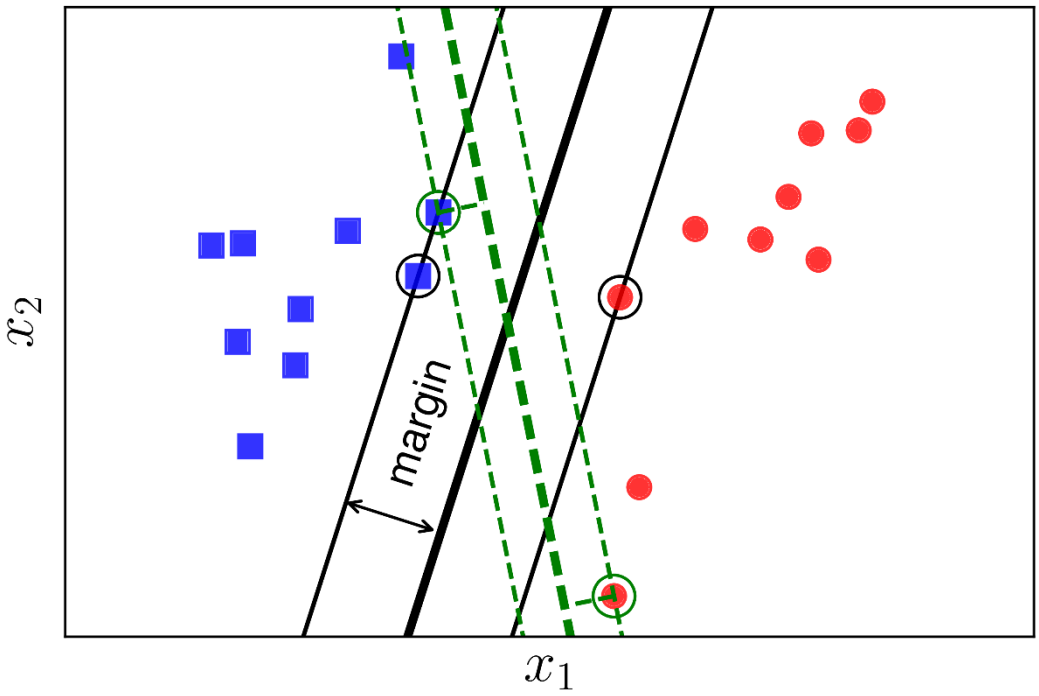


Hình 2.5 Mô hình skip-gram trong Word2vec

2.3 Các phương pháp học máy xây dựng mô hình phân lớp

2.3.1 Phương pháp SVM

Mô hình Support Vector Machine. Mô hình SVM là mô hình hết sức kinh điển trong bài toán phân loại. Tư tưởng của SVM [3] là định nghĩa ra một siêu mặt phẳng có thể phân tách các tập dữ liệu cần phân loại sao cho khoảng cách (margin) từ siêu mặt phẳng đến các tập cần phân loại là tương đương nhau và lớn nhất. Thuật toán SVM ban đầu được thiết kế để giải quyết bài toán phân lớp nhị phân với ý tưởng chính như sau:



Hình 2.6 Khoảng cánh margin của 2 phân lớp là bằng nhau và lớn nhất [3]

Trong không gian hai chiều đã biết khoảng cách từ một điểm có tọa độ (x_0,y_0) tới đường thẳng có phương trình $w_1x + w_2y + b = 0$ được tính bằng:

$$h = \frac{|\, w_1 x_0 + w_2 y_0 + b \,|}{\sqrt{w_1^2 + w_2^2}}$$

Công thức (2.4) Tính khoảng khách không gian 2 chiều

Trong không gian ba chiều khoảng cách từ một điểm có tọa độ (x_0,y_0,z_0) tới một mặt phẳng có phương trình w1x + w2y + w3z +b = 0 được tính bằng: w_1

$$h = \frac{|w_1 x_0 + w_2 y_0 + w_3 z_0 + b|}{\sqrt{w_1^2 + w_2^2 + w_3^2}}$$

Công thức (2.5) Tính khoảng cách không gian 3 chiều

Nhận thấy nếu bỏ dấu giá trị tuyệt đối của tử số thì có thể xác định được điểm đang xét nằm về phía nào của đường thẳng hay mặt phẳng. Không làm mất tính tổng quát thì những biểu thức trong dấu giá trị tuyệt đối nếu mang dấu dương

thì nằm cùng một phía dương còn những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang dấu âm thì nằm về phía âm. Những điểm nằm trên đường thẳng/ mặt phẳng sẽ làm cho giá trị của tử số bằng 0 hay khoảng cách bằng 0. Tổng quát trên không gian nhiều chiều thì sẽ phức tạp hơn so với việc biểu diễn bởi không gian 2 chiều (đường thẳng) hay không gian 3 chiều (mặt phẳng). Khái niệm này được gọi là siêu mặt phẳng có công thức $w^T x + b = 0$. Khoảng cách được tính bằng:

$$h = \frac{|w^T x_0 + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

Công thức (2.6) Tính khoảng cách trong không gian d chiều

d là số chiều của không gian.

Chất lượng của siêu phẳng được đánh giá bởi khoảng cách h giữa hai lớp, khoảng cách càng lớn thì siêu phẳng quyết định càng tốt và chất lượng phân lớp càng cao. Giả sử rằng các cặp dữ liệu của training set là (x_1, y_1), (x_2, y_2),…, (x_n, y_n) trong đó x_i là vector đầu vào của một điểm dữ liệu và y_i là nhãn của điểm dữ liệu đó. Giả sử nhãn của điểm dữ liệu có 2 giá trị là 1 và -1.

Khi đó khoảng cách từ điểm đến mặt phân chia $w_1x_1 + w_2x_2 + b = 0$ là

$$h = \frac{y_n(w^T x_n + b)}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

Công thức (2.7) Tính khoảng cách từ điểm đến mặt phân chia

Margin được tính là khoảng cách gần nhất của 1 điểm tới mặt phân chia

$$margin = \min_n \frac{y_n(w^T x_n + b)}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

Công thức (2.8) Tính khoảng cách gần nhất của 1 điểm tới mặt phân chia

Bài toán tối ưu trong SVM là bài toán tìm w và b sao cho margin này đạt giá trị lớn nhất:

$$(w, b) = arg\max_{w,b}\left\{\frac{1}{\sqrt{\sum_{i=1}^{d} \mathrm{w}_i^2}}\min_{n} y_n(w^T x_n + \mathrm{b})\right\}$$

Công thức (2.9) Tối ưu bài toán tính margin

Đối với bài toán phân lớp với số phân lớp d > 2 thì sử dụng chiến lược one-vsrest bằng cách chuyển về bài toán phân lớp nhị phân giữa 1 lớp và (d-1) lớp còn lại. Tức là sẽ phải thực hiện bài toán SVM nhị phân d lần giữa phân lớp thứ i và (d-1) phân lớp còn lại.

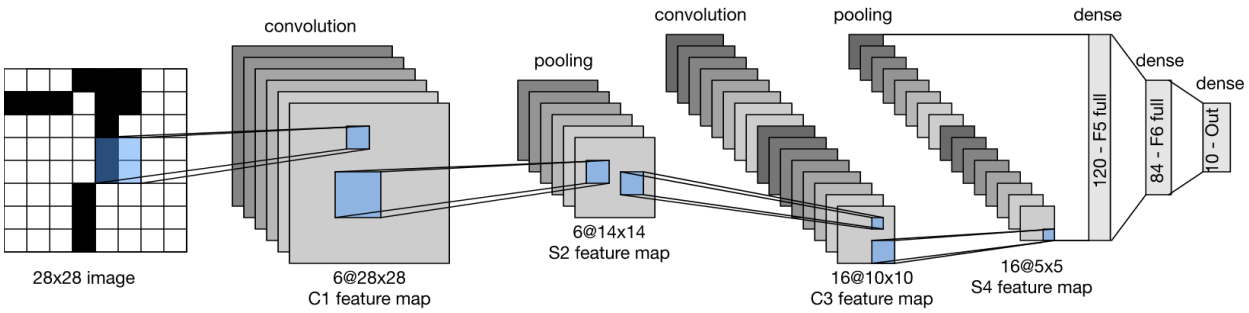
2.3.2 Kiến trúc mạng nơron tích chập (CNN)

Mạng nơron tích chập [18] là một trong những mạng truyền thẳng đặc biệt. Mạng nơron tích chập là một mô hình học sâu phổ biến và tiên tiến nhất hiện nay. Hầu hết các hệ thống nhận diện và xử lý ảnh hiện nay đều sử dụng mạng nơron tích chập vì tốc độ xử lý nhanh và độ chính xác cao. Trong mạng nơron truyền thống, các tầng được coi là một chiều, thì trong mạng nơron tích chập, các tầng được coi là 3 chiều, gồm: chiều cao, chiều rộng và chiều sâu. Mạng nơron tích chập có hai khái niệm quan trọng: kết nối cục bộ và chia sẻ tham số. Những khái niệm này góp phần giảm số lượng trọng số cần được huấn luyện, do đó tăng nhanh được tốc độ tính toán.

Có ba tầng chính để xây dựng kiến trúc cho một mạng nơron tích chập:

1. Tầng tích chập
2. Tầng gộp (pooling layer)
3. Tầng được kết nối đầy đủ (fully-connected).

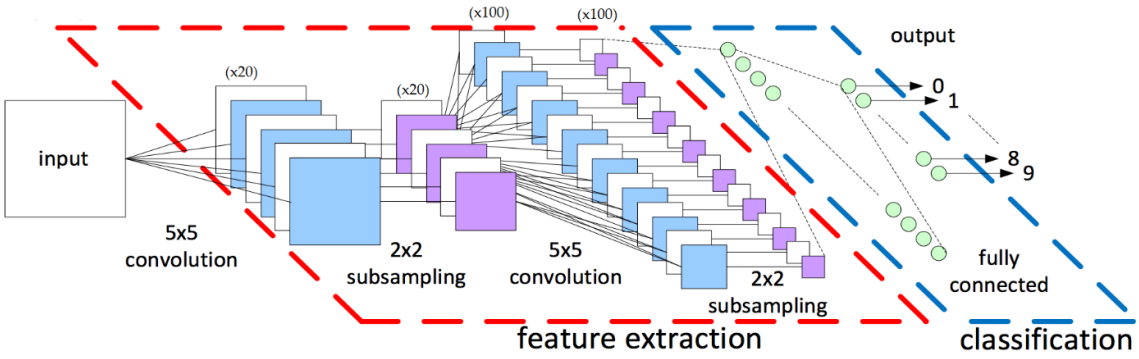
Tầng kết nối đầy đủ giống như các mạng nơron thông thường, và tầng chập thực hiện tích chập nhiều lần trên tầng trước. Tầng gộp có thể làm giảm kích thước mẫu trên từng khối 2x2 của tầng trước đó. Ở các mạng nơron tích chập, kiến trúc mạng thường chồng ba tầng này để xây dựng kiến trúc đầy đủ. Ví dụ minh họa về một kiến trúc mạng nơron tích chập đầy đủ:



Hình 2.7 Kiến trúc mạng LeNet [18]

Sau quá trình tìm hiểu và tham khảo, với điều kiện thiết bị thực nghiệm còn hạn chế, với kiến trúc CNN, luận văn quyết định áp dụng 2 convolutional layers với các thông số sau:

- Convolutional layer 1:
  - 20 Feature maps
  - Patch size 5x5
  - Pool size 2x2
- Convolutional layer 2:
  - 100 Feature maps
  - Patch size 5x5
  - Pool size 2x2

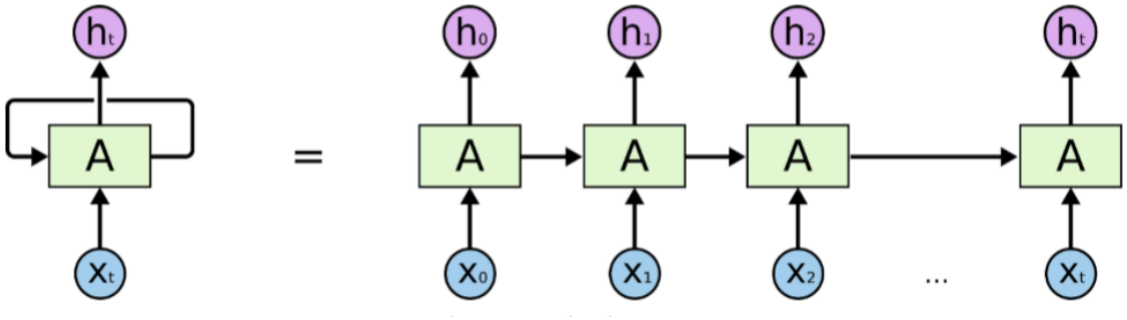


Hình 2.8 Mô hình CNN luận văn sử dụng

2.3.3 Kiến trúc mạng nơron hồi quy (RNN)

a. Giới thiệu mạng nơron hồi quy RNN

Mạng nơron hồi quy RNN được mô hình để giải quyết vấn đề mô phỏng về mặt thời gian của dữ liệu chuỗi. Do đó, mạng RNN rất phù hợp cho việc mô hình hóa xử lý ngôn ngữ. Trong đó, mỗi từ trong chuỗi đầu vào sẽ được liên kết với một bước thời gian cụ thể. Trong thực tế, số bước thời gian sẽ bằng với độ dài tối đa của chuỗi.



Hình 2.9 Mô hình mạng RNN [18]

Hình 2.4 là mô tả cơ bản của mạng RNN. Hàm A nhận đầu vào x_t tại thời điểm t và đầu ra là giá trị vector ẩn h_t. Nhận thấy, hàm A cho phép thông tin được lặp lại truyền từ một bước của mạng tới bước tiếp theo. Sử dụng mạng RNN có rất nhiều ứng dụng như nhận dạng giọng nói, mô hình hóa ngôn ngữ, dịch, nhận dạng ảnh.

Tuy nhiên, mạng RNN có vấn đề lưu trữ thông tin ngữ cảnh phụ thuộc lâu dài. Xét 2 trường hợp ví dụ sau đây:

1. Trên đường nhiều xe cộ.
2. lớn lên ở Hà Nội, có thể nhớ hết danh lam thắng cảnh tại Hà Nội.

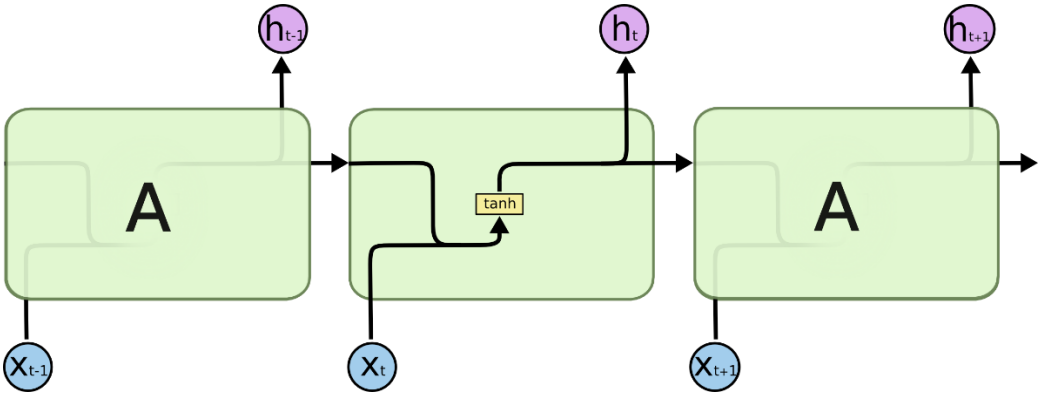
Với ví dụ 1, ta không cần thông tin ngữ cảnh, nhưng trong trường hợp 2, các thông tin phía trước đó gợi ý rằng từ tiếp theo có thể liên quan đến tên của một thành phố. Trong trường hợp 2, khoảng cách giữa 2 phụ thuộc này là lớn hơn. Để đưa ra dự đoán này, bắt buộc mạng RNN phải lưu trữ toàn bộ các từ vào trong bộ nhớ. Trong phạm vi khoảng cách phụ thuộc này thấp thì có thể khả thi, nhưng nếu

với khoảng cách cực lớn, đoạn văn dài thì việc lưu trữ của RNN trở nên nặng nề và không hợp lý. Đây chính là vấn đề lưu trữ thông tin phụ thuộc lâu dài.

Trên lý thuyết, mạng RNN có thể phát sinh bộ nhớ đủ để xử lý vấn đề lưu trữ phụ thuộc dài. Tuy nhiên, trong thực tế thì không phải vậy. Vấn đề này đã được Hochreiter (1991) đưa ra như thách thức của mạng RNN. Và mạng Long short-term memory (LSTM) được phát biểu năm 1997 đã giải quyết được vấn đề này.

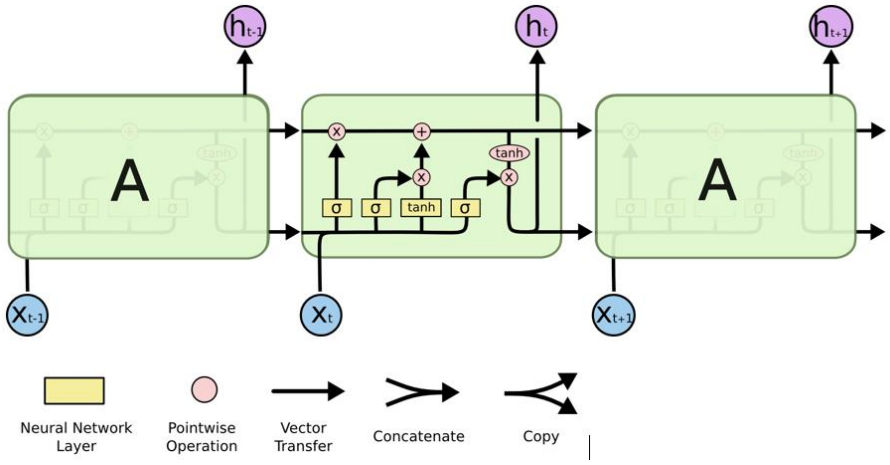
b. Mạng Long short-term memory (LSTM)

Long short term memory là cải tiến của mạng RNN nhằm giải quyết vấn đề học, lưu trữ thông tin ngữ cảnh phụ thuộc dài, cùng xem xét cách LSTM [11] cải tiến hơn so với mạng RNN. Trong mô hình RNN, tại thời điểm t thì giá trị của vector ẩn ht chỉ được tính bằng một hàm tanh



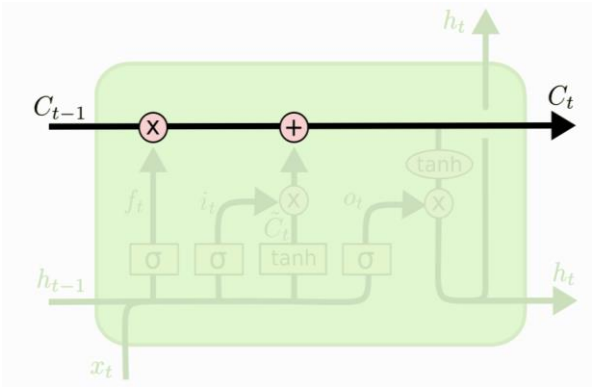
Hình 2.10 Module xử lý h_t của RNN [18]

LSTM cũng có cấu trúc mắt xích tương tự, nhưng các module lặp có cấu trúc khác hẳn. Thay vì chỉ có một layer neural network, thì LSTM có tới bốn layer, tương tác với nhau theo một cấu trúc cụ thể.



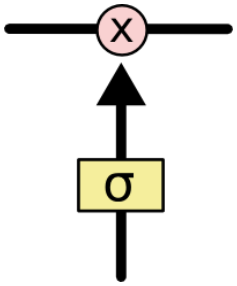
Hình 2.11 Module lặp của mạng LSTM [18]

Mấu chốt của LSTM là cell state (trạng thái nhớ), đường kẻ ngang chạy dọc ở trên cùng của hình 2.11. Cell state giống như băng chuyền, chạy xuyên thẳng toàn bộ mắc xích, chỉ một vài tương tác nhỏ tuyến tính (minor linear interaction) được thực hiện. Điều này giúp cho thông tin ít bị thay đổi xuyên suốt quá trình lan truyền.



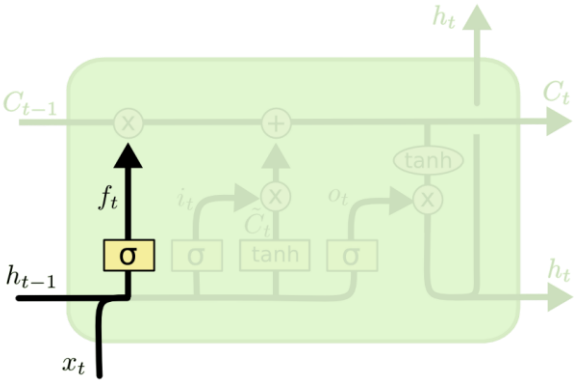
Hình 2.12 Cell state của LSTM giống như một băng chuyền [18]

LSTM có khả năng thêm hoặc bớt thông tin vào cell state, được quy định một cách cẩn thận bởi các cấu trúc gọi là cổng (gate). Các cổng này là một cách (tuỳ chọn) để định nghĩa thông tin băng qua. Chúng được tạo bởi hàm sigmoid và một toán tử nhân pointwise.



Hình 2.13 Cổng trạng thái LSTM [18]

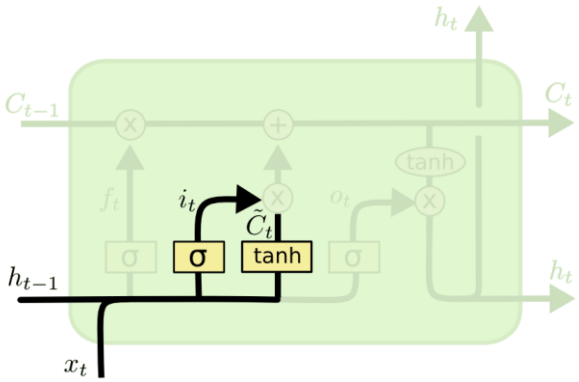
Hàm kích hoạt Sigmoid có giá trị từ 0 – 1, mô tả độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu thu được zero điều này có nghĩa là "không cho bất kỳ cái gì đi qua", ngược lại nếu thu được giá trị là một thì có nghĩa là "cho phép mọi thứ đi qua". Một LSTM có ba cổng như vậy để bảo vệ và điều khiển cell state. Quá trình hoạt động của LSTM được thông qua các bước cơ bản sau. Bước đầu tiên của mô hình LSTM là quyết định xem thông tin nào cần loại bỏ khỏi cell state. Tiến trình này được thực hiện thông qua một sigmoid layer gọi là "forget gate layer" – cổng chặn. Đầu vào là h_{t-1} và x_t, đầu ra là một giá trị nằm trong khoảng [0, 1] cho cell state C_{t-1}. 1 tương đương với "giữ lại thông tin", 0 tương đương với "loại bỏ thông tin".



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

Hình 2.14 Cổng chặn f_t [18]

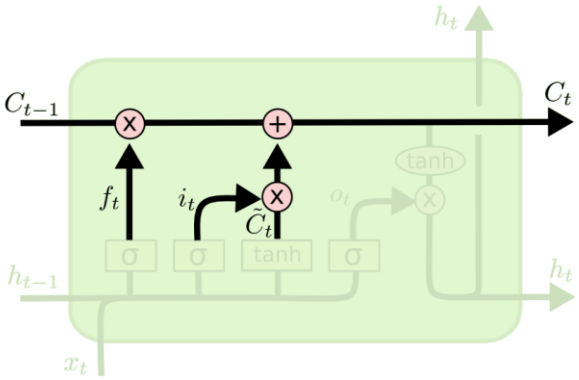
Bước tiếp theo, cần quyết định thông tin nào cần được lưu lại tại cell state, có hai phần là single sigmoid layer được gọi là "input gate layer"- cổng vào quyết định các giá trị sẽ cập nhật. Tiếp theo, một tanh layer tạo ra một vector ứng viên mới được $\tilde{C}_t$ thêm vào trong cell state.



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

Hình 2.15 Cổng vài i_t và tanh Ct [18]

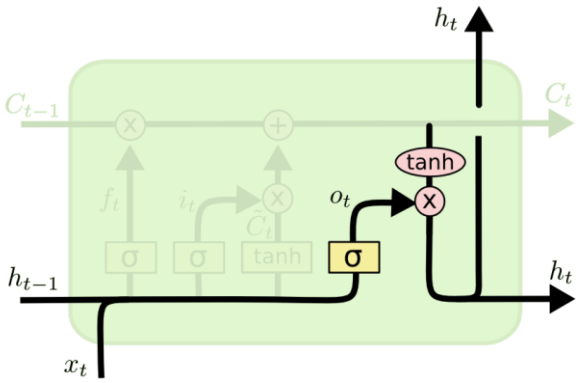
Ở bước tiếp theo, sẽ kết hợp hai thành phần này lại để cập nhật vào cell state. Lúc cập nhật vào cell state cũ, C_{t-1}, vào cell state mới C_t, đưa state cũ hàm f_t , để quên đi những gì trước đó. Sau đó, thêm $i_t * \tilde{C}_t$. Đây là giá trị ứng viên mới, co giãn (scale) số lượng giá trị mà muốn cập nhật cho mỗi state.



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

Hình 2.16 Giá trị state C_t [18]

Cuối cùng, cần quyết định xem thông tin output là gì. Output này cần dựa trên cell state, nhưng sẽ được lọc bớt thông tin. Đầu tiên, áp dụng single sigmoid layer để quyết định xem phần nào của cell state dự định sẽ output. Sau đó, sẽ đẩy cell state qua tanh (đẩy giá trị vào khoảng -1 và 1) và nhân với một "output sigmoid gate" cổng ra, để giữ lại những phần muốn output ra ngoài

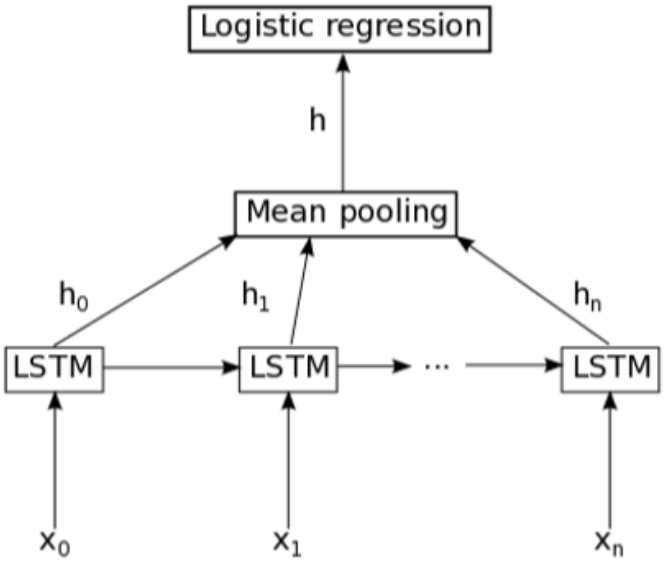


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Hình 2.17 Giá trị cổng ra và vector trạng thái ẩn h_t [18]

Mạng LSTM theo các công thức kể trên mà được lặp lại qua từng thời điểm t. Thông tin của cell state được điều khiển bởi cấu trúc các cổng chặn f_t, cổng vào it và cổng ra o_t . Trong đó cổng chặn ft chính là tư tưởng chủ đạo của mạng LSTM khi cho phép điều khiển lượng thông tin đầu vào h_{t-1} từ các thời điểm trước.

Với ưu điểm về lưu trữ phụ thuộc dài, model sử dụng để huấn luyện trong luận văn này là model LSTM. Mô hình được luận văn sử dụng được mô tả trong hình 2.17 gồm một lớp LSTM duy nhất sau đó là một lớp tổng hợp trung bình (full-connection) và một lớp hồi quy logistic. Các từ được vector hóa sử dụng mô hình Word2vec.



Hình 2.18 Mô hình LSTM luận văn sử dụng

2.4 Kết luận chương

Chương 2 đã trình bày về quá trình tìm hiểu và áp dụng thuật toán TF-IDF, N-Gram để trích xuất đặc trưng. Bên cạnh đó, chương này cũng đã trình bày giới thiệu về thuật toán SVM, mạng nơron tích chập, mạng nơron hồi quy để phân lớp dữ liệu.

Với những kiến thức đã tìm hiểu và trình bày tại chương, luận văn sẽ áp dụng kiến trúc mạng nơron hồi quy – LSTM, kiến trúc mạng CNN và so sánh với SVM.

Chương 3 sẽ tiến hành thiết lập thực nghiệm dữ liệu với phương pháp đã đề xuất trên các kịch bản khác nhau, sau đó sẽ đánh giá độ chính xác và đưa ra đề xuất định hướng tiếp theo.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

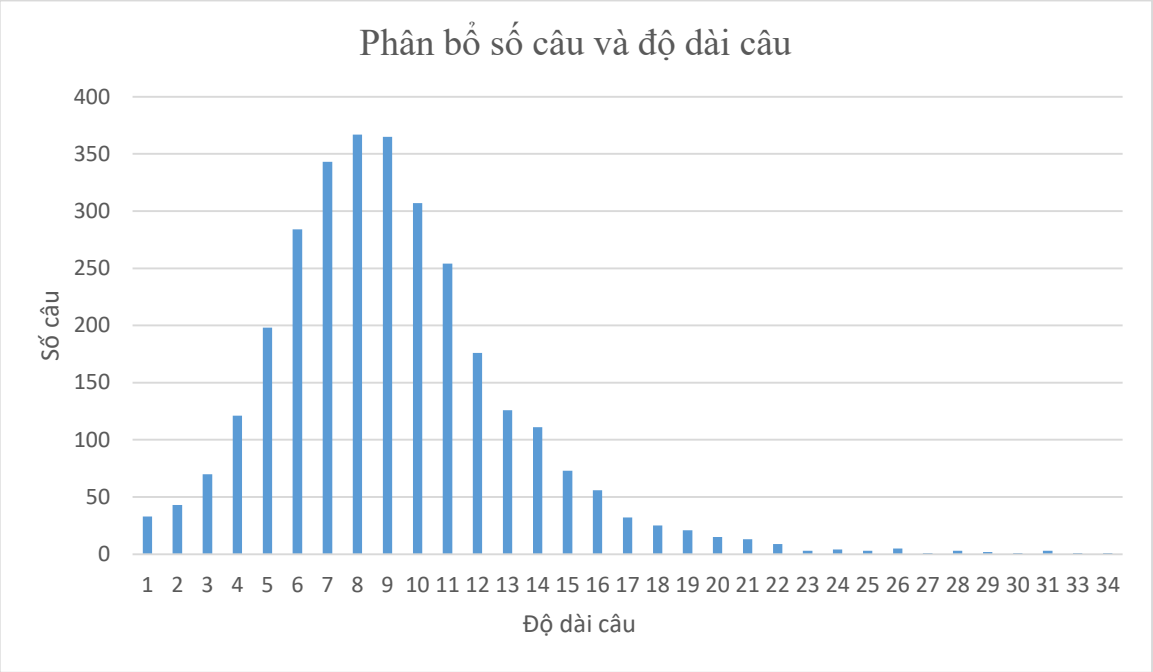
3.1 Dữ liệu thực nghiệm

Luận văn sử dụng dữ liệu thực nghiệm được thu thập từ: ***Kênh thông tin trực tuyến, Khoa Quốc tế, Đại học quốc gia Hà Nội*** với tổng số lượng là **3069** câu hỏi. Quá trình gán nhãn cho tệp dữ liệu gồm 3 bạn tham gia, 2 bạn gán nhãn và 1 bạn kiểm tra lại việc gán nhãn. Sau khi thực hiện gán nhãn, các câu hỏi được đưa về các lớp ý định sau: *Thông tin về trường, thông tin liên lạc, thông tin về khoa, cơ hội nghề nghiệp, điều kiện tiếng Anh, học phí, điểm chuẩn, nhập học, thủ tục, học bổng, nghiên cứu khoa học, tài liệu, từ chối/ không đồng ý, đồng ý, khác*. Số lượng cụ thể thu được sau quá trình gán nhãn ý định được mô tả tại bảng 3.1.

Nội dung ý định	Số lượng
Thông tin về trường	150
Thông tin liên lạc	91
Thông tin về khoa	569
Cơ hội nghề nghiệp	73
Điều kiện tiếng Anh	84
Học phí	192
Điểm chuẩn	83
Nhập học	275
Thủ tục	502
Học bổng	379
Nghiên cứu khoa học	300
Tài liệu	86
Từ chối, không đồng ý	100
Đồng ý	100
Khác	85

Bảng 3.1 Bảng mô tả dữ liệu thực nghiệm

Làm khảo sát với tập dữ liệu này, luận văn có được biểu đồ phân bố số lượng từ trong câu như biểu đồ 3.1



Hình 3.1 Biểu đồ phân bố số câu và độ dài câu

Dựa vào biểu đồ trên ta có thể thấy:

- Số lượng câu tập trung phần lớn khoảng 5 đến 12 từ
- Số lượng câu trên 100 khá nhiều, toàn bộ các câu có số lượng từ từ 4 đến 14 đều trên 100.
- Số lượng câu có độ dài 8 từ là nhiều nhất: 367 câu.
- Không có câu nào có độ dài 32 từ.
- Số lượng câu có độ dài 27, 30, 33, 34 thấp nhất: 1 câu.

3.2 Thiết lập thực nghiệm

Quá trình thực nghiệm thuật toán gồm 3 giai đoạn chính:

- Tiền xử lý dữ liệu: Loại bỏ các dư thừa, các từ vô nghĩa trong câu.
- Vector hóa và trích chọn đặc trưng: Sử dụng 2 thuật toán TF-IDF, N-Grams với n lần lượt chọn các giá trị 1, 2, 3.
- Xây dựng bộ phân lớp dữ liệu: Sử dụng LSTM, CNN và SVM.

Tiền xử lý dữ liệu: Luận văn sử dụng ngôn ngữ python để xử lý các dữ liệu dư thừa, loại bỏ các stopwords.

Vector hóa: Luận văn sử dụng filter StringToVector có sẵn trong Weka để thiết lập và trích chọn đặc trưng của dữ liệu.

Mô hình phân lớp: Mô hình mà luận văn sử dụng được mô tả trong phần 2.3.2 về mô hình CNN và phần 2.3.3 về mô hình LSTM.

Thiết lập tham số với Weka:

- Filter **StringToVector** với đặc trưng **Unigrams**:
  - IDFTransform: False
  - TFTransform: False
  - Tokenizer: NGramTokenizer
  - NGramMaxSize: 1
  - NGramMinSize: 1
- Filter **StringToVector** với đặc trưng **Bigrams**:
  - IDFTransform: False
  - TFTransform: False
  - Tokenizer: NGramTokenizer
  - NGramMaxSize: 2
  - NGramMinSize: 2
- Filter **StringToVector** với đặc trưng **Trigrams**:
  - IDFTransform: False
  - TFTransform: False
  - Tokenizer: NGramTokenizer
  - NGramMaxSize: 3
  - NGramMinSize: 3
- Filter **StringToVector** với đặc trưng **TD-IDF**:
  - IDFTransform: False
  - TFTransform: False
  - Tokenizer: WordTokenizer

- **Classifiers** function **LibSVM**:
  - SVMType: C-SVC
  - batchSize: 100
  - cacheSize: 40.0
  - coef0: 0.0
  - cost: 1.0
  - eps: 0.001
  - loss: 0.1
  - kernelType: linear: u'*v
- **Classifiers** function **Dl4jMlpClassifier** (*LSTM*):
  - Layer: LSTM + Output
  - batchSize: 100
- **Classifiers** function **NeuralNetwork** (CNN):
  - batchSize: 100
  - hiddenLayers: 20-5-5-2-2,100-5-5-2-2
  - learningRate: 0.01
  - maxInterations: 10

Sau quá trình nghiên cứu và tìm hiểu các phương pháp đánh giá thực nghiệm, luận văn đề xuất sử dụng phương pháp K-fold Cross Validation. K-fold cross validation có các đặc điểm sau:

- Tập toàn bộ các ví dụ D được chia ngẫu nhiên thành k tập con không giao nhau (gọi là "fold") có kích thước xấp xỉ nhau.
- Mỗi lần (trong số k lần) lặp, một tập con được sử dụng làm tập kiểm thử, và (k-1) tập con còn lại được dùng làm tập huấn luyện.
- k giá trị lỗi (mỗi giá trị tương ứng với một fold) được tính trung bình cộng để thu được giá trị lỗi tổng thể.

Để đánh giá chính xác hơn chất lượng của mô hình ta sử dụng thêm 2 độ đo là Precision và Recall.

- Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive (TP + FP).

$$Precision = \frac{TP}{TP + FP}$$

Công thức (3. 1) Tính Precision

- Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN).

$$Recall = \frac{TP}{TP + FN}$$

Công thức (3. 2) Tính Recall

Thực tế thì hai độ đo trên không phải lúc nào cũng tăng giảm tương ứng với nhau, có trường hợp Recall cao còn Precision thấp và ngược lại, để cho đánh giá tổng quát hơn ta dùng độ đo F-measure là trung bình điều hòa của 2 độ đo trên với hệ số 0.5 (tầm quan trọng của 2 hệ số ngang nhau):

$$F_1 = \frac{2}{\frac{1}{pricision} + \frac{1}{recall}} = 2\frac{pricision\ .recall}{pricision + recall}$$

Công thức (3. 3) Tính F_1

3.3 Công cụ thực nghiệm

3.3.1 Môi trường thực nghiệm

Thành phần	Thông số
CPU	CPU Intel Core i5 3.3GHz
RAM	RAM 8GB
Hệ điều hành (OS)	Windows 10 Professional 64bit

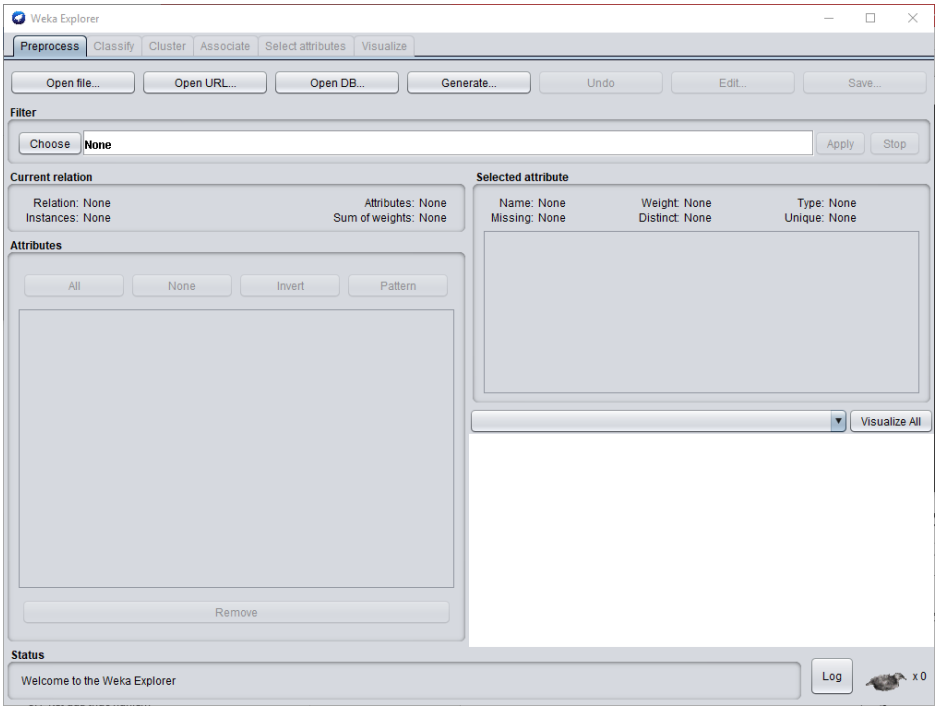
Bảng 3.2 Môi trường thực nghiệm

3.3.2 Công cụ phần mềm

Tên	Mô tả
PyCharm	IDE sử dụng Python để tiền xử lý dữ liệu. https://www.jetbrains.com/pycharm/
Weka 3.8	Công cụ tích hợp hỗ trợ các thuật toán học máy. https://www.cs.waikato.ac.nz/ml/weka/
Package WekaDeeplearning4j	Gói thư viện deep learning dành cho Weka. https://deeplearning.cms.waikato.ac.nz/user-guide/getting-started/
Package LibSVM	Gói thư viện thuật toán SVM cho Weka. http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/LibSVM.html
Packge NeuralNetwork	Gói thư viện hỗ trợ Neural Network cho Weka. https://github.com/amten/NeuralNetwork

Bảng 3.3 Công cụ phần mềm

Giới thiệu về Weka



Hình 3.2 Giao diện của Weka Explorer

Weka cung cấp 5 môi trường làm việc nhằm hỗ trợ người sử dụng hai chức năng chính là khai phá dữ liệu và thực nghiệm, đánh giá các mô hình học máy. Cụ thể:

- **Explorer**: Môi trường cho phép tiến hành khai phá dữ liệu với các tính năng tiền xử lý dữ liệu (Preprocess), phân lớp (Classify), phân cụm (Cluster), khai thác luật kết hợp (Associate). Ngoài ra, nó còn cung cấp thêm tính năng hỗ trợ lựa chọn thuộc tính (Select attributes) và mô hình hóa dữ liệu (Visualize).
- **Experimenter**: Môi trường cho phép thực nghiệm (Setup, Run), so sánh, phân tích (Analyse) các mô hình học máy.
- **KnowledgeFlow**: Môi trường này hỗ trợ các tính năng cơ bản giống như Explorer nhưng với một giao diện kéo thả để hỗ trợ học tập gia tăng.
- **Simple CLI**: Cung cấp một giao diện dòng lệnh đơn giản cho phép thực thi trực tiếp các lệnh của WEKA cho các hệ điều hành không cung cấp giao diện dòng lệnh riêng.
- **Workbench**: Môi trường này là sự kết hợp của 4 môi trường nêu trên, người sử dụng có thể tùy ý chuyển đổi mà không cần phải quay lại cửa sổ "Weka GUI Chooser".

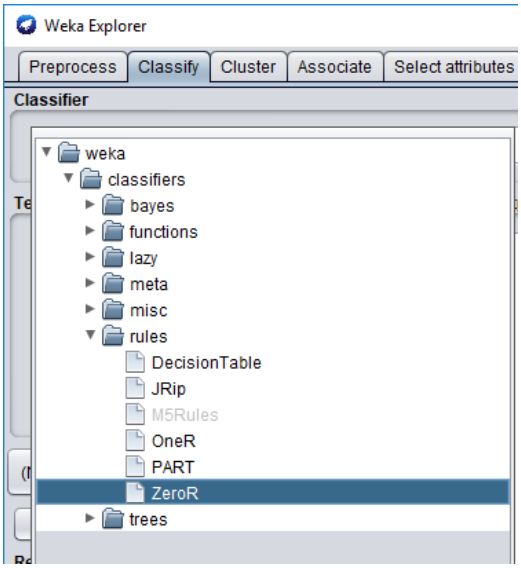
Dữ liệu đầu vào của WEKA được định dạng chuẩn ARFF với phần mở rộng "***.arff**". Tuy nhiên, WEKA cung cấp bộ chuyển đổi dữ liệu từ các định dạng "***.csv**", "***.names**", "**.data**", "***.json**", "***.libsvm**", "***.m**", "***.dat**", "***.bsi**" sang dạng "***.arff**". Ngoài ra, cũng có thể bổ sung các định dạng khác bằng cách thêm bộ chuyển đổi tập tin vào package "**weka.core.converters**". Người sử dụng cần mở tập tin dữ liệu ban đầu, tùy chỉnh dữ liệu rồi lưu lại với định dạng "***.arff**". Một tập tin ARFF là một tập tin văn bản theo bảng mã ASCII mô tả một danh sách các thể hiện (instances) của tập các thuộc tính.

```
@relation 1
@attribute name {John,Peter,Marry}
@attribute birthday date "yyyy-MM-dd HH:mm:ss"
@attribute math numeric
@attribute sentence string
@data
John,"2014-07-02 12:00:00",7,'aaa'
Peter,"2014-07-03 12:00:00",8,'aa b'
Marry,"2014-07-04 12:00:00",5,'Acvc aa1'
```

Một tập tin ARFF đơn giản có dạng:

Để xây dựng và đánh giá mô hình, Weka hỗ trợ người sử dụng thông qua tính năng **Classify** của Explorer. Người sử dụng cần thiết lập ba đối tượng cụ thể sau:

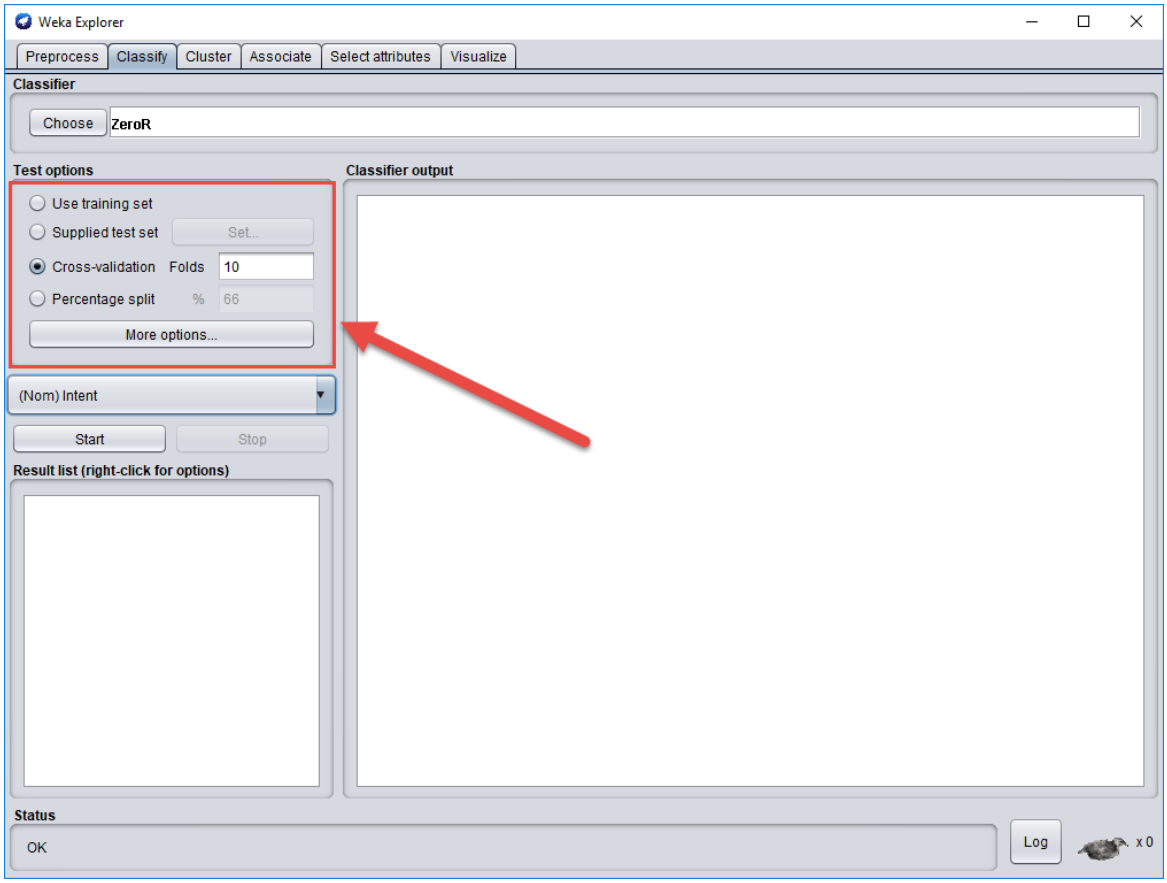
Bước đầu tiên, chọn bộ phân lớp (**Classifer**) như danh sách mô tả tại hình 3.3, việc xây dựng mô hình hồi quy tuyến tính được WEKA thực hiện trên cơ sở phương pháp bình phương tối thiểu. Có thể thực hiện lựa chọn thuộc tính bằng phương thức tham lam sử dụng loại bỏ lạc hậu hoặc xây dựng một mô hình đầy đủ từ tất cả các thuộc tính rồi loại bỏ dần các thuộc tính cho đến khi đạt được tiêu chí chấm dứt AIC. Ngoài ra, việc xây dựng mô hình được thực hiện với cơ chế phát hiện các thuộc tính đa cộng tuyến và cơ chế ổn định các trường hợp thoái hóa, giảm tình trạng quá tải thông bằng cách xử phạt các hệ số lớn.



Hình 3.3 Bộ phân lớp trong Weka Explorer

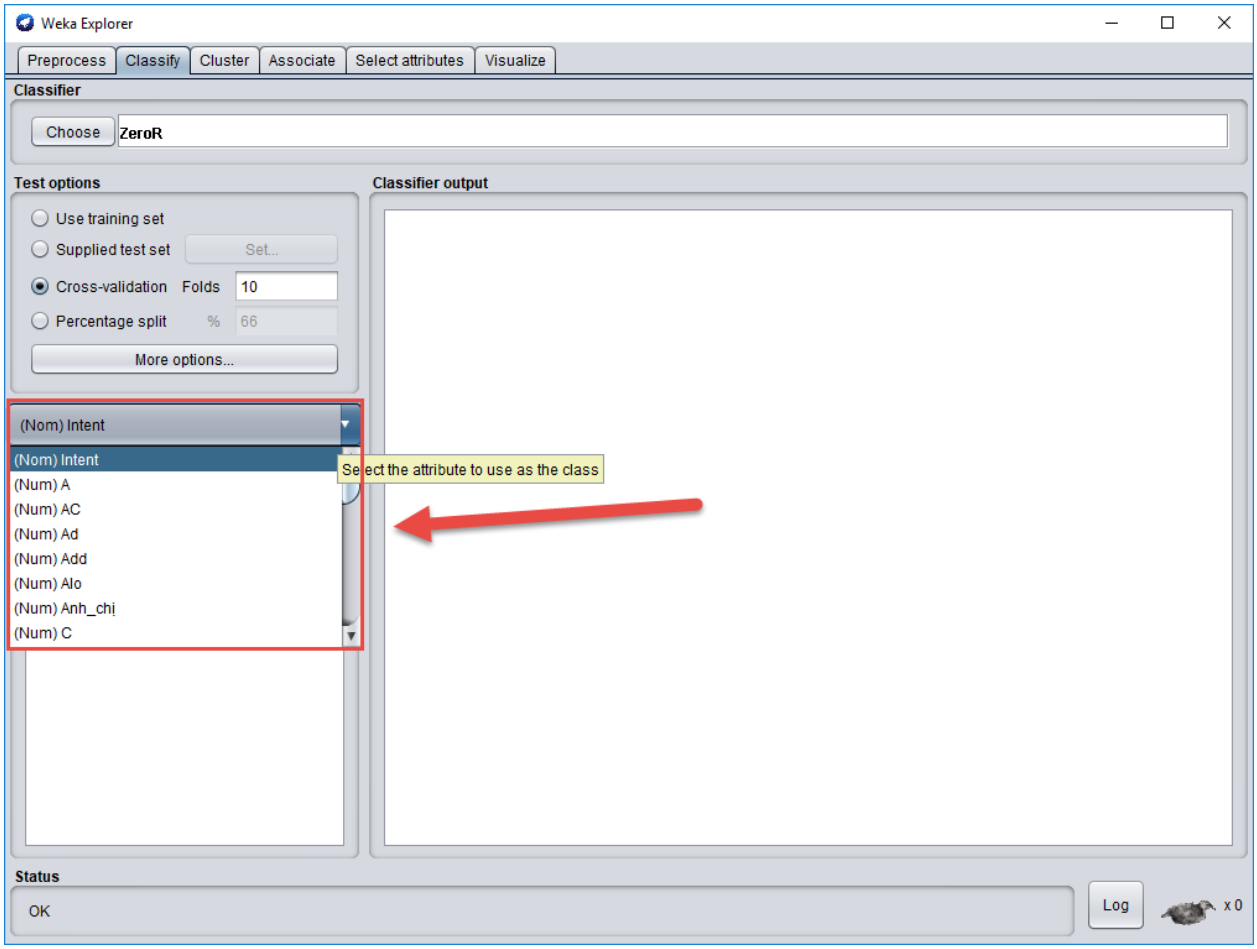
Bước thứ hai, cấu hình các tùy chọn kiểm thử (**Test options**): Tùy chọn phương pháp kiểm thử. Weka cung cấp 4 phương pháp như mô tả tại hình 3.4, gồm:

- **Use training set**: Sử dụng tập dữ liệu mà bộ phân loại đã được huấn luyện.
- **Supplied test set**: Cung cấp tập dữ liệu kiểm thử. Người sử dụng có thể lựa chọn tập dữ liệu kiểm thử bằng cách nháy vào nút "Set…"
- **Cross-validation**: Tiến hành xác nhận chéo.
- **Percentage split**: Chia tập dữ liệu thành 2 phần, huấn luyện trên 1 phần và kiểm thử trên phần còn lại. Phân chia tập dữ liệu theo tỷ lệ phần trăm do người sử dụng cài đặt.



Hình 3.4 Các tùy chọn kiểm thử của Weka

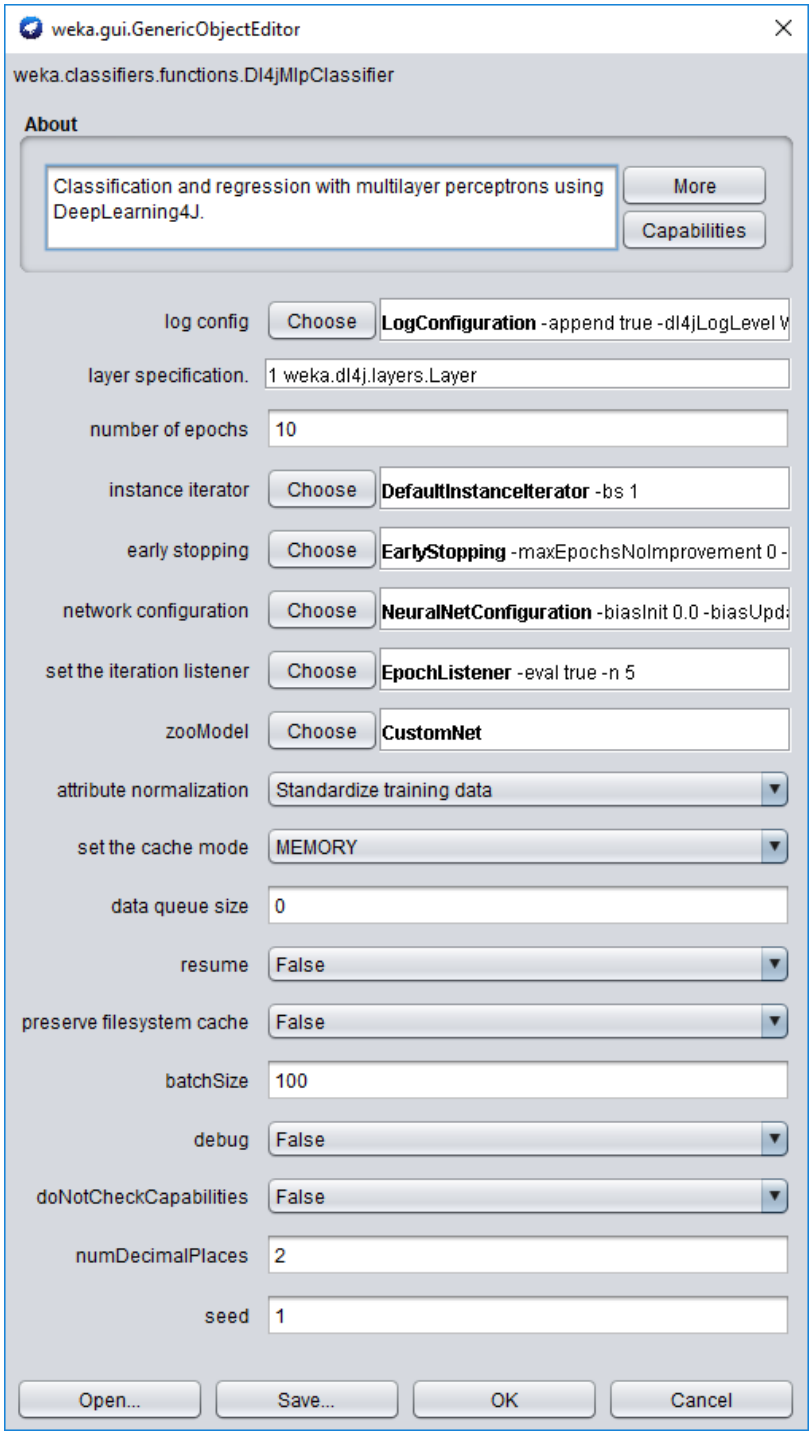
Bước cuối cùng là lựa chọn thuộc tính được dự đoán (biến phụ thuộc) được mô tả tại hình 3.5.



Hình 3.5 Lựa chọn thuộc tính dự đoán phụ thuộc

Giới thiệu về Package WekaDeeplearning4j

Gói ***WekaDeeplearning4j*** là một gói thư viện học sâu cho tích hợp với công cụ hỗ trợ Weka. Nó được phát triển để kết hợp các kỹ thuật hiện đại của việc học sâu vào Weka và được xây dựng trên ngôn ngữ Java.

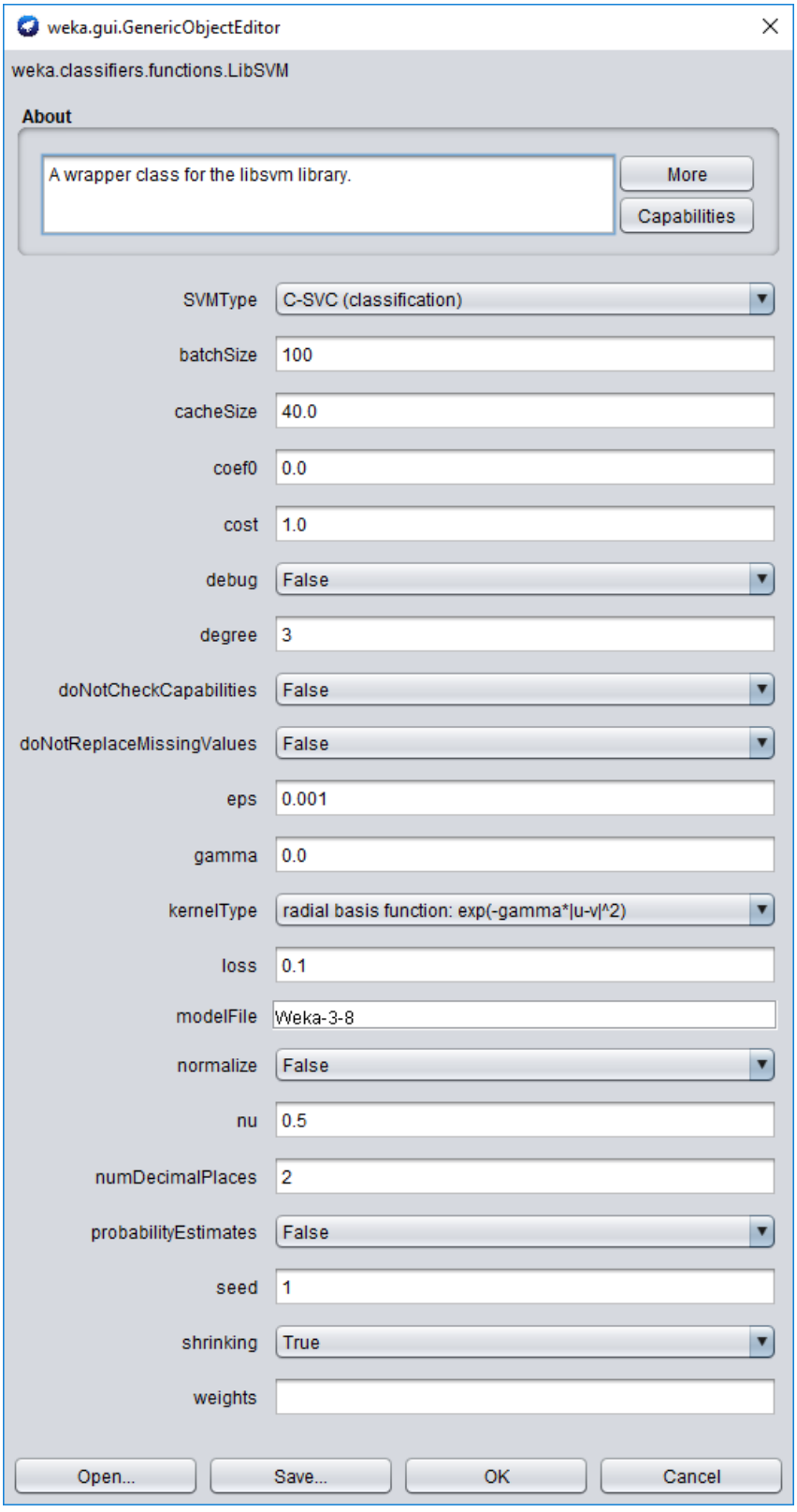


Hình 3.6 Giao diện WekaDl4j trên Weka GUI

Tất cả các chức năng của WekaDeeplearning4j đều có thể truy cập được thông qua giao diện của Weka hình 3.6 hoặc qua các dòng lệnh lập trình trong Java. Các lớp mạng nơron sau đây đều có sẵn trong gói WekaDeeplearning4j:

- **ConvolutionLayer**: áp dụng mạng nơron tích chập, hữu ích cho hình ảnh và nhúng văn bản.
- **DenseLayer**: tất cả các đơn vị được kết nối với tất cả các đơn vị của lớp cha của nó.
- **SubamplingLayer**: mẫu phụ từ các nhóm đơn vị của lớp cha theo các chiến lược khác nhau (trung bình, tối đa, v.v.)
- **LSTM**: sử dụng phương pháp tiếp cận bộ nhớ ngắn hạn
- **GlobalPoolingLayer**: áp dụng nhóm theo thời gian cho RNN và gộp nhóm cho CNN được áp dụng theo trình tự
- **OutputLayer**: Tạo đầu ra phân loại / hồi quy.

Giới thiệu về Package LibSVM



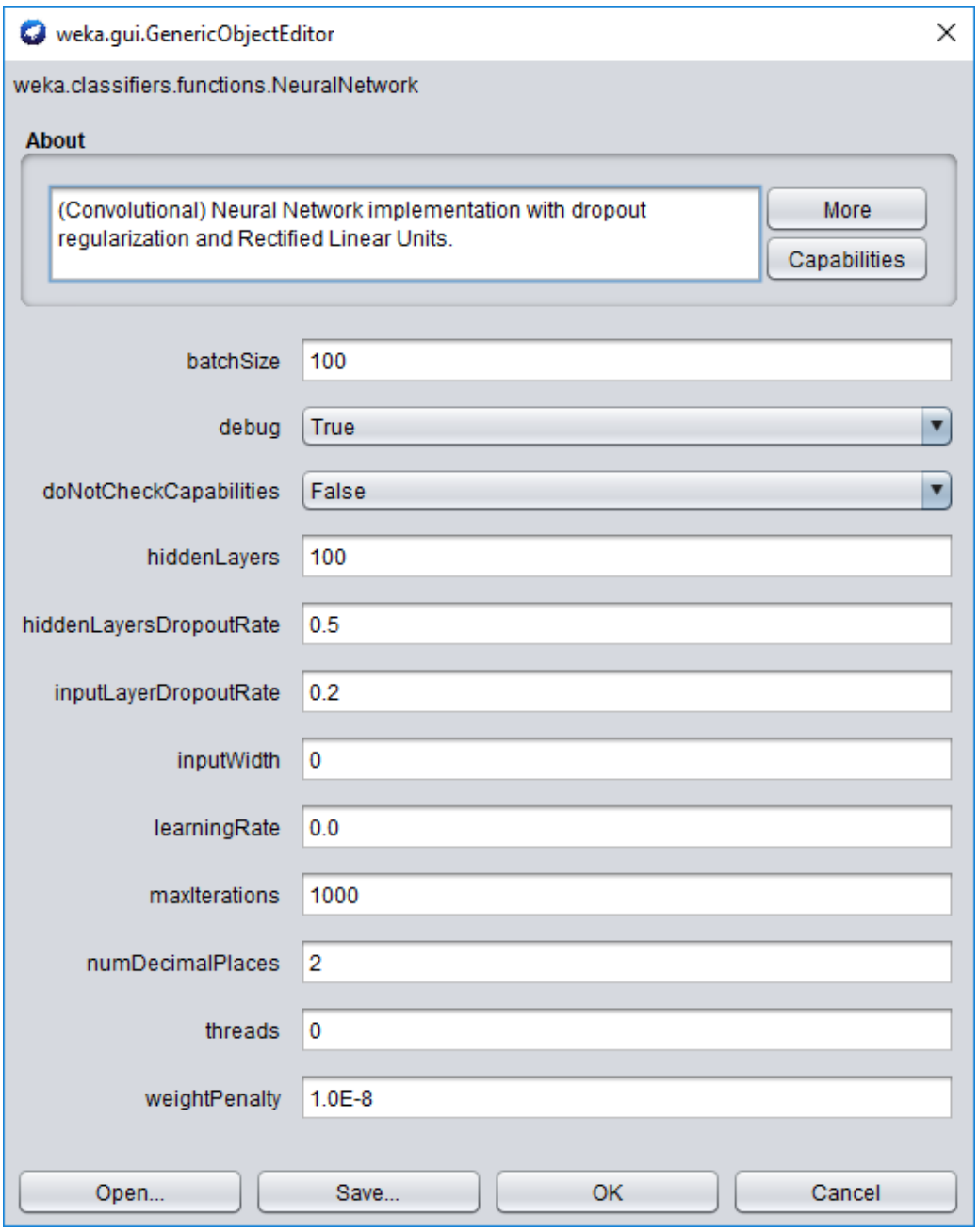
Hình 3.7 Giao diện LibSVM trên Weka GUI

Package LibSVM được mô tả tại hình 3.7 là một lớp wrapper cho các công cụ libsvm hỗ trợ và cho phép người dùng thử nhiệm với phương pháp One-class SVM, Regressing SVM và nu-SVM được hỗ trở bởi LibSVM. Để sử dụng được gói LibSVM, các thuộc tính dữ liệu cần được chuẩn hóa trước nếu thực hiện SVM hồi quy và bật chuẩn hóa để có kết quả tốt nhất.

Với package LibSVM, ta có thể thiết lập các thông số cơ bản sau:

- modelFile: vị trí lưu lại file mô hình sau khi training và test.
- kernelType: loại nhân mô hình sẽ sử dụng
- numDecimalPlaces: Số chữ số thập phân sử dụng trong đầu ra của mô hình.
- batchSize: số batch size
- cacheSize: Giá trị bộ nhớ đệm
- cost: Giá trị cost
- SVMType: Loại SVM sử dụng
- nu: giá trị cho nu-SVC, one-class SVM and nu-SVR.

Giới thiệu về Package Neural Network



Hình 3.8 Package Neural Network trên Weka GUI

Package Neural Network là một plugin cho Weka để sử dụng mạng nơron tích chập được phát triển bởi tác giả Amten. Ta có thể thiết lập các lớp trong mạng nơron qua tham số hidden layers, thiết lập các thông số về learning rate, iterations, batch size (Hình 3.8).

3.4 Kết quả thực nghiệm

3.4.1 Kết quả

LSTM	Unigrams			Bigrams			Trigrams			TF-IDF		
Acc (%)	85.14			72.47			54.58			85.04		
Độ đo / Ý định	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Thông tin về trường	84.1	74.0	78.7	84.1	64.0	67.1	70.6	24.0	35.8	82.8	74.0	78.2
Thông tin liên lạc	83.7	79.1	81.4	83.7	46.2	56.8	73.3	12.1	20.8	83.7	79.1	81.4
Thông tin về khoa	85.0	85.8	85.4	85.0	87.0	68.9	39.0	**81.5**	52.8	84.9	85.8	85.3
Cơ hội nghề nghiệp	71.8	76.7	74.2	71.8	34.2	43.9	50.0	9.6	16.1	71.4	75.3	73.3
Điều kiện tiếng Anh	88.4	90.5	89.4	88.4	61.9	69.8	**91.7**	26.2	40.7	88.4	90.5	89.4
Học phí	83.4	89.1	86.1	83.4	68.2	70.8	61.0	43.2	50.6	83.4	89.1	86.1
Điểm chuẩn	70.4	60.2	64.9	70.4	33.7	44.4	55.0	13.3	21.4	70.4	60.2	64.9
Nhập học	81.1	87.3	84.1	81.1	77.1	71.5	66.5	68.7	67.6	81.1	87.3	84.1
Thủ tục	89.8	93.4	91.6	89.8	85.3	84.9	48.8	69.7	57.4	89.8	93.4	91.6
Học bổng	94.3	91.0	92.6	94.3	82.3	83.1	81.9	57.3	67.4	94.3	91.0	92.6
Nghiên cứu khoa học	**96.6**	**94.3**	**95.4**	**96.6**	**87.3**	**88.7**	87.1	74.0	**80.0**	**96.6**	**94.3**	**95.4**
Tài liệu	82.2	86.0	84.1	82.2	54.7	63.1	91.4	37.2	52.9	82.0	84.9	83.4
Từ chối, không đồng ý	80.8	59.0	68.2	80.8	46.0	56.1	63.0	17.0	26.8	79.7	59.0	67.8
Đồng ý	79.4	81.0	80.2	79.4	43.0	56.6	48.0	12.0	19.2	78.6	81.0	79.8
Khác	40.0	44.7	42.2	40.0	5.9	9.9	16.7	2.4	4.1	40.2	43.5	41.8

Bảng 3.4 Kết quả mô hình LSTM

Dựa trên bảng kết quả 3.4 của mô hình **LSTM**, ta có thể thấy:

- Đặc trưng **Unigrams** và đặc trưng **TF-IDF** cho kết quả cao nhất, lần lượt là *85.14%* và *85.04%*
- Đặc trưng **Bigrams** và **Trigrams** có kết quả thấp hơn, lần lượt là *72.47%* và *54.58%*
- Chênh lệch độ chính xác accuracy giữa đặc trưng cao nhất (**Unigrams**) và thấp nhất (**Trigrams**) là *30.59%*
- Độ chính xác cao nhất tập trung vào ý định ***"Nghiên cứu khoa học"*** với độ đo trung bình điều hòa F1 là *95.4%* với đặc trưng **Unigrams** và **TD-IDF**

➔ Với mô hình **LSTM**, ta có thể thấy được kết quả khả quan nếu sử dụng đặc trưng **Unigrams** hay **TF-IDF**.

CNN	Unigrams			Bigrams			Trigrams			TF-IDF		
Acc (%)	**85.76**			82.37			72.79			81.23		
Độ đo / Ý định	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Thông tin về trường	68.9	81.3	74.6	73.5	76.0	74.8	88.5	92.7	90.6	67.6	78.0	72.4
Thông tin liên lạc	87.2	90.1	88.6	96.7	95.6	96.1	98.9	98.9	98.9	76.1	76.9	76.5
Thông tin về khoa	82.5	83.8	83.2	67.1	83.0	74.2	81.8	59.9	69.2	80.5	77.9	79.2
Cơ hội nghề nghiệp	88.9	76.7	82.4	90.3	89.0	89.7	93.2	93.2	93.2	74.6	72.6	73.6
Điều kiện tiếng Anh	93.1	96.4	94.7	**98.8**	**96.4**	**97.6**	**100**	**100**	**100**	**91.6**	90.5	91.0
Học phí	91.0	89.6	90.3	78.5	79.7	79.1	81.8	56.3	66.7	85.1	86.5	85.8
Điểm chuẩn	85.1	75.9	80.3	89.3	80.7	84.8	82.8	92.8	87.5	64.3	75.9	69.6
Nhập học	85.5	90.2	87.8	77.3	72.0	74.6	81.5	69.1	74.8	84.5	85.5	85.0
Thủ tục	92.1	90.8	91.5	86.8	84.1	85.4	50.7	80.7	62.3	90.7	89.0	89.8
Học bổng	94.0	**95.8**	94.9	92.2	81.5	86.6	53.4	66.2	59.1	90.9	92.1	91.5
Nghiên cứu khoa học	**96.9**	95.3	**96.1**	93	88.0	90.4	93.4	70.7	80.5	87.1	**97.0**	**91.8**
Tài liệu	92.9	90.7	91.8	95.3	94.2	94.7	98.8	95.3	97.0	71.0	76.7	73.7
Từ chối, không đồng ý	61.6	69.0	65.1	81.2	69.0	74.6	90.3	56.0	69.1	52.7	58.0	55.2
Đồng ý	61.4	62.0	61.7	88.4	76.0	81.7	83.6	61.0	70.5	59.0	46.0	51.7
Khác	38.6	20.0	74.6	87.5	82.4	84.8	93.3	82.4	87.5	35.1	15.3	21.3

Bảng 3.5 Kết quả mô hình CNN

Dựa trên bảng kết quả 3.5 của mô hình **CNN**, ta có thể thấy:

- Đặc trưng **Unigrams** và đặc trưng **Bigrams** cho kết quả cao nhất, lần lượt là *85.76%* và *82.37%*
- Đặc trưng **TD-IDF** và **Trigrams** có kết quả thấp hơn, lần lượt là *81.23%* và *72.79%*
- Chênh lệch độ chính xác accuracy giữa đặc trưng cao nhất (**Unigrams**) và thấp nhất (**Trigrams**) là *12.97%*
- Độ chính xác cao nhất tập trung vào ý định ***"Điều kiện tiếng Anh"*** với độ đo trung bình điều hòa F1 là *100%* với đặc trưng **Trigrams**

➔ Với mô hình **CNN**, ta có thể thấy được kết quả khả quan nếu sử dụng đặc trưng **Unigrams, Bigrams** hay **TF-IDF**.

So với mô hình **LSTM**, độ chênh lệnh chính xác accuracy giữa đặc trưng cao nhất và thấp nhất của CNN nhỏ hơn đáng kể, chỉ có *12.97%* so với *30.59%*.

SVM	Unigrams			Bigrams			Trigrams			TF-IDF		
Acc (%)	**88.89**			70.22			51.48			87.59		
Độ đo / Ý định	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Thông tin về trường	84.7	81.3	83.0	69.5	60.7	64.8	90.6	19.3	31.9	78.4	80.0	79.2
Thông tin liên lạc	96.3	86.8	91.3	94.1	35.2	51.2	**100**	8.8	16.2	97.3	80.2	88.0
Thông tin về khoa	86.2	91.2	88.6	46.6	**88.8**	61.1	29.9	**91.9**	45.1	87.9	89.5	88.7
Cơ hội nghề nghiệp	88.7	75.3	81.5	95.2	27.4	42.6	80.0	5.5	10.3	90.0	74.0	81.2
Điều kiện tiếng Anh	97.5	91.7	94.5	**97.0**	76.2	85.3	97.9	56.0	71.2	97.4	90.5	93.8
Học phí	92.3	93.2	92.7	82.8	57.8	68.1	71.1	33.3	45.4	91.3	92.7	92.0
Điểm chuẩn	92.9	78.3	85.0	83.3	24.1	37.4	83.3	6.0	11.2	92.5	74.7	82.7
Nhập học	87.1	88.7	87.9	74.8	71.3	73.0	80.5	55.6	65.8	85.2	90.2	87.6
Thủ tục	94.5	95.0	94.7	84.2	82.7	83.4	71.0	57.6	63.6	94.7	92.0	93.3
Học bổng	**98.4**	94.7	96.5	78.8	84.4	81.5	79.1	52.0	62.7	95.2	**93.9**	94.6
Nghiên cứu khoa học	97.3	**96.7**	**97.0**	96.5	82.7	**89.0**	92.9	70.0	**79.8**	98.0	96.3	97.1
Tài liệu	97.3	82.6	89.3	92.7	59.3	72.3	**100**	26.7	42.2	**98.8**	93.0	**95.8**
Từ chối, không đồng ý	54.6	89.0	67.7	76.3	45.0	56.6	78.9	15.0	25.2	47.2	91.0	62.1
Đồng ý	84.0	79.0	81.4	90.0	36.0	51.4	81.3	13.0	22.4	78.7	70.0	74.1
Khác	39.7	27.1	32.2	25.0	1.2	2.2	00.0	00.0	00.0	39.2	23.5	29.4

Bảng 3.6 Kết quả phương pháp SVM

Dựa trên bảng kết quả 3.5 của mô hình **SVM**, ta có thể thấy:

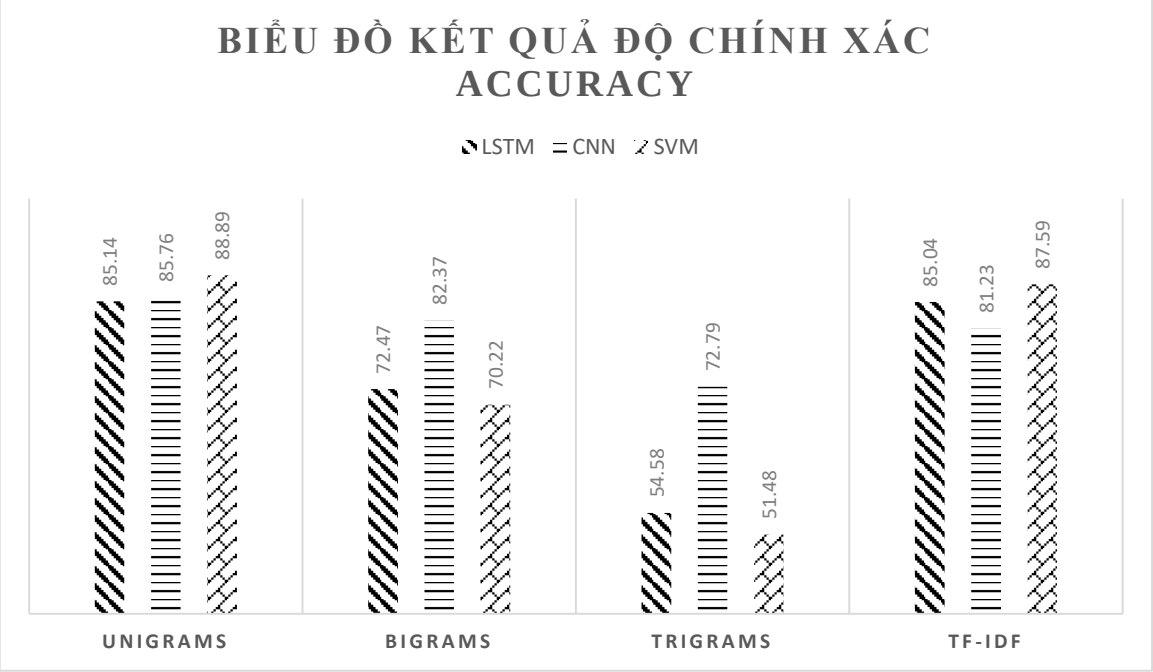
- Đặc trưng **Unigrams** và đặc trưng **TD-IDF** cho kết quả cao nhất, lần lượt là *88.89%* và *87.59%*
- Đặc trưng **TD-IDF** và **Trigrams** có kết quả thấp hơn, lần lượt là *70.22%* và *51.48%*
- Chênh lệch độ chính xác accuracy giữa đặc trưng cao nhất (**Unigrams**) và thấp nhất (**Trigrams**) là *37.41%*
- Độ chính xác cao nhất phân bố không đồng đều, không tập trung vào ý định nào. Độ trung bình điều hòa F1 cao nhất là *97%*, với ý định ***"Nghiên cứu khoa học"***

➔ Với phương pháp **SVM**, ta có thể thấy được kết quả khả quan nếu sử dụng đặc trưng **Unigrams** hay **TF-IDF**.

So với mô hình **LSTM**, **CNN** độ chênh lệnh chính xác accuracy giữa đặc trưng cao nhất và thấp nhất của phương pháp **SVM** cao nhất, lên tới *37.41%*.

3.4.2 Đánh giá kết quả

a. So sánh độ chính xác của các phương pháp trích chọn đặc trưng



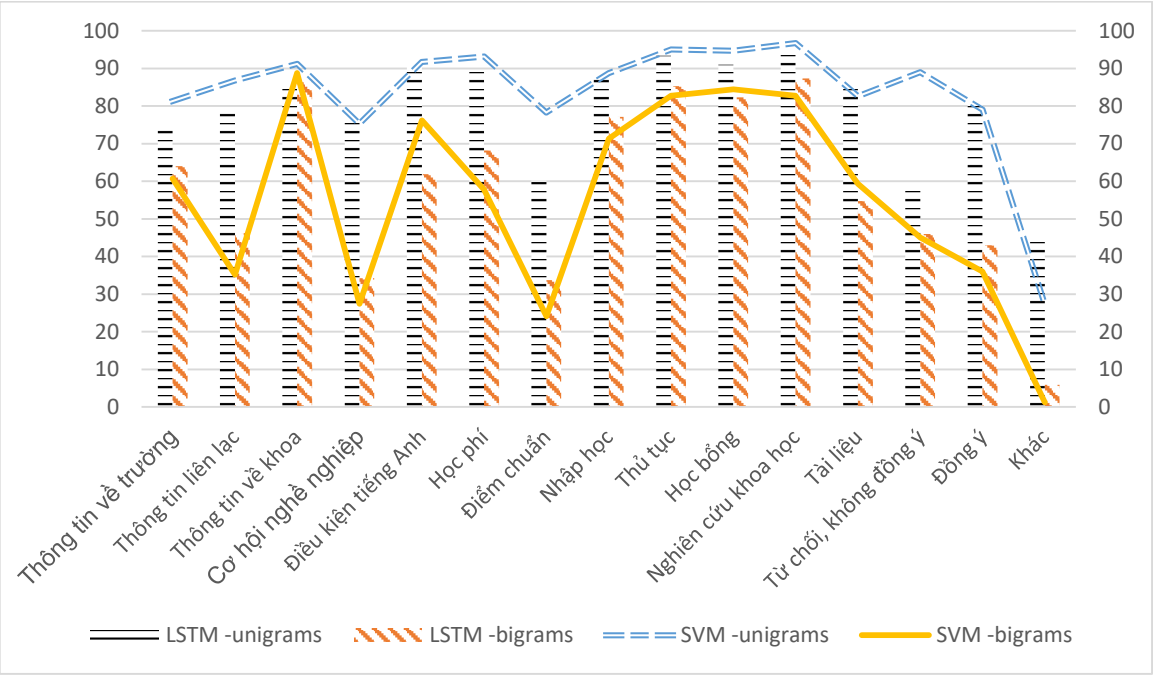
Hình 3.9 Biểu đồ so sánh kết quả accuracy của các mô hình với đặc trưng

Qua biểu đồ 3.9 về độ chính xác accuracy của các mô hình với các trích chọn đặc trưng của dữ liệu khác nhau ta có thể thấy:

- Đặc trưng **unigrams** cho kết quả trung bình độ chính xác cao nhất (*86.60%*)
- Phương pháp SVM cho độ chính xác cao nhất của **unigrams** và **TF-IDF** lần lượt là: *88.89%* và *87.59%.*
- Mô hình **CNN** cho kết quả độ chính xác cao nhất với đặc trưng **bigrams** và **trigrams**, lần lượt là: *82.37%* và *72.79%.*
- Với đặc trưng **trigrams**, mô hình **CNN** cho kết quả tốt nhất, hơn *18.21%* so với **LSTM** và *21.31%* so với **SVM**.
- Chênh lệch độ chính xác giữa các đặc trưng cao nhất là **SVM**: *37.41%.*
- Chênh lệch độ chính xác giữa các đặc trưng thấp nhất là **CNN**: *12.97%.*

Với mô hình **LSTM**, độ chính xác trung bình cao nhất tập trung vào phân lớp *"Nghiên cứu khoa học"* (Bảng 3.4). Mô hình CNN cho kết quả độ chính xác cao nhất tập trung vào phân lớp "Điều kiện tiếng Anh" (Bảng 3.5). Còn với phương pháp SVM, độ chính xác không đồng đều, không tập trung vào lớp nào (Bảng 3.6).

b. So sánh đặc trưng unigrams và bigrams LSTM và SVM



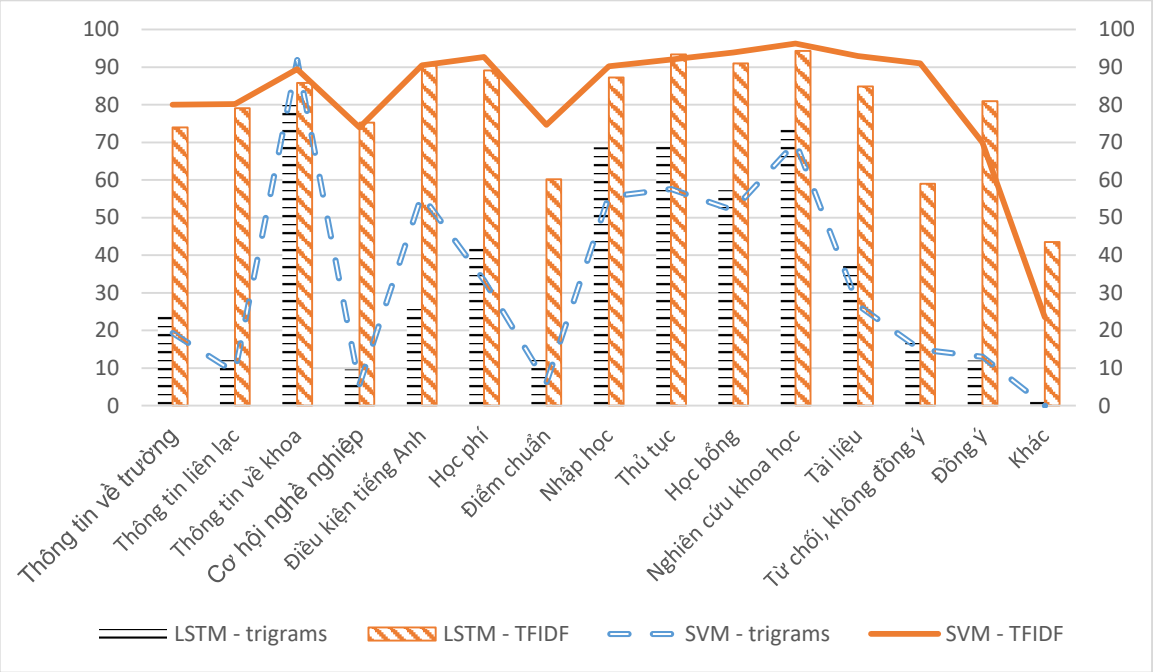
Hình 3.10 Biểu đồ đặc trưng unigrams và bigrams với mô hình LSTM và SVM

Qua biểu đồ về độ chính xác của 2 đặc trưng unigrams và bigrams với 2 mô hình học máy LSTM và SVM ta có thể thấy:

- Độ biến thiên của biểu đồ có hình dạng giống nhau, với các xu hướng đi lên, đi xuống theo các loại ý định là giống nhau;
- Độ chính xác cao nhất của mô hình LSTM là 94.3% (LSTM với đặc trưng unigrams – ý định "Nghiên cứu khoa học")
- Độ chính xác cao nhất của mô hình SVM là 96.7% (SVM với đặc trưng unigrams – ý định "Nghiên cứu khoa học")
- Độ chính xác thấp nhất của mô hình LSTM là 5.9% (LSTM với đặc trưng bigrams – ý định "Khác")
- Độ chính xác cao nhất của mô hình SVM là 1.2% (SVM với đặc trưng bigrams – ý định "Khác")
- Với đặc trưng unigrams, mô hình SVM cho kết quả trung bình độ chính xác lớn hơn với mô hình LSTM (88.88% > 85.14%)

- Với đặc trưng bigrams, mô hình SVM cho kết quả trung bình độ chính xác nhỏ hơn với mô hình LSTM (70.21% < 72.46%)

c. So sánh đặc trưng trigrams và tf-idf LSTM và SVM



Hình 3.11 Biểu đồ đặc trưng trigrams và tf-idf với mô hình LSTM và SVM

Qua biểu đồ về độ chính xác của 2 đặc trưng trigrams và tf-idf với 2 mô hình học máy LSTM và SVM ta có thể thấy:

- Độ biến thiên của biểu đồ có hình dạng không đồng đều, với các xu hướng đi lên, đi xuống theo các loại ý định là không giống nhau;
- Độ chính xác cao nhất của mô hình LSTM là 94.3% (LSTM với đặc trưng td-idf – ý định "Nghiên cứu khoa học")
- Độ chính xác cao nhất của mô hình SVM là 96.3% (SVM với đặc trưng td-idf – ý định "Nghiên cứu khoa học")
- Độ chính xác thấp nhất của mô hình LSTM là 2.4% (LSTM với đặc trưng trigrams – ý định "Khác")
- Độ chính xác cao nhất của mô hình SVM là 0% (SVM với đặc trưng trigrams – ý định "Khác")

- Với đặc trưng trigrams, mô hình SVM cho kết quả trung bình độ chính xác nhỏ hơn với mô hình LSTM (51.48% < 54.57 %)
- Với đặc trưng tf-idf, mô hình SVM cho kết quả trung bình độ chính xác lớn hơn với mô hình LSTM (87.58% > 85.04%)

Dựa trên các số liệu và hình dáng phía trên, để lựa chọn mô hình áp dụng cho đề tài phát hiện ý định người dùng trong hệ thống hỏi đáp của trường Đại học giữa 2 đặc trưng trigrams và tf-idf ta còn phụ thuộc vào yếu tố dữ liệu. Ví dụ như trường hợp ý định "Từ chối, không đồng ý" và ý định "Đồng ý":

- Với đặc trưng tf-idf, mô hình SVM cho kết quả độ chính xác lớn hơn với mô hình LSTM (91% > 59%) nhưng với đặc trưng trigrams thì ngược lại (15% < 17%)
- Với đặc trưng tf-idf, mô hình LSTM cho kết quả độ chính xác lớn hơn với mô hình SVM (81% > 70%) nhưng với đặc trưng trigrams thì ngược lại (12% < 13%)

Dựa trên các số liệu phía trên, để lựa chọn mô hình áp dụng cho đề tài phát hiện ý định người dùng trong hệ thống hỏi đáp của trường Đại học, đề xuất và đánh giá cao mô hình **CNN** hơn cả, vì đặc trưng của ngôn ngữ tiếng Việt khó phân tích, và thường được dùng bigrams để phân tích hình thái. Bên cạnh đó, các số liệu trung bình cũng như độ chênh lệch độ chính xác của mô hình **CNN** cho kết quả khả quan nhất.

3.5 Kết luận chương

Nội dung chương này trình quá trình thực nghiệm luận văn phát hiện ý định người dùng trong hệ thống hỏi đáp trên bộ dữ liệu thu tập được từ "*Kênh thông tin trực tuyến, Khoa Quốc tế, Đại học quốc gia Hà Nội"*. Dựa trên số liệu kết quả thực nghiệm ở chương này luận văn đưa ra phân tích đánh giá về phương pháp thực hiện. Các kết quả cho thấy việc sử dụng các đặc trưng khác nhau mang lại độ chính xác khác nhau. Sau khi quan sát bộ dữ liệu, có rất nhiều từ được viết theo văn phong riêng và sai chính tả (Ví dụ: "add" – ý hỏi admin, ad) hay viết tắt (Ví dụ: k thay cho không) dù đã loại bỏ stopwords. Đây thực sự là thách thức trong việc xây dựng hệ thống phát hiện ý định với ngôn ngữ tự nhiên, đặc biệt bằng tiếng Việt.

KẾT LUẬN

Nghiên cứu về xử lý ngôn ngữ tự nhiên nói chung, về bài toán phát hiện ý định người dùng nói riêng với là công nghệ mới, thời gian nghiên cứu còn ngắn nên vẫn còn nhiều vấn đề chưa thực sự nắm bắt tốt. Tuy nhiên qua quá trình nghiên cứu, luận văn đã tìm hiểu sâu về các giai đoạn từ tiền xử lý dữ liệu đến việc chọn các phương pháp biểu diễn đặc trưng của văn bản (N-grams, TF-IDF), phương pháp học máy để xây dựng mô hình phân lớp dữ liệu mạng nơron (kiến trúc LSTM và CNN trong luận văn đề xuất) và so sánh với phương pháp SVM.

Sử dụng mạng nơron nói chung hay mô hình LSTM và CNN nói riêng trong Deep Learning là một hướng đi có kỹ thuật và hiệu quả trong bài toán xử lý chuỗi và hiện đang được các nhà nghiên cứu sử dụng rất nhiều. Tuy nhiên, LSTM và CNN không phải là một kỹ thuật vạn năng mà cứ bài toán về NLP là lại áp dụng được. Nó còn căn cứ vào nhiều yếu tố như tập ngữ liệu, đặc tính của tập ngữ liệu. Vì đôi khi sử dụng một thuật toán SVM lại cho ra kết quả tốt hơn.

Trong tương lai, luận văn có thể được phát triển nghiên cứu các mô hình khác, thay đổi cấu trúc mạng norơn nhiều lớp hơn hoặc kết hợp các loại mạng nơron với nhau để nâng cao độ chính xác và cải thiện tốc độ xử lý đối với việc phát hiện ý định người dùng chính xác hơn. Luận văn cũng là tiền đề xây dựng hệ thống tư vấn, quảng cáo trong hệ thống hỏi đáp của trường Đại học phù hợp, với lượng người quan tâm cao và hỗ trợ nhanh chóng giải đáp đúng các vấn đề trong hệ thống hỏi đáp.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Ngo Xuan Bach, Tu Minh Phuong, "Leveraging User Ratings for Resource-Poor Sentiment Classification", In Proceedings of the 19th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES), Procedia Computer Science, pp. 322–331, 2015.

[2] Nguyen Thi Duyen, Ngo Xuan Bach, Tu Minh Phuong, "An Empirical Study on Sentiment Analysis for Vietnamese". In Proceedings of the International Conference on Advanced Technologies for Communications (ATC), Special session on Computational Science and Computational Intelligence (CSCI), pp. 309-314, 2014.

[3] Vũ Hữu Tiệp, Blog Machine Learning Cơ bản tại địa chỉ https://machinelearningcoban.com.

[4] Kim Đình Sơn, Đặng Ngọc Thuyên, Phùng Văn Chiến, Ngô Thành Đạt, Các mô hình ngôn ngữ N-gram và Ứng dụng, 2013.

[5] https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF, truy nhập ngày 18/10/2019.

Tiếng Anh

[6] Ahmed Husseini Orabi, Prasadith Buddhitha, Mahmoud Husseini Orabi, Diana Inkpen, "*Deep Learning for Depression Detection of Twitter Users*", 2018.

[7] Awais Athar, Simone Teufel, *"Detection of Implicit Citations for Sentiment Detection"*, 2012.

[8] B. Liu (2009), Handbook Chapter: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing, Handbook of Natural Language Processing. Marcel Dekker, Inc. New York, NY, USA.

[9] Bratman, Michael, "Intention, plans, and practical reason.", 1987.

[10] Google (2013), Word2vec model https://code.google.com/archive/p/word2vec/.

[11] Hochreiter and Schmidhuber (1997), Long short-term memory.

[12] Iryna Haponchyk, Antonio Uva1, Seunghak Yu, Olga Uryupina, Alessandro Moschitti, *"Supervised Clustering of Questions into Intents for Dialog System Applications"*, 2018.

[13] Maria Karanasou, Christos Doulkeridis, Maria Halkidi, *"DsUniPi: An SVM-based Approach for Sentiment Analysis of Figurative Language on Twitter"*, 2015.

[14] Peng Chen, Zhongqian Sun Lidong Bing, Wei Yang, *"Recurrent Attention Network on Memory for Aspect Sentiment Analysis"*, 2017.

[15] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, Bo Xu, *"Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling"*, 2016.

[16] Zheng Chen, Fan Lin, Huan Liu, Yin Liu, Wei-Ying Ma and Liu Wenyin, *"User Intention Modeling in Web Applications Using Data Mining"*, 2002.

[17] Zhiyuan Chen, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh, "*Identifying Intention Posts in Discussion Forums*", 2013.

[18] http://colah.github.io/posts/2015-08-Understanding-LSTMs/, truy nhập ngày 18/10/2019.

[19] https://d2l.ai/chapter_convolutional-neural-networks/lenet.html, truy nhập ngày 18/10/2019.

[20] http://karpathy.github.io/2015/05/21/rnn-effectiveness/, truy nhập ngày 18/10/2019.