

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

PHAN KHÔI NGUYỄN

KHÓA LUẬN TỐT NGHIỆP
NGHIÊN CỨU XÂY DỰNG CHATBOT TỰ ĐỘNG TRẢ LỜI
CÂU HỎI TRẮC NGHIỆM TIẾNG ANH DẠNG ĐIỀN KHUYẾT
MỘT CHỖ TRỐNG

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2017

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

PHAN KHÔI NGUYỄN – 13520564

KHÓA LUẬN TỐT NGHIỆP
NGHIÊN CỨU XÂY DỰNG CHATBOT TỰ ĐỘNG TRẢ LỜI
CÂU HỎI TRẮC NGHIỆM TIẾNG ANH DẠNG ĐIỀN KHUYẾT
MỘT CHỖ TRỐNG

CỬ NHÂNNGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN
NGUYỄN VĂN TOÀN

TP. HỒ CHÍ MINH, 2017

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số ngày
..... của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.

MỤC LỤC

Chương 1. GIỚI THIỆU	14
1.1. Nội dung đề tài	14
1.2. Phát biểu bài toán	16
1.3. Đối tượng và phạm vi nghiên cứu.....	16
Chương 2. TỔNG QUAN.....	17
2.1. Giới thiệu vấn đề.....	17
2.1.1. Các hướng tiếp cận.....	17
2.1.1.1. Hướng tiếp cận theo kiểm tra ngữ pháp.....	17
2.1.1.2. Mô hình n-gram	18
2.1.1.3. Mô hình cây phụ thuộc (Dependency models)	19
2.1.1.4. Continuous Space Models	19
2.1.1.5. PMI Model.....	20
2.1.1.6. Kiểm nghiệm độ chính xác giữa các hướng tiếp cận	20
Chương 3. MÔ HÌNH.....	23
3.1. Cơ sở lý thuyết.....	23
3.1.1. Ngôn ngữ học.....	23
3.1.2. Đặc trưng ngôn ngữ.....	23
3.1.3. Đặc điểm ngôn ngữ của tiếng Anh.....	24
3.1.4. Ngôn ngữ học thống kê	24
3.1.4.1. Mô hình N-gram.....	24
3.1.5. Ngôn ngữ học ngữ liệu	26
3.1.5.1. Tiền xử lý	26
3.1.5.2. Ngữ liệu	28

3.2. Mô hình đề xuất.....	29
3.2.1. Thu thập dữ liệu.....	30
3.2.2. Giải thuật	31
3.2.2.1. Tiền xử lý văn bản.....	31
3.2.2.2. Thuật giải chọn đáp án.....	32
Chương 4. CÀI ĐẶT – THỬ NGHIỆM	36
4.1. Cài đặt.....	36
4.1.1. Giới thiệu hệ thống Azure	36
4.1.1.1. Tổng quan.....	36
4.1.1.2. Giới thiệu về Azure Machine Learning	39
4.1.2. Hiện thực mô hình n-grams bằng Azure Machine Learning	41
4.1.2.1. Trích xuất dữ liệu.....	41
4.1.2.2. Hiện thực việc chọn đáp án của một câu	49
4.1.3. Xuất dữ liệu sang môi trường SQL.....	51
4.1.4. Câu hỏi thử nghiệm hệ thống	52
4.1.5. Hiện thực các hàm liên quan	53
4.1.6. Hiện thực hệ thống Web Service API và chatbot.....	55
4.1.6.1. Giới thiệu về Heroku.....	55
4.1.6.2. Hiện thực chat bot trên Facebook Messenger.....	56
4.2. Thử nghiệm.....	58
4.2.1. Kết quả khảo sát.....	58
4.2.2. Kiểm nghiệm tốc độ giải câu hỏi trên máy bàn nội bộ	59
4.3. Mô hình n - gram	60
4.4. Ngữ liệu	62

4.5. Hệ thống chương trình	66
Chương 5. KẾT LUẬN.....	68
5.1. Kết luận.....	68
5.2. Hướng phát triển.....	68

DANH MỤC HÌNH VẼ

Hình 3.1: Nội dung của ngữ liệu của statmt.org	30
Hình 3.2: Sơ đồ thể hiện quá trình chọn đáp án.....	34
Hình 4.1: Sơ đồ thể hiện quá trình làm việc với Azure Machine Learning Studio (hình ảnh được cung cấp bởi Microsoft).....	40
Hình 4.2: Sơ đồ thể hiện quá trình làm việc với dữ liệu nhỏ	42
Hình 4.3: Cài đặt cho các module để xử lý trên bộ ngữ liệu nhỏ	43
Hình 4.4: Kết quả đạt được sau quá trình tách dữ liệu	44
Hình 4.5: Gắn thêm module Execute Python Code vào sơ đồ.....	45
Hình 4.6: Kết quả sau khi gắn thêm module Python.....	45
Hình 4.7: Sơ đồ các module để trích xuất n-gram từ bộ ngữ liệu lớn.....	47
Hình 4.8: Sơ đồ các module để đưa dữ liệu vào database Azure SQL	48
Hình 4.9: Thử truy xuất lên database đã tạo trên Azure SQL.....	48
Hình 4.10: Thử truy xuất lên database đã tạo trên Azure SQL	48
Hình 4.11: Sơ đồ các module để thực hiện việc chọn đáp án	49
Hình 4.12: Kết quả sau khi tiền xử lý.....	50
Hình 4.13: Kết quả sau khi trích xuất các n-gram trong câu.....	50
Hình 4.14: Xuất dữ liệu thu được sang môi trường Azure SQL	52
Hình 4.15: Kết quả sau khi đưa toàn bộ dữ liệu lên Azure SQL	52
Hình 4.16: Hiện thực hàm tiền xử lý và trích xuất n-gram với các tùy chỉnh tương tự trước đó và xuất ra thành Web API.....	53
Hình 4.17: Chương trình Java sử dụng lần lượt các câu hỏi để truy vấn lấy câu trả lời từ hệ thống.....	54
Hình 4.18: Sơ đồ thiết kế hệ thống ứng dụng	55
Hình 4.19: Kết quả sau khi hiện thực chatbot.....	57
Hình 4.20: Chương trình kiểm nghiệm hiệu suất của hệ thống.....	59
Hình 4.21: Mô tả hệ thống chương trình	66

DANH MỤC BẢNG

Bảng 2.1: Bảng kiểm nghiệm độ chính xác giữa các mô hình thuật giải khác nhau để giải quyết bài toán hoàn thành câu dựa trên đề kiểm nghiệm MSR Sentence Completion Challenge. [3].....	21
Bảng 2.2: Kết quả giải đề thi TOEIC dựa trên bộ ngữ liệu n-gram của Google với các n-gram khác nhau. [4]	21
Bảng 2.3: Kết quả khảo sát dựa trên đề thi SAT của các thuật giải khác nhau. [5]	22
Bảng 3.1: Mẫu câu hỏi	32
Bảng 4.1: Bảng số lượng câu hỏi của các đề thi từ khóa luận [1]	36
Bảng 4.2: Kiểm nghiệm độ chính xác trên bộ ngữ liệu thu thập đến năm 2007	58
Bảng 4.3: Kiểm nghiệm độ chính xác trên bộ ngữ liệu thu thập đến năm 2014	59
Bảng 4.4: Bảng so sánh độ chính xác giữa các n-gram trong việc giải đề thi TOEIC dựa trên bộ n-gram của Google trong nghiên cứu [4]	61
Bảng 4.5: Bảng khảo sát độ chính xác khi kết hợp 3 và 4 gram trên bộ ngữ liệu năm 2014.....	62
Bảng 4.6: Kết quả khảo sát lần 1 với một nửa bộ ngữ liệu của khóa luận [1].....	64
Bảng 4.7: Kết quả khảo sát lần 2 với toàn bộ ngữ liệu của khóa luận [1]	64
Bảng 4.8: Kết quả khảo sát trên bộ ngữ liệu thu thập đến năm 2007 không tiền xử lý như đã đề ra và loại bỏ stopword	65
Bảng 4.9: Kết quả khảo sát trên sử dụng kết hợp 4, 3, 2 và 1-gram trên bộ ngữ liệu 2014	65

DANH MỤC TỪ VIẾT TẮT

TÓM TẮT KHÓA LUẬN

Đóng góp của khoá luận:

- Tìm hiểu tổng quan về tình hình nguyên cứu hệ thống này trong thời gian gần đây.
 - Chúng tôi đã tiến hành khảo sát hệ thống đã có trước nhằm nâng cao độ chính xác của mô hình và đã có nhiều cải tiến nâng độ chính xác lên :
 - Chúng tôi đã tìm hiểu và phát hiện hệ thống trước đã bỏ qua các thông tin hữu ích ở phần tiền xử lý nhằm nâng cao độ chính xác của hệ thống.
Chúng tôi đã cải tiến phần tiền xử lý: không bỏ các stopword có ý nghĩa mà hệ thống trước đã bỏ đi.
 - Chúng tôi phát hiện ra sự chưa tương thích giữa ngữ liệu học mà hệ thống trước đã dùng và các bộ test. Điều này đã góp phần làm cho độ chính xác của hệ thống trước chưa được như mong muốn.
 - Chúng tôi đã khảo sát trên cùng bộ test (4.000 câu) để chọn lại bộ ngữ liệu thích hợp hơn.
 - Dựa trên các phân tích của hệ thống trước, chúng tôi cũng nghi ngờ và tiếp tục thử nghiệm:
 - Khảo sát lại tính chính xác của mô hình 2,3,4-gram so với mô hình kết hợp 2-3 gram của khoá luận trước.
 - Khảo sát tính chính xác của mô hình ngôn ngữ của deep learning trên Azure
- Để đưa ra kết luận dùng mô hình 4-gram sẽ có độ chính xác cao hơn.
- Và cuối cùng hệ thống của chúng tôi đạt độ chính xác cao hơn hệ thống cũ là 12-16%.
- Chúng tôi đã tìm hiểu sử dụng và đề xuất dùng hệ thống Azure thích hợp cho phát triển khóa luận.

- Triển khai ứng dụng trên nền tảng ChatBot: thân thiện hơn với người sử dụng, không phụ thuộc vào thiết bị cuối, phù hợp với xu thế công nghệ hiện nay.

Chương 1. GIỚI THIỆU

1.1. Nội dung đề tài

Trong môi trường hội nhập quốc tế hiện nay, nhu cầu sử dụng tiếng Anh trở nên bức thiết. Từ đó các bài thi quốc tế ra đời nhằm mục đích đánh giá khả năng sử dụng tiếng Anh của một người nào đó. Các văn bằng chương trình tiếng Anh thực hành A, B, C hoặc các chứng chỉ tiếng Anh quốc tế TOEIC, TOEFL, ... có một đặc điểm chung là sử dụng bài tập trắc nghiệm ghi điểm làm đánh giá khả năng của người học. Hiện nay, việc tự học có được lợi thế cao từ mặt thời gian và độ linh động. Vì thế, việc thường xuyên rèn luyện trước các kỳ thi tiếng Anh là một trong các phương pháp nâng cao trình độ và bổ sung kiến thức. Tuy nhiên, khó khăn của việc tự học và làm bài tập là cần có được sự hướng dẫn trực tiếp. Ngày nay, đã có một số ứng dụng được đưa ra để người tự học có thể có được lời giải thích cho một câu hỏi thuộc các lĩnh vực khác nhau như: Toán, Vật Lý, Hóa Học, ... Tuy nhiên, chưa có công trình hay ứng dụng nào nhằm vào mục tiêu giải câu hỏi tiếng Anh. Một câu hỏi tiếng Anh, để có thể giải được cần một người có kinh nghiệm và hiểu biết để chọn đáp án chính xác và giải thích cho người học hiểu, đây là hạn chế. Vì thế khóa luận đưa ra ý tưởng sử dụng máy tính để giải và đưa ra câu trả lời gợi ý mang tính tham khảo cho người học tiếng Anh.

Đến thời điểm này có ít các nghiên cứu và giải pháp tập trung vào tự động hóa quá trình chọn câu trả lời gợi ý cho câu hỏi trắc nghiệm tiếng Anh. Việc ít các công trình nghiên cứu do độ khó trong việc giải quyết bài toán kết hợp tri thức ngữ nghĩa và suy luận. Hiện nay, đây vẫn đang là bài toán thách thức nếu sử dụng phương pháp mô hình ngữ nghĩa. Các mô hình đòi hỏi đánh giá ngữ nghĩa cả câu.

Đồng thời vào ngày nay, mạng xã hội ngày càng phổ biến và đang được quan tâm bởi rất nhiều người. Việc tiếp cận với mạng xã hội giúp cho ứng dụng dễ đạt được lượng người dùng lớn, với độ hiệu quả cao và dễ tiếp cận. Trong mạng xã hội, con người giao tiếp với nhau thông qua các mẫu đoạn văn bản nhỏ. Chatbot là

một công cụ mà ngày nay được sử dụng rộng rãi làm kênh giao tiếp giữa các cửa hàng, các công ty với người dùng bằng việc tự động trả lời tin nhắn bằng hệ thống.

Để hiện thực hóa ý tưởng, khóa luận chọn xây dựng hệ thống giải tự động câu hỏi trắc nghiệm tiếng Anh dạng điền khuyết. Khóa luận bao gồm:

- Xây dựng hệ thống giải bài tập tiếng Anh dạng điền khuyết dựa trên phương pháp xác suất thống kê
- Xây dựng n-grams từ bộ ngữ liệu tin tức ở statmt.org
- Xây dựng hệ thống phân tích ngữ liệu và xử lý câu hỏi, chọn đáp án trên hệ thống Azure Machine Learning
- Xây dựng Web Service API Server để phù hợp việc phát triển nhiều ứng dụng trên nhiều nền tảng, thiết bị khác nhau.
- Hiện thực chatbot để giao tiếp với người dùng giải câu hỏi tiếng Anh dạng điền khuyết

1.2. Phát biểu bài toán

Bài tập trắc nghiệm tiếng Anh có nhiều dạng khác nhau như:

- Bài tập điền khuyết
- Tìm lỗi sai trong câu
- Đọc hiểu văn bản chọn câu đúng nhất
- Chọn từ thích hợp cho đoạn văn
- Chọn từ có trọng âm khác với từ còn lại
- Chọn từ đồng nghĩa
- ...

Các bài tập trắc nghiệm tiếng Anh sẽ cho trước từ 3 đến 5 câu trả lời gợi ý bên dưới mỗi câu hỏi. Người làm bài tập sẽ chọn một đáp án làm đáp án chính xác cho bài tập đó.

1.3. Đối tượng và phạm vi nghiên cứu

Khóa luận giới hạn trong việc trả lời câu hỏi trắc nghiệm tiếng Anh dạng điều khuyết với một vị trí điền khuyết cùng với các lựa chọn phương án. Hệ thống sẽ xử lý và chọn một đáp án được cho là đúng.

Ví dụ:

Certain clear patterns in the metamorphosis of a butterfly indicate that the process is ____.

- (A) systematic
- (B) voluntary
- (C) spontaneous
- (D) experimental
- (E) clinical

Chương 2. TỔNG QUAN

2.1. Giới thiệu vấn đề

Có ít các công trình nghiên cứu tập trung trực tiếp vào việc giải quyết bài toán “trả lời câu hỏi tiếng Anh dạng điền khuyết”. Bài toán quy về một số hướng tiếp được đưa ra trước đây cho các vấn đề liên quan như: grammar check, sentence completion, ...

Tuy nhiên vẫn có một số nghiên cứu trực tiếp tập trung vào việc giải quyết bài toán.

2.1.1. Các hướng tiếp cận

2.1.1.1. Hướng tiếp cận theo kiểm tra ngữ pháp

Hướng tiếp cận này giải quyết bài toán “trả lời câu hỏi tiếng Anh dạng điền khuyết” dựa vào các mô hình được dùng để kiểm tra ngữ pháp và chính tả của câu. Ngữ pháp của một ngôn ngữ tự nhiên được biểu diễn bằng các cú pháp và hình thái từ. Do đó, kiểm tra ngữ pháp có thể hiểu là việc kiểm tra tính chính xác của cú pháp và hình thái đối với ngôn ngữ đang xét. Có nhiều phương pháp khác nhau để kiểm tra tính chính xác về ngữ pháp trên một đoạn văn bản. Từ các dữ liệu nhập vào, chương trình sẽ lần lượt thế các phương án vào chỗ trống, từ đó tìm ra phương án được cho là thích hợp nhất trả về cho người dùng. [1]

Các công trình sử dụng *grammar check* dựa trên ý tưởng chính là lần lượt thế các đáp án vào vị trí trống. Chọn ra đáp án có tần số xuất hiện cao nhất dựa trên ngữ liệu đã học được. Một sinh viên trước đây ở trường cũng đã có khóa luận tốt nghiệp về vấn đề này. Anh tiếp cận đề tài bằng cách quy bài toán về vấn đề grammar check và giải quyết bài toán bằng n-gram kết hợp gán nhãn chủ ngữ với chủ ngữ là ngôi 1, ngôi thứ 2 và ngôi thứ 3. Việc gom nhóm chủ ngữ này giúp tăng tần số xuất hiện của các trường hợp tương đồng, giảm bớt sự phân tán tần số xuất hiện không đáng có cho các chủ ngữ khác nhau nhưng cùng ngữ pháp chia động từ. Ở bước kiểm tra so sánh để tìm ra đáp án ta cũng thực hiện việc

gom nhóm chủ ngữ tương tự, nhờ đó với n-grams rơi vào các trường hợp chung sẽ cho ra kết quả chính xác hơn. [1]

Ở một số nghiên cứu khác, bài toán *grammar check* cũng được giải quyết bằng mô hình n-gram và xác suất thống kê như “Mô hình kiểm tra ngữ pháp và chính tả dựa trên xác suất”. Ý tưởng chính dựa trên xác suất, thu thập cái bi- tri- quad- và pentagram của một ngôn ngữ thông qua quá trình huấn luyện dữ liệu. Trong quá trình huấn luyện, thu thập xác suất của các n-gram. Sử dụng “*Word Class Agreements*” nhằm giải quyết 2 vấn đề đặc thù trong kiểm tra ngữ pháp và chính tả tiếng Anh là: *Adverb-verb-agreement* và *Adjective-noun-agreement* bằng cách lưu trữ song song các từ thường đi chung với nhau. Ví dụ: trạng từ “yesterday” sẽ được lưu trữ chung với tag động từ “verb (past tense)”. [2]

2.1.1.2. Mô hình n-gram

Lợi thế trong việc sử dụng mô hình n-gram là khả năng tính toán được xác suất xuất hiện của một chuỗi *token*. Điều này dễ cho việc huấn luyện trên các bộ ngữ liệu không được dán nhãn. Tuy nhiên mô hình n-gram bị giới hạn do sử dụng nguồn dữ liệu đã thông qua huấn luyện dẫn đến đánh giá dựa trên những câu đã được huấn luyện, không thể phân tích những câu phức tạp, mang tính ngữ nghĩa cao do khoảng cách lớn giữa các token trong câu. [3]

Một công trình nghiên cứu khác sử dụng trực tiếp ngữ liệu n-gram của Google để chọn đáp án đúng trong câu hỏi multiple question tiếng Anh. Nghiên cứu chọn câu trả lời gợi ý thông qua việc tách các n-gram xung quanh khoảng trống để tra khảo trong cơ sở dữ liệu n-gram và chọn kết quả nào có số lần xuất hiện cao nhất. Sử dụng đề thi TOEIC để làm dataset kiểm nghiệm hệ thống. Nhóm tác giả đề xuất sử dụng lần lượt quad-gram và tri-gram để giải quyết bài toán sau khi lần lượt sử dụng các n-gram khác nhau để tính toán độ chính xác của từng n-gram với bài toán cụ thể. [4]

2.1.1.3. Mô hình cây phụ thuộc (*Dependency models*)

Dependency models giải quyết được giới hạn của mô hình n-gram bằng cách biểu diễn mỗi từ bằng 1 node trong cây phụ thuộc. Mô hình *cây phụ thuộc không dán nhãn* coi mỗi từ là mỗi từ là một từ độc lập một cách có điều kiện so với những từ phía trước, được xử lý độc lập với mỗi quan hệ ngữ nghĩa.

Để giải quyết việc tính toán giá trị của câu, 2 câu khác nhau về trật tự giữa động từ và đối số của nó, mô hình *labeled dependency language* coi mỗi từ độc lập một cách có điều kiện và được gán nhãn bên ngoài.

Ưu điểm là đưa ra được hiệu suất cao hơn so với mô hình n-gram, lợi thế của cách biểu diễn nằm bao gồm việc huấn luyện và ước tính dễ dàng cũng như khả năng tận dụng phương pháp làm mịn chuẩn (standard smoothing methods). Tuy nhiên, kết quả của mô hình phụ thuộc vào phương pháp *automatic dependency extraction* và sự thừa thớt trong dữ liệu được thu thập. [3]

2.1.1.4. Continuous Space Models

Mạng neural giảm thiểu vấn đề thừa thớt dữ liệu bằng cách học các biểu diễn phân tán của các từ, chứng minh mô hình nổi trội trong việc bảo tồn những qui luật tuyến tính giữa các token. Mặc dù nhược điểm bao gồm độ mờ, xu hướng *overfitting*, và tăng yêu cầu tính toán. *Neural language models* đã vượt trội hơn mô hình n-gram và *dependency models*.

Mô hình kiến trúc Log-linear đã được đề xuất để giải quyết chi phí tính toán cho mô hình mạng neural. Mô hình *continuous bag-of-words* cố gắng đoán từ hiện tại bằng cách sử dụng n từ trong tương lai và n từ trong quá khứ làm ngữ cảnh. Ngược lại, *continuous skip-gram model* sử dụng từ hiện tại làm đầu vào để dự đoán những từ xung quanh. Sử dụng kiến trúc tổng thể bao gồm *skip-gram model* và mạng *neural*, đạt được hiệu suất cao trong *MSR Sentence Completion Challenge*.

2.1.1.5. PMI Model

Cách tiếp cận mô hình PMI dựa trên pointwise mutual information. Mô hình được thiết kế nhằm vào nguồn thông tin gần và xa để tính toán tổng thể sự gắn kết trong câu. PMI dựa trên lý thuyết đo đặc thông tin. PMI thể hiện sự tương quan giữa 2 từ i và j bằng cách so sánh xác suất của chúng dựa trên quan sát các từ trong cùng bối cảnh so với xác suất của việc quan sát các từ một cách độc lập.

Việc đầu tiên trong việc áp dụng mô hình PMI lên công việc hoàn thành câu đòi hỏi phải tạo được một word-context chứa các ma trận là các tần suất xuất hiện của từ trong bộ ngữ liệu. Các context phải độc lập và chứa các từ trong một câu nhằm mục tiêu để kiểm tra được độ liên quan giữa các từ ở cấp độ câu nhằm đạt được hiệu suất tối đa. Trong quá trình huấn luyện, các từ trước khi được đưa vào xử lý phải thông qua quá trình tiền xử lý bao gồm xóa stop-word, thay thế từ viết tắt, ... Các từ khi được đưa vào word-context cần phải được gắn tag để thể hiện rõ vai trò của từ trong câu. Tuy nhiên việc này làm giảm hiệu suất đáng kể và tăng yêu cầu rất lớn về mặt tài nguyên hệ thống, khiến cho tốc độ thực thi rất kém.

2.1.1.6. Kiểm nghiệm độ chính xác giữa các hướng tiếp cận

Ở nghiên cứu [3] có đưa ra kết quả độ chính xác trong việc hoàn thành câu (sentence completion) giữa các mô hình thuật toán với nhau. Bài kiểm tra dựa trên data set của Microsoft Research Sentence Completion Challenge - bộ tổng hợp 1040 câu chứa khoảng trống và có đáp án được rút trích từ tác phẩm Sherlock Holmes. Kết quả cho thấy mô hình PMI cho kết quả tốt hơn rất nhiều so với các mô hình tiền nhiệm trước đó.

Language Model	MSR
Random chance	20.00
N-gram [Zweig (2012b)]	39.00
Skip-gram [Mikolov (2013)]	48.00
LSA [Zweig (2012b)]	49.00
Labeled Dependency [Gubbins (2013)]	50.00
Dependency RNN [Mirowski (2015)]	53.50
RNNs [Mikolov (2013)]	55.40
Log-bilinear [Mnih (2013)]	55.50
Skip-gram + RNNs [Mikolov (2013)]	58.90
PMI	61.44

Bảng 2.1: Bảng kiểm nghiệm độ chính xác giữa các mô hình thuật giải khác nhau để giải quyết bài toán hoàn thành câu dựa trên đề kiểm nghiệm MSR Sentence Completion Challenge. [3]

Ở nghiên cứu [4], công trình nghiên cứu sử dụng bộ ngữ liệu n-gram của Google để chọn đáp án đúng trong câu hỏi multiple question tiếng Anh. Nghiên cứu chọn option thông qua việc tách các n-gram xung quanh khoảng trống (___) để tra khảo trong cơ sở dữ liệu n-gram và chọn kết quả nào có số lần xuất hiện cao nhất. Trong đó, tác giả sử dụng đề thi TOEIC để làm dataset kiểm nghiệm hệ thống. Kết quả cho thấy khi sử dụng lần lược quad-gram và tri-gram thì xác suất chính xác và hiệu suất tăng lên.

	Measurement	Vocabulary	Grammar	Total
5gram	Recall(%)	56.8	46.667	53
	Precision(%)	78.873	100	85.849
	F1-measure	66.041	63.636	65.538
4gram	Recall(%)	90.16	79.92	85
	Precision(%)	85.455	86.667	85.882
	F1-measure	87.746	881.504	85.438
Trigram	Recall(%)	100	98.611	99.5
	Precision(%)	75.781	85.915	79.397
	F1-measure	86.222	91.826	88.318
Trigram & 4gram	Recall(%)	100	97.436	99
	Precision(%)	83.607	86.842	84.848
	F1-measure	91.071	91.831	91.379

Bảng 2.2: Kết quả giải đề thi TOEIC dựa trên bộ ngữ liệu n-gram của Google với các n-gram khác nhau. [4]

Ở nghiên cứu [5], năm 2015 công trình khảo sát các thuật giải trong việc lựa chọn đáp án cho câu hỏi tiếng Anh dạng điền khuyết đề thi SAT. Công trình sử dụng nhiều thuật giải khác nhau bao gồm mô hình n-gram, mô hình PMI, mô hình Word2Vec, ... cùng với bộ ngữ liệu GloWbE để giải quyết bài toán. Kết quả cho thấy, sử dụng mô hình n-gram cho về kết quả có xác suất cao nhất.

Method	% Correct	% Incorrect by		% Incorrect by					Avg. Error Margin	Avg. Error Rank	Avg. Correctness Margin
		No. Blanks		Difficulty							
		1	2	1	2	3	4	5			
NPMI	30	77	67	76	61	83	57	72	53	3.15	28
Co-occ. Freq.	52	50	46	43	56	53	43	44	67	3.17	44
LSA	39	63	58	48	67	73	64	52	54	3.17	35
CBOW	48	47	58	48	33	73	50	44	55	3.09	36
CSKIP	48	45	60	48	44	57	57	52	45	2.93	26

Bảng 2.3: Kết quả khảo sát dựa trên đề thi SAT của các thuật giải khác nhau. [5]

Chương 3. MÔ HÌNH

3.1. Cơ sở lý thuyết

3.1.1. Ngôn ngữ học

Ngôn ngữ là gì ? Ngôn ngữ được định nghĩa như sau: *“là một hệ thống những đơn vị vật chất và những quy tắc hoạt động chúng, dùng làm công cụ giao tiếp của con người, được phản ánh trong ý thức cộng đồng và trừu tượng hóa khỏi bất kỳ một tư tưởng, cảm xúc và ước muốn cụ thể nào”*.

Ngôn ngữ có thể tồn tại dưới dạng văn viết và lời nói. Thành phần cấu tạo của một ngôn ngữ gồm nhiều tầng như: bài viết, đoạn văn, câu, từ, ... Trong đó, với câu được cấu tạo bởi nhiều từ, một đoạn văn hay bài viết được cấu tạo bởi nhiều câu. Cú pháp, ngữ nghĩa của một câu là cách kết hợp các từ, dưới một trật tự thứ tự và quy luật riêng biệt để tạo thành câu. Vậy, ta có thể định nghĩa ngữ pháp và ngữ nghĩa của một câu trong ngôn ngữ tự nhiên là một tập hợp các luật về cú pháp và sự biến đổi của các từ, trật tự của các từ trong ngôn ngữ để tạo thành một câu có nghĩa. Các ngôn ngữ khác nhau có cấu trúc ngữ pháp khác nhau.

Về bản chất, ngôn ngữ là một hiện tượng xã hội cũng là phương tiện giao tiếp quan trọng nhất của con người. Là tái hiện trực tiếp của tư tưởng của con người, là công cụ của tư duy.

3.1.2. Đặc trưng ngôn ngữ

Ước tính hiện nay có khoảng 5600 ngôn ngữ trên thế giới. Được phân bố không đồng đều, phụ thuộc vào sự phân biệt khá tùy ý giữa các ngôn ngữ chính và ngôn ngữ địa phương. Các ngôn ngữ khác nhau về cấu trúc, hình thái, ngữ pháp, quy luật. Độ phức tạp của ngôn ngữ phụ thuộc vào nhiều yếu tố như: sự biến đổi giữa các từ, độ mập mờ về ngữ nghĩa của một câu, ... Ví dụ như tiếng Việt phức tạp hơn tiếng Anh ở việc mập mờ ngữ nghĩa của câu trong khi tiếng Đức lại phức tạp hơn tiếng Anh ở việc cấu trúc của một câu và có nhiều lựa chọn để viết thành một câu có nghĩa.

Các ngôn ngữ khác nhau, các câu và từ có độ dài khác nhau. Ví dụ trong tiếng Đức, câu dài được sử dụng phổ biến trong khi tiếng Anh không phổ biến nhưng câu thường trên 3 chữ. Trong tiếng Hoa một câu viết rất dài nhưng có thể rút gọn lại thành một câu tiếng Anh rất ngắn. Chiều dài trung bình của các ngôn ngữ khác nhau cũng khác nhau.

3.1.3. Đặc điểm ngôn ngữ của tiếng Anh

Tiếng Anh là ngôn ngữ quốc tế hiện nay, được sử dụng rộng rãi khắp nơi trên toàn thế giới. Có nguồn gốc Ấn Âu. Tiếng Anh sử dụng chữ cái latin với 26 chữ khác nhau. Tiếng Anh được xếp vào loại hình ngôn ngữ hòa kết (*flexion*). Các từ có thể biến đổi hình thái để biểu diễn ngữ nghĩa khác nhau. Phương pháp biến đổi chủ yếu là thêm phụ tố (*affix*). Việc sử dụng phụ tố để cấu tạo nên từ mới là hiện tượng rất phổ biến, ví dụ như bicycle (bi-cycle), preprocessing (pre-processing), investment (invest-ment), Chính về thế, số lượng từ vựng trong tiếng Anh là rất lớn. Số lượng từ vựng trích xuất từ một tác phẩm văn học thông thường là khoảng hơn 3 triệu từ, riêng với bộ ngữ liệu của Google thống kê từ tất cả các sách điện tử cho thấy có đến 3 tỷ từ với 1 tỷ họ từ.

3.1.4. Ngôn ngữ học thống kê

Ngôn ngữ học thống kê nhằm vào mục tiêu giải quyết các vấn đề ứng dụng của máy tính lên ngôn ngữ như: dịch máy, tìm kiếm, phân tích ngữ nghĩa ... dựa trên các lý thuyết về xác suất.

3.1.4.1. Mô hình N-gram

Trong xử lý ngôn ngữ tự nhiên theo hướng tiếp cận xác suất thống kê, n-gram là mỗi chuỗi có n-token được tách từ một chuỗi lớn hơn. Không phân biệt là chữ, dấu câu hay là số. Ví dụ:

Từ một câu “This is a modern house” ta có thể tách thành các n-gram như sau:

- 1 – gram: “this”, “is”, “a”, “modern”, “house”
- 2 – gram: “this is”, “is a”, “a modern”, “modern house”

- 3 – gram: “this is a”, “is a modern”, “a modern house”
- 4 – gram: “this is a modern”, “is a modern house”
- ...

Kích thước của n-gram nằm trong khoảng từ 1 đến 5. Với mỗi giá trị n sẽ có tên gọi khác nhau, ví dụ:

- n = 1 gọi là unigram
- n = 2 gọi là bigrams
- n = 3 gọi là trigrams
- n = 4 gọi là tetragrams
- n = 5 gọi là pentagrams

Ta thấy các ứng dụng dịch ngôn ngữ như hiện nay ví dụ như Google Translate có thể phỏng đoán một câu sai chính tả như sau: “Bài viết tiếng Việc” được gợi ý sửa lại là “Bài viết tiếng Việt”. Các ứng dụng tương tự có rất nhiều, vậy dựa vào đâu để làm điều này ?

Ứng dụng n-gram vào dữ liệu lớn có độ chính xác cao, ta có thể lấy được các từ, các cụm từ thông dụng trong văn bản con người. Điều này phục vụ được bài toán nêu trên.

- Lấy ví dụ: cụm “Trường Đại Học Công Nghệ Thông Tin” ta thấy cụm “Đại Học” và cụm “Công Nghệ Thông Tin” là phổ biến.
- Tương tự như ví dụ ở trên, cụm từ “tiếng Việt” thông dụng hơn cụm từ “tiếng Việc” nên nó được gợi ý chỉnh sửa câu.

Công thức tính toán xác suất của một câu là:

$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$

Với: W: câu ta đang xét

$w_1, w_2, w_3, \dots, w_n$: các chữ thành lập nên câu

Ví dụ:

$$P(\text{"Công Nghệ Thông Tin"}) = P(\text{"Công"}, \text{"Nghệ"}, \text{"Thông"}, \text{"Tin"})$$

Lưu ý:

$P(A, B) = P(A) * P(B | A)$ – điều này xảy ra khi A diễn ra trước khi B diễn ra.

$$\Rightarrow P(W) = P(w_1, w_2, w_3, \dots, w_n) = P(w_1) \times P(w_2 | w_1) \times P(w_3 | w_1, w_2) \times \dots \times P(w_n | w_1, w_2, w_3, \dots, w_{n-1})$$

Ví dụ:

- $P(\text{"Đại Học"}) = P(\text{"Đại"}) \times P(\text{"Học"} | \text{"Đại"})$
- $P(\text{"Công Nghệ Thông Tin"})$
 $= P(\text{"Công"}) \times P(\text{"Nghệ Thông Tin"} | \text{"Công"})$
 $= P(\text{"Công"}) \times P(\text{"Nghệ"}) \times P(\text{"Thông Tin"} | \text{"Công Nghệ"})$
 $= P(\text{"Công"}) \times P(\text{"Nghệ"}) \times P(\text{"Thông"})$
 $\times P(\text{"Tin"} | \text{"Công Nghệ Thông"})$

Điều này có nghĩa là xác suất thành lập một cụm từ có n chữ phụ thuộc vào xác suất thành lập của cụm từ n-1 chữ đứng trước đó. Từ đó ta có thể quy ra được một cụm từ có xác suất chính xác là bao nhiêu và so sánh được sau khi phân tích từ dữ liệu lớn (text mining).

Dựa vào công thức Toán học, ta có công thức như sau dùng để so sánh xác suất giữa 2 chuỗi mà không phải thực hiện phép toán nhân số nhỏ quá nhiều lần:

$$P_1 * P_2 * P_3 * \dots * P_n \rightarrow \log P_1 + \log P_2 + \log P_3 + \dots + \log P_n$$

3.1.5. Ngôn ngữ học ngữ liệu

3.1.5.1. Tiền xử lý

Tiền xử lý là một quá trình quan trọng trong các bước của việc giải một bài toán. Mục tiêu chính của tiền xử lý là nhằm giảm bớt độ nhiễu của dữ liệu đầu vào hoặc tăng độ chính xác của dữ liệu đầu vào nhằm mục tiêu cải thiện độ chính xác của

hệ thống. Tiền xử lý thông thường có các công việc như: làm sạch văn bản, phát hiện câu, tách token, phân tích câu thành mệnh đề, ... Tùy thuộc bài toán cần giải quyết mà ta sẽ tiền xử lý văn bản ở các giai đoạn khác nhau và những công việc tiền xử lý khác nhau.

Token là một dãy tuần từ các ký tự có thể là chữ cái, số, dấu câu, khoảng cách, ... Quá trình tách token ra khỏi câu được thực hiện bởi bộ phận tách token. Nhiệm vụ của bộ phận tách token là chia đầu vào thành các token rời rạc. Quá trình tách token là một bước quan trọng nếu muốn tiếp cận xử lý ngôn ngữ tự nhiên theo hướng tiếp cận xác suất thống kê.

Cách để lấy token ra khỏi văn bản có thể khác nhau tùy biến theo ứng dụng muốn tiếp cận cũng như việc tách token là một quá trình phức tạp, dễ xuất hiện nhiều trường hợp đặc biệt khác nhau. Ví dụ: nếu ta xét sau mỗi dấu chấm là kết thúc câu thì lúc này, các chữ viết tắt trong chuỗi "T. A. Thomas" sẽ được xem như là 3 câu khác nhau hoặc "wouldn't" có thể sử dụng làm token hoặc thay thế từ viết tắt thành "would not" rồi mới xem nó là một token. Một trường hợp khác là việc xem một chuỗi số là một token hoặc loại bỏ luôn cả chuỗi số. Tùy vào độ phức tạp và tính ứng dụng mà tokenization mang lại cho ứng dụng.

Quá trình tách token là một công việc khó, trước khi tách token, thường văn bản sẽ trải qua một bước tiền xử lý để văn bản sẽ dễ hơn cho bộ phận tách token thực hiện. Quá trình tiền xử lý gồm rất nhiều bước và các bước khác nhau và cũng có những cách tiếp cận khác nhau. Các vấn đề trong tiền xử lý bao phải giải quyết trước khi tách token gồm những trường hợp phức tạp như: loại bỏ dấu câu hiệu quả, loại bỏ số, loại bỏ các ký tự đặc biệt, loại bỏ chuỗi đặc biệt như email hay tên miền, thay thế từ viết tắt, thay thế từ về dạng kinh điển, ... và vẫn giữ được hình thái, cấu trúc toàn vẹn của câu trước khi tách token.

Tiền xử lý và tách token là một quá trình phức tạp, hiện nay đã có nhiều công trình nghiên cứu tập trung giải quyết hai vấn đề trên, với độ phức tạp cao và độ

chính xác cao. Từ đó cũng xuất hiện nhiều công cụ hỗ trợ cho tiền xử lý và tách token được thực hiện dễ dàng hơn.

3.1.5.2. Ngữ liệu

Ngữ liệu, hay còn gọi là “*corpus*” là những dữ liệu của ngôn ngữ với những chứng cứ thực tế đã được số hóa, được lưu trữ có cấu trúc. Các ngữ liệu có thể là các dữ liệu bằng một ngôn ngữ (ngữ liệu đơn ngữ) hoặc nhiều thứ tiếng (ngữ liệu đa ngữ).

Ngữ liệu thường được chia hai dạng: ngữ liệu thô ngữ liệu có dán nhãn chú thích (*annoatited corpus*). Xây dựng ngữ liệu thô thường đơn giản và đã có sẵn. Ngữ liệu có dán nhãn chú thích sẽ tốn thêm thời gian và công sức để gán thêm thông tin.

Các ngôn ngữ khác nhau thường sẽ có các bộ ngữ liệu khác nhau. Có nhiều bộ ngữ liệu khác nhau cho nhiều loại ngôn ngữ. Tiếng Anh là một ngôn ngữ phổ biến, vì thế có nhiều bộ ngữ liệu tiếng Anh được cấp miễn phí và đã được tổ chức ACL thống kê lại đăng tải trên trang của của ACL [6].

Một số bộ ngữ liệu nổi tiếng ta có thể biết đến như:

- **Google n-grams** – bộ ngữ liệu này khác biệt so với các bộ ngữ liệu còn lại ở chỗ nó chỉ chứa danh sách các n-grams chứ không phải văn bản. Nguồn dữ liệu của Google n-grams được thu thập từ các trang web tiếng Anh, và được phân tích thành unigrams, bigram, . . . đến pentagrams. Mỗi n-grams đều có thông tin về tần số xuất hiện
- **WMT** – bộ ngữ liệu chứa nội dung là các tin tức được thu thập trên báo chí từ năm 2006 đến nay. Bộ ngữ liệu được sử dụng tập trung để giải quyết vấn đề dịch máy. Vì thế bộ ngữ liệu có đến 6 thứ tiếng như: Đức, Anh, Nga, Pháp, ... Đến nay đã được 12 phiên bản với các phiên bản khác nhau như News Crawl, Development Set, Europarl, ...

- **GloWbE** – được phát hành vào năm 2013 dựa trên 1.9 tỷ từ được lấy từ các văn bản tiếng Anh ở các nước khác nhau. Ngữ liệu chứa chủ yếu gồm các thông tin từ các trang blog, báo chí, tạp chí, các thông tin trên các trang web của các công ty, các tiểu thuyết, diễn văn, ... Nội dung là các văn bản tiếng Anh được lấy từ các nước sử dụng tiếng Anh thông dụng như: Mỹ, Anh, Úc, ... Bộ ngữ liệu ngoài dữ liệu thô còn có các đánh dấu *part of speech*, họ từ cho từng từ.
- **American National Corpus (ANC)** – hiện có hơn 20 triệu từ vựng tiếng Anh mà người Mỹ sử dụng và được quản lý bởi Linguistic Data Consortium và được giới thiệu vào những năm 1990. Dự án vẫn còn đang trong quá trình phát triển và dự kiến lúc kết thúc sẽ có hơn 100 triệu từ. Ngữ liệu bao gồm ngữ liệu thô, chú thích *part of speech*, họ từ, ... Hiện tại, bộ ngữ liệu được cung cấp miễn phí trên trang chủ của American National Corpus.
- **Brown Corpus** – là bộ ngữ liệu triệu từ đầu tiên của tiếng Anh. Brown Corpus được giới thiệu vào năm 1961 tại trường Đại học Brown. Bộ ngữ liệu này chứa khoảng 1 triệu từ được thu thập từ các ấn phẩm bằng tiếng Anh của Mỹ trong suốt năm 1961. Tính đến thời điểm hiện tại Brown Corpus đã có 6 phiên bản, với các dữ liệu từ hơn 500 nguồn khác nhau như: tin tức, biên tập, tôn giáo, ... Brown Corpus hiện tại miễn phí với bộ ngữ liệu thô. Với bộ ngữ liệu đã được chú thích có chứa hơn 80 *part of speech* là phiên bản đặc biệt, giới hạn.

3.2. Mô hình đề xuất

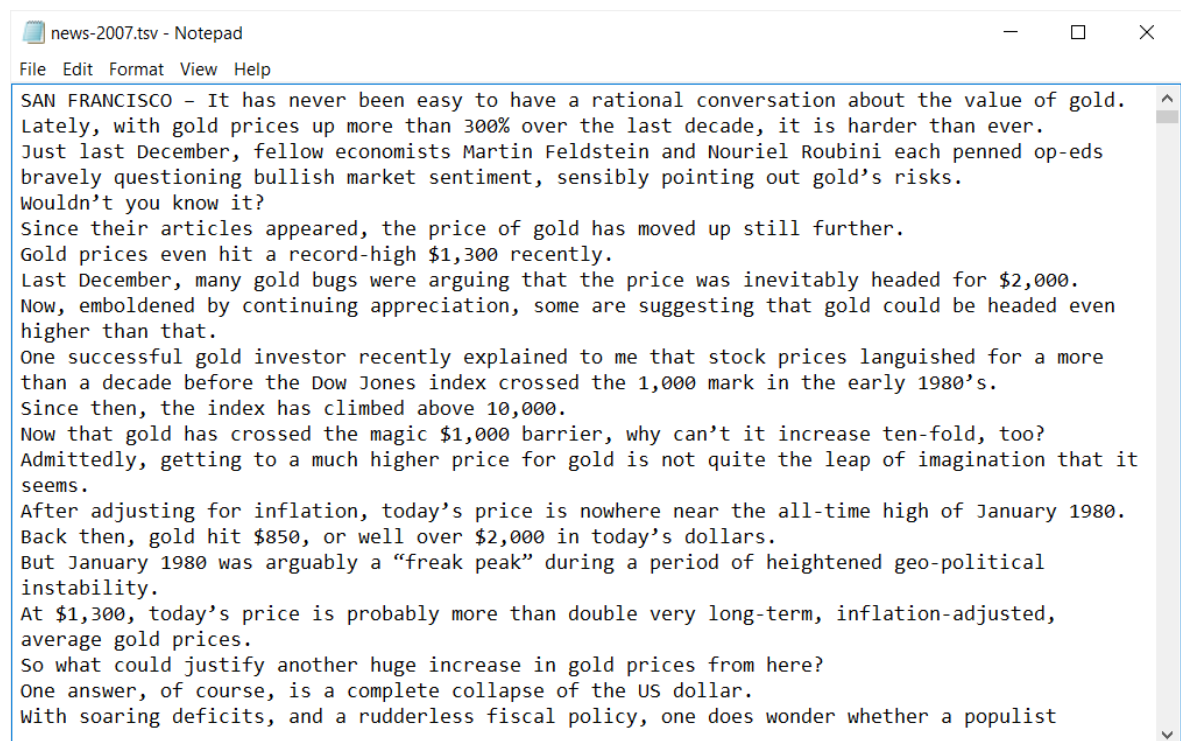
Hệ thống do khóa luận đề xuất dựa trên mô hình xác xuất thống kê. Theo đó, khi người dùng nhập vào một câu tiếng Anh có chỗ trống và các đáp án. Hệ thống có thể lấy đáp án, điền vào chỗ trống, trích xuất các n-grams liên quan và lấy được tần số xuất hiện của các n-grams này. Phương án nào cho được tổng logarit lớn nhất được xem là đáp án đúng.

Hệ thống cần một lượng lớn các dữ liệu text mẫu để huấn luyện và phải đảm bảo các dữ liệu mẫu này chứa các câu đúng, gần sát với nội dung thi TOEIC để đảm

bảo độ chính xác cao. Các đoạn văn bản này phải được trải qua các công đoạn xử lý như tách câu, tách token, đơn giản hóa từ, thay thế từ viết tắt, ... rồi sau đó trích xuất n-grams. Sản phẩm thu được từ các công đoạn trên sẽ được lưu trữ trên cơ sở dữ liệu, cùng với thông tin về số lần xuất hiện n-grams đang xét trong suốt quá trình huấn luyện.

Hệ thống chương trình được hiện thực và cung cấp Web API Service để phù hợp cho nhu cầu phát triển chatbot và các ứng dụng mở rộng sau này.

3.2.1. Thu thập dữ liệu



Hình 3.1: Nội dung của ngữ liệu của statmt.org

Bộ ngữ liệu khóa luận sử dụng là bộ ngữ liệu tin tức được thu thập từ năm 2014 đến nay của statmt.org. Được cung cấp miễn phí tại trang chủ của statmt.org [7].

Bộ ngữ liệu đơn giản là một tệp tin với các mẫu tin tức khác nhau. Các tin tức được phân thành hàng với mỗi hàng là một câu. Bộ ngữ liệu có nhiều thứ tiếng tuy nhiên khóa luận sử dụng bộ ngữ liệu tiếng Anh để phù hợp về yêu cầu của hệ thống.

Từ bộ ngữ liệu này, khóa luận tiến hành tiền xử lý như: đơn giản hóa từ, thay thế từ viết tắt, xóa các dữ liệu thừa sau đó thu thập các n-grams. Để dễ cài đặt và làm nhẹ hệ thống, ta sử dụng bộ ngữ liệu được thu thập đến năm 2007 (dung lượng 462MB – 3, 782, 549 dòng) sau đó sử dụng bộ ngữ liệu lớn hơn, được thu thập đến năm 2014 (4,1 GB) để triển khai ứng dụng.

3.2.2. Giải thuật

3.2.2.1. Tiền xử lý văn bản

Quá trình tiền xử lý văn bản để làm sạch và đơn giản hóa văn bản. Bằng việc tiền xử lý, ta sẽ có thể lấy được các n-grams có ý nghĩa hơn, giúp việc sử dụng n-grams để giải quyết vấn đề cho được hiệu quả cao hơn.

Trong tiền xử lý văn bản, ta có nhiều vấn đề như:

- Loại bỏ stop-words (Ví dụ: bỏ các từ: “the”, “a”, “about”, “all”, “didn’t”, ...)
- Đơn giản hóa định dạng từ về dạng kinh điển (Ví dụ: “them, their” thành “they”, “died” thành “die”, “fruits” thành “fruit”, ...)
- Thêm thành phần để phát hiện bắt đầu câu (Ví dụ: “I am a man” thành “<P> I am a man”)
- Loại bỏ dấu câu (Ví dụ: xóa các dấu “.”, “,”, “!”, ...)
- Loại bỏ các thành phần đặc biệt (Ví dụ: loại bỏ các email như “13520564@gm.uit.edu.vn” hoặc “example@host.com”. Loại bỏ số như: loại bỏ các số điện thoại “0121 2234 1909” hoặc “1990’s” thành “ ’s ”. Loại bỏ các đường dẫn đến địa chỉ website như: “https://www.google.com”)
- Thay thế từ viết tắt (Ví dụ: “wouldn’t” thành “would not”, “let’s” thành “let us”, “I’ve” thành “I have”)
- ...

Vì yếu tố khóa luận muốn giải quyết các bài toán về ngữ pháp và giải các bài tập liên quan đến tiếng Anh. Vì thế việc loại bỏ stop word sẽ khiến cho kết quả không được như mong muốn vì các thành phần ngữ pháp của tiếng Anh được cấu tạo

từ stop word rất nhiều. Việc loại bỏ stop word lại có thể khiến câu khó hiểu hơn và không đủ dữ liệu để tính toán xác suất của câu. Tương tự như việc đơn giản hóa các từ về dạng kinh điển cũng khiến không đủ dữ liệu để tính toán xác suất của một câu.

Vì thế, khóa luận đề xuất việc tiền xử lý văn bản gồm:

- Thêm thành phần để phát hiện bắt đầu câu.
- Loại bỏ dấu câu và các ký tự đặc biệt, thay thế bằng dấu khoảng cách
- Loại bỏ các thành phần đặc biệt.
- Thay thế từ viết tắt.

Việc tiền xử lý sẽ được thực hiện 2 lần, lần thứ nhất là tiền xử lý văn bản trước khi được vào bộ huấn luyện, trích xuất n-gram. Tiền xử lý thứ 2 nằm trong quá trình trả lời câu hỏi. Khi nhận được câu hỏi từ người dùng và thay thế các đáp án, hệ thống sẽ tiền xử lý các câu này trước khi truy vấn tìm xác suất của các token. Để đảm bảo tính đồng nhất, việc tiền xử lý trước quá trình trích xuất n-gram và tiền xử lý trước khi truy vấn câu hỏi phải được cài đặt giống nhau.

3.2.2.2. Thuật giải chọn đáp án

Hệ thống đề xuất mẫu cho câu hỏi trắc nghiệm để tự động trả lời câu hỏi trắc nghiệm tiếng Anh dạng điền khuyết như bên dưới.

I ____ with mom in 1980's.
was
be
am
been

Bảng 3.1: Mẫu câu hỏi

Thuật giải chọn đáp án gồm:

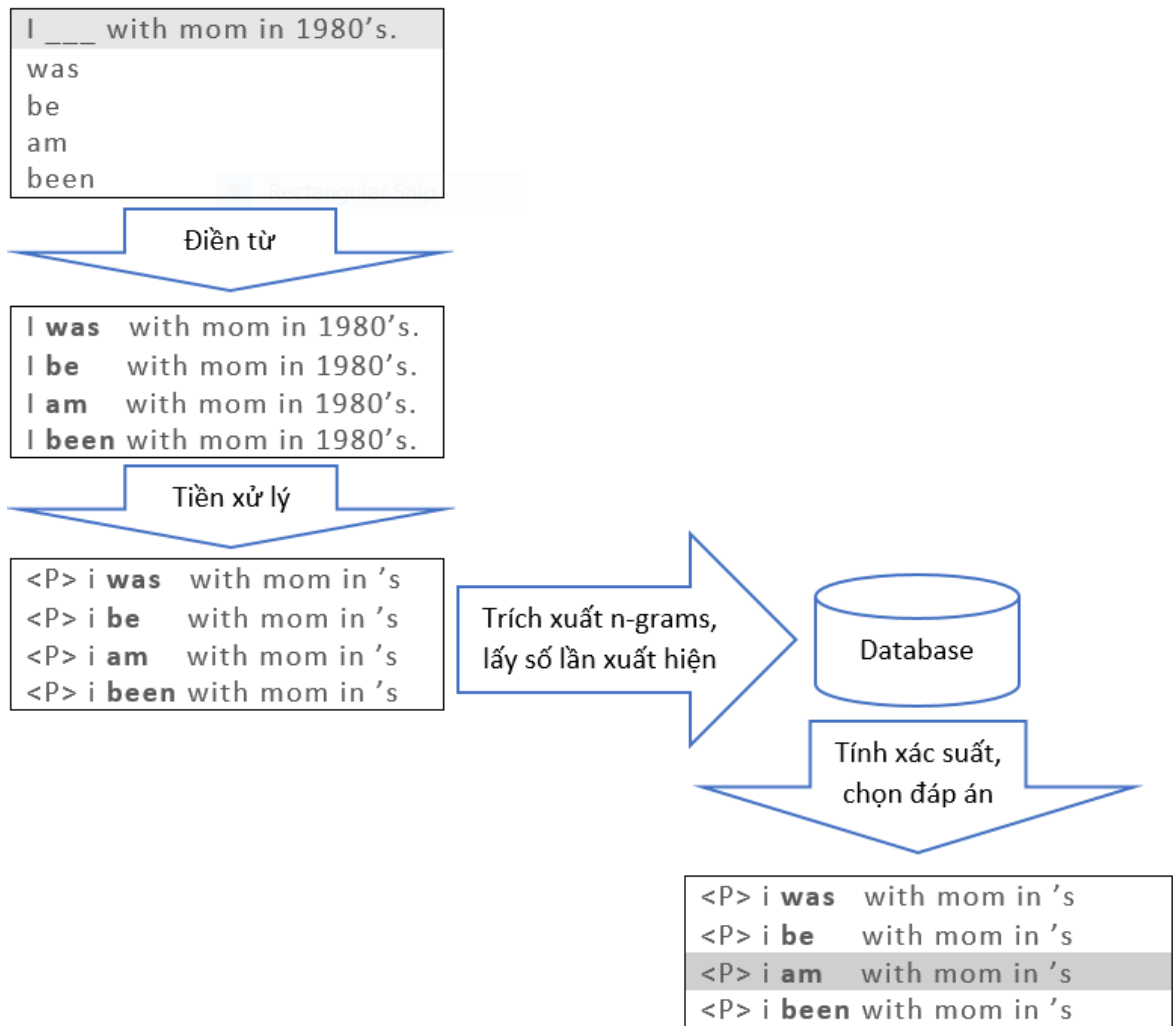
- **Tiền xử lý trước khi chọn đáp án:** trước khi chọn đáp án, ta lần lượt thay thế các câu trả lời gợi ý vào vị trí điền khuyết, sau đó thực hiện các công đoạn tiền xử lý tương tự như lúc thu thập ngữ liệu để đảm bảo nội dung của câu hỏi có định dạng tương đương với bộ ngữ liệu.

- **Kiểm tra tần số n-grams:** sau khi tiền xử lý, ta tách thành các n-grams và sử dụng để tính toán xác suất của một câu dựa trên công thức log và minimum add-one như bên dưới. Sau đó chọn câu trả lời có tần số xuất hiện cao nhất
- Công thức tính xác suất tồn tại của câu dựa trên log:

$$P(s_0, s_k) = \sum_{i=0}^k \log \left(\frac{t_i + 1}{\max(t_0 \dots t_k) + 1} \right)$$

Với:

- $P(s_0, s_k)$: xác suất tồn tại của một câu, chứa các token từ s_0 đến s_k
- t_i : số lần xuất hiện của token đó trong toàn bộ ngữ liệu đã huấn luyện trước đó
- $\max(t_0 \dots t_k)$: giá trị số lần xuất hiện lớn nhất của một trong toàn bộ ngữ liệu



Hình 3.2: Sơ đồ thể hiện quá trình chọn đáp án

Toàn bộ quá trình chọn đáp án có thể tóm gọn trong đoạn mã giả sau:

```
findAnswerQuest(sentence, options, isCover, n)

    sentencesAfterReplace = sentence.replace(options)

    allToken = []

    for i: 1 -> sentencesAfterReplace.length
        allToken.put(textPreprocess(sentencesAfterReplace[i]))

    allCount = getCount(allToken)

    percisions = []
    covers = []

    foreach i: 1 -> sentencesAfterReplace.lenght
```

```

        isCoverThisOption = true
        percisionOfThisOption = 0.0

        foreach token in sentencesAfterReplace[i]
            thisTokenCount = 1

            if (allCount.contains(token))
                thisTokenCount = allCount[token] + 1
            else
                isCoverThisOption = false

            percisionOfThisOption += Math.log(thisTokenCount / BigCount +

1)

        percisions.push(percisionOfThisOption)
        covers.push(isCoverThisOption)

    isCover = covers[percisions.maxIndex]
    return options[percisions.maxIndex]

```

Với:

- Hàm **findAnswerQuest** tìm câu trả lời cho câu hỏi dựa trên n-gram
- Hàm **textPreprocess** tiền xử lý và trả về danh sách token của câu đó
- Hàm **getCount** trả về danh sách số đôi với token và lần xuất hiện của token đó
- **sentence** câu hỏi có chứa dấu khoảng cách
- **options** danh sách các câu gợi ý
- **n** số gram trong một token được sử dụng để tính toán
- **isCover** trả về câu có được bao phủ hoàn toàn bởi n-gram hay không
- **BigCount** giá trị số lần xuất hiện lớn nhất của một token trong toàn bộ ngữ liệu

Sau đoạn mã giả này, ta thu được câu trả lời được cho là chính xác nhất cùng với việc kiểm tra toàn bộ câu này có được bao phủ bởi các gram của hệ thống hay không.

Chương 4. CÀI ĐẶT – THỬ NGHIỆM

Khóa luận tiến hành khảo sát độ chính xác của các bộ 1, 2, 3 và 4 grams để tìm ra phương pháp số n-grams cho về độ chính xác cao. Để tiến hành khảo sát, khóa luận sử dụng lại bộ câu hỏi của đề tài trước bao gồm các đề thi bằng A, bằng B, bằng C và đề thi TOEFL.

Đề thi	Số lượng câu hỏi
Bằng A	800
Bằng B	460
Bằng C	2020
TOEFL	820

Bảng 4.1: Bảng số lượng câu hỏi của các đề thi từ khóa luận [1]

Nhằm kiểm nghiệm độ ảnh hưởng của độ lớn corpus lên độ chính xác, khóa luận chia ra làm 2 lần kiểm nghiệm. Lần kiểm nghiệm thứ nhất với bộ ngữ liệu tin tức đến năm 2007 (462MB). Lần kiểm nghiệm thứ hai với bộ ngữ liệu tin tức dựa thu thập đến năm 2014 (4.1G).

Sau khi tìm được khảo sát được độ chính xác giữa các bộ n-gram, khóa luận tiến hành hiện thực các ứng dụng liên quan cũng như cài đặt server để các ứng dụng có thể chạy được như ý muốn.

4.1. Cài đặt

4.1.1. Giới thiệu hệ thống Azure

Khóa luận sử dụng hệ thống Azure được cung cấp bởi tập đoàn Microsoft để hiện thực khóa luận.

4.1.1.1. Tổng quan

Microsoft Azure là một dịch vụ điện toán đám mây (*cloud computing*) được tạo ra và cung cấp bởi tập đoàn Microsoft. Hệ thống nhằm vào việc xây dựng, thử nghiệm, triển khai và quản lý các dịch vụ công nghệ thông tin thông qua một trung tâm được điều hành bởi Microsoft. Hệ thống hỗ trợ nhiều ngôn ngữ lập

trình khác nhau, các framework, máy ảo và các module hệ thống khác nhau để phù hợp cho sản phẩm cần được phát triển. Hiện tại, những dịch vụ tiêu biểu mà Microsoft Azure có cung cấp như sau:

- Web service
- SQL Database
- Máy ảo
- Azure Machine Learning
- ...

Dịch vụ tính toán Microsoft Azure có thể chạy nhiều kiểu ứng dụng khác nhau. Mục tiêu chính của kiến trúc này, là hỗ trợ các ứng dụng có lượng người sử dụng truy cập đồng thời cực lớn. Microsoft Azure được thiết kế để hỗ trợ ứng dụng tốt nhất, chạy nhiều bản sao của cùng một mã nguồn trên nhiều máy chủ khác nhau. Ứng dụng Microsoft Azure có thể có nhiều thực thể, thực thể được thực thi trên một máy ảo.

Để chạy một ứng dụng, lập trình viên truy cập Microsoft Azure portal thông qua trình duyệt, sử dụng tài khoản Windows Live ID đăng nhập. Từ đó, Lập trình viên có thể upload ứng dụng của mình hoặc sử dụng các ứng dụng, module có sẵn bên trong hệ thống. Lập trình viên, có thể thấy được trạng thái của ứng dụng đã được triển khai, thông qua Microsoft Azure portal. Một khi ứng dụng được triển khai, nó hoàn toàn được quản lý bởi Microsoft Azure. Các thông số sử dụng cho ứng dụng, còn lại, việc triển khai, tính mở rộng, tính sẵn sàng, nâng cấp, chuẩn bị phần cứng server đều được thực hiện bởi Microsoft Azure cho các ứng dụng đám mây.

Cơ sở dữ liệu SQL Azure cung cấp một hệ thống quản lý cơ sở dữ liệu dựa trên đám mây. Công nghệ này cho phép ứng dụng và đám mây lưu trữ dữ liệu quan hệ và những kiểu dữ liệu khác trên các máy chủ trong trung tâm dữ liệu Microsoft. Ứng dụng yêu cầu chi trả cho những gì người dùng sử dụng. Cơ sở dữ liệu SQL Azure được xây dựng trên Microsoft SQL Server. Cho qui mô lớn, công nghệ này cung cấp môi trường SQL Server trong đám mây, bổ sung với Index,

View, Store Procedure, Trigger,...và còn nữa. Dữ liệu này có thể được truy xuất bằng ADO.Net và các giao tiếp truy xuất dữ liệu Windows khác. Ngoài ra, SQL Azure hoàn toàn có thể kết nối với các module khác còn lại trong hệ thống Microsoft Azure như Azure Machine Learning.

Khi ứng dụng sử dụng Cơ sở dữ liệu SQL Azure thì yêu cầu về quản lý sẽ được giảm đáng kể. Thay vì lo lắng về cơ chế, như giám sát việc sử dụng đĩa và theo dõi tập tin nhật ký, người sử dụng Cơ sở dữ liệu SQL Azure có thể tập trung vào dữ liệu. Microsoft sẽ xử lý các chi tiết hoạt động. Và giống như các thành phần khác của nền tảng Windows Azure, để sử dụng Cơ sở dữ liệu SQL Azure chỉ cần đến Microsoft Azure Web Portal và cung cấp các thông tin cần thiết. Ứng dụng có thể dựa vào SQL Azure với nhiều cách khác nhau. Một ứng dụng Microsoft Azure có thể lưu trữ dữ liệu trong Cơ sở dữ liệu SQL Azure. Trong khi bộ lưu trữ Microsoft Azure không hỗ trợ các bảng dữ liệu quan hệ, mà nhiều ứng dụng đang tồn tại sử dụng cơ sở dữ liệu quan hệ. Vì vậy lập trình viên có thể chuyển ứng dụng đang chạy sang ứng dụng Microsoft Azure với lưu trữ dữ liệu trong Cơ sở dữ liệu SQL Azure.

Storage services trong Microsoft Azure là dịch vụ lưu trữ mở rộng vô cùng tiện ích cho các lập trình viên với 100 TB mỗi tài khoản, tự động thu gọn để truy xuất các dữ liệu băng thông rộng. Storage services hỗ trợ 3 kiểu dịch vụ lưu trữ bảng: blob, table, queue. Các kiểu dịch vụ này hỗ trợ cục bộ cũng như truy cập trực tiếp thông qua REST services.

Cách đơn giản nhất để lưu trữ dữ liệu trong Microsoft Azure storage là sử dụng Blob. Một blob chứa dữ liệu nhị phân. Cấu trúc lưu trữ của Blob đơn giản như sau: Mỗi tài khoản lưu trữ có một hoặc nhiều container, mỗi container chứa một hoặc nhiều blob. Kích thước Blob có thể lớn đến 50GB, chúng có thể chứa thêm metadata. Ví dụ: nơi chụp của tấm ảnh, hay ca sĩ thể hiện bài hát trong file MP3...

Bộ lưu trữ Microsoft Azure cũng cung cấp Table. Tuy nhiên, nó không phải là bảng quan hệ như trong SQL. Thực tế, dữ liệu lưu trữ bên trong nó là một hệ

thống các thực thể với các thuộc tính. Hơn cả việc sử dụng SQL, một ứng dụng có thể truy cập dữ liệu của Table bằng ADO.NET data Service hoặc LINQ. Một bảng có thể sẽ rất lớn, với hàng tỉ thực thể chứa hàng terabyte dữ liệu. Bộ lưu trữ Microsoft Azure có thể phân vùng cho nó qua nhiều máy chủ khác nhau để tăng hiệu suất.

Ngoài ra, Storage còn có dịch vụ lưu trữ dạng Drives, là cơ chế cho phép một Virtual Hard Drives trong một blob có thể gắn kết như là một ổ đĩa dạng NTFS vào chức năng Compute. Bộ lưu trữ Microsoft Azure có thể được truy cập từ một ứng dụng Microsoft Azure hoặc từ một ứng dụng khác. Trong cả 2 trường hợp, cả ba cách lưu trữ của dịch vụ lưu trữ Microsoft Azure đều có thể sử dụng REST để truy xuất dữ liệu. Mọi thứ đều được đặt tên qua URL và được truy xuất thông qua các thao tác HTTP chuẩn.

4.1.1.2. Giới thiệu về Azure Machine Learning

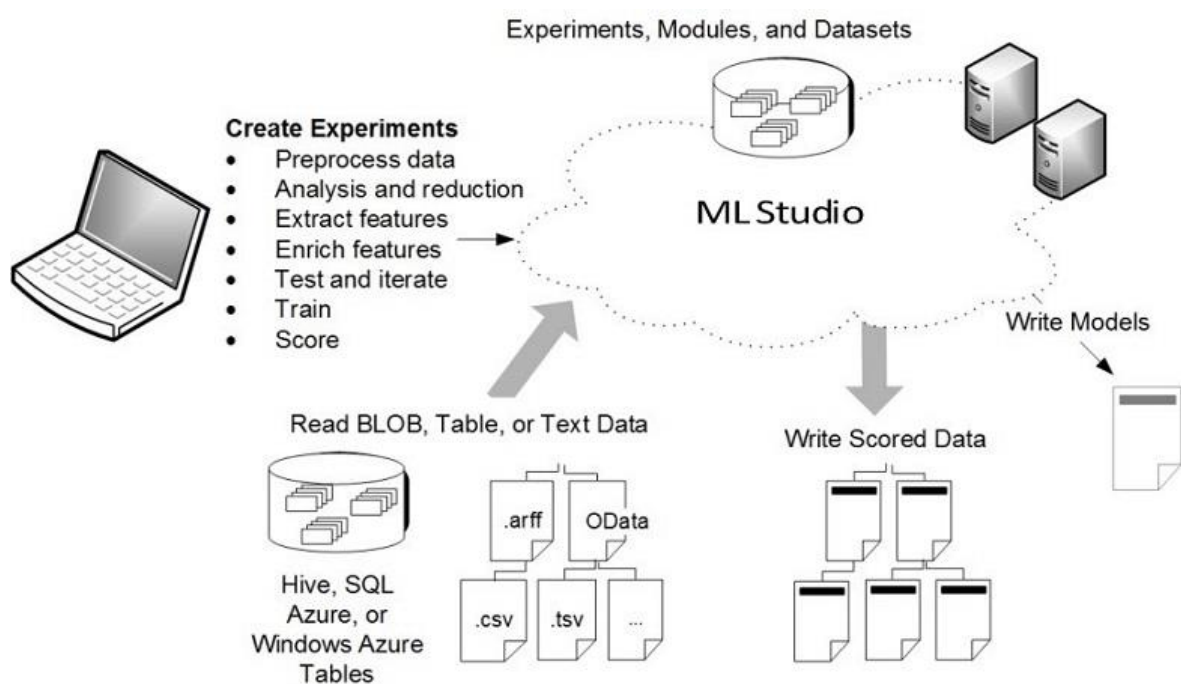
Azure Machine Learning là một hệ thống cho phép phân tích, xử lý, tính toán bằng hệ thống đám mây mà Azure cung cấp. Bên trong Azure Machine Learning có cung cấp sẵn nhiều module có sẵn phù hợp cho các bài toán liên quan đến Machine Learning như *Linear Regression*, *Two Class Regression*, ...

Machine Learning trước kia yêu cầu các phần mềm phức tạp, hệ thống máy tính cao cấp và các nhà khoa học đầy kinh nghiệm để hiểu nó. Đối với các công ty startup hoặc ngay cả các doanh nghiệp lớn quá đắt đỏ và phức tạp. Azure Machine Learning đã thổi luồng không khí mới vào dịch vụ machine learning, giúp nó trở nên dễ tiếp cận hơn. Azure Machine Learning cho phép người dùng không có hiểu biết sâu về khoa học dữ liệu cũng có thể truy cập dữ liệu cho mục đích dự đoán và dự báo.

Đồng thời với Azure Machine Learning, chúng ta không cần phải bận tâm về phần mềm hay phần cứng, môi trường và các dịch vụ đi kèm. Chỉ với trình duyệt và kết nối Internet, chúng ta có thể truy cập vào Azure và bắt đầu phát triển các mô

hình dự đoán và mô hình phân tích trong thời gian nhanh nhất. Azure Machine Learning cũng cho phép chúng ta lưu trữ không giới hạn số lượng file trên Azure Storage, và kết nối đồng bộ với các dịch vụ liên quan đến Azure, bao gồm: HDInsight, giải pháp và dữ liệu lớn dựa trên nền Hadoop, SQL Server database và máy ảo.

Machine Learning Studio, góp phần quan trọng trong toàn bộ giải pháp Machine Learning trên Azure. Azure Machine Learning Studio cung cấp môi trường làm việc trực quan, dễ dàng xây dựng kiểm tra và xây dựng mô hình phân tích, dự đoán mà không cần đòi hỏi phải biết lập trình. Chúng ta có thể kéo thả các dataset và các module phân tích một cách trực quan. Tuy nhiên để có thể mở rộng hơn bạn cần phải sử dụng các ngôn ngữ lập trình như R hoặc Python.



Hình 4.1: Sơ đồ thể hiện quá trình làm việc với Azure Machine Learning Studio (hình ảnh được cung cấp bởi Microsoft)

4.1.2. Hiện thực mô hình n-grams bằng Azure Machine Learning

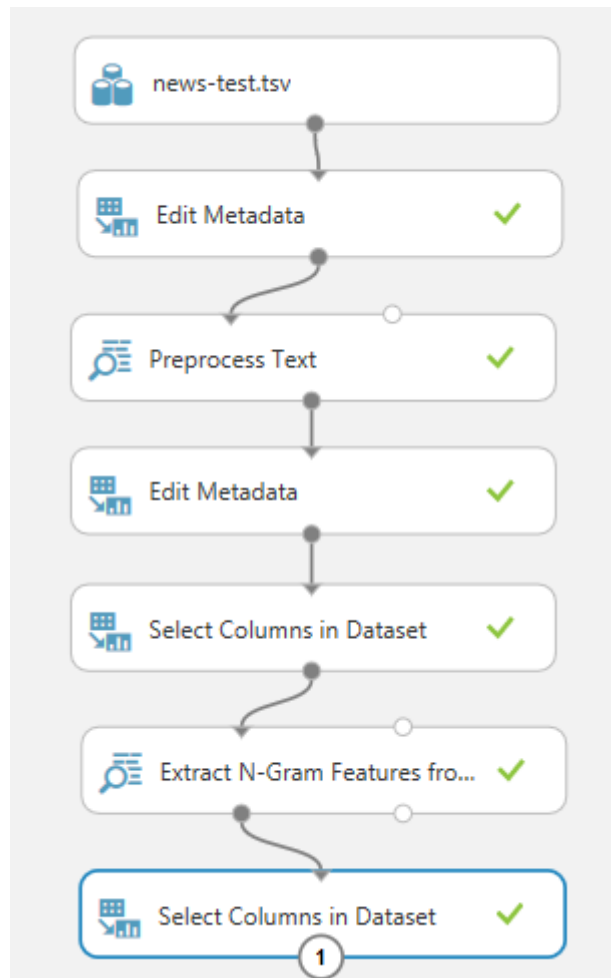
4.1.2.1. Trích xuất dữ liệu

Vì Azure Machine Learning giới hạn về dung lượng cũng như do thời gian thực thi của một module tốn thời gian. Cho nên ta tách bộ ngữ liệu thu được thành một corpus nhỏ. Thực thi trên bộ ngữ liệu đó trước sau đó thay thế vào bộ ngữ liệu lớn hơn.

Để hoàn thành việc trích xuất n-grams, ta sử dụng các module có sẵn trong Azure Machine Learning như sau:

- Preprocess Text: module cho phép ta thực hiện các thao tác tiền xử lý văn bản trước khi đưa vào qua trình trích xuất n-grams
- Extract N-gram Features from Text: module cho phép ta tách n-gram, trả về một bảng chứa số lần xuất hiện của một từ trong câu đó. Với mỗi dòng là một câu.
- Execute Python Script: các module của Azure Machine Learning không cung cấp đủ khả năng tùy biến để thực hiện đủ các thao tác cần thiết. Ta sử dụng thêm module này, và viết script python để thực hiện các thao tác ta muốn
- Các module để xử lý bảng như: Edit Metadata để đổi tên cột; Select Columns in Dataset để chọn cột và Split Data chia dữ liệu ra các phần nhỏ khác nhau.

Ta có sơ đồ sau trong Azure Machine Learning Studio để xử lý trên dữ liệu nhỏ:



Hình 4.2: Sơ đồ thể hiện quá trình làm việc với dữ liệu nhỏ
Với các tùy chỉnh sau trong từng module để phù hợp với giải thuật đã nêu trước đó.

Preprocess Text

Language
English

Remove by part of speech
False

Text column to clean
Selected columns:
Column names: text
Launch column selector

- ☐ Remove stop words
- ☐ Lemmatization
- ☐ Detect sentences
- ☒ Normalize case to lowercase
- ☒ Remove numbers
- ☒ Remove special characters
- ☐ Remove duplicate characters
- ☒ Remove email addresses
- ☒ Remove URLs
- ☒ Expand verb contractions
- ☒ Normalize backslashes to slashes
- ☒ Split tokens on special characters

Custom regular expression

Custom replacement string

Extract N-Gram Features from Text

Text column
Selected columns:
Column names: T
Launch column selector

Vocabulary mode
Create

N-Grams size
4

K-Skip size
0

Weighting function
TF Weight

Minimum word length
1

Maximum word length
2500000

Minimum n-gram document absolute frequency
5

Maximum n-gram document ratio
1000000000

- ☐ Detect out-of-vocabulary rows
- ☒ Mark begin-of-sentence
- ☐ Normalize n-gram feature vectors

Use filter-based feature selection
False

Hình 4.3: Cài đặt cho các module để xử lý trên bộ ngữ liệu nhỏ

rows	columns														
25	14														
		T.[prices]	T.[more]	T.[than]	T.[and]	T.[price]	T.[in]	T.[it]	T.[to]	T.[is]	T.[a]	T.[of]	T.[that]	T.[the]	T.[gold]
view as															
		0	0	0	0	0	0	1	1	0	1	1	0	1	1
		1	1	2	0	0	0	1	0	1	0	0	0	1	1
		0	0	0	1	0	0	0	0	0	0	0	0	0	1
		0	0	0	0	0	0	1	0	0	0	0	0	0	0
		0	0	0	0	1	0	0	0	0	0	1	0	1	1
		1	0	0	0	0	0	0	0	0	1	0	0	0	1
		0	0	0	0	1	0	0	0	0	0	0	1	1	1
		0	0	1	0	0	0	0	0	0	0	0	2	0	1
		1	1	1	0	0	1	0	1	0	2	0	1	3	1
		0	0	0	0	0	0	0	0	0	0	0	0	1	0
		0	0	0	0	0	0	1	0	0	0	0	1	1	1

Hình 4.4: Kết quả đạt được sau quá trình tách dữ liệu

Có thể thấy đó là module hoạt động chính xác như ý muốn. Ta viết thêm thêm một script python để tính tổng số lần xuất hiện của 1 gram.

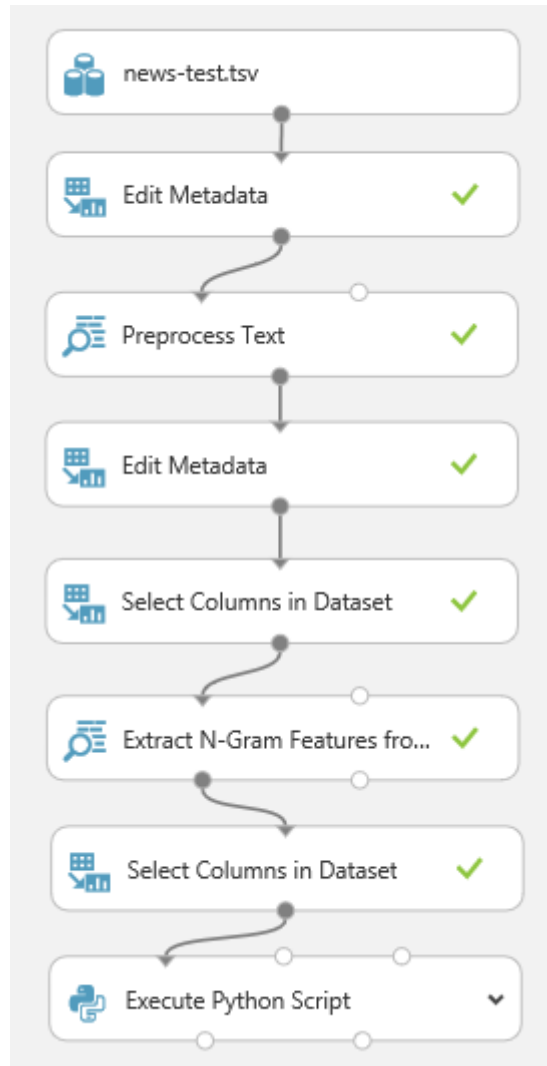
Sum up N gram

```
import pandas as pd
import numpy as np
```

Param<dataframe2>: a pandas.DataFrame contain N-Gram after extraction

```
def azureml_main(dataframe1, dataframe2 = None):
    result=pd.DataFrame(dataframe1.sum(axis = 0))
    result=result.T
    return result
```

Và gắn module Excute Python Script vào chương trình cùng với đoạn code này:



Hình 4.5: Gắn thêm module Excute Python Code vào sơ đồ

rows	columns
1	14
	T.[prices] T.[more] T.[than] T.[and] T.[price] T.[in] T.[it] T.[to] T.[is] T.[a] T.[of] T.[that] T.[the] T.[gold]
view as	
	5 5 6 5 5 6 6 6 6 11 9 9 19 16

Hình 4.6: Kết quả sau khi gắn thêm module Python

Sau khi có được kết quả như ý muốn với bộ bộ ngữ liệu nhỏ. Ta bắt đầu thực thi trên bộ bộ ngữ liệu thật. Trên bộ corpus tin tức thu thập từ năm 2006 đến năm 2007 có khối lượng 462 MB. Bộ bộ ngữ liệu chứa chính xác 3, 782, 550 dòng với mỗi dòng là một câu hoàn chỉnh. Do yêu cầu không cần sử dụng chính xác mỗi dòng một câu, module của Azure cũng đã có hỗ trợ nhận biết câu. Vì thế, ta viết

một đoạn chương trình C# sử dụng mã giả sau để nối cứ 10,000 câu lại thành một hàng để hệ thống Azure không bị quá tải trong quá trình làm việc.

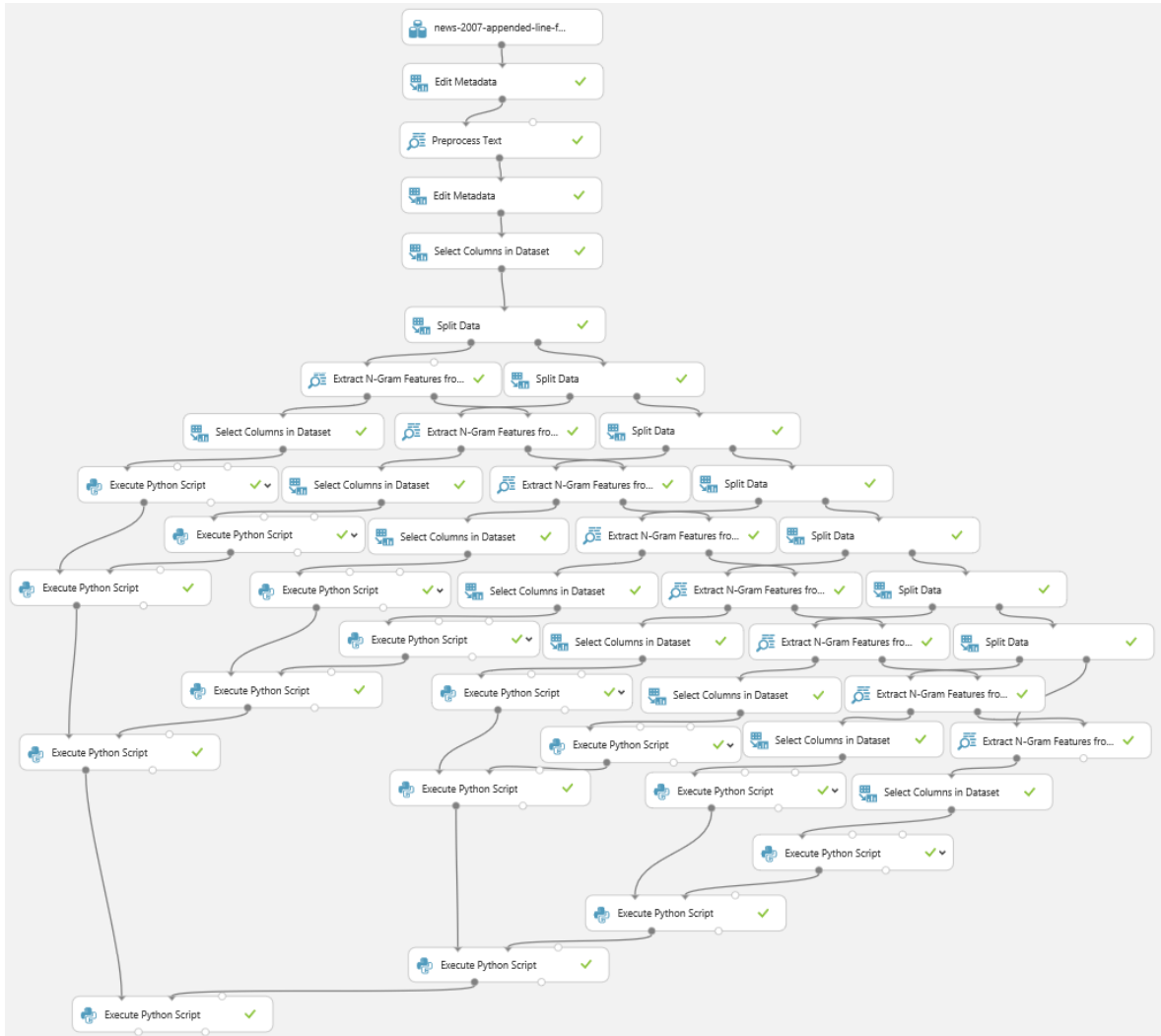
```
string[] readText = đọc file từ một được dẫn
for i: 0 -> readText.Length {
    if (readText[i] kết thúc câu không phải dấu ".") {
        thêm dấu chấm vào cuối câu
    }
}
List<string> newText = new List<string>();
for i: 0 -> readText.Length {
    string newLine = "";
    for j: i -> i + 10000 {
        newLine += readText[j]
        if (j >= readText.Length)
            break;
    }
    newText.Add(newLine);
}
File.WriteAllLines(pathToSave, newText);
}
```

Kết quả, ta thu về được tệp tin chứa 38 dòng với mỗi dòng chứa 10000 câu. Tuy nhiên, vì giới hạn trên hệ thống Azure Machine Learning vẫn tồn tại khiến ta không thực thi một lúc cả bộ ngữ liệu được. Vì thế, ta sử dụng module *Split Data* để tiếp tục phân nhỏ dữ liệu ra và chạy lần lượt, cuối cùng sử dụng một đoạn script Python để nối các bảng lại với nhau:

```
# Join table
import pandas as pd
import numpy as np

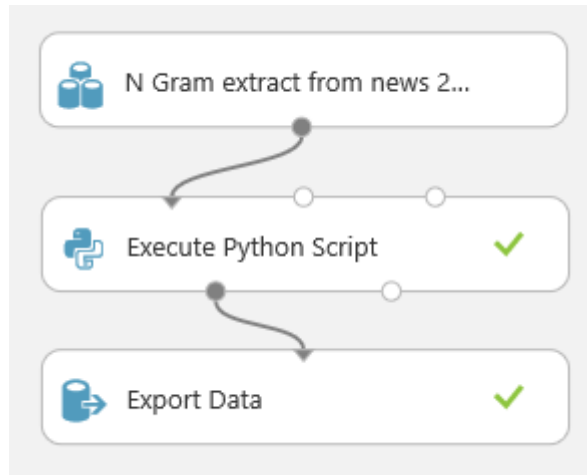
# Param<dataframe1>: a pandas.DataFrame N gram after sum up
# Param<dataframe2>: a pandas.DataFrame N gram after sum up
def azureml_main(dataframe1, dataframe2):
    result = pd.concat([dataframe1, dataframe2], axis = 0)
```

```
result = pd.DataFrame(result.sum(axis = 0))
result = result.T
return result
```



Hình 4.7: Sơ đồ các module để trích xuất n-gram từ bộ ngữ liệu lớn

Sau quá trình trích xuất dữ liệu, dữ liệu được lưu lại làm thành dataset. Vì lý do mỗi lần ta cần excute một lệnh lên dataset, hệ thống Azure Machine Learning sẽ load hết dataset này lên RAM khiến cho chương trình thực thi lâu. Vì thế, ta sử dụng hệ thống Microsoft Azure, tạo một database Azure SQL và lưu trữ toàn bộ dataset đã được trích xuất vào data base này.



Hình 4.8: Sơ đồ các module để đưa dữ liệu vào database Azure SQL

```
SELECT TOP 1000 [keyWord]
, [countWord]
FROM [dbo].[ngram]
ORDER BY [countWord] DESC
```

100 %

Results Messages

	keyWord	countW...
1	T.[the]	77205
2	T.[to]	34759
3	T.[of]	33112
4	T.[and]	32888
5	T.[a]	31127
6	T.[in]	28512
7	T.[of_the]	22925
8	T.[in_the]	20875

Hình 4.9: Thử truy xuất lên database đã tạo trên Azure SQL

```
SELECT COUNT(countWord) FROM ngram
```

100 %

Results Messages

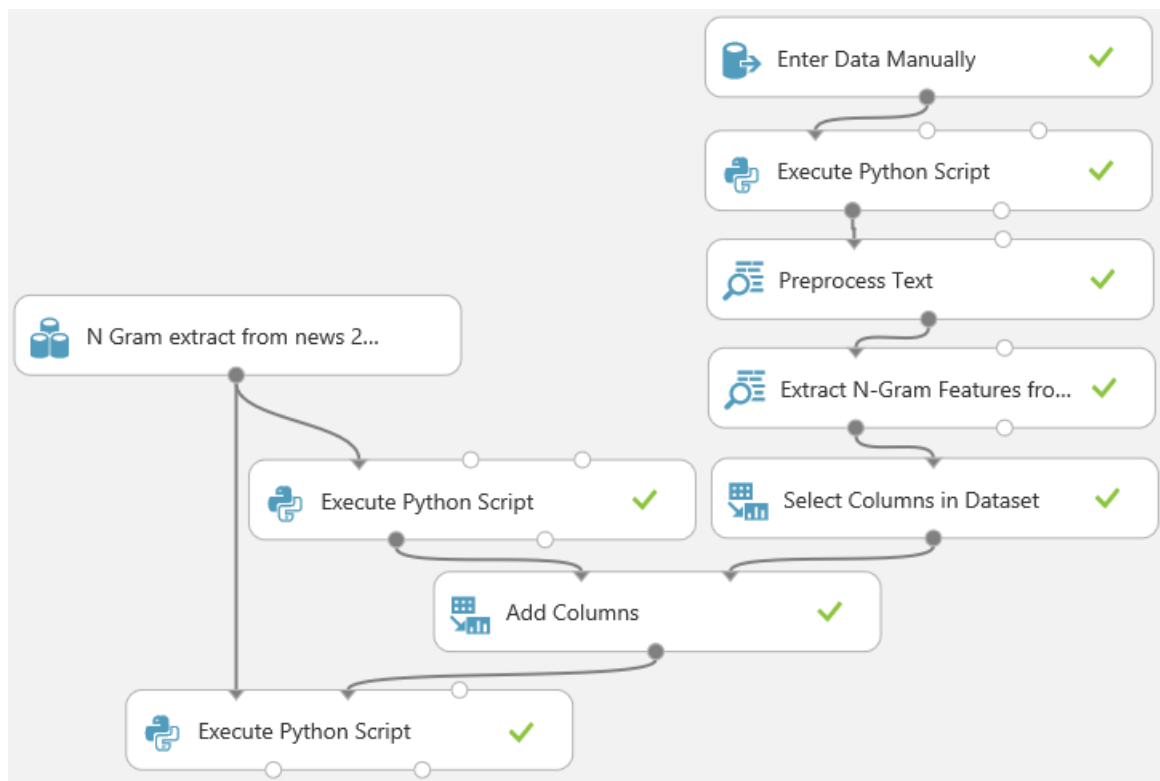
	(No column na...
1	2260378

Hình 4.10: Thử truy xuất lên database đã tạo trên Azure SQL


4.1.2.2. Hiện thực việc chọn đáp án của một câu

Sau khi thu thập được các n-grams từ bộ ngữ liệu, ta tiến hành hiện thực việc giải một câu tiếng Anh với bộ dataset sau khi thu thập được. Ta sử dụng các module của Azure Machine Learning để thực hiện giải thuật đã đề ra trước đó, bao gồm:


- Thay thế các từ vào câu
- Tiền xử lý các câu trong văn bản
- Rút trích n-grams
- Tính toán xác suất hiện của một câu dựa trên công thức tổng logarit và add-one



Hình 4.11: Sơ đồ các module để thực hiện việc chọn đáp án

rows	columns				
4	4				
		NGram	key	text	Preprocessed text
view as					
					
		3	was	I was with mom in 1980's.	i was with mom in ' s.
		3	be	I be with mom in 1980's.	i be with mom in ' s.
		3	am	I am with mom in 1980's.	i am with mom in ' s.
		3	been	I been with mom in 1980's.	i been with mom in ' s.

Hình 4.12: Kết quả sau khi tiền xử lý

rows	columns				
3	4				
		maxCount	NGram	key	NGramsString
view as					
					
		77205	3	seen	["i","have","seen","<P>_i","i_have","have_seen","<P>_i_have","i_have_seen"]
			3	saw	["i","have","saw","<P>_i","i_have","have_saw","<P>_i_have","i_have_saw"]
			3	see	["i","have","see","<P>_i","i_have","have_see","<P>_i_have","i_have_see"]

Hình 4.13: Kết quả sau khi trích xuất các n-gram trong câu

Sau đó tiến hành chọn đáp án dựa trên mã giả sau:

```
function getTokensCount(dataSet, tokens)
    tokenCounts = []
    for token in tokens:
        count = 1;
        if token in dataSet:
            count = dataSet[token]['count'] + 1;
        tokenCounts.add(count);
```

```

return tokenCounts

main:
    listOfNGrams = danh sách các danh sách chứa n-gram;
    dfFromSQL = lấy danh sách từ và số lần xuất hiện của các n-gram trong
listOfNGrams;
    for NGrams in listOfNGrams:
        tokens = getTokenByNGram(NGrams, N);
        tokenCounts = getTokensCount(dfFromSQL, tokens);

        result = 0;
        for count in tokenCounts:
            result += log(count / maxCount);
        results.append(result);

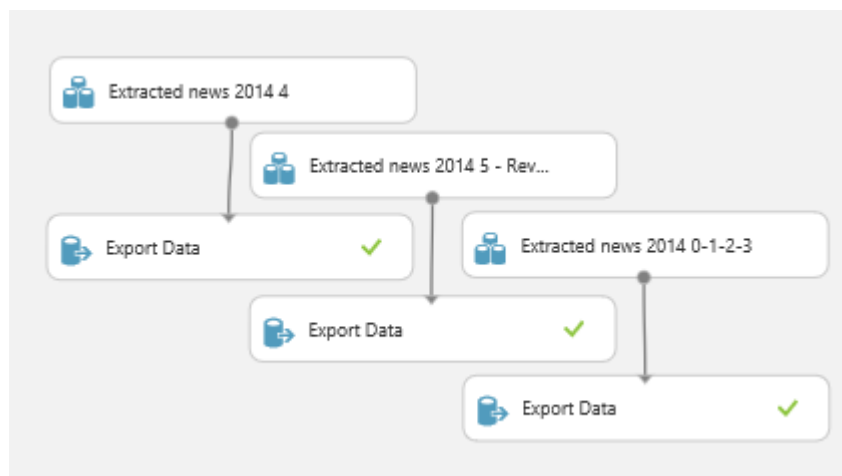
    maxIndex = max(results);
    return maxIndex;

```

4.1.3. Xuất dữ liệu sang môi trường SQL

Để tăng tốc độ truy xuất đến các gram, khóa luận đề xuất đưa toàn bộ dữ liệu lên môi trường SQL để có thể truy vấn với tốc độ cao. Ở khóa luận [2] và khóa luận [1] cũng sử dụng SQL để truy vấn n-gram cho thấy tốc độ cải thiện đáng kể thay cho việc lưu trên file và truy xuất thủ công.

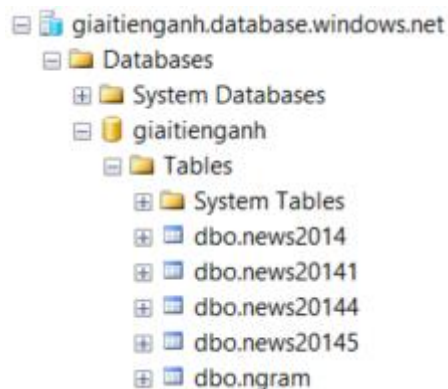
Nhằm kiểm nghiệm hệ thống, toàn bộ dữ liệu n-gram sau khi được trích xuất sẽ được xuất sang Azure SQL sau đó lưu về nội bộ ở máy tính bàn để truy vấn nhằm cải thiện tốc độ kiểm nghiệm của hệ thống. Hệ thống Azure Machine Learning cung cấp công cụ để ta có thể xuất dữ liệu sang Azure SQL.



Hình 4.14: Xuất dữ liệu thu được sang môi trường Azure SQL

Nhằm để giảm tải hệ thống Azure SQL, ta chia nhỏ bộ dữ liệu sau khi được trích xuất ra làm 3 phần và thay phiên gửi lên Azure SQL. Sau đó trên hệ thống SQL ta sử dụng một đoạn script để gom nhóm 3 dữ liệu này lại.

```
insert into news2014 ([keyWord], [countWord])
select
    C.[keyWord],
    sum((COALESCE(news20141.[countWord],0) +
COALESCE(news20144.[countWord],0) + COALESCE(news20145.[countWord],0)))
[countWord]
from
    (select [keyWord] from news20141
    union
    select [keyWord] from news20144
    union
    select [keyWord] from news20145
    ) C left join news20141 on C.[keyWord] = news20141.[keyWord] left join
news20144 on C.[keyWord] = news20144.[keyWord] left join news20145 on
C.[keyWord] = news20145.[keyWord]
group by C.[keyWord]
```



Hình 4.15: Kết quả sau khi đưa toàn bộ dữ liệu lên Azure SQL

4.1.4. Câu hỏi thử nghiệm hệ thống

Các câu hỏi của đề tài trước để lại bao gồm các file Excel chứa các cột:

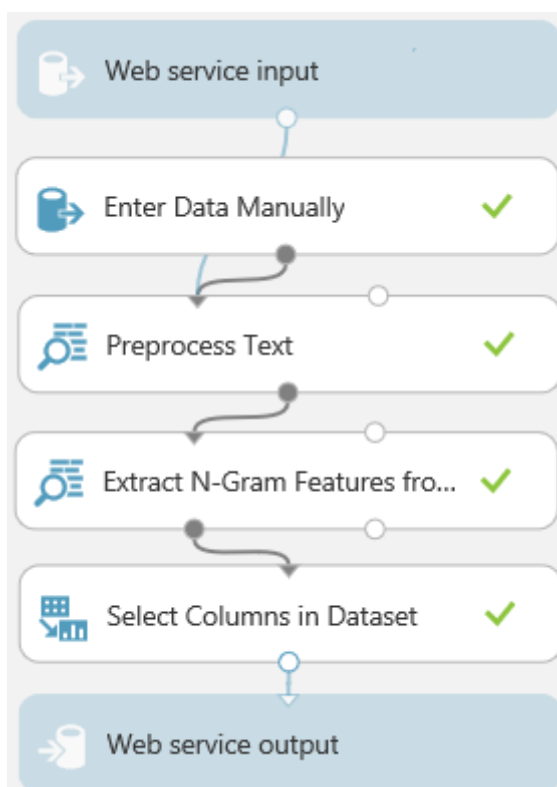
- **Content** – Câu hỏi có chứa dấu khoảng trắng
- **A, B, C, D** – các câu trả lời gợi ý
- **Key** – đáp án cho câu hỏi này

- **Machine Key** – đáp án cho câu hỏi này

Từ nội dung này, khóa luận viết một đoạn mã bằng ngôn ngữ Java để lần lượt load các file Excel lấy nội dung câu hỏi và các câu trả lời gợi ý truy vấn vào hệ thống hiện tại mà khóa luận đang hiện thực và lấy câu trả lời

4.1.5. Hiện thực các hàm liên quan

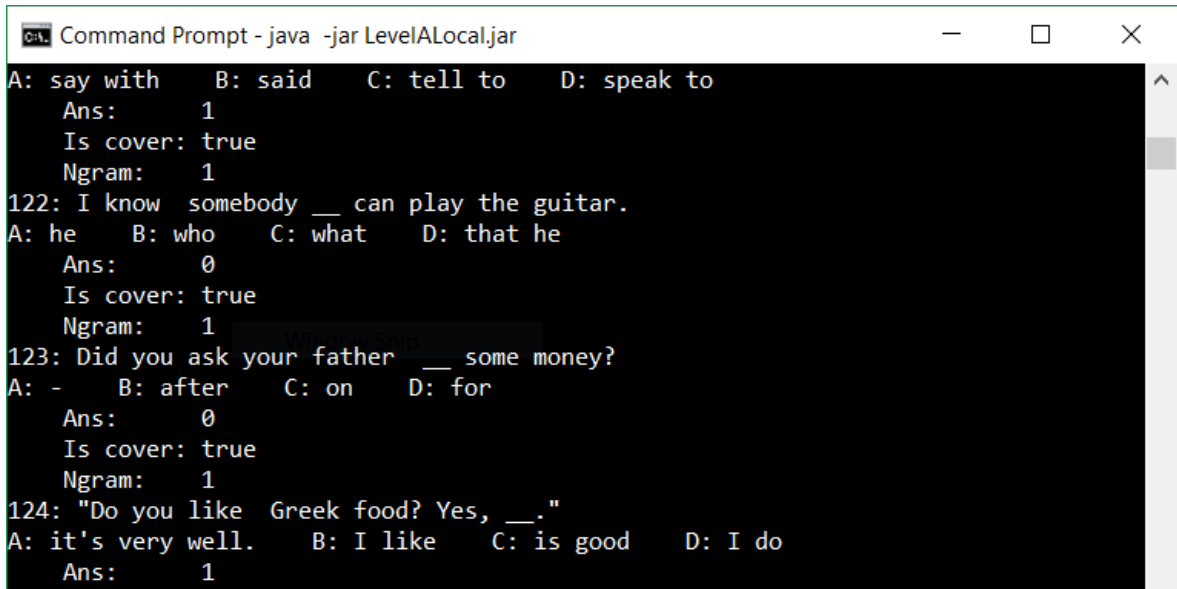
Sau khi đưa dữ liệu sang môi trường SQL và có chương trình Java cùng bộ câu hỏi ở nội bộ máy tính. Tuy nhiên, một số chức năng cơ bản của hệ thống Azure Machine Learning như tiền xử lý câu và trích xuất gram của câu vẫn cần để thực hiện quá trình khảo sát. Khóa luận tiến hành tạo các Web API mà hệ thống Azure Machine Learning cung cấp. Việc tiền xử lý văn bản và trích xuất n-gram trong hàm này phải chính xác tương khớp với toàn bộ các tùy chỉnh trong lúc trích xuất dữ liệu trước đó nhằm đảm bảo tính đồng nhất giữa n-gram trích xuất được từ câu hỏi sau khi thể đáp án và các n-gram đang được lưu trữ trên hệ thống.



Hình 4.16: Hiện thực hàm tiền xử lý và trích xuất n-gram với các tùy chỉnh tương tự trước đó và xuất ra thành Web API

Tiến hành khảo sát

Ta sử dụng Java để hiện thực chương trình khảo sát giống với thuật toán và mã giả đã đề ra trước đó.

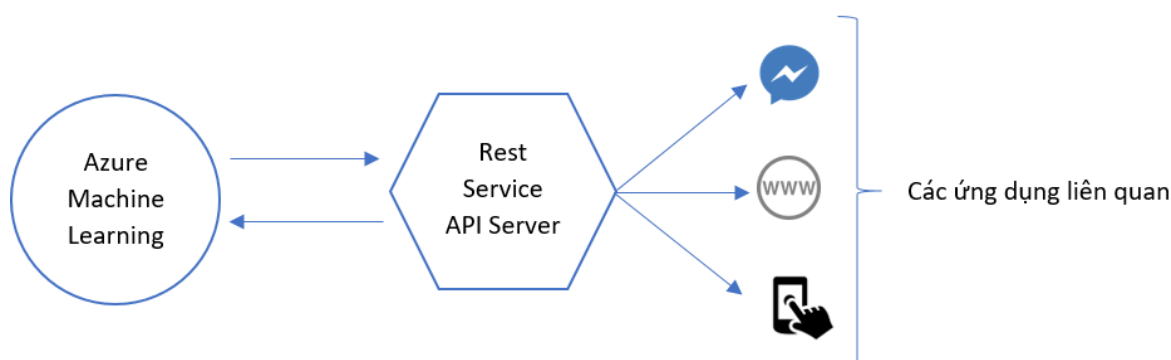


```
Command Prompt - java -jar LevelALocal.jar
A: say with    B: said    C: tell to    D: speak to
Ans:          1
Is cover: true
Ngram:        1
122: I know somebody __ can play the guitar.
A: he    B: who    C: what    D: that he
Ans:      0
Is cover: true
Ngram:    1
123: Did you ask your father __ some money?
A: -    B: after    C: on    D: for
Ans:      0
Is cover: true
Ngram:    1
124: "Do you like Greek food? Yes, __."
A: it's very well.    B: I like    C: is good    D: I do
Ans:                  1
```

Hình 4.17: Chương trình Java sử dụng lần lượt các câu hỏi để truy vấn lấy câu trả lời từ hệ thống

4.1.6. Hiện thực hệ thống Web Service API và chatbot

Để triển khai ứng dụng, hệ thống Azure Machine Learning cung cấp cho ta một Web API Service của các experiment. Các Web API Service nhận vào một lệnh HTTP Post Request chứa body là một JSON chứa truy vấn và trả về một JSON chứa câu trả lời. Ở đây, JSON chứa truy vấn đó là câu hỏi, các câu trả lời gợi ý và số n-gram cần truy vấn. Mục tiêu của khóa luận đó là một hệ thống có thể triển khai được nhiều ứng dụng. Vì thế khóa luận quyết định xây dựng ứng dụng là một Rest Service API Service chứa các API để giải câu hỏi cùng với các ứng dụng liên quan, bao gồm: chat bot giải tiếng Anh, một web site và một ứng dụng điện thoại di động thông minh. Sơ đồ của hệ thống như sau:



Hình 4.18: Sơ đồ thiết kế hệ thống ứng dụng

Để xây dựng một Rest Service API Service, khóa luận sử dụng hệ thống server Heroku được cung cấp miễn phí và service xây dựng trên nền tảng Java.

4.1.6.1. Giới thiệu về Heroku

Heroku là một dịch vụ nền tảng đám mây hỗ trợ một số ngôn ngữ lập trình được sử dụng như một mô hình triển khai ứng dụng web. Heroku là một trong những nền tảng đám mây đầu tiên được phát triển từ tháng 6 2007, thời điểm đó chỉ hỗ trợ mỗi ngôn ngữ Ruby. Ngày nay, Heroku đã hỗ trợ nhiều ngôn ngữ khác

nhau như Java, Node.js, Scala, Python, PHP, ... Ngoài ra, ngày nay Heroku còn hỗ trợ giao thức chuẩn HTTPS trên các server miễn phí.

Tại thời điểm hiện tại, server miễn phí do Heroku cung cấp có giới hạn, đó là server sẽ tự động tắt sau 30 phút nếu không có bất cứ lệnh request nào được gửi đến server. Tuy nhiên server sẽ tự khởi động lại nếu có request gửi đến server. Ngoài ra, heroku còn hỗ trợ người dùng các tính năng như log, lưu files, ...

4.1.6.2. Hiện thực chat bot trên Facebook Messenger

Ngày nay, Facebook đã cho ra mắt Messenger Platform chatbot API giúp cho lập trình viên có thể tự tạo chatbot cho riêng mình. Mô hình chatbot thân thiện với người dùng giúp người dùng dễ tiếp cận với ứng dụng hơn. Ngày nay, một số tổ chức nổi tiếng như trang tin tức CNN, hay một số cửa hàng nhỏ cũng đã sử dụng chatbot như một công cụ để giao tiếp với người dùng.

Để hiện thực chatbot trên Facebook Messenger, ta cần một địa chỉ webhook với giao thức chuẩn HTTPS. Facebook Messenger Platform cung cấp một token để server có thể giao tiếp được với chatbot và gửi thông tin cần thiết. Sau quá trình cài đặt, ta hiện thực chatbot với mã giả như sau:

```
if (Người dùng gửi "start"))
    Bắt đầu nhận câu hỏi

else
    if (Đang nhận câu trả lời)
        if (Người dùng gửi "~~~" hoặc "!!!")
            if (Người dùng gửi "!!!") {
                if (trước đó không có tin nhắn nào khác)
                    Dừng việc nhận câu hỏi và câu trả lời

            }

        if (Có câu hỏi và câu trả lời)
            Gửi câu hỏi và các câu trả lời gợi ý lên hệ thống lấy kết quả
            Gửi trả kết quả cho người dùng

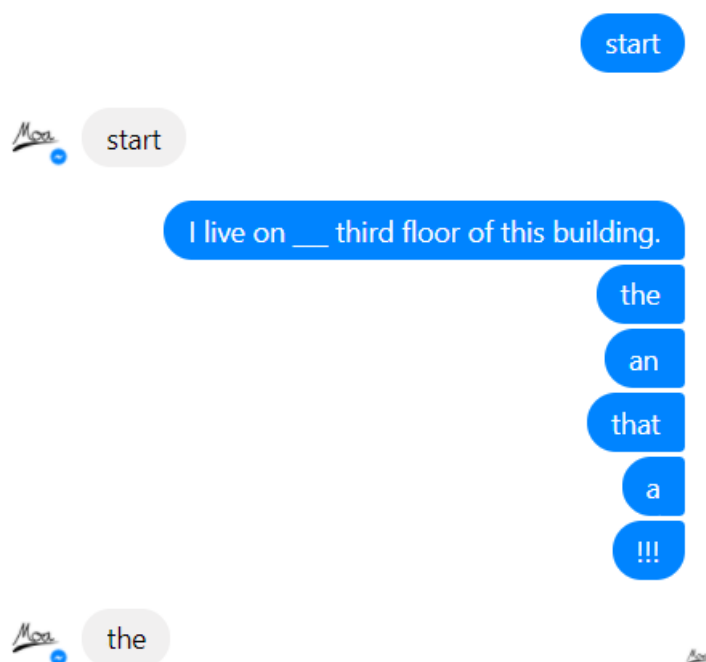
        else if (lần đầu gửi tin nhắn sau khi gửi "start")
            lưu tin nhắn vừa rồi làm câu có khoảng điền khuyết

    else
```


lưu tin nhắn vừa rồi làm câu trả lời gợi ý

else

Trò chuyện cùng người dùng



Hình 4.19: Kết quả sau khi hiện thực chatbot

4.2. Thử nghiệm

4.2.1. Kết quả khảo sát

Khóa luận tiến hành khảo sát độ chính xác và độ bảo phủ toàn câu hỏi của hệ thống trên bộ ngữ liệu được thu thập đến năm 2007 và bộ ngữ liệu được thu thập đến năm 2014 của statmt.org. Công thức tính độ chính xác như sau:

$$\text{Độ chính xác} = \frac{\text{Số câu trả lời đúng}}{\text{Toàn bộ số câu trong bộ đề}} \times 100$$

	1-gram	2-gram	3-gram	4-gram
	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)
Bảng A (800 câu)	23.75	32.88	35.25	30.50
Bảng B (460 câu)	25.22	33.04	34.78	30.65
Bảng C (2020 câu)	23.71	36.53	35.59	29.46
TOEFL (820 câu)	28.66	38.17	35.85	32.20
Trung bình	25.34	35.16	35.37	30.70

Bảng 4.2: Kiểm nghiệm độ chính xác trên bộ ngữ liệu thu thập đến năm 2007

	1-gram	2-gram	3-gram	4-gram
	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)
Bảng A (800 câu)	23.5	42.375	50.25	44.625
Bảng B (460 câu)	22.17	38.26	45.87	44.13
Bảng C (2020 câu)	24.75	46.14	57.52	52.23
TOEFL (820 câu)	21.71	39.39	44.27	38.05
Trung bình	23.03	41.54	49.48	44.76

Bảng 4.3: Kiểm nghiệm độ chính xác trên bộ ngữ liệu thu thập đến năm 2014

4.2.2. Kiểm nghiệm tốc độ giải câu hỏi trên máy bàn nội bộ

Khóa luận sử dụng lại chương trình Java lúc tiến hành khảo sát, giả lập trường hợp nhiều người sử dụng liên tục bằng cách cho 4 chương trình giải bộ đề C lần lượt với 1, 2, 3, 4 - gram chạy cùng một lúc và lấy thời gian thực thi của cả 3 chương trình. Như vậy 4 chương trình tổng cộng thực hiện 32, 320 câu hỏi. Như kết quả chương trình bên dưới, ta lấy thời gian lớn nhất trong 4 chương trình, hệ thống tốn tổng cộng 1171 giây để giải 32, 320 câu hỏi. Vậy trong một giây, hệ thống có thể giải được 27.6 câu hỏi. Cho thấy hiệu suất của hệ thống rất lớn nếu triển khai trên ứng dụng thật.

```
Command Prompt
Ngram: 4
2019: "If I had seen this film on TV, I __ to
the cinema to see it."
A: don't go B: didn't go C: wouldn't hav
e gone D: wouldn't go
Ans: 0
Is cover: false
Ngram: 4
2020: What __ for the last two weeks?
A: were you doing B: have you been doing
C: did you do D: are you doing
Ans: 3
Is cover: false
Ngram: 4
Total Excute time: 1170216 milliseconds.
C:\Users\Nguyen>
```

Hình 4.20: Chương trình kiểm nghiệm hiệu suất của hệ thống

4.3. Mô hình n - gram

Từ bộ ngữ liệu, khóa luận tiến hành tiền xử lý như: đơn giản hóa từ, mở rộng cuối động, xóa các dữ liệu thừa sau đó thu thập các n-grams. Để dễ cài đặt và làm nhẹ hệ thống, cũng như do hệ thống Azure Machine Learning có giới hạn dung lượng và yêu cầu trả phí nếu vượt mức; vì thế khóa luận sử dụng bộ ngữ liệu tin tức được thu thập đến năm 2007 với dung lượng là 462 MB. Sau đó đăng ký thử nghiệm hệ thống để sử dụng ở mức cao hơn trong giới hạn và sử dụng bộ ngữ liệu thu thập đến năm 2014 với dung lượng là 4.1 GB để khảo sát lần thứ 2. Trong điều kiện khóa luận có thể phát triển thành ứng dụng lớn, ta có thể tiếp tục trả phí cho hệ thống Azure để có thể sử dụng trích xuất các n-gram của các bộ ngữ liệu lớn hơn nữa và cộng dồn vào dữ liệu trước đó của hệ thống để cải thiện độ chính xác.

Trong đó, khóa luận đề xuất việc tiền xử lý văn bản gồm:

- Thêm thành phần để phát hiện bắt đầu câu.
- Loại bỏ dấu câu và các ký tự đặc biệt, thay thế bằng dấu khoảng cách
- Loại bỏ các thành phần đặc biệt.
- Thay thế từ viết tắt.

Mục tiêu của tiền xử lý nhằm đưa văn bản về dạng thống nhất, loại bỏ các thành phần đặc biệt nhưng không làm mất đi tính chất ngữ pháp, ngữ nghĩa của văn bản.

Trong nghiên cứu [4] có đưa ra bảng đánh giá kết quả khi tác giả thực thi hệ thống giải bài tập tiếng Anh TOEIC dạng điền khuyết dựa trên bộ n-gram được cung cấp bởi Google. Trong nghiên cứu có báo cáo về việc 4-gram cho về độ bao phủ thấp, tuy nhiên các câu mà 4-gram bao phủ được đều giải chính xác rất cao. Còn 3-gram lại cho về độ bao phủ cao hơn tuy nhiên độ chính xác thấp hơn so với sử dụng 4-gram. Trong nghiên cứu này, tác giả thử trích xuất các câu không

thể bao phủ bởi 4-gram thì thấy đa số các câu do là câu chỉ có 3 chữ hoặc một số câu đặc biệt mà tác giả chưa tiền xử lý dẫn đến việc không thể trích xuất 4-gram từ các câu này. Vì thế, khóa luận đề xuất việc thêm thành phần đánh dấu đầu câu ở mỗi câu để đảm bảo các câu sẽ có từ 4 token trở lên, thuận lợi cho việc tính toán xác suất chính xác hơn.

	Measurement	Vocabulary	Grammar	Total
5gram	Recall(%)	56.8	46.667	53
	Precision(%)	78.873	100	85.849
	F1-measure	66.041	63.636	65.538
4gram	Recall(%)	90.16	79.92	85
	Precision(%)	85.455	86.667	85.882
	F1-measure	87.746	88.1504	85.438
Trigram	Recall(%)	100	98.611	99.5
	Precision(%)	75.781	85.915	79.397
	F1-measure	86.222	91.826	88.318
Trigram & 4gram	Recall(%)	100	97.436	99
	Precision(%)	83.607	86.842	84.848
	F1-measure	91.071	91.831	91.379

Bảng 4.4: Bảng so sánh độ chính xác giữa các n-gram trong việc giải đề thi TOEIC dựa trên bộ n-gram của Google trong nghiên cứu [4]

Ở khóa luận tiền nhiệm của đề tài này ở trường [1] trong quá trình tiền xử lý văn bản, tác giả lược bỏ các con số, đường dẫn và các ký tự dấu câu, sử dụng bộ ngữ liệu lớn hơn là Open American National Corpus, vì thế trả về số lượng token rất lớn. Tuy nhiên, trong quá trình tiền xử lý, tác giả có loại bỏ stop word, là thành phần quan trọng trong việc phát hiện ngữ pháp và cấu tạo ngữ pháp của câu Tác giả lại chỉ sử dụng 2-gram và 3-gram để làm khảo sát và như ở bảng báo cáo của nghiên cứu [4] ta thấy rõ 3-gram cho về độ chính xác thấp dù đã được huấn luyện trên bộ n-gram rất lớn của Google (13 GB). Vì thế độ chính xác của ứng dụng mà tác giả tiền nhiệm của đề tài này đạt mức rất thấp, không vượt qua mức 50%.

Sau khi tham khảo bảng (4.4) Khóa luận tiến hành khảo sát phương pháp kết hợp 4-gram và 3-gram bằng cách sử dụng 4-gram khảo giải đáp án trước, nếu 4-

gram không phủ cả câu thì sử dụng 3-gram để giải. Vì ta thấy độ chính xác của 2-gram thấp hơn 3-gram nhiều (18-21%) vì thế nếu 3-gram không bao phủ được câu, ta vẫn dùng 3-gram để giải để đảm bảo độ chính xác cao. Kết quả cho thấy độ chính xác có cải thiện nhưng rất ít. Có thể phân tích là do độ phủ của 4-gram chưa tốt, vì bộ ngữ liệu dùng để train vẫn còn bé để có thể sử dụng ở môi trường thực tế khi đưa ra ứng dụng. Vì thế ta có thể phỏng đoán độ chênh lệch này sẽ tăng nếu đưa thêm nhiều ngữ liệu hơn nữa. Chi phí để giải quyết bài toán kết hợp 3 và 4 gram là gần gấp đôi đối so với ban đầu chỉ sử dụng một loại gram vì độ bao phủ cả câu của 4-gram không cao. Và sau khi khảo sát độ chính xác của các n-gram, khóa luận quyết định chọn 3-gram để hiện thực ứng dụng.

	3 và 4-gram
	Độ chính xác (%)
Bảng A (800 câu)	49,63
Bảng B (460 câu)	46.74
Bảng C (2020 câu)	57.78
TOEFL (820 câu)	44.27
Trung bình	49.21

Bảng 4.5: Bảng khảo sát độ chính xác khi kết hợp 3 và 4 gram trên bộ ngữ liệu năm 2014

4.4. Ngữ liệu

Mục tiêu của khóa luận đó là giải các câu hỏi tiếng Anh dạng điền khuyết có một chỗ trống. Ưu tiên cho các câu hỏi từ các đề quốc tế phổ biến hiện nay như TOEIC, TOEFL. Các đề quốc tế hiện nay xoay quanh các câu hỏi mang đề tài kinh tế, báo chí, xã hội và những mẫu đối thoại trong văn phòng.

Bộ ngữ liệu khóa luận sử dụng để huấn luyện hệ thống là bộ ngữ liệu tin tức được thu thập từ năm 2006 đến nay của statmt.org. Được cung cấp miễn phí tại trang chủ của statmt.org.

Bộ ngữ liệu đơn giản là một tệp tin với các mẫu tin tức khác nhau. Các tin tức được phân thành hàng với mỗi hàng là một câu. Bộ ngữ liệu có nhiều thứ tiếng tuy nhiên khóa luận sử dụng bộ ngữ liệu tiếng Anh để phù hợp về yêu cầu của hệ thống.

Ở khóa luận của người tiền nhiệm đề tài này ở trường [1], tác giả sử dụng bộ ngữ liệu Open American National Corpus là bộ ngữ liệu chứa các vấn đề lịch sử và các báo cáo như: báo cáo 911, các bài hướng dẫn du lịch, các bài viết về lịch sử, ... Vì thế bộ ngữ liệu không gần với vấn đề cần được giải quyết đó là các bài thi tiếng Anh thường xoay quanh chủ đề kinh tế, xã hội, các mẫu đối thoại trong văn phòng, Vì thế độ chính xác của ứng dụng mà tác giả tiền nhiệm của đề tài này đạt mức rất thấp, không vượt qua mức 50%.

Vì thế, bộ ngữ liệu khóa luận sử dụng là bộ ngữ liệu tin tức được thu thập từ năm 2006 đến năm 2007. Các tin tức xoay quanh các vấn đề kinh tế, thời sự và vài mẫu thông tin về các vấn đề văn phòng, một vài mẫu phỏng vấn giữa người đưa tin và người dân. Bộ ngữ liệu này sẽ phù hợp để xây dựng ứng dụng giải các câu hỏi tiếng Anh dạng điền khuyết.

Trong khóa luận tiền nhiệm của đề tài này [1], tác giả đã có khảo sát dựa trên bộ ngữ liệu Open American National Corpus. Tác giả khảo sát 2 lần, với lần 1 là khảo sát trên một nửa bộ ngữ liệu và lần 2 là dựa trên toàn bộ bộ ngữ liệu. Tác giả kết luận độ phụ thuộc của độ lớn của bộ ngữ liệu trên độ chính xác của thuật giải là có ảnh hưởng nhưng không nhiều (chênh lệch với ban đầu 0.1-2.7%).

Đề thi	Tổng số câu	Số câu đúng	Tỉ lệ (%)
Bảng A	800	299	37.36
Bảng B	460	181	39.35
Bảng C	2020	858	42.46
TOELF	820	273	33.29
Đề thi đại học	100	35	35

Bảng 4.6: Kết quả khảo sát lần 1 với một nửa bộ ngữ liệu của khóa luận [1]

Đề thi	Tổng số câu	Số câu đúng	Tỉ lệ (%)
Bảng A	800	317	39.63
Bảng B	460	181	39.35
Bảng C	2020	915	45.30
TOELF	820	278	33.90
Đề thi đại học	100	36	36

Bảng 4.7: Kết quả khảo sát lần 2 với toàn bộ ngữ liệu của khóa luận [1]

Tuy nhiên, từ kết quả khảo sát của khóa luận cho thấy, lần khảo sát thứ 1. Khóa luận sử dụng bộ ngữ liệu nhỏ (462 MB) đã cho về kết quả ở đề thi A, B, C gần bằng với kết quả tốt nhất của khóa luận [1]. Riêng đề thi TOEFL đã cao hơn 5.17%. Sau đó khóa luận khảo sát trên bộ ngữ liệu lớn hơn, cho thấy độ chính xác ở 3-gram cao hơn từ 6-12% so với kết quả tốt nhất từ khóa luận [1].

Có thể nói do quá trình tiền xử lý văn bản trước khi trích xuất dữ liệu của khóa luận và tiền xử lý câu hỏi trước khi truy vấn tốt hơn so với khóa luận tiền nhiệm trước đó giúp độ chính xác cải thiện đáng kể dù chỉ trên bộ ngữ liệu nhỏ. Thêm vào đó, ở đề thi TOEFL là các đề thi mang tính quốc tế cao, sử dụng nhiều từ môi trường làm việc, văn phòng, báo chí. Vì thế bộ ngữ liệu ở statmt.org đã phục vụ tốt cho việc giải câu hỏi từ các đề thi này.

Để khẳng định cho lựa chọn tiền xử lý đã đề ra trước đó, khóa luận đã thực hiện một khảo sát bằng cách thực hiện giải đáp áp trên bộ ngữ liệu được thu thập đến 2007 nhưng loại bỏ stop word bằng ngân hàng stop word SMART [8] và loại bỏ các options tiền xử lý đã nêu trước đó. Kết quả khảo sát cho thấy, độ chính xác giảm rất thấp, thậm chí ở bộ đề C, 3-gram độ chính xác dưới 25% là mức ngẫu nhiên chọn đáp án của câu hỏi.

	1-gram	2-gram	3-gram	4-gram
	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)	Độ chính xác (%)
Bảng A (800 câu)	23.88	24.88	25.13	24.88
Bảng B (460 câu)	26.52	27.83	26.52	26.96
Bảng C (2020 câu)	24.06	25.59	23.96	24.21
TOEFL (820 câu)	29.51	29.88	29.27	29.39
Trung bình	25.99	27.05	26.22	26.36

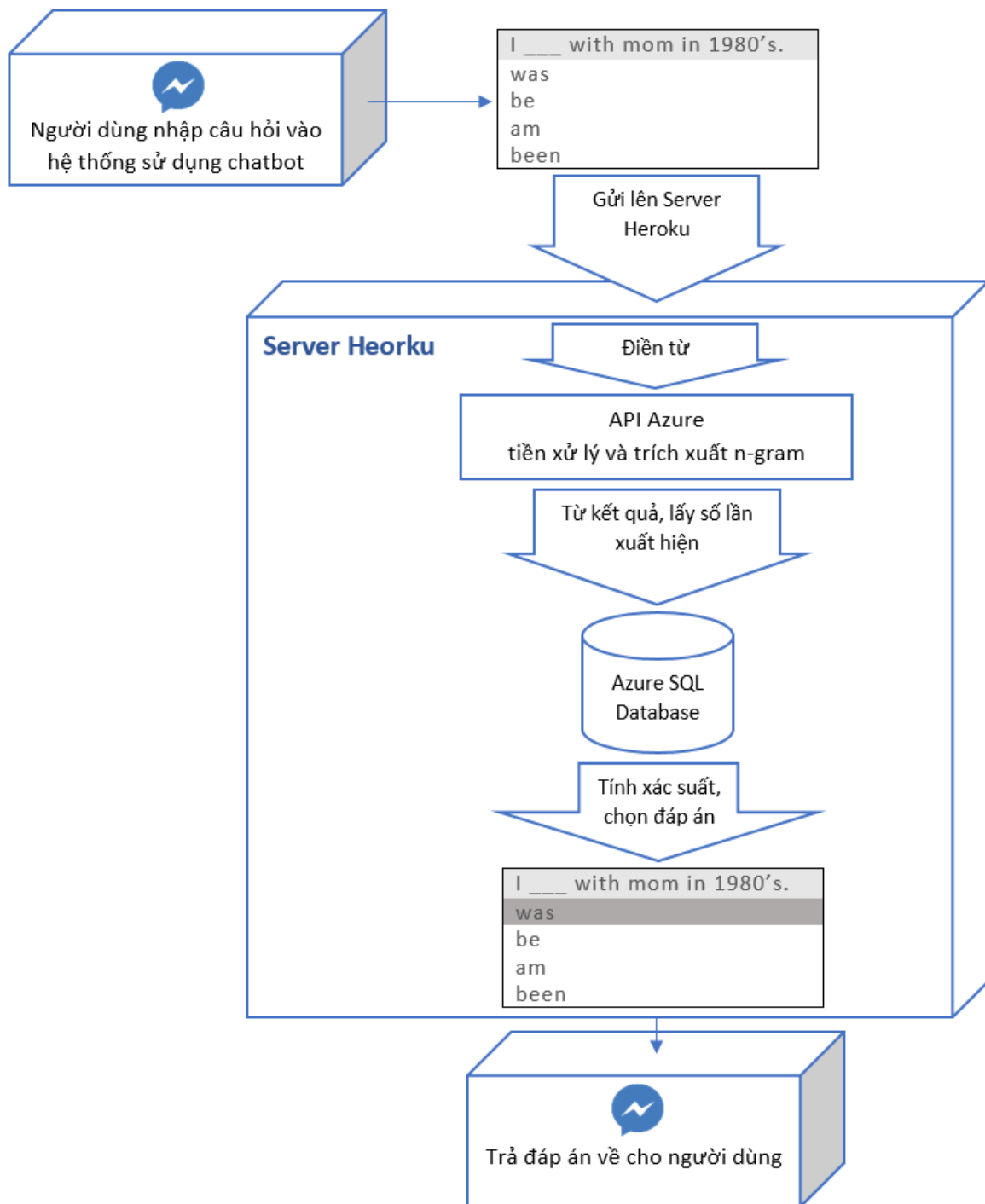
Bảng 4.8: Kết quả khảo sát trên bộ ngữ liệu thu thập đến năm 2007 không tiền xử lý như đã đề ra và loại bỏ stopword

Sau khảo sát, tôi nhận thấy, các câu trả lời bằng 4-gram bị sai thì 3-gram có thể trả lời đúng, và tương tự như thế đối với 3-gram trả lời sai thì 2-gram có thể trả lời đúng. Các câu hỏi mà 4-gram không trả lời được là vì tồn tại 2 hoặc nhiều câu trả lời đưa về xác suất tương tự, khiến hệ thống chọn ngẫu nhiên câu trả lời trong nhiều câu. Vì thế, tôi tiến hành thêm một khảo sát nữa. Đó là trong trường hợp 4-gram trả về nhiều hơn 1 câu trả lời, ta tiếp tục dùng nhiều câu trả lời này gửi xuống 3-gram lấy đáp án. Và tiếp tục như thế cho đến 1-gram. Kết quả được thể hiện ở bảng sau. Nhận thấy xác suất tăng không cao, nhưng đây cũng là một cách tiếp cận tận dụng được vấn đề cần giải quyết là câu hỏi tiếng Anh.

	4, 3, 2 và 1-gram
	Độ chính xác (%)
Bảng A (800 câu)	49.00
Bảng B (460 câu)	46.70
Bảng C (2020 câu)	59.85
TOEFL (820 câu)	42.80
Trung bình	49.59

Bảng 4.9: Kết quả khảo sát trên sử dụng kết hợp 4, 3, 2 và 1-gram trên bộ ngữ liệu 2014

4.5. Hệ thống chương trình



Hình 4.21: Mô tả hệ thống chương trình

Hệ thống chương trình chạy ổn định. Tuy nhiên do sử dụng nhiều công nghệ miễn phí với mức dùng thử nên hệ thống không thực thi nhanh được. Ví dụ với hệ thống Azure SQL, mỗi lệnh truy vấn dữ liệu tiêu tốn nhiều thời gian vì bị giới hạn DTU hoặc hệ thống server Heroku tự động ngắt khi không có tương tác sau 30 phút. Khi có truy cập vào server Heroku trở lại, hệ thống cần 10s để khởi động về trạng thái chuẩn bị nhận tin nhắn từ người dùng.

Để có thể triển khai ứng dụng ra thành ứng dụng cho người dùng sử dụng được, cần tiêu tốn một lượng chi phí để mua thêm băng thông và dung lượng trên cơ sở dữ liệu trong trường hợp sử dụng một bộ ngữ liệu lớn hơn để huấn luyện hệ thống. Và một khoảng chi phí để sử dụng server Heroku luôn chạy với băng thông chấp nhận được.

Ngoài ra, có thể cải thiện thêm việc tương tác với người dùng bằng cách trò chuyện, gợi ý học tiếng Anh, những câu nói chuyện đơn giản mà trong đoạn mã giả của chatbot, khóa luận có nhắc đến việc tương tác với người dùng.

Chương 5. KẾT LUẬN

5.1. Kết luận

Thông qua việc xây dựng chatbot tự động trả lời câu hỏi tiếng Anh dạng điền khuyết, chúng tôi đã thu được các kết quả như sau:

- Nắm được kiến thức cơ bản để giải quyết bài toán xử lý ngôn ngữ tự nhiên dựa trên hướng tiếp cận xác suất thống kê.
- Tìm hiểu được nhiều công trình nghiên cứu trước và gần đây đang giải quyết vấn đề này với các mô hình, thuật giải và độ chính xác khác nhau.
- Cài đặt thành công phương pháp thống kê để giải tự động câu hỏi tiếng Anh dạng điền khuyết.
- Nâng cao khả năng lập trình, linh hoạt chuyển đổi giữa các ngôn ngữ, hệ thống, tăng cao khả năng giải quyết vấn đề nhằm hướng đến giải quyết được vấn đề toàn cục.
- Hiểu và thiết kế, triển khai mô hình hệ thống để phù hợp với ứng dụng.
- Xây dựng được Web Service API Server, chatbot, dựng thí nghiệm trên hệ thống Azure Machine Learning, sử dụng cơ sở dữ liệu Azure SQL.

Tóm lại, kết quả do hệ thống cung cấp dù cao hơn so với khóa luận tiền nhiệm. Khóa luận, thể hiện khả năng ứng dụng của Xử lý ngôn ngữ tự nhiên theo hướng tiếp cận Xác suất thống kê lên giải quyết vấn đề thực tế là giải tự động câu hỏi tiếng Anh dạng điền khuyết. Bên cạnh đó, khóa luận sử dụng hệ thống Azure là cách triển khai phù hợp với chi phí thấp, giảm thiểu rủi ro. Cũng như khóa luận sử dụng nhiều ngôn ngữ lập trình khác nhau với các công nghệ khác nhau nhằm giải quyết vấn đề chung của cả hệ thống.

5.2. Hướng phát triển

Đa số, các câu hỏi mà người dùng muốn hệ thống giải giúp là những câu hỏi khó, có độ phức tạp cao. Trong khi đó, những câu mà hệ thống giải sai thường rơi vào những câu có độ phức tạp cao. Vì thế ta cần một giải thuật mới giúp tăng độ chính

xác cao hơn. Trong thực tế, người dùng còn mong muốn hệ thống có thể giải thích được vì sao hệ thống lại chọn đáp án đó và giải thích cho người dùng hiểu để người dùng có thể trao đổi thêm kiến thức.

TÀI LIỆU THAM KHẢO

- [1] L. Q. Khải, "Xây dựng hệ thống tự động trả lời câu hỏi trắc nghiệm tiếng Anh dạng điền khuyết", Trường Đại học Công nghệ Thông Tin - Đại học Quốc gia thành phố Hồ Chí Minh, Hồ Chí Minh, 2013.
- [2] V. H. a. T. Reuter, "LISGrammarChecker: Language Independent Statistical Grammar Checking," University of Applied Sciences, Hochschule Darmstadt, 2009.
- [3] A. M. Woods, "Exploiting Linguistic Features for Sentence Completion," Carnegie Mellon University, Pittsburgh, PA 15213, USA, 2016.
- [4] D. Choi, M. Hwang, B. Ko and a. P. Kim, "Solving English Questions through Applying Collective Intelligence," Dept. Of Computer Engineering Chosun University; Korea Institute of Science and Technology Information, Gwangju, South Korea; Daejeon, South Korea, 2011.
- [5] E. T. S. b. P. C. Fellbaum, "Assessing the Effectiveness of Corpus-based Methods in Solving SAT Sentence Completion Questions," Independent Work Report Spring, 2015, 2015.
- [6] "ACL Corpora for English," ACL, [Online]. Available: https://aclweb.org/aclwiki/Corpora_for_English.
- [7] "Shared Task: Machine Translation," statmt.org, 17-18 September 2015. [Online]. Available: <http://www.statmt.org/wmt15/translation-task.html>.
- [8] A. McCallum, "Builtin stoplist words (from SMART," 1997. [Online]. Available: <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/naive-bayes/bow-0.8/stopwords.c>.

