

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

CỬ 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

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	10
1.1. Nội dung đề tài	10
1.2. Phát biểu bài toán	12
1.3. Đối tượng và phạm vi nghiên cứu.....	12
Chương 2. TỔNG QUAN.....	13
2.1. Giới thiệu vấn đề.....	13
2.1.1. Các hướng tiếp cận.....	13
2.1.1.1. Hướng tiếp cận theo kiểm tra ngữ pháp.....	13
2.1.1.2. Mô hình n-gram	14
2.1.1.3. Mô hình cây phụ thuộc (Dependency models)	15
2.1.1.4. Continuous Space Models	15
2.1.1.5. PMI Model.....	16
2.1.2. Kiểm nghiệm độ chính xác giữa các hướng tiếp cận.....	16
Chương 3. MÔ HÌNH.....	18
3.1. Cơ sở lý thuyết.....	18
3.1.1. Ngôn ngữ tự nhiên	18
3.1.2. Đặc trưng ngôn ngữ.....	18
3.1.3. Khái niệm token, tokenization, tokenizer.....	18
3.1.4. Mô hình N-gram	19
3.1.4.1. Ngữ liệu	21
3.2. Mô hình đề xuất.....	23
3.3. Thu thập dữ liệu	23
3.4. Giải thuật.....	24

3.4.1.	Tiền xử lý văn bản	24
3.4.2.	Thuật giải chọn đáp án	25
3.5.	Hiện thực.....	27
3.5.1.	Giới thiệu hệ thống Azure	27
3.5.2.	Giới thiệu về Azure Machine Learning.....	30
3.5.3.	Hiện thực mô hình n-grams bằng Azure Machine Learning	31
3.5.3.1.	Trích xuất dữ liệu.....	31
3.5.3.2.	Hiện thực việc chọn đáp án của một câu	40
Chương 4.	CÀI ĐẶT – THỬ NGHIỆM	43
4.1.	Cài đặt.....	43
4.1.1.	Giới thiệu về Heroku	46
4.1.2.	Hiện thực chat bot trên Facebook Messenger	47
4.2.	Mô hình n - gram	48
4.3.	Ngữ liệu	50
4.4.	Hệ thống chương trình	52
Chương 5.	KẾT LUẬN.....	53
5.1.	Kết luận.....	53
5.2.	Hướng phát triển.....	53

DANH MỤC HÌNH VẼ

Hình 3.1: Nội dung của ngữ liệu của statmt.org	24
Hình 3.2: Sơ đồ thể hiện quá trình chọn đáp án.....	27
Hình 3.3: Sơ đồ thể hiện quá trình làm việc với Azure Machine Learning Studio (hình ảnh được cung cấp bởi Microsoft).....	31
Hình 3.4: Sơ đồ thể hiện quá trình làm việc với dữ liệu nhỏ	33
Hình 3.5: Cài đặt cho các module để xử lý trên bộ ngữ liệu nhỏ	34
Hình 3.6: Kết quả đạt được sau quá trình tách dữ liệu	35
Hình 3.7: Gắn thêm module Excute Python Code vào sơ đồ.....	36
Hình 3.8: Kết quả sau khi gắn thêm module Python.....	36
Hình 3.9: Sơ đồ các module để trích xuất n-gram từ bộ ngữ liệu lớn.....	38
Hình 3.10: Sơ đồ các module để đưa dữ liệu vào database Azure SQL.....	39
Hình 3.11: Thử truy xuất lên database đã tạo trên Azure SQL	39
Hình 3.12: Thử truy xuất lên database đã tạo trên Azure SQL	39
Hình 3.12: Sơ đồ các module để thực hiện việc chọn đáp án	40
Hình 3.14: Kết quả sau khi tiền xử lý.....	41
Hình 3.15: Kết quả sau khi trích xuất các n-gram trong câu.....	41
Hình 4.1: Dataset 100 câu hỏi	44
Hình 4.2: Sơ đồ thiết kế hệ thống ứng dụng.....	45
Hình 4.3: Sơ đồ các module để triển khai experiment thành Web Service	46
Hình 4.4: Kết quả sau khi hiện thực chatbot.....	48

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]	17
Bảng 2.2: Kết quả giải đề thi TOEIC dựa trên bộ ngữ liệu n-gram của Google với các n-gram khác nhau. [5]	17
Bảng 3.1: Mẫu câu hỏi	25
Bảng 4.1: Bảng độ chính xác và độ bao phủ của các n-gram trên bộ dataset kiểm nghiệm	44
Bảng 4.2: 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 [5]	49
Bảng 4.3: Kết quả khảo sát lần 1 với một nửa bộ ngữ liệu của khóa luận [1]	51
Bảng 4.4: Kết quả khảo sát lần 1 với toàn bộ ngữ liệu của khóa luận [1]	52

DANH MỤC TỪ VIẾT TẮT

TÓM TẮT KHÓA LUẬN

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, TOEFT, ... 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 phân tích ngữ liệu và xử lý câu hỏi, chọn đáp án trên hệ thống Azure Machine Learning
- Xây dựng Rest Service API Server để phù hợp việc phát triển nhiều ứng dụng trên nhiều nền tảng, thiết bị khác nhau.
- 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
- 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)”. [3]

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*. Dễ trong 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. [2]

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 từ 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. [2]

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, mở rộng từ, ... 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.

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

Ở nghiên cứu [2] 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. [5]

Chương 3. MÔ HÌNH

3.1. Cơ sở lý thuyết

3.1.1. Ngôn ngữ tự nhiên

Ngôn ngữ là công cụ con người giao tiếp với nhau, 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 các từ trong ngôn ngữ để tạo thành một câu có nghĩa. Các ngôn ngữ khác nhau có cấu trúc ngữ pháp khác nhau.

3.1.2. Đặc trưng ngôn ngữ

Mỗi ngôn ngữ khác nhau về hình thái, cấu trúc, 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. Khái niệm token, tokenization, tokenizer

Mỗi thành phần trong câu đều được gọi là một token. Việc tách token ra khỏi câu hay còn gọi là tokenization được thực hiện bởi tokenizer. Tokenization 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. Ví dụ “wouldn’t” có thể sử dụng làm token hoặc tiền xử lý 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.

Tokenization là một quá trình khó, trước khi tokenization, thường văn bản sẽ trải qua một bước tiền xử lý để văn bản sẽ dễ hơn cho tokenizer thực hiện. Quá trình tiền xử lý gồm rất nhiều bước và các cách khác nhau và cũng như tokenization cũng có những cách tiếp cận khác nhau. Các thách thức trong tiền xử lý bao 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, mở rộng từ, thay thế từ về dạng kinh điển, ... và vẫn giữ được hình thái, cấu trúc toàn vẹn của câu.

Tiền xử lý và tokenization 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.

3.1.4. 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 Việt” 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 Việt” 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:

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

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:

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

3.1.4.1. Ngữ liệu

Ngữ liệu, hay còn gọi là corpus là một tập hợp các văn bản có thể là viết hoặc nói dựa trên một ngôn ngữ tự nhiên nhất định. Ngữ liệu thường là dữ liệu đã được số hóa và lưu trên máy tính, máy tính có thể đọc được. Một bộ người liệu thường mang trong mình:

- Các văn bản

- Các metadata để diễn giải lại ý nghĩa của các thẻ hoặc diễn giải tóm gọn lại về văn bản
- Các annotations liên quan đến văn bản

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](#).

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

- **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. Dự án này vẫn còn đang trong quá trình phát triển và dự kiến lúc kết thúc sẽ có hơn 100 triệu từ.
- **Brown Corpus** – Standard Corpus do Present-Day American English xây dựng (còn gọi là Brown Corpus). 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, tất cả các phiên bản này đều giống nhau về nội dung nhưng khác nhau về định dạng và cách tổ chức
- **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, ...

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.

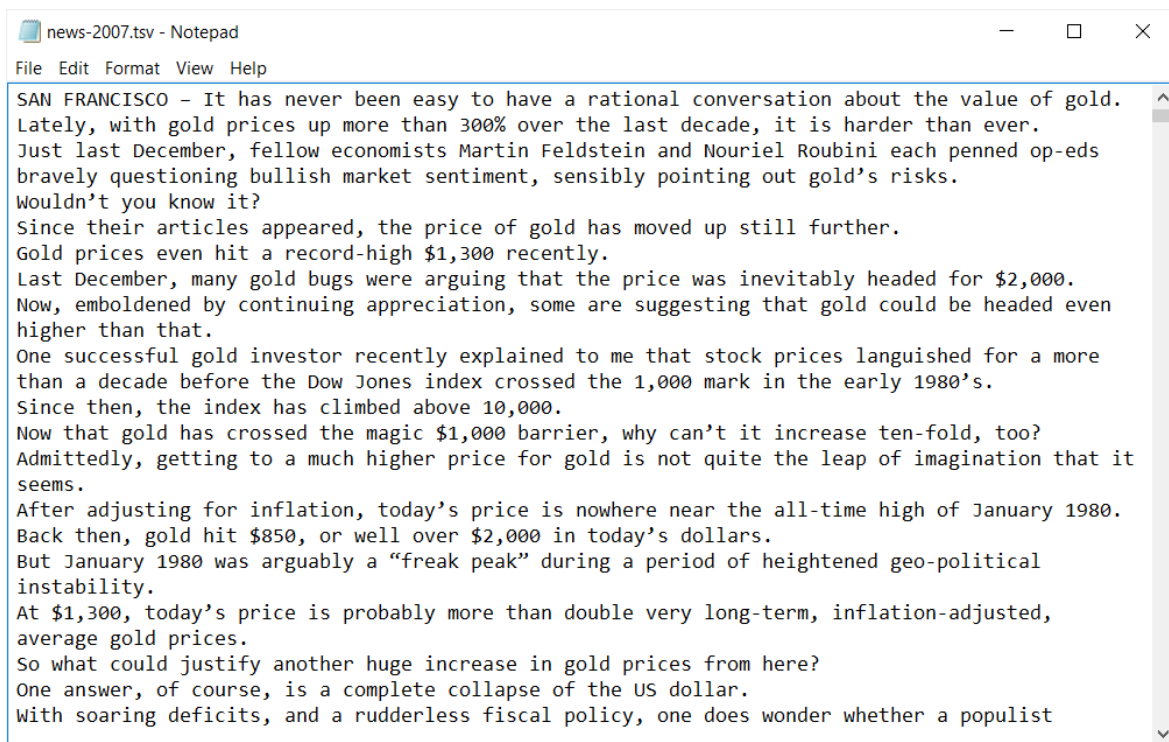
Hệ thống cần một lượng lớn các dữ liệu text mẫu để huấn luyện và phải đảm bảo các dữ liệu mẫu này chứa các câu đúng, gần sát với nội dung thi TOEIC để đảm bảo độ chính xác cao. Các đoạn văn bản này phải được trải qua các công đoạn xử lý như tách câu, tách token, đơn giản hóa từ, mở rộng ngữ pháp cuối động từ, ... 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.

3.3. Thu thập dữ liệu

Bộ ngữ liệu khóa luận sử dụng 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.

Từ bộ ngữ liệu này, 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, ta sử dụng bộ ngữ liệu được thu thập đến năm 2007 (dung lượng 462MB).



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

3.4. Giải thuật

3.4.1. 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”)

- Mở rộng các động từ (Ví dụ: “wouldn’t” thành “would not”)
- ...

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.
- Mở rộng các động từ.

3.4.2. 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, 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.

Công thức tính xác suất tồn tại của câu dựa trên log:

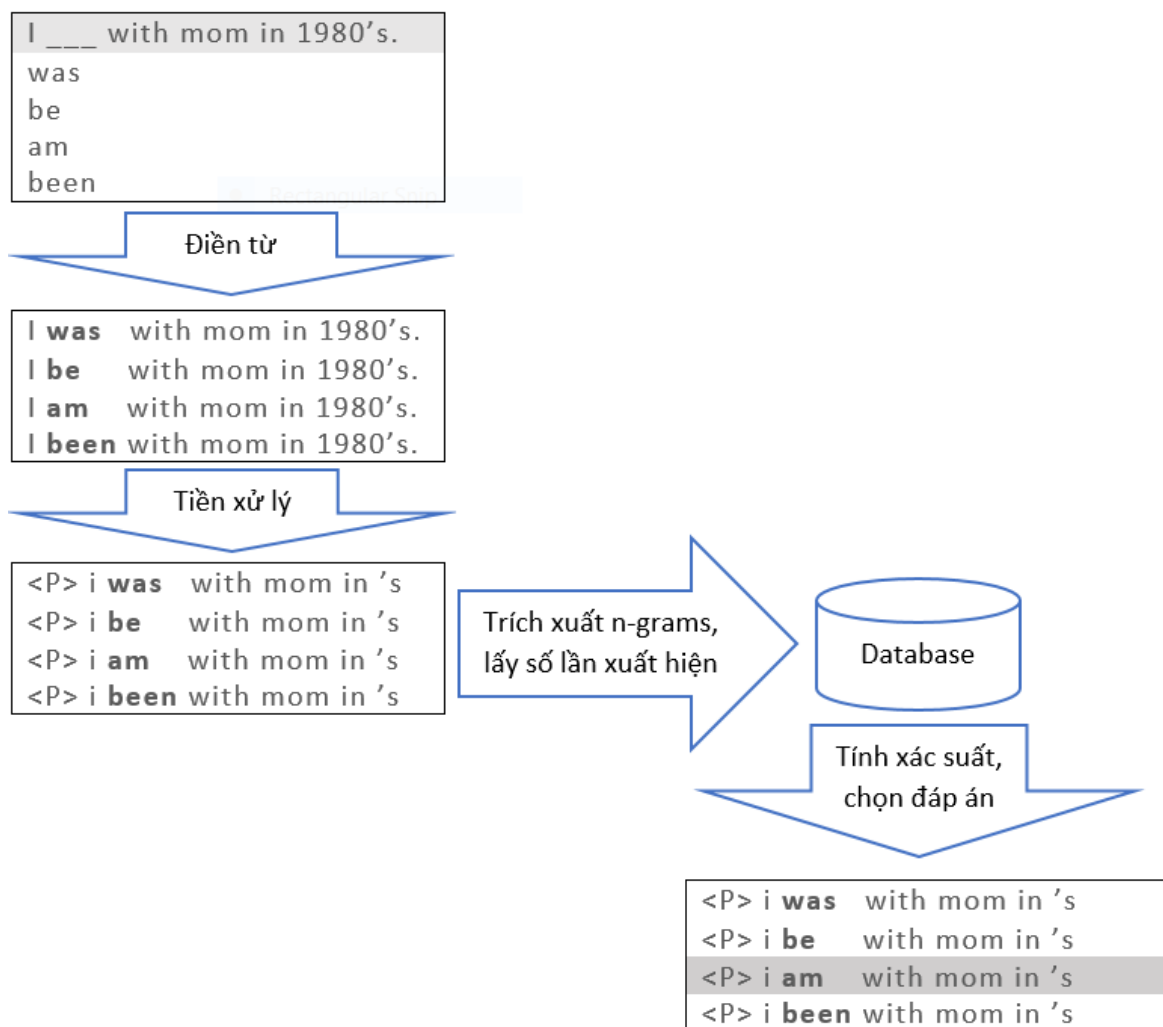
$$P(s_0, s_k) = \sum_{i=0}^k -\log \left(\frac{t_i + 1}{\max(t_0 \dots t_k) + 1} \right)$$

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

$\min(t_0 \dots t_k)$: 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

3.5. Hiện thực

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

3.5.1. Giới thiệu hệ thống Azure

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.

3.5.2. Giới thiệu về Azure Machine Learning

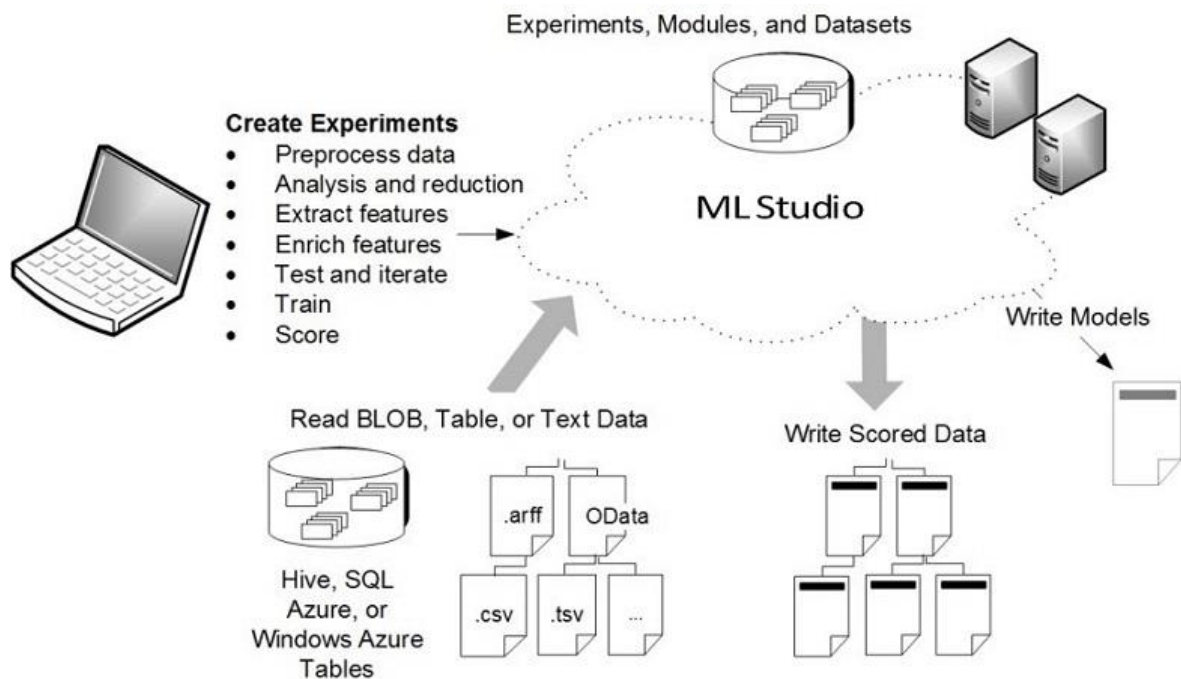
Azure Machine Learning là một hệ thống cho phép phân tích, xử lý, tính toán bằng hệ thống đám mây mà Azure cung cấp. Bên trong Azure Machine Learning có cung cấp sẵn nhiều module có sẵn phù hợp cho các bài toán liên quan đến Machine Learning như *Linear Regression*, *Two Class Regression*, ...

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

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

3.5.3.1. Trích xuất dữ liệu

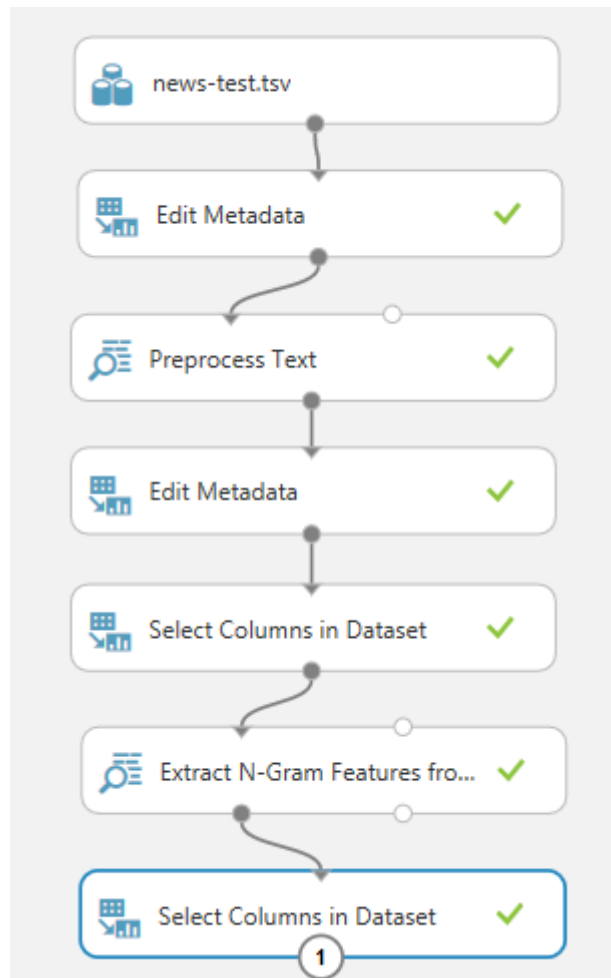
Vì Azure Machine Learning giới hạn về dung lượng cũng như do thời gian thực thi của một module tốn thời gian. Cho nên ta tách bộ ngữ liệu thu được thành

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 3.4: 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 3.5: 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 3.6: 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.

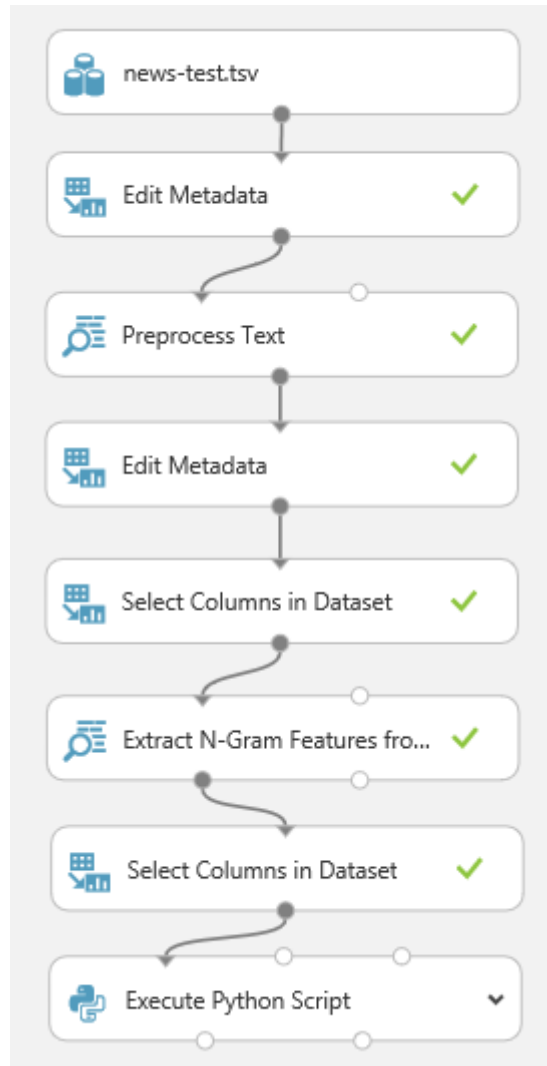
Sum up N gram

```
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 3.7: 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 3.8: 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.

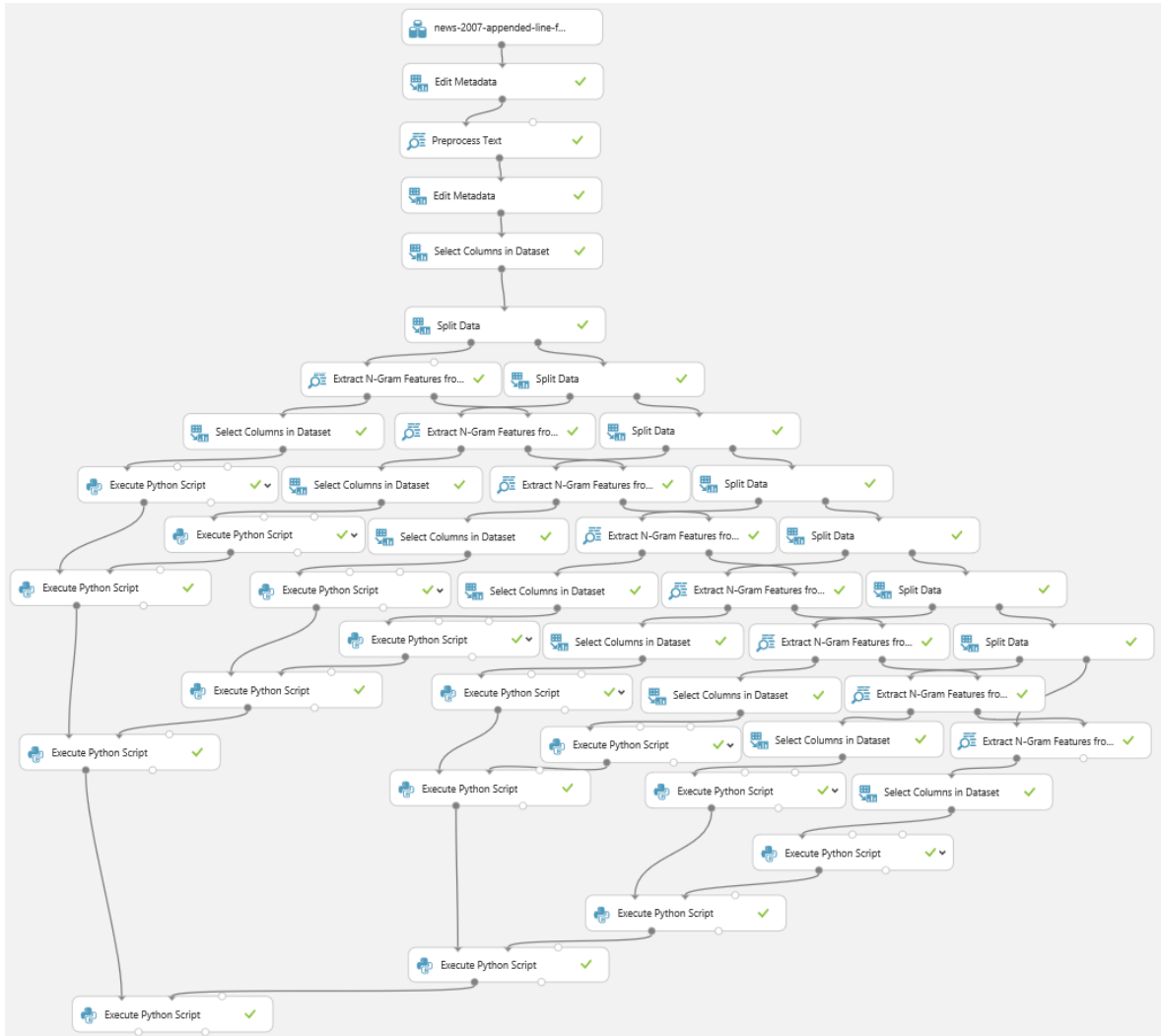
```
string[] readText = đọc file từ một được dẫn
for i: 0 -> readText.Length {
    if (readText[i] kết thúc câu không phải dấu ".") {
        thêm dấu chấm vào cuối câu
    }
}
List<string> newText = new List<string>();
for i: 0 -> readText.Length {
    string newLine = "";
    for j: i -> i + 10000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

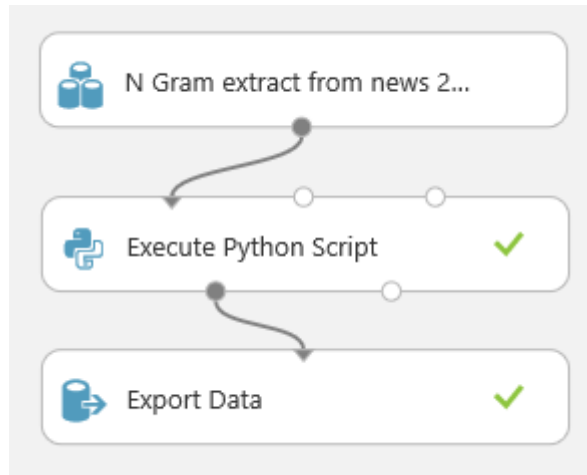
# Param<dataframe1>: a pandas.DataFrame N gram after sum up
# Param<dataframe2>: a pandas.DataFrame N gram after sum up
def azureml_main(dataframe1, dataframe2):
    result = pd.concat([dataframe1, dataframe2], axis = 0)
```

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



Hình 3.9: 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 3.10: 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 3.11: Thử truy xuất lên database đã tạo trên Azure SQL

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

100 % <

Results Messages

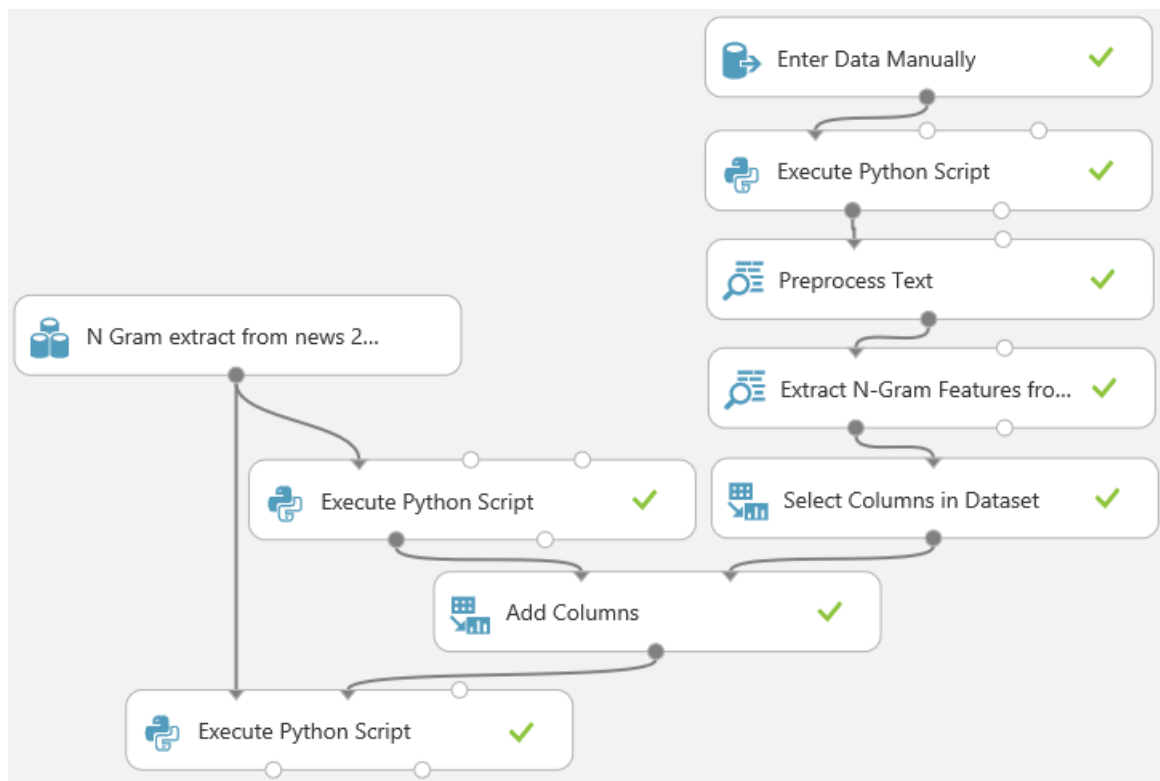
	(No column na...
1	2260378

Hình 3.12: Thử truy xuất lên database đã tạo trên Azure SQL

3.5.3.2. 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 3.13: 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 3.14: 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 3.15: 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 = getTokenCounts(dfFromSQL, tokens);

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

    maxIndex = max(results);
    return maxIndex;

```

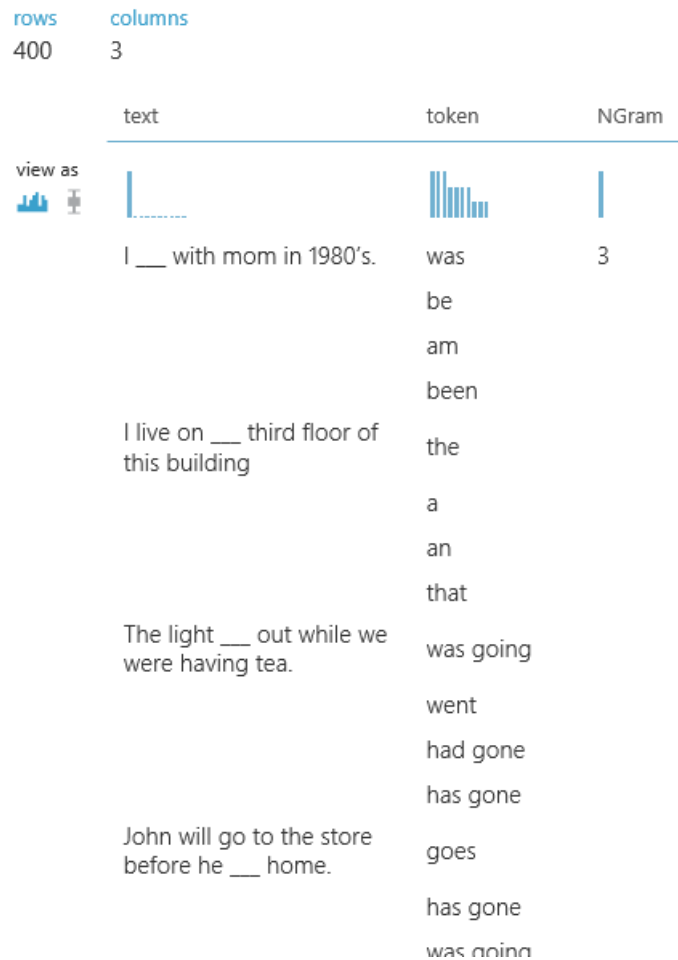
Chương 4. **CÀI ĐẶT – THỬ NGHIỆM**

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 lập ra bộ đề tiếng Anh được lấy nguồn từ internet với 100 câu hỏi trắc nghiệm đủ các lĩnh vực như ngữ pháp, chính tả, mạo từ, ... Sau đó tiến hành cài đặt dựa trên mẫu câu hỏi đã đề ra từ trước, cài đặt lên hệ thống Azure Machine Learning và tiến hành khảo sát.

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

Để kiểm nghiệm độ chính xác, ta tạo thêm một experiment trên project Azure Machine Learning đang thực hiện. Trong đó, ta tạo một dataset chứa 100 câu hỏi với định dạng tương tự với mẫu câu hỏi đã nêu trước đó. Sau đó tiến hành điền từ và tiền xử lý văn bản. Ở đây, khóa luận lấy 100 câu hỏi với mỗi câu hỏi bao gồm 4 đáp án khác nhau để đảm bảo tính thống nhất trong quá trình kiểm tra.



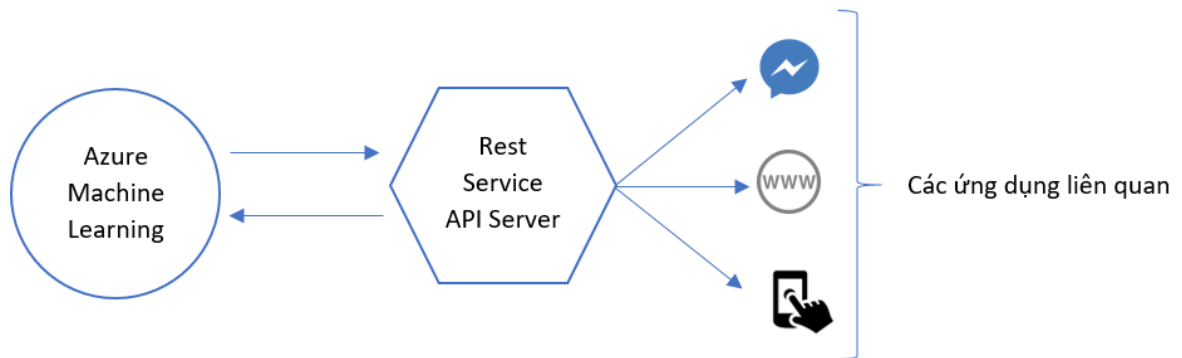
Hình 4.1: Dataset 100 câu hỏi

Sau khi tiền xử lý, và kiểm nghiệm, lấy được kết quả, ta tiến hành so khớp với đáp án để lấy được độ chính xác giữa các n-gram khác nhau và đưa vào bảng khảo sát.

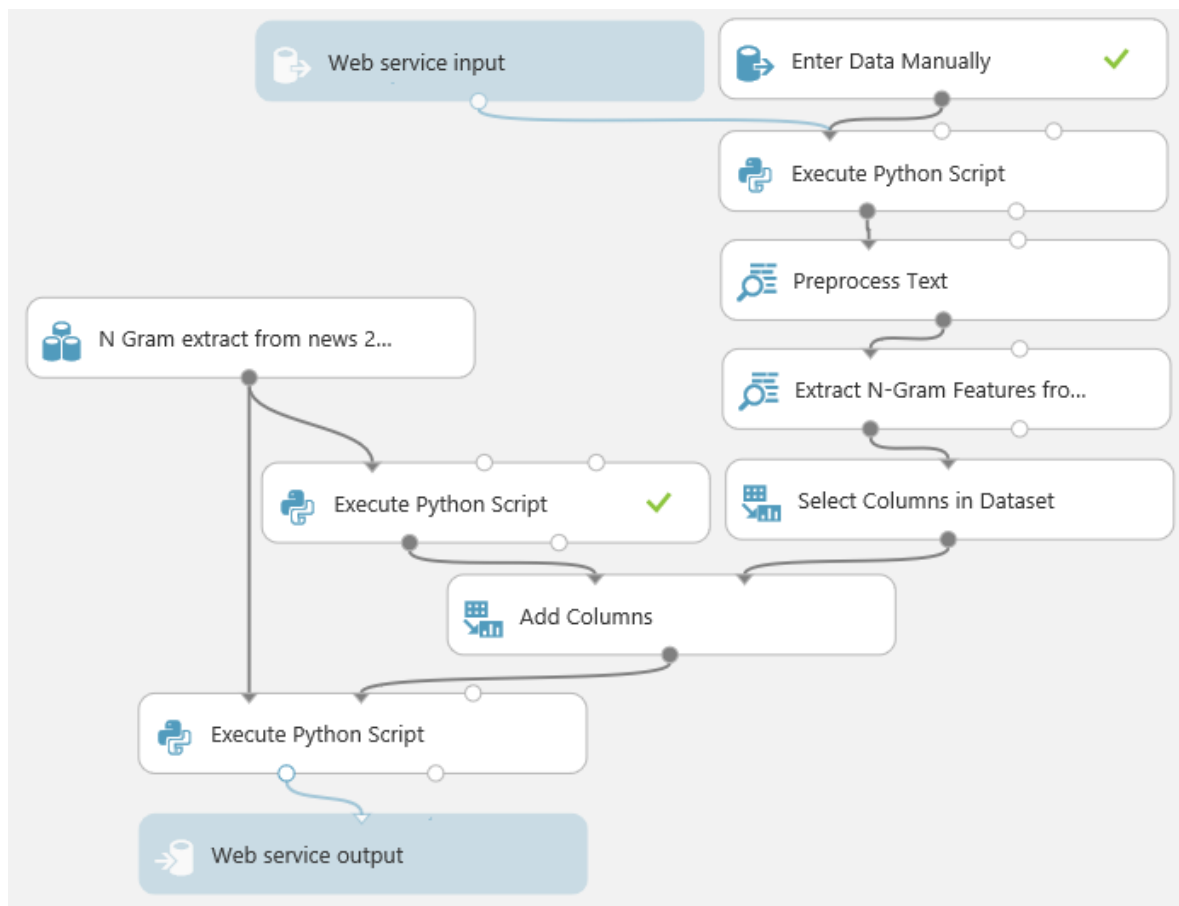
N-gram	Độ chính xác	Độ bao phủ
1-gram	56%	100%
2-gram	64%	100%
3-gram	81%	100%
4-gram	85%	100%

Bảng 4.1: Bảng độ chính xác và độ bao phủ của các n-gram trên bộ dataset kiểm nghiệm

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 4.2: Sơ đồ thiết kế hệ thống ứng dụng



Hình 4.3: Sơ đồ các module để triển khai experiment thành Web Service
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.

4.1.1. 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, ...

4.1.2. Hiện thực chat bot trên Facebook Messenger

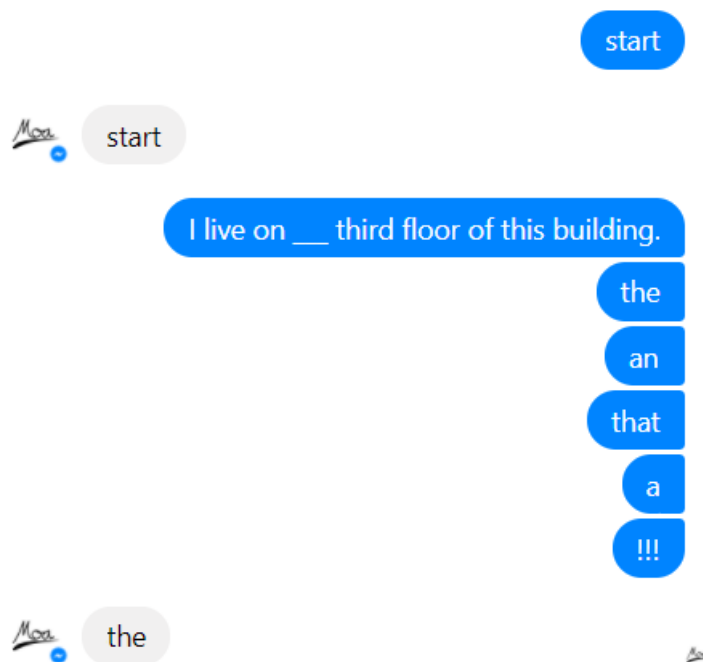
Ngày nay, Facebook đã cho ra mắt Messenger Platform chatbot API giúp cho lập trình viên có thể tự tạo chatbot cho riêng mình. Mô hình chatbot thân thiện với người dùng giúp người dùng dễ tiếp cận với ứng dụng hơn. Ngày nay, một số tổ chức nổi tiếng như trang tin tức CNN, hay một số cửa hàng nhỏ cũng 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 (receiveMessage(sender) == "start")
    listOfAnswer = []
    sentence = receiveMessage(sender)
    nextMessage = receiveMessage(sender)
    while (nextMessage != "!!!")
        listOfAnswer.add(nextMessage)
        nextMessage = receiveMessage(sender)

    answer = getAnswerFromAzure(sentence, listOfAnswer)

    sendReply(sender, answer)
```



Hình 4.4: Kết quả sau khi hiện thực chatbot

4.2. 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. Trong điều kiện khóa luận có thể phát triển thành ứng dụng lớn, ta có thể xem xét sử dụng bộ ngữ liệu mới gần đây nhất (1.7 GB dữ liệu đã nén tức ~>5 GB dữ liệu sau khi giải nén).

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.
- Mở rộng các động từ.

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 [5] 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. 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	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 4.2: 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 [5]

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 [5] 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 thực hiện khảo sát dựa trên 1, 2, 3 và 4-gram, khóa luận nhận thấy độ bao phủ và độ chính xác của 4-gram cao và tốc độ truy xuất phù hợp với ứng dụng, vì thế khóa luận chọn 4-gram để làm cơ sở dữ liệu cho ứng dụng.

4.3. 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, TOELF. 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 tiên 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 tiên 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. Kết quả cho ta thấy toàn độ phụ thuộc của độ lớn của bộ ngữ liệu trên độ chính xác của thuật giải là có ảnh hưởng nhưng không nhiều. Vì thế, tương tự với độ lớn của bộ ngữ liệu mà khóa luận đưa ra vẫn đủ lớn để bao phủ toàn bộ các câu hỏi của các đề thi tiếng Anh.

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 4.3: 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 4.4: Kết quả khảo sát lần 1 với toàn bộ ngữ liệu của khóa luận [1]

4.4. 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 27 giây hoặc hệ thống server Heroku tự động ngắt sau mỗi 30'. 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.

Chương 5. KẾT LUẬN

5.1. Kết luận

Thông qua việc xây dựng hệ thống tự động trả lời câu hỏi tiếng Anh dạng điền khuyết, tôi đã nghiệm 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 Rest 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, tuy nhiên độ chính xác vẫn chưa đủ để có thể đưa ra được ứng dụng thực tế. Tuy nhiên 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

Hiện tại, độ chính xác của hệ thống đưa về là chưa đủ cao để có thể hiện thực thành ứng dụng cho người dùng sử dụng. Vì 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] A. M. Woods, "Exploiting Linguistic Features for Sentence Completion," Carnegie Mellon University, Pittsburgh, PA 15213, USA, 2016.
- [3] V. H. a. T. Reuter, "LISGrammarChecker: Language Independent Statistical Grammar Checking," University of Applied Sciences, Hochschule Darmstadt, 2009.
- [4] M. H. B. K. a. a. P. K. D. Choi, "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] 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.