LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn đến Ban Giám hiệu trường Đại học Công nghệ - Đại học Quốc gia Hà Nội đã tạo cho em môi trường tốt để em có thể học tập và tiếp thu được những kiến thức quý báu trong những năm qua.

Em xin gửi lời cảm ơn sâu sắc đến Thầy giáo, Tiến sĩ Nguyễn Ngọc Cương đã nhiệt tình hướng dẫn, định hướng, hỗ trợ em trong suốt quá trình thực hiện luận văn, giúp đỡ em tiếp cận với cách tư duy, giải quyết và trình bày một vấn đề cần nghiên cứu. Những điều này đã giúp em khắc phục được những hạn chế của bản thân và những khó khăn để hoàn thành luận văn thành công, đúng thời hạn.

Em cũng gửi lời cảm ơn chân thành tới các thầy cô trong trường, đặc biệt các thầy cô trong Khoa Công nghệ thông tin đã giảng dạy em trong suốt thời gian học tập tại trường. Với những kiến thức, bài học có được sẽ là hành trang giúp em tự tin hơn trong công việc, cuộc sống và những mục tiêu trong tương lai.

Tôi cũng xin được cảm ơn tới gia đình, những người thân, các đồng nghiệp và bạn bè đã thường xuyên quan tâm, động viên; cảm ơn Tiến sĩ Ngô Quốc Dũng đã chia sẻ kinh nghiệm, cung cấp các tài liệu hữu ích trong thời gian học tập, nghiên cứu cũng như trong suốt quá trình thực hiện luận văn tốt nghiệp.

Mặc dù em đã cố gắng hoàn thành luận văn bằng tất cả sự nỗ lực và khả năng của mình, nhưng chắc chắn vẫn còn nhiều hạn chế và thiếu sót. Em mong nhận được sự cảm thông và góp ý quý giá từ các thầy cô và các bạn.

Hà Nội, tháng 11 năm 2017

Học viên

Bùi Đức Anh

LỜI CAM ĐOAN

Tôi xin cam đoan, những kiến thức trình bày trong luận văn là do tôi tìm hiểu, nghiên cứu và trình bày lại. Trong quá trình làm luận văn tôi có tham khảo các tài liệu có liên quan và đã ghi rõ nguồn tài liệu tham khảo đó. Những kết quả mới trong luận văn là của riêng tôi, không sao chép từ bất kỳ một công trình nào khác. Nếu có điều gì không trung thực, tôi xin hoàn toàn chịu trách nhiệm.

Tác giả

Bùi Đức Anh

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ
AI	Artificial Intelligence *(Trí tuệ nhân tạo)*
AIML	Artificial Intelligence Markup Language
CKY	Cocke-Kasami-Younger
CNF	Chomsky Normal Form
CFG	Context-Free Grammar *(Văn phạm phi ngữ cảnh)*
DARPA	Defense Advanced Research Projects Agency
DT	Determiner *(Từ hạn định/Từ chỉ định)*
IBM	International Business Machines
IN	Prepostion *(Giới từ)*
NN	Noun *(Danh từ)*
NP	Noun Phrase *(Cụm danh từ)*
PCFGs	Probabilistic Context-Free Grammars *(Văn phạm phi ngữ cảnh hướng thống kê)*
PP	Prepositional Phrase *(Cụm giới từ)*
S	Sentence *(Câu)*
SCFG	Stochastic Context-Free Grammar
Vi	Intrasitive Verb *(Nội động từ)*
VP	Verb Phrase *(Cụm động từ)*
Vt	Transitive Verb *(Ngoại động từ)*

DANH MỤC CÁC HÌNH ẢNH

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Trong nhiều năm trở lại đây, với nhu cầu về hội nhập ngày càng cao giữa các quốc gia cả về kinh tế lẫn văn hóa, yêu cầu về ngoại ngữ, đặc biệt là tiếng Anh, trở thành một vấn đề cấp thiết với mỗi người. Nhưng nhiều người không có đủ thời gian cũng như điều kiện để tham gia các lớp học thêm hoặc các câu lạc bộ để nâng cao trình độ của mình. Chính vì vậy cần có những phần mềm, công cụ để hỗ trợ người học tiếng Anh ở bất cứ nơi đâu, vào bất cứ thời gian nào. Hiện nay đã có nhiều công cụ hướng tới mục đích đó, song mỗi công cụ, phần mềm đều có những hạn chế riêng, đặc biệt là tính thụ động. Người học hầu như chỉ tham gia vào các hoạt động được thiết kế từ trước trên công cụ, ít có sự tương tác hai chiều.

Với sự phát triển của khoa học công nghệ, việc mỗi người sở hữu cho mình một chiếc điện thoại thông minh hiện nay là rất phổ biến. Dựa trên nền tảng điện thoại thông minh mà đã và đang xuất hiện các ứng dụng hỗ trợ học tiếng Anh có sự tương tác cao giữa người và máy; một trong những ứng dụng phổ biến nhất hiện nay có thể kể tới là các ứng dụng dựa trên Chatbot. Tuy nhiên, phần lớn các ứng dụng Chatbot mới tập trung vào phần từ vựng, từ điển, trắc nghiệm... hoặc các ứng dụng luyện kỹ năng nghe, kỹ năng đọc..., có rất ít các ứng dụng có thể giúp người dùng kiểm tra chính tả cũng như cú pháp của câu để chỉ ra lỗi sai cho người đọc và chỉnh sửa nó, mặc dù đây là một bài toán tương đối quan trọng.

Vì vậy, Chatbot đáp ứng được các yêu cầu, chạy trên điện thoại thông minh để hỗ trợ người học tiếng Anh sẽ là một giải pháp có hiệu quả để nâng cao chất lượng học tập tiếng Anh.

Chính vì lý do đó, tác giả đã lựa chọn đề tài: *"Nghiên cứu mô hình PCFGs và ngôn ngữ AIML trong xây dựng chatbot hỗ trợ học tiếng Anh"*.

2. Mục tiêu nghiên cứu

Nghiên cứu cơ sở lý thuyết nền tảng của bài toán kiểm tra chính tả và cú pháp của câu trong tiếng Anh; ứng dụng cài đặt, đánh giá giải thuật và xây dựng một ứng dụng hỗ trợ các tính năng như kiểm tra chính tả, ngữ pháp, cú pháp thông qua hội thoại giữa người dùng và máy trên nền tảng Android.

3. Đối tượng và phạm vi nghiên cứu của đề tài

Đối tượng nghiên cứu

- Khái quát về trí tuệ nhân tạo

- Mô hình PCFGs, ứng dụng xây dựng cây cú pháp

- Ngôn ngữ AIML và kỹ thuật xây dựng chatbot

Phạm vi nghiên cứu

Chatbot trên điện thoại thông minh sử dụng hệ điều hành Android được xây dựng dựa trên AIML và mô hình PCFGs có khả năng thực hiện hội thoại với người dùng, phát hiện và sửa những lỗi chính tả và cú pháp.

4. Phương pháp nghiên cứu

- Khảo sát, phân tích và hệ thống hóa nội dung các tài liệu khoa học liên quan đến chatbot hỗ trợ học tiếng Anh

- Đối sánh nội dung nghiên cứu của đề tài với các nội dung nghiên cứu đã thực hiện để vừa phát triển áp dụng các kết quả khoa học - công nghệ đã có cho đề tài vừa tìm ra các nội dung mới cần được nghiên cứu và thi hành.

- Thiết kế mô hình và thực nghiệm đánh giá các kỹ thuật, bài toán đã đề xuất để chứng minh tính hiệu quả.

5. Ý nghĩa khoa học, ý nghĩa thực tiễn của đề tài

Ý nghĩa khoa học

- Nghiên cứu, nắm vững về trí tuệ nhân tạo và ngôn ngữ AIML

- Vận dụng trí tuệ nhân tạo để tạo ra sự giao tiếp thân thiện, gần gũi giữa người và máy tính

- Tìm hiểu về chatbot và ứng dụng chatbot để cung cấp thông tin

Ý nghĩa thực tiễn

- Tạo ra được công cụ hỗ trợ học tiếng Anh theo hình thức hội thoại giữa người và máy

- Giúp phát hiện và sửa những lỗi thường gặp về chính tả và cú pháp trong quá trình giao tiếp (viết, nói) bằng tiếng Anh.

- Nâng cao hiệu quả học tiếng Anh.

6. Kết cấu luận văn

- Chương 1: Các vấn đề tổng quan: Giới thiệu tổng quan lý thuyết về trí tuệ nhân tạo, xu hướng phát triển của trí tuệ nhân tạo, lĩnh vực xây dựng chatbot hỗ trợ học tiếng Anh, bài toán phân tích cú pháp, kiểm tra chính tả, ngữ pháp và các vấn đề liên quan.

- Chương 2: Mô hình PCFGs và ngôn ngữ AIML: Nghiên cứu văn phạm phi ngữ cảnh, tính mập mờ trong phân tích cú pháp và đề xuất giải pháp sử dụng văn phạm phi ngữ cảnh hướng thống kê PCFGs; nghiên cứu mã nguồn mở AIML trong xây dựng chatbot.

- Chương 3: Phân tích thiết kế, cài đặt ứng dụng: Trình bày cơ bản về thiết kế của ứng dụng và kết quả đạt được thông qua một số mẫu kiểm thử.

- Kết luận: Trình bày điểm mạnh và hạn chế trong luận văn. Đồng thời nêu ra hướng phát triển tiếp theo trong tương lai.

CHƯƠNG 1: CÁC VẤN ĐỀ TỔNG QUAN

Chương 1 của luận văn giới thiệu tổng quan về trí tuệ nhân tạo, xu hướng phát triển của trí tuệ nhân tạo, lĩnh vực xây dựng chatbot hỗ trợ học tiếng Anh và bài toán phân tích cú pháp, kiểm tra chính tả, ngữ pháp và các vấn đề liên quan.

1.1. Chatbot

1.1.1. Trí tuệ nhân tạo

1.1.1.1 Định nghĩa

Khái niệm về trí tuệ nhân tạo (Artificial Intelligence - viết tắt là AI) có thể được nhìn nhận theo nhiều cách khác nhau, chưa có định nghĩa nào được thừa nhận chung. Trên thế giới hiện có nhiều định nghĩa về trí tuệ nhân tạo, cụ thể:

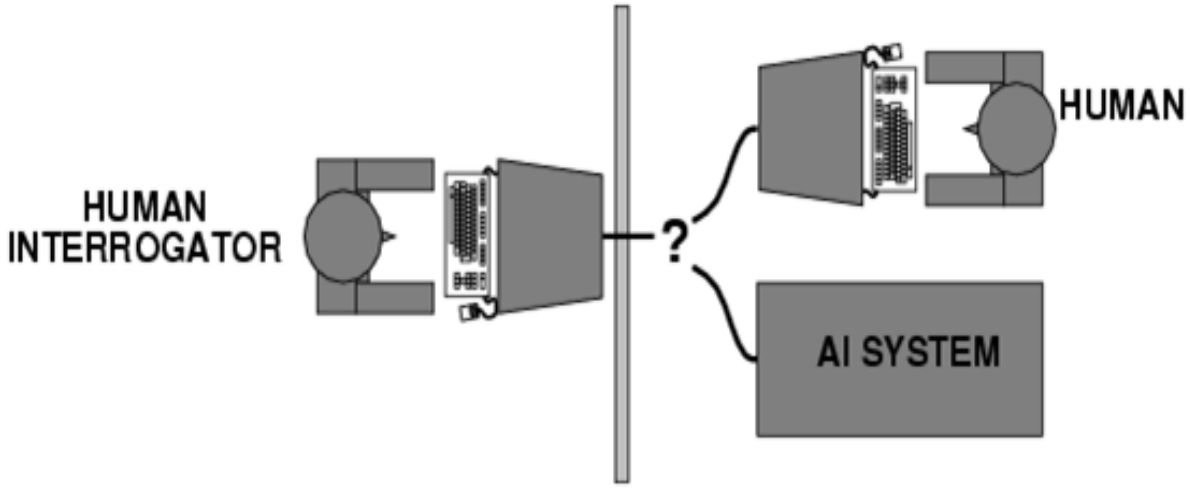
- Theo Wikipedia, trí tuệ nhân tạo là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo.
- Bellman (1978) định nghĩa: trí tuệ nhân tạo là tự động hoá các hoạt động phù hợp với suy nghĩ con người, chẳng hạn các hoạt động ra quyết định, giải bài toán...
- Rich anh Knight (1991) thì cho rằng: Trí tuệ nhân tạo là khoa học nghiên cứu xem làm thế nào để máy tính có thể thực hiện những công việc mà hiện con người còn làm tốt hơn máy tính.
- Winston (1992) cho rằng trí tuệ nhân tạo là lĩnh vực nghiên cứu các tính toán để máy có thể nhận thức, lập luận và tác động.
- Nilsson (1998): trí tuệ nhân tạo nghiên cứu các hành vi thông minh mô phỏng trong các vật thể nhân tạo.

Mỗi khái niệm, định nghĩa đều có điểm đúng riêng, nhưng để đơn giản chúng ta có thể hiểu trí tuệ nhân tạo là một ngành khoa học máy tính. Nó xây dựng trên một nền tảng lý thuyết vững chắc và có thể ứng dụng trong việc tự động hóa các hành vi thông minh của máy tính; giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi[18].

1.1.1.2. Quá trình hình thành và phát triển

Ý tưởng xây dựng một chương trình AI xuất hiện lần đầu vào tháng 10/1950, khi nhà bác học người Anh Alan Turing xem xét vấn đề "liệu máy tính có khả năng suy nghĩ hay không?". Để trả lời câu hỏi này, ông đã đưa ra khái niệm

"phép thử bắt chước" mà sau này người ta gọi là "phép thử Turing". Phép thử được thực hiện dưới dạng một trò chơi. Theo đó, có ba đối tượng tham gia trò chơi (gồm hai người và một máy tính). Một người (người thẩm vấn) ngồi trong một phòng kín tách biệt với hai đối tượng còn lại. Người này đặt các câu hỏi và nhận các câu trả lời từ người kia (người trả lời thẩm vấn) và từ máy tính. Cuối cùng, nếu người thẩm vấn không phân biệt được câu trả lời nào là của người, câu trả lời nào là của máy tính thì lúc đó có thể nói máy tính đã có khả năng "suy nghĩ" giống như người[18].



Hình 1.1. Mô hình Turing Test

Năm 1956, tại Hội nghị do Marvin Minsky và John McCarthy tổ chức với sự tham dự của vài chục nhà khoa học tại trường Dartmouth (Mỹ), tên gọi "Artificial Intelligence" được chính thức công nhận và được sử dụng cho đến ngày nay. Cũng tại đây, bộ môn nghiên cứu trí tuệ nhân tạo đầu tiên đã được thành lập.

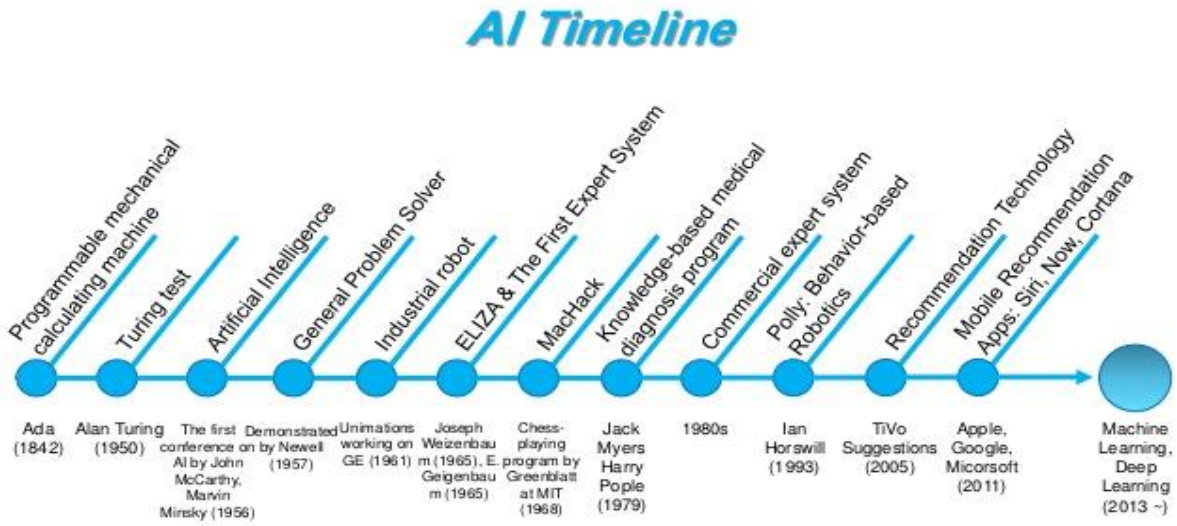
Những năm sau đó, các nhà khoa học như John McArthy, Marvin Minsky, Allen Newell và Herbert Simon cùng với những cộng sự đã viết nên những chương trình máy tính giải được những bài toán đại số, chứng minh các định lý và nói được tiếng Anh. Một số thành tựu ban đầu của giai đoạn này có thể kể đến như: chương trình chơi cờ của Samuel; chương trình lý luận Logic của Newell & Simon; chương trình chứng minh các định lý hình học của Gelernter[18].

Trong các thập kỷ tiếp theo, lĩnh vực trí tuệ nhân tạo đã trải qua những thăng trầm khi một số vấn đề nghiên cứu tỏ ra khó khăn hơn so với dự kiến và một số nghiên cứu đã chứng minh không thể vượt qua với các công nghệ tại thời điểm đó[16]. Tuy nhiên, trí tuệ nhân tạo cũng đã đạt được không ít những thành tựu đáng kể.

Thập kỷ 60, 70 Joel Moses viết chương trình toán học Macsyma sử dụng cơ sở tri thức đầu tiên thành công. Marvin Minsky và Seymour Papert đưa ra các

chứng minh đầu tiên về giới hạn của các mạng nơ-ron đơn giản. Ngôn ngữ lập trình logic Prolog ra đời và được phát triển bởi Alain Colmerauer. Ted Shortliffe xây dựng thành công một số hệ chuyên gia đầu tiên trợ giúp chẩn đoán trong y học, các hệ thống này sử dụng ngôn ngữ luật để biểu diễn tri thức và suy diễn.

Vào đầu những năm 1980, những nghiên cứu thành công liên quan đến AI như các hệ chuyên gia (expert systems) - một dạng của chương trình AI mô phỏng tri thức và các kỹ năng phân tích của một hoặc nhiều chuyên gia con người. Đến những năm 1990 và đầu thế kỷ 21, AI đã đạt được những thành tựu to lớn nhất, AI được áp dụng trong logic, khai phá dữ liệu, chẩn đoán y học và nhiều lĩnh vực ứng dụng khác trong công nghiệp. Sự thành công dựa vào nhiều yếu tố: tăng khả năng tính toán của máy tính, tập trung giải quyết các bài toán con cụ thể, xây dựng các mối quan hệ giữa AI và các lĩnh vực khác giải quyết các bài toán tương tự, và một sự chuyển giao mới của các nhà nghiên cứu cho các phương pháp toán học vững chắc và chuẩn khoa học chính xác[5][15].



Hình 1.2. Quá trình hình thành và phát triển của trí tuệ nhân tạo

Ngày nay, sự phát triển mạnh mẽ của khoa học công nghệ, cùng những bộ dữ liệu phong phú, các công cụ phát triển phần mềm miễn phí hoặc giá rẻ đã hỗ trợ rất nhiều cho các nhà nghiên cứu. Từ đó thúc đẩy sự phát triển các nghiên cứu về trí tuệ nhân tạo, giúp cho mảnh đất AI thu hút đông đảo các ông lớn như: Facebook, Google, Microsoft... tham gia nghiên cứu, phát triển sản phẩm và mở ra kỷ nguyên mới cho trí tuệ nhân tạo.

1.1.1.3. Một số ứng dụng

Hiện tại, trí tuệ nhân tạo được ứng dụng trong đời sống theo hai hướng: Dùng máy tính để bắt chước quá trình xử lý của con người và thiết kế những máy tính thông minh độc lập với cách suy nghĩ của con người.

Một số ứng dụng của trí tuệ nhân tạo trong cuộc sống thực tiễn có thể kể đến như: nhận dạng chữ viết, nhận dạng tiếng nói, dịch tự động, tìm kiếm thông tin, khai phá dữ liệu và phát triển tri thức, lái xe tự động, robot[18].

Nhận dạng chữ viết

Nhận dạng chữ viết ứng dụng trong lĩnh vực nhận dạng chữ in hoặc chữ viết tay và lưu thành văn bản điện tử. Ở Việt Nam, phần mềm VnDOCR do Phòng Nhận dạng & Công nghệ tri thức, Viện Công nghệ Thông tin xây dựng có thể nhận dạng trực tiếp tài liệu bằng cách quét thông qua máy scanner thành các tệp ảnh, chuyển đổi thành các tệp có định dạng *.doc, *.xls, *.txt, *.rtf, giúp người sử dụng không phải gõ lại tài liệu vào máy. Tương tự với phần mềm nhận dạng chữ viết trong thư viện, người ta cũng có thể dễ dàng chuyển hàng ngàn đầu sách thành văn bản điện tử một cách nhanh chóng[14].

Nhận dạng tiếng nói

Nhận dạng tiếng nói đóng vai trò quan trọng trong giao tiếp giữa người và máy. Nó giúp máy móc hiểu và thực hiện các hiệu lệnh của con người. Một ứng dụng trong lĩnh vực này là hãng sản xuất xe hơi BMW (Đức) đang tiến hành phát triển một công nghệ mới cho phép các tài xế có thể soạn email, tin nhắn bằng giọng nói trong khi đang lái xe. Một ứng dụng khác là phần mềm lồng phụ đề vào các chương trình truyền hình. Đây là một công việc khá buồn tẻ và đòi hỏi phải có những người ghi tốc ký chuyên nghiệp. Nhờ có những tiến bộ trong công nghệ nhận dạng tiếng nói, các nhà cung cấp dịch vụ truyền hình gần đây đã gia tăng đáng kể số lượng các chương trình được lồng phụ đề của họ[14].

Dịch tự động

Dịch tự động là công việc thực hiện dịch một ngôn ngữ sang một hoặc nhiều ngôn ngữ khác, không có sự can thiệp của con người trong quá trình dịch. Tuy nhiên, để làm cho máy hiểu được ngôn ngữ là một trong những vấn đề khó nhất của trí tuệ nhân tạo. Thí dụ câu: "ông già đi nhanh quá" cũng có nhiều cách hiểu khác nhau: với cách phân tách từ và cụm từ thành ông già/đi/nhanh quá và ông/già đi/nhanh quá… thì việc dịch câu kiểu như thế này từ tiếng Việt sang tiếng Anh đòi hỏi máy không những phải hiểu đúng nghĩa câu tiếng Việt mà còn phải tạo ra

được câu tiếng Anh tương ứng. Các phần mềm dịch tự động hiện nay còn phải tiếp tục nghiên cứu nhiều hơn nữa để có được những hệ dịch tốt[19].

Tìm kiếm thông tin

Thông tin trên mạng hàng ngày được gia tăng theo cấp số nhân. Việc tìm kiếm thông tin mà người dùng quan tâm bây giờ là tìm đúng thông tin mình cần và phải đáng tin cậy. Theo thống kê, có đến hơn 90% số lượng người Việt Nam lên mạng internet để thực hiện việc tìm kiếm thông tin. Các máy tìm kiếm (search engine) hiện nay chủ yếu thực hiện tìm kiếm dựa theo từ khóa. Thí dụ, Google hay Yahoo chỉ phân tích nội dung một cách đơn giản dựa trên tần suất của từ khoá, thứ hạng của trang và một số tiêu chí đánh giá khác. Kết quả là rất nhiều tìm kiếm không nhận được câu trả lời phù hợp, thậm chí bị dẫn tới một liên kết không liên quan gì do thủ thuật đánh lừa nhằm giới thiệu sản phẩm hoặc lại nhận được quá nhiều tài liệu không phải thứ ta mong muốn, trong khi đó lại không tìm ra tài liệu cần tìm. Hiện nay, các nhà nghiên cứu đang cải tiến các công cụ tìm kiếm trực tuyến để một ngày nào đó, nó có thể hiểu và trả lời cả những câu hỏi cụ thể, thí dụ như "giá tour du lịch rẻ nhất từ Hà Nội đi Đà Lạt trong ba ngày của tháng này là bao nhiêu?". Tuy vậy, thực tế cho đến bây giờ chưa có máy tìm kiếm nào có thể làm hài lòng người dùng kiểu như vậy[14].

Khai phá dữ liệu và phát hiện tri thức

Đây là lĩnh vực cho phép xử lý từ rất nhiều dữ liệu khác nhau để phát hiện ra tri thức mới. Ngoài ra, ứng dụng trong lĩnh vực này cũng cần phải biết trả lời câu hỏi của người sử dụng chúng từ việc tổng hợp dữ liệu thay vì máy móc chỉ đáp trả những gì có sẵn trong bộ nhớ. Thực tế để làm được điều này rất khó, nó gần như là mô phỏng quá trình học tập, khám phá khoa học của con người. Ngoài ra, dữ liệu thường có số lượng rất lớn, với nhiều kiểu (số, văn bản, hình ảnh, âm thanh, video...) và không ngừng thay đổi. Để tìm ra tri thức thì các chương trình phải đối mặt với vấn đề độ phức tạp tính toán... Đây là lĩnh vực vẫn còn đang trong giai đoạn đầu phát triển[14].

Lái xe tự động

Theo Sebastian Thrun, Giáo sư ngành máy tính và kỹ thuật điện của Đại học Carnegie Mellon: ưu điểm lớn nhất của xe tự lái là khả năng loại bỏ sai sót của con người - nguyên nhân dẫn đến 95% số vụ tử vong mỗi năm tại Mỹ do tai nạn giao thông. "Chúng tôi có thể giảm bớt 50% số vụ tai nạn do nguyên nhân này", ông Sebastian Thrun khẳng định. Chế tạo được ôtô tự lái và an toàn cao cũng là một mục tiêu được Cục nghiên cứu các dự án công nghệ cao Bộ quốc

phòng Mỹ DARPA (Defense Advanced Research Projects Agency) khởi xướng và hỗ trợ dưới dạng một cuộc thi mang tên "thách thức lớn của DARPA" (DARPA grand challenge). Chúng ta hy vọng sẽ đến một ngày, những chiếc ôtô chạy trên đường không cần người lái. Chỉ nói nơi muốn đến, xe sẽ đưa ta đi và đi an toàn[14].

Robot

Nhiều đề án nghiên cứu về robot thông minh và các lĩnh vực liên quan được ứng dụng trong đời sống. Các đề án này hướng đến các sáng tạo công nghệ có nhiều ý nghĩa trong văn hóa, xã hội và công nghiệp, đòi hỏi phải tích hợp nhiều công nghệ, như nguyên lý các tác tử, biểu diễn tri thức về không gian, nhận biết chiến lược, lập luận thời gian thực, nhận dạng và xử lý các chuỗi hình ảnh liên tục trong thời gian thực... Một trong những ứng dụng đó là đề án RoboCup: tổ chức thi đấu bóng đá giữa các đội robot. Mục tiêu hướng đến của đề án này là đến năm 2050, sẽ chế tạo được một đội robot có thể thắng đội bóng đá vô địch thế giới. Ứng dụng quan trọng khác của lĩnh vực này là chế tạo robot đối phó và dò tìm nạn nhân trong các thảm họa. Trong sự cố hư hỏng tại nhà máy điện hạt nhân xảy ra sau trận động đất và sóng thần ngày 11 tháng 3 năm 2011 ở Nhật Bản, người ta gửi robot có tên Quince để hoạt động tại những khu vực khó tiếp cận do độ phóng xạ cao của nhà máy Fukushima. Được điều khiển từ xa, Quince có thể làm việc trong nhiều giờ đồng hồ để chụp hình và đo độ phóng xạ trong những tòa nhà bị lây nhiễm chất phóng xạ, nơi mà các kỹ thuật viên không thể vào bên trong[14].

Trong tương lai, trí tuệ nhân tạo với sự quan tâm và phát triển của các ông lớn trong ngành công nghệ, dự kiến sẽ mở rộng hơn nữa phạm vi ứng dụng sang các lĩnh vực như: y tế, xây dựng, ngân hàng, công nghệ siêu vi...

Đến nay, trí tuệ nhân tạo đã góp phần không nhỏ trong việc giúp con người tiết kiệm sức lao động, đẩy nhanh quá trình tự động hóa và số hóa nền kinh tế của nhân loại, với chi phí khá rẻ. Mặc dù, vẫn có nhiều ý kiến lo ngại về công ăn việc làm của con người khi trí tuệ nhân tạo phát triển. Nhưng thiết nghĩ, nếu chúng ta có những chính sách phù hợp thì trí tuệ nhân tạo sẽ là nền tảng đưa loài người bước lên một tầm cao mới.

1.1.1.4. Xu thế nghiên cứu và phát triển của trí tuệ nhân tạo hiện đại

Cho đến thời điểm chuyển giao thiên niên kỷ, sự lôi cuốn của trí tuệ nhân tạo chủ yếu ở hứa hẹn cung cấp của nó, nhưng trong hơn mười lăm năm qua, nhiều lời hứa đó đã được thực hiện. Các công nghệ trí tuệ nhân tạo đã thâm nhập vào cuộc sống của chúng ta. Khi chúng trở thành một lực lượng trung tâm trong xã

hội, lĩnh vực này đang chuyển từ những hệ thống chỉ đơn giản là thông minh sang chế tạo các hệ thống có nhận thức như con người và đáng tin cậy.

Một số yếu tố đã thúc đẩy cuộc cách mạng trí tuệ nhân tạo. Quan trọng nhất trong số đó là sự trưởng thành của máy học, được hỗ trợ một phần bởi nguồn tài nguyên điện toán đám mây và thu thập dữ liệu rộng khắp dựa trên web. Máy học đã đạt tiến bộ đáng kể bằng "học sâu", một dạng đào tạo các mạng lưới thần kinh nhân tạo thích nghi sử dụng phương pháp gọi là lan truyền ngược. Bước nhảy vọt này trong việc thực hiện các thuật toán xử lý thông tin đã được hỗ trợ bởi các tiến bộ đáng kể trong công nghệ phần cứng cho các hoạt động cơ bản như cảm biến, nhận thức, và nhận dạng đối tượng. Các nền tảng và thị trường mới cho các sản phẩm nhờ vào dữ liệu, và các khuyến khích kinh tế để tìm ra các sản phẩm và thị trường mới, cũng góp phần cho sự ra đời công nghệ dựa vào trí tuệ nhân tạo[17].

Tất cả những xu hướng này thúc đẩy các lĩnh vực nghiên cứu về trí tuệ nhân tạo trong những năm qua và cả trong tương lai không xa, cụ thể[2]:

- Nhận dạng mẫu
- Xử lý ảnh
- Mạng nơron
- Xử lý ngôn ngữ tự nhiên
- Robot học
- Chatbot...

1.1.1. Chatbot là gì?

Chatbot (có thể được gọi là chatter robot) là một lĩnh vực của trí tuệ nhân tạo. Chatbot là một hệ thống thực hiện sự trao đổi thông tin giữa hai hay nhiều đối tượng theo một quy chuẩn nhất định, quá trình trao đổi thông tin có thể bằng ngôn ngữ nói, ngôn ngữ viết hoặc kí hiệu[2].

Chatbot có thể hiểu đơn giản là một chương trình máy tính mà người dùng có thể giao tiếp với máy thông qua các ứng dụng nhắn tin. Một chatbot có thể nói và hiểu tiếng nói và sẽ phân tích những gì con người nói và cố gắng hiểu một yêu cầu đưa ra. Chatbot sau đó giao tiếp với các máy khác, truyền đạt câu hỏi sau đó trả lời con người.

Chatbot giúp cho con người tiết kiệm thời gian, chi phí thông qua ứng dụng trong việc chăm sóc khách hàng (tự động hóa quy trình...), hay nâng cao năng suất

lao động (các bot giúp đặt lịch...) hay thậm chí chăm sóc đời sống con người (các bot chăm sóc sức khỏe...).

Chatbot có thể được phân loại thành 3 loại chính[2]:

- Chatbot giữa người với người

- Chatbot giữa máy với máy

- Chatbot giữa người và máy

Mặc dù chatbot là chủ đề "nóng" trong thời gian gần đây, nhưng thực ra chatbot đã có mặt từ cách đây 50 năm. Năm 1950, từ ý tưởng của Turing là đưa ra một thiết bị thông minh sẽ thay thế con người thực hiện các cuộc hội thoại. Ý tưởng này giúp hình thành nền tảng cho cuộc cách mạng chatbot. Sau đó, Eliza là chương trình chatbot đầu tiên được phát triển năm 1966. Chương trình được tạo ra để "đóng vai" nhà trị liệu trả lời các câu hỏi đơn đơn giản với các cấu trúc câu xác định. Chương trình được phát triển bởi ông Joseph Weizenbaum, Viện Công nghệ Massachusetts, Mỹ.

Ngày nay với sự xuất hiện của máy tính ở mọi nơi và dựa trên kho cơ sở dữ liệu đa dạng và đồ sộ được lưu trữ trên máy tính. Để có thể khai thác được kho dữ liệu đa dạng và đồ sộ này máy tính cần có khả năng xử lý thông tin trong quá trình trao đổi thông tin (hội thoại). Với khả năng hội thoại thông minh, chatbot có thể đáp ứng được yêu cầu trên để trở thành một chương trình tư vấn trợ giúp cho mọi người.

1.1.2. Chatbot hỗ trợ học tiếng Anh

1.1.2.1. Miki

Miki là một chatbot trên Facebook, được hoạt động sau khi Facebook chính thức hỗ trợ một nền tảng dành cho bot trên Messenger. Chatbot này có rất nhiều tính năng chủ yếu về lĩnh vực giải trí, tra cứu và học tập, trong đó có tính năng hỗ trợ học tiếng Anh khá thú vị.

Sự tiện lợi khi sử dụng chatbot này đó là người sử dụng không cần phải cài thêm bất kì ứng dụng nào, chỉ cần bật Messenger và chat với chatbot. Các tính năng học tính Anh được hỗ trợ trên Miki:

- Tra từ điển Anh Việt

- Tra câu song ngữ Anh Việt

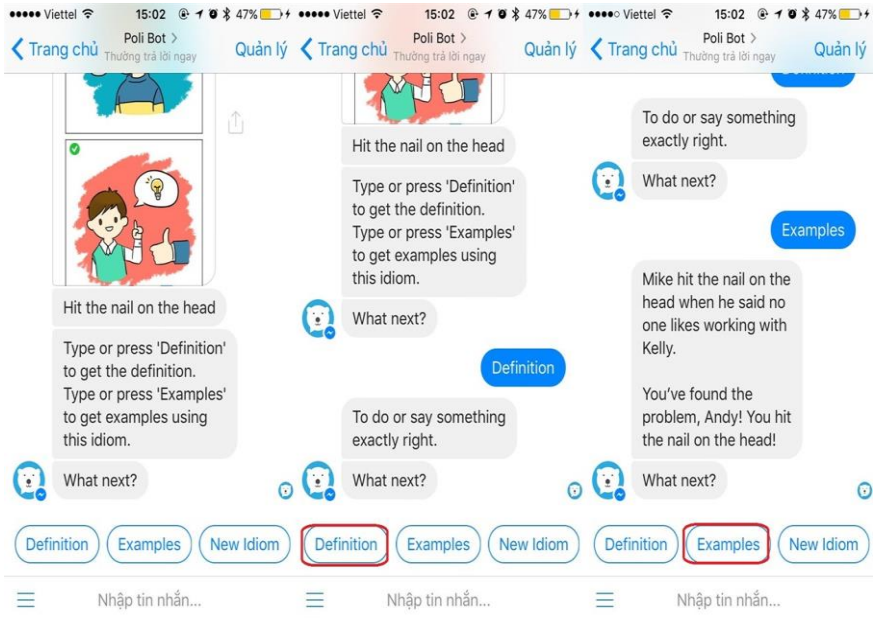
- Dịch đoạn văn

Hình 1.3. Chatbot Miki

1.1.2.2. Poli Bot

Poli là một chatbot chuyên dạy thành ngữ tiếng Anh. Poli hoạt động tương tự như một cuốn từ điển số, cung cấp các thành ngữ tiếng Anh phổ biến kèm theo hình minh họa dễ nhớ để người dùng có hứng thú học hỏi hơn. Một số tính năng của Poli:

- Cung cấp các thành ngữ tiếng Anh
- Xem định nghĩa
- Xem các ví dụ về cách dùng

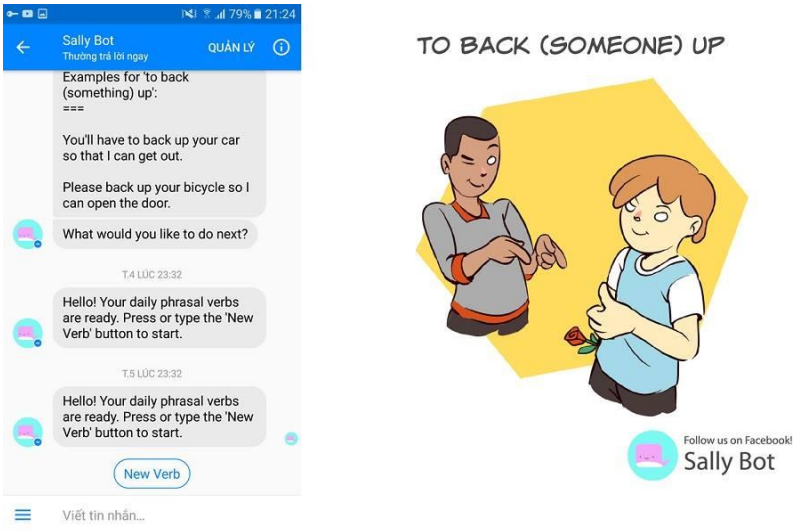


Hình 1.4. Chatbot Poli

1.1.2.3. Sally Bot

Cùng được sáng lập bởi Airpoli như Poli Bot, Sally được xây dựng để người dùng hiểu hơn về các cụm động từ trong tiếng Anh. Các tính năng của Sally:

- Học cụm động từ mới
- Định nghĩa cụm từ đã cho
- Đưa ví dụ liên quan đến cụm từ đã cho
- Đưa cụm từ đã cho áp dụng vào đoạn hội thoại

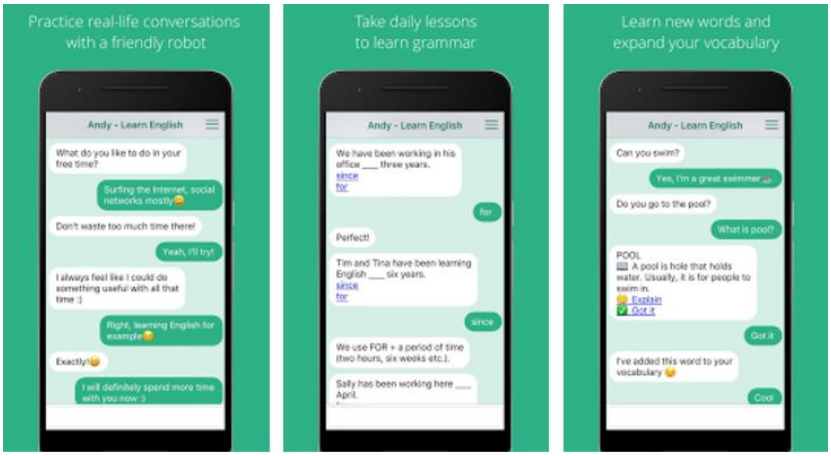


Hình 1.5. Chatbot Sally

1.1.2.4. Andy English

Các tính năng của Andy English:

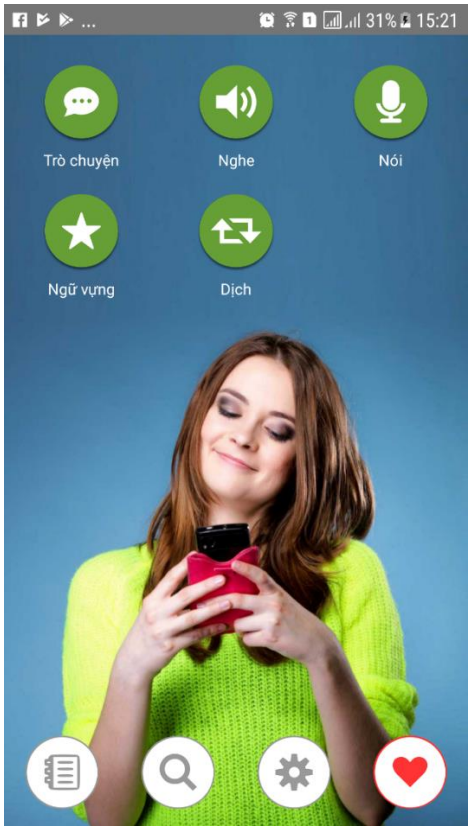
- Hội thoại bằng tiếng Anh, thảo luận về các chủ đề khác nhau
- Học ngữ pháp
- Học thêm từ mới để mở rộng vốn từ



Hình 1.6. Chatbot Andy English

1.1.2.5. Acobot

Acobot là một ứng dụng hỗ trợ học tiếng Anh với các tính năng giúp người sử dụng luyện các kỹ năng đọc, viết, nghe, nói, đàm thoại, phát âm, dịch thuật, ngữ pháp và từ vựng.



Hình 1.7. Chatbot Acobot

Qua nghiên cứu các chatbot trên, có thể thấy rằng, hầu như tất cả các chatbot đều tập trung vào việc tập trung vào phần luyện từ vựng, ứng dụng từ điển, trắc nghiệm... hoặc các ứng dụng luyện kỹ năng nghe, kỹ năng đọc; gần như chưa có ứng dụng nào hỗ trợ người sử dụng trong việc kiểm tra chính tả, ngữ pháp. Đó cũng chính là lý do chính để tác giả lựa chọn đề tài này.

1.2. Ngữ pháp tiếng Anh

1.2.1. Các khái niệm cơ bản

Ngữ pháp

Ngữ pháp là thuật ngữ dịch từ *grammaire* (tiếng Pháp), *grammar* (tiếng Anh) mà gốc là *grammatikè technè* (nghệ thuật viết) của tiếng Hi Lạp[11]. Thuật ngữ này có hai nghĩa:

(1) là một bộ phận của cấu trúc ngôn ngữ, nó có đơn vị khác với đơn vị của từ vựng và ngữ âm;

(2) là một ngành của ngôn ngữ học nghiên cứu sự hoạt động, sự hành chức theo những quy tắc nhất định để biến các đơn vị ngôn ngữ thành các đơn vị giao tiếp.

Để phân biệt rạch ròi hai nghĩa trên có thể dùng thuật ngữ "ngữ pháp" cho nghĩa (1) và "ngữ pháp học" cho nghĩa (2). Với ý nghĩa đó mà nói thì ngữ pháp học là khoa học nghiên cứu về ngữ pháp[4].

Ngữ pháp là quy tắc chủ yếu trong cấu trúc ngôn ngữ. Việc tạo ra các quy tắc chính cho một ngôn ngữ riêng biệt là ngữ pháp của ngôn ngữ đó, vì vậy mỗi ngôn ngữ có một ngữ pháp riêng biệt của nó. Ngữ pháp là một phần trong nghiên cứu ngôn ngữ hay còn gọi là ngôn ngữ học. Ngữ pháp là một cách thức để hiểu về ngôn ngữ. Mặt khác, ngữ pháp còn là một công cụ để quản lý từ ngữ, làm cho từ ngữ từ một từ hay nhiều từ thành một câu đúng ý nghĩa và thực sự hữu ích.

Ngữ pháp, theo cách hiểu của hầu hết các nhà ngôn ngữ học hiện đại bao gồm ngữ âm, âm học, hình thái ngôn ngữ, cú pháp, ngữ nghĩa. Tuy nhiên, theo truyền thống, ngữ pháp chỉ bao gồm hình thái ngôn ngữ và cú pháp[11].

Kiểm tra ngữ pháp là quá trình kiểm tra một văn bản có phù hợp với ngữ pháp của ngôn ngữ đó hay không.

Cú pháp

Cú pháp là một phần trong ngữ pháp. Cú pháp bao gồm tập các luật, nguyên tắc và các quá trình biến đổi để ta có thể xây dựng cấu trúc của một câu trong một ngôn ngữ theo một thứ tự nhất định.

Các lớp từ (nhãn từ) trong tiếng Anh

Theo Jurafsky và Martin[5], các từ thường được phân nhóm thành các lớp được gọi là thành phần văn bản (POS), các lớp từ, các lớp hình thái học hoặc các nhãn từ. Trong ngữ pháp truyền thống thường chỉ có một vài thành phần (danh từ, động từ, tính từ, giới từ, trạng từ, kết hợp...). Những mô hình ngữ pháp gần đây lại có số lượng lớp từ lớn hơn (45 lớp theo Penn Treebank, 87 với văn thể Brown, và 146 với mục tiêu của C7). Ví dụ, những tập nhãn riêng biệt giữa các tính từ sở hữu (my, your, his...) với các đại từ nhân xưng (I, you, he...). Việc biết được nhãn của từ có thể giúp ta xác định được mục đích cũng như ngữ cảnh sử dụng của chúng, điều này rất hữu ích cho việc nhận dạng ngôn ngữ.

Các thành phần ngữ pháp có thể được chia thành 2 mảng lớn: đóng và mở. Các lớp đóng là các lớp có tập thành phần cố định. Ví dụ, giới từ là lớp đóng vì đã có một tập các giới từ cố định trong tiếng Anh; rất hiếm khi một giới từ mới

được tạo ra. Mặt khác, danh từ và động từ là các lớp mở vì nhiều danh từ và động từ mới vẫn đang được tạo ra hoặc được mượn từ ngôn ngữ khác. Lớp các từ đóng nhìn chung cũng là các từ chức năng, các từ chức năng là các từ ngữ pháp như of, it, and, hoặc you, những từ ngắn, được sử dụng thường xuyên và đóng một vai trò quan trọng trong ngữ pháp.

Có 4 lớp từ mở chính: Danh từ (nouns), động từ (verbs), tính từ (adjectives) và trạng từ (adverbs). Tuy nhiên, điều này đúng với tiếng Anh nhưng không phải với tất cả các ngôn ngữ, nhiều ngôn ngữ không có tính từ.

Tất cả các ngôn ngữ của loài người đã được biết đến đều có ít nhất 2 lớp từ là danh từ và động từ. Danh từ là tên của lớp từ vựng mà ở đó các từ chủ yếu được chỉ cho người, địa điểm, và các điều xảy ra. Nhưng vì các lớp từ ngữ giống danh từ được định nghĩa chức năng hơn là ngữ nghĩa nên nhiều từ dành cho người, địa điểm và các vật có thể không phải là danh từ, và nhiều danh từ chuyển đổi có thể không phải dành cho người, địa danh...

Danh từ được chia làm 2 nhóm: danh từ riêng (ví dụ: Regina, Colorado...) và danh từ chung được dùng để chỉ các đối tượng chung, không chỉ cụ thể một đối tượng với tên cụ thể. Trong danh từ chung lại chia làm 2 nhóm: danh từ đếm được và danh từ không đếm được.

Động từ bao gồm phần lớn các từ chỉ hành động và tiến trình, bao gồm các động từ chính như draw, provide, differ và go. Một lớp con của lớp động từ được gọi là bổ ngữ.

Lớp từ mở thứ ba là tính từ, xét về mặt ngữ nghĩa lớp này bao gồm nhiều từ mô tả các thuộc tính, tính chất và chất lượng. Đa số các ngôn ngữ đều có tính từ với các chủ đề như màu sắc, tuổi và giá trị, nhưng cũng có ngôn ngữ không có tính từ.

Lớp từ mở cuối cùng là trạng từ, nó khá phức tạp và có tính hỗn hợp cả về ngữ nghĩa và hình thái.

Các lớp từ đóng khác nhau giữa các ngôn ngữ khác nhau hơn so với các lớp mở. Dưới đây là tổng quát một vài lớp từ đóng quan trọng trong tiếng Anh cùng ví dụ của chúng:

- Giới từ (Prepositions): on, under, over, near, by, from, to, with...

- Mạo từ (Determiners): a, an, the...

- Đại từ nhân xưng (Pronouns): I, she, he, who...

- Liên từ (Conjunctions): and, but, or, as, if, when...

- Trợ động từ (Auxiliary verbs): can, may, should, are...

- Particles: up, down, on, off, in, out, at, by...

- Số đếm (numerals): one, two, three...

Giới từ thường đứng trước cụm danh từ; về mặt ngữ nghĩa, nó thường chỉ mỗi quan hệ về thời gian và không gian, có thể cả về nghĩa đen hoặc nghĩa ẩn dụ. Tuy nhiên chúng thường thể hiện một số các quan hệ khác.

Một particle là một từ giống với một giới từ hoặc một trạng từ, và chúng thường kết hợp với các động từ trong các mẫu gồm nhiều thành phần gọi là cụm động từ.

1.2.2. Phân loại lỗi

Lỗi chính tả (Spelling errors)

Được định nghĩa như một lỗi xác định một từ không tồn tại trong danh sách các từ đã có, từ này coi như là không đúng đối với tập từ đó. Tuy nhiên, trong một tập từ khác nếu có từ đó, đó lại không phải là lỗi chính tả nữa. Vì vậy, có thể nói lỗi chính tả là tương đối với tập từ quy định, tuy nhiên đa phần tập từ được xây dựng dựa trên từ điển đã bao gồm gần như toàn bộ các từ, nên các lỗi xác định được xác định thường là do người dùng nhập sai hoặc nhớ sai gây ra.

Ví dụ: *spekeas*, *tkuk* là các từ không có trong từ điển tiếng Anh.

Lỗi ngữ pháp (Grammar errors)

Được định nghĩa khi một câu không phù hợp với những tập luật của tiếng Anh. Không giống lỗi chính tả, lỗi ngữ pháp được xác định cần dựa vào toàn bộ các từ trong câu. Lỗi ngữ pháp được chia làm lỗi có cấu trúc và lỗi không cấu trúc. Lỗi có cấu trúc là các lỗi chỉ có thể được sửa bằng cách chèn, xóa hoặc chuyển một hoặc nhiều từ trong câu. Trong khi lỗi không cấu trúc là các lỗi mà có thể được sửa bằng cách thay thế một từ có sẵn trong câu bằng một từ khác.

Ví dụ: *I goes to school*; *goes* là động từ đi với ngôi thứ 3 số ít trong khi *I* là ngôi số 1, câu này có thể được sửa thành *I go to school*.

Lỗi phong cách dùng từ (Style errors)

Được định nghĩa là lỗi khi sử dụng các từ không thông dụng hoặc các câu có cấu trúc phức tạp làm văn bản trở lên khó hiểu.

1.2.3. Một số lỗi ngữ pháp trong tiếng Anh

Số luật ngữ pháp rất lớn và khó có thể kiểm soát được hết, do vậy luận văn chỉ tìm hiểu một số ít luật trong số đó. Ta sẽ phân tích một số luật cũng như các lỗi về ngữ pháp dưới đây:

- Lỗi chia động từ (Subject-Verb Agreement)[10]: Trong tiếng Anh, chủ ngữ và động từ luôn phải phù hợp với nhau theo ngôi hoặc theo số lượng.

Ví dụ: *He are a doctor*, chủ ngữ *(He)* và động từ *(are)* là được chia không chính xác (*He* là ngôi thứ ba số ít và phải đi với động từ tobe là *is*).

- Lỗi dùng mạo từ không xác định a/an[10]: Mạo từ *an* được sử dụng nếu danh từ đi theo sau được bắt đầu với các nguyên âm (a, e, i, o, u).

Ví dụ: a car, a test, an uniform, an engineer...

Tuy nhiên, cũng có một số trường hợp đặc biệt như: a university, a European initiative; an hour, an honor...

- Câu hỏi đuôi (Tag questions)[10]: Câu hỏi đuôi thường được sử dụng trong văn nói để khẳng định lại một mệnh đề. Nó được dùng bằng cách sử dụng một mẫu câu bị động với một trợ động từ và chủ ngữ của câu.

Ví dụ: *It's warm today* trở thành *It's warm today, isn't it?*

Khi động từ trong câu là bị động, trợ động từ sẽ ở dạng chủ động. Câu hỏi đuôi là một câu đặc biệt và có luật đặc biệt, do đó thường bị bỏ qua khi xét đến.

- Những lỗi khác: Nhiều lỗi khác thường là lỗi do kĩ thuật, thường bị gây ra do chỉnh sửa văn bản nhưng lại quên một số từ.

Ví dụ: *He said thought that it was a big cat*. Tác giả có thể muốn thay thế *said* bằng *thought* nhưng lại quên việc xóa đi động từ *said*.

Một số lỗi cũng có thể được gây ra do nhầm lẫn một số từ giống nhau như Than - Then.

1.3. Tổng quan bài toán kiểm tra ngữ pháp tiếng Anh

Để giải quyết bài toán kiểm tra ngữ pháp tiếng Anh, chúng ta cần phải thực hiện 2 nhiệm vụ:

- Phân tích cú pháp

- Kiểm tra ngữ pháp

1.3.1. Phân tích cú pháp

Đã từ lâu, con người luôn mong muốn phát minh ra một chiếc máy có khả năng nghe và thực hiện các mệnh lệnh của con người. Cho đến nay, một hệ thống như vậy vẫn còn trong ước mơ bởi máy móc vẫn gặp khó khăn trong việc nhận biết ngôn ngữ của con người, từ việc nghe đúng cho đến việc hiểu đúng được lời nói của con người rất là khó khăn. Tuy nhiên, con người đang tích cực nghiên cứu phát triển ra công nghệ mới để thực hiện được một hệ thống thông minh như con người, lĩnh vực đó là xử lý ngôn ngữ tự nhiên[6]. Trong đó, phân tích cú pháp là một bài toán trung tâm và được sử dụng trong rất nhiều ứng dụng của xử lý ngôn ngữ tự nhiên, một xu thế nghiên cứu và phát triển của trí tuệ nhân tạo hiện đại. Độ chính xác của bộ phân tích cú pháp có ảnh hưởng lớn tới kết quả của các ứng dụng xử lý ngôn ngữ khác. Các nghiên cứu về xây dựng phân tích cú pháp tự động đã được phát triển từ rất sớm và đã có nhiều bộ phân tích cú pháp với chất lượng rất tốt cho các ngôn ngữ như tiếng Anh, tiếng Trung, tiếng Việt...

1.3.1.1. Xử lý ngôn ngữ tự nhiên và các vấn đề chính

Xử lý ngôn ngữ tự nhiên là lĩnh vực trong khoa học máy tính, nhiệm vụ của nó là xây dựng một hệ thống có thể phân tích, hiểu được ngôn ngữ của con người, không những thế hệ thống này còn có khả năng phản hồi lại bằng chính ngôn ngữ của con người. Như vậy ta có một mô hình đơn giản về một hệ thống xử lý ngôn ngữ tự nhiên như sau:

Hình 1.8. Mô hình xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên có rất nhiều ứng dụng trong thực tế, có thể kể ra ở đây một vài ứng dụng của xử lý ngôn ngữ tự nhiên như là dịch máy (machine translation), tìm kiếm thông tin (information retrieval), trích chọn thông tin (information retrieval) hay như là nhận dạng tiếng nói (speech recognition)[6].

- Dịch máy (machine translation) là một ứng dụng có nhiệm vụ dịch một văn bản từ một ngôn ngữ (ví dụ như tiếng Anh) sang một ngôn ngữ khác (chẳng hạn là tiếng Việt), giống như người phiên dịch.

- Tìm kiếm thông tin (information retrieval): ở đây ta có thể thấy một ví dụ rất điển hình đó là web search engine, www.google.com, website này là một dạng của tìm kiếm thông tin, tức là khi cần một thông tin, hệ thống sẽ thực hiện việc tìm kiếm trong dữ liệu (tập rất nhiều các văn bản) một hay nhiều văn bản tương tự với thông tin ta cần tìm kiếm.

- Trích chọn thông tin (information extraction): khi đưa vào một tập văn bản, hệ thống này có thể trả về cho ta những đoạn trong văn bản đó miêu tả thông tin chúng ta quan tâm. Một ví dụ đơn giản ở đây là khi gặp một trang blog ta cần xác định một số thông tin về cá nhân sở hữu blog như tên, giới tính, địa chỉ... thì hệ thống trích chọn thông tin có nhiệm vụ trả về cho ta các thông tin này.

- Nhận dạng tiếng nói (speech recognition): Khi bạn nói một câu, chúng ta đã có những hệ thống có thể ghi lại những âm thanh này ở dạng dữ liệu số, mục tiêu của ứng dụng này là chuyển được sóng âm thanh này thành dữ liệu văn bản.

Trên đây là một số ứng dụng của xử lý ngôn ngữ tự nhiên và trong thực tế còn nhiều ứng dụng khác đang được nghiên cứu và phát triển. Tuy nhiên, các ứng dụng ngôn ngữ tự nhiên đều có chung một số bài toán cơ sở như là phân tích từ tố, phân tích cú pháp, phân tích ngữ nghĩa. Trong đó, phân tích cú pháp đóng vai trò trung tâm trong ứng dụng xử lý ngôn ngữ tự nhiên và là mục tiêu của luận văn này hướng tới.

1.3.1.2. Phân tích cú pháp

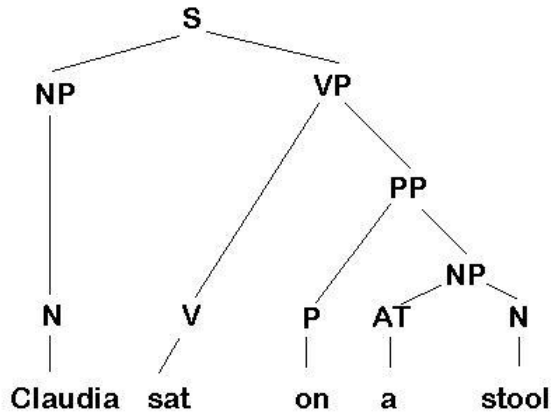
Phân tích cú pháp liên quan đến việc nghiên cứu về cấu trúc của một câu và mối quan hệ đa dạng giữa các từ trong câu đó[7]. Phân tích cú pháp là bước xử lý quan trọng trong các bài toán hiểu ngôn ngữ tự nhiên. Nó cung cấp một nền tảng vững chắc cho việc xử lý văn bản thông minh như các hệ thống hỏi đáp, khai phá văn bản và dịch máy[1].

Phân tích cú pháp (parsing analysis hay syntatic analysis) là quá trình phân tích một chuỗi từ tố (chuỗi từ tố này là kết quả của quá trình phân tích từ tố, thông thường đối với xử lý ngôn ngữ là các từ), nhằm đưa ra các cấu trúc ngữ pháp của chuỗi từ đó dựa vào một văn phạm nào đó. Thông thường cấu trúc ngữ pháp được chọn ở đây thường là dạng cây, bởi thông qua dạng này sự phụ thuộc của các thành phần là trực quan[6].

Nói cách khác, phân tích cú pháp là quá trình dựa vào văn phạm để xây dựng một cây cú pháp.

Trong tiếng Anh, phân tích cú pháp cụ thể là phân tích một câu và xây dựng một cây cú pháp dựa trên một văn phạm, văn phạm đó thường là dựa trên tập luật ngữ pháp trong tiếng Anh. Ta sẽ kiểm tra câu hoặc văn bản có thỏa mãn các luật ngữ pháp trong tiếng Anh hay không. Nếu một câu không thể xây dựng thành công cây cú pháp, có nghĩa câu văn đó là lỗi.

Ví dụ: Phân tích cú pháp câu: *Claudia sat on a stool*



Hình 1.9. Cú pháp câu "*Claudia sat on a stool*"

Việc phân tích cú pháp câu có thể được chia làm 2 bước chính: Tách, gán nhãn từ loại và xây dựng cấu trúc cây cú pháp[1].

1.3.1.3. Vai trò của phân tích cú pháp trong xử lý ngôn ngữ tự nhiên

Có thể nói phân tích cú pháp là bài toán cơ sở, xuất hiện rất nhiều trong các ứng dụng của xử lý ngôn ngữ tự nhiên. Ứng dụng đầu tiên ta có thể thấy ngay đó là áp dụng phân tích cú pháp trong kiểm tra lỗi ngữ pháp. Đối với việc kiểm tra lỗi ngữ pháp ta cần thực hiện việc phân tích cú pháp câu đầu vào, xem cấu trúc có đúng không? Trong dịch máy, hiện nay, có ba chiến lược dịch cơ bản là dịch trực tiếp, dịch chuyển đổi và dịch liên ngữ. Đối với dịch trực tiếp, cách dịch này dựa vào bộ từ điển song ngữ để dịch, không sử dụng đến phân tích cú pháp. Tuy nhiên trong dịch chuyển đổi và dịch liên ngữ, quá trình phân tích cú pháp là một bước quan trọng. Tư tưởng chung ở đây là đều phân tích câu nguồn trở thành cây cú pháp sử dụng bộ phân tích cú pháp. Đối với dịch chuyển đổi, hệ thống sẽ xây dựng cây cú pháp tương đương trong ngôn ngữ đích và cuối cùng đưa cây cú pháp thành câu cần đưa ra. Đối với dịch liên ngữ, cây cú pháp ở ngôn ngữ nguồn được đưa thành một biểu diễn chung giữa hai ngôn ngữ sau đó dạng biểu diễn chung này được chuyển về cây cú pháp ở ngôn ngữ đích, cuối cùng trả về câu cần dịch. Trong

lĩnh vực như nhận dạng tiếng nói (speech recoginition) sử dụng phân tích cú pháp có thể giúp sửa sai quá trình nhận dạng. Trong tổng hợp tiếng nói, phân tích cú pháp giúp đặt trọng âm vào đúng vị trí trong câu[6].

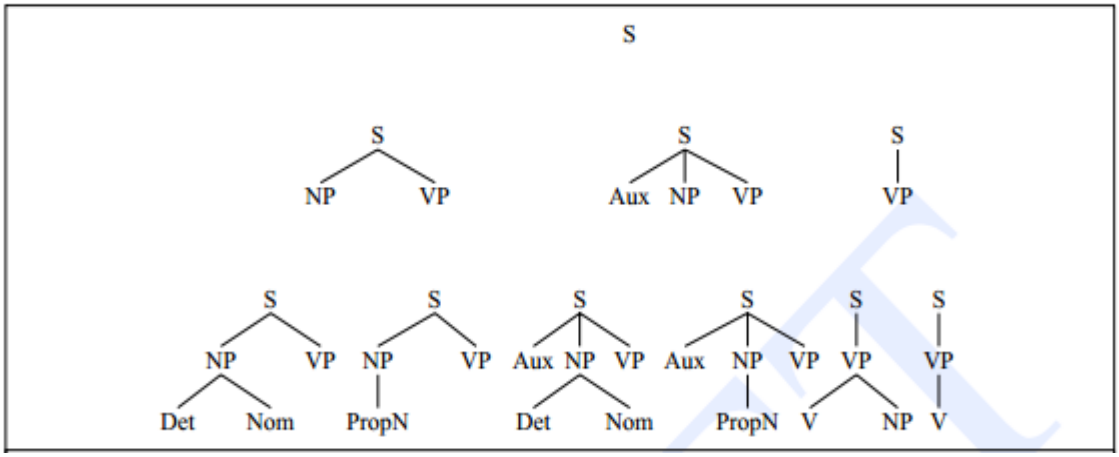
Những ví dụ ở trên đây đã khẳng định được vai trò của phân tích cú pháp trong xử lý ngôn ngữ tự nhiên. Vì vậy, để kiểm tra và phát hiện được lỗi ngữ pháp trong tiếng Anh thì cần phải giải quyết được bài toán cơ sở và trọng tâm là phân tích cú pháp cho tiếng Anh.

1.3.1.4. Các hướng tiếp cận

Để tiếp cận bài toán, có 2 hướng chính: Phương pháp phân tích từ trên xuống (Top - Down Parsing) và phương pháp phân tích từ dưới lên (Bottom - Up Parsing). Những thuật ngữ này là dựa vào thứ tự xây dựng các nút trong cây phân tích cú pháp. Phương pháp Top - Down là bắt đầu xây dựng từ gốc tiến hành hướng xuống các nút lá, còn phương pháp Bottom - Up là tiếp cận từ các lá tiến về gốc.

Phương pháp phân tích cú pháp từ trên xuống (Top - Down)

Phương pháp này tìm kiếm một cây cú pháp phù hợp bằng cách cố gắng xây dựng dần dần từ nút gốc S xuống các nút lá. Ta sẽ thử hết tất cả các nút là có thể sinh ra từ một nút gốc S mà có mặt trong văn phạm, từ đó lại tiếp tục lấy các nút lá đó làm gốc để tiếp tục quá trình. Nói cách khác, tại mỗi bước ta sẽ thử lần lượt từng cây có thể sinh ra.



Hình 1.10. Phương pháp Top - Down

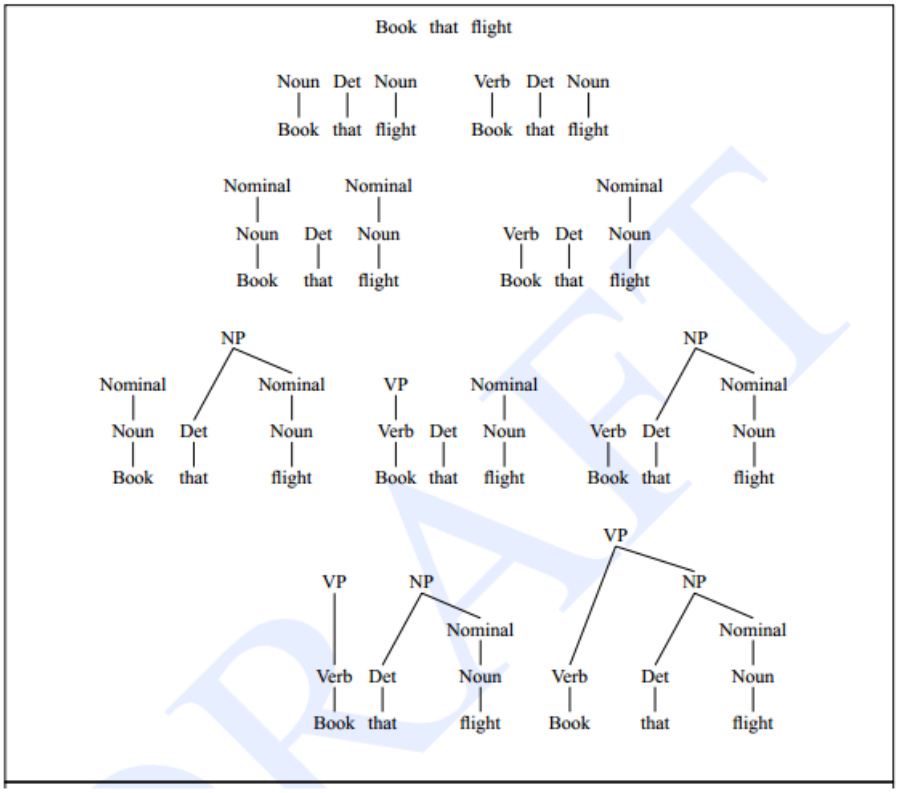
Phương pháp này có ưu điểm là sẽ chỉ tập trung vào những cây cú pháp có gốc là S, do vậy sẽ không lãng phí thời gian vào những cây cú pháp mà không có kết quả mong muốn. Tuy nhiên, do chỉ có thể tập trung vào những cây cú pháp có gốc cụ thể, phương pháp này sẽ không phải lúc nào cũng cho ra kết quả, hơn nữa,

do tiến hành xây dựng từ gốc, cây cú pháp tạo ra có thể có những nút là không phù hợp với dữ liệu đầu vào.

Phương pháp phân tích từ dưới lên (Bottom - Up)

Đây là phương pháp được biết đến sớm nhất, được đưa ra lần đầu từ năm 1955. Cách tiếp cận này sẽ thực hiện bắt đầu từ những từ trong dữ liệu đầu vào, và cố gắng xây dựng cây cú pháp dần lên trên để tạo thành gốc qua các luật, văn phạm.

Ví dụ: Book that flight



Hình 1.11. Phương pháp Bottom – Up

Ưu điểm của phương pháp này là ta chỉ xét những cây có thể được sinh ra từ dữ liệu đầu vào. Tuy nhiên, ta lại có thể xây dựng ra những cây cú pháp có gốc không phải là S như mong muốn vì ta phải xây dựng tất cả các cây có thể từ dữ liệu đầu vào.

1.3.2. Bài toán kiểm tra ngữ pháp tiếng Anh

Có nhiều các tiếp cận bài toàn kiểm tra ngữ pháp, dưới đây ta sẽ tìm hiểu cơ bản về những phương pháp này.

- Kiểm tra dựa vào cú pháp (Syntax-based checking): Được đưa ra bởi Jensen vào năm 1993, ở cách tiếp cận này, một văn bản được xem là một cây cú pháp (parser) đầy đủ, các câu sẽ được phân tích và mỗi câu sẽ được xét ứng với

một cây cú pháp. Văn bản sẽ được xem xét là không đúng nếu quá trình phân tích cây cú pháp không thể thành công.

- Kiểm tra dựa vào thống kê (Statistics-based checking): Đưa ra bởi Attwell vào năm 1987. Với cách này, mỗi câu sẽ được xây dựng một văn thể với mỗi từ đều được gán nhãn để tạo ra một chuỗi các nhãn. Nhiều chuỗi sẽ hợp lệ nhưng một số khác có thể sẽ không thể xảy ra. Các chuỗi xảy ra thường xuyên trong các văn thể có thể được xem xét là đúng trong một văn bản khác, và các chuỗi còn lại có thể là lỗi.

- Kiểm tra dựa vào luật (Rule-based checking): Ở phương pháp này, một tập luật được so khớp với một văn bản đã được gán nhãn. Cách này giống với phương pháp kiểm tra dựa vào thống kê nhưng tất cả các luật đã được xây dựng bằng tay.

Ưu điểm của phương pháp tiếp cận dựa trên cú pháp là việc kiểm tra ngữ pháp sẽ luôn hoàn thành nếu bản thân ngữ pháp của nó là hoàn chỉnh, bất cứ câu nào không chính xác đều sẽ được xác định. Tuy nhiên, phương pháp này chỉ có thể xác định một câu là có chính xác hay không mà không thể chỉ ra được chính xác lỗi sai là gì và ở đâu. Bởi vậy, những luật mở rộng là cần thiết để có thể hoàn thành việc phân tích cú pháp câu. Nếu một câu chỉ có thể được phân tích cú pháp với một luật mở rộng, câu đó là lỗi.

Tuy nhiên, vấn đề chính của phương pháp dựa trên cú pháp là nó yêu cầu một ngữ pháp đầy đủ có thể bao phủ tất cả các văn bản muốn kiểm tra. Tuy nhiên, vẫn chưa có một bộ ngữ pháp có đủ bao phủ đủ rộng và mạnh mẽ được công bố cho đến hiện nay. Hơn nữa, do tính mập mờ của ngôn ngữ tự nhiên, một câu có thể được phân tích thành nhiều cách và tạo thành nhiều cây cú pháp.

Mặc khác, phương pháp dựa trên thống kê lại đối mặt với nhược điểm là kết quả rất khó để tương tác và giải thích. Thêm vào đó là việc lựa chọn tham số ngưỡng (threshold) để phân biệt giữa câu thông thường (common) và câu bất thường (uncommon).

Với phương pháp dựa vào luật, khác với dựa trên cú pháp, việc kiểm tra sẽ không bao giờ kết thúc, nó cũng có nhiều ưu điểm như:

- Một câu sẽ không cần hoàn thành để kiểm tra, thay vào đó ta có thể kiểm tra văn bản ngay trong khi đang được nhập vào và đưa ra phản hồi.

- Dễ dàng để điều chỉnh các tập luật.

- Có thể chỉ ra cụ thể thông tin về lỗi và giải thích lỗi.

- Dễ dàng mở rộng bằng người dùng.

- Có thể được xây dựng dần dần, bắt đầy với chỉ một luật rồi mở rộng tập luật sau đó theo thời gian.

1.4. Kết luận chương

Chương một luận văn đã trình bày những vấn đề cơ bản nhất về trí tuệ nhân tạo, lịch sử hình thành và phát triển cùng với xu hướng phát triển của trí tuệ nhân tạo hiện nay và trong tương lai, trong đó đi vào tìm hiểu xu hướng chatbot và lĩnh vực xây dựng chatbot hỗ trợ học tiếng Anh; tìm hiểu, đánh giá những chatbot hỗ trợ học tiếng Anh hiện có, từ đó đặt ra bài toán xây dựng công cụ hỗ trợ xây dựng cây cú pháp, kiểm tra ngữ pháp; tìm hiểu bài toán phân tích cú pháp, kiểm tra chính tả, ngữ pháp và các vấn đề liên quan.

CHƯƠNG 2: MÔ HÌNH PCFGs VÀ NGÔN NGỮ AIML

2.1. Mô hình PCFGs

Một hướng tiếp cận trong việc xây dựng bộ phân tích cú pháp là sử dụng phương pháp thống kê. Bài toán phân tích cú pháp giống như một bài toán trong học máy, thông qua quá trình huấn luyện xây dựng một mô hình xác suất để thực hiện việc lựa chọn cây cú pháp phù hợp nhất. Trong phần này chúng ta sẽ tiếp cận văn phạm phi ngữ cảnh hướng thống kê PCFGs (Probabilistic Context Free Grammar) để giải quyết vấn đề đó.

2.1.1. Văn phạm phi ngữ cảnh

Các khái niệm cơ bản

Để có thể thực hiện được phân tích cú pháp trước tiên ta phải biểu diễn được ngôn ngữ đó bằng máy tính. Ngôn ngữ được định nghĩa là tập các xâu mà mỗi xâu này được tạo ra bởi một tập hữu hạn các phần từ không rỗng gọi là bảng chữ cái[6], ví dụ như ngôn ngữ tiếng Anh và bảng chữ cái tiếng Anh.

Một văn phạm phi ngữ cảnh (CFG) là một tập 4 thành phần chính G = (N, ∑, R, S), trong đó:

- N là tập chứa hữu hạn các phần tử được gọi là phần tử không kết thúc

- ∑ là tập chứa hữu hạn các phần tử được gọi là phần tử kết thúc

- R là tập các luật ngữ pháp có dạng $X \rightarrow Y_1Y_2...Y_n$, $X \in N$, $n \geq 0$, $Y_i \in (N \cup \sum)$ với i = 1...n.

- S là một trong những phần tử ∈ N được gọi là ký tự bắt đầu.

Xét một CFG đơn giản như sau:

N = {S, NP, VP, PP, DT, Vi, Vt, NN, IN}
S = S
Σ = {sleeps, saw, man, woman, dog, telescope, the, with, in}

R =

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
NN	→	dog
DT	→	the
IN	→	with
IN	→	in

Hình 2.1. CFG đơn giản[9]

Hình 2.1 chỉ ra một CFG đơn giản trong tiếng Anh. Trong trường hợp này, tập các phần tử không kết thúc N đặc tả một số cú pháp cơ bản: S đại diện cho "Sentence", NP đại diện cho "Noun Phrase", VP cho "Verb Phrase",… ∑ chứa tập các từ trong câu đã được phân tách. Phần tử bắt đầu cho văn phạm này là S, có nghĩa ta sẽ tiến hành xây dựng một cây cú pháp có gốc là S. Cuối cùng, chúng ta có tập luật phi ngữ cảnh R gồm luật **S → NP VP** hay **NN → man**

Ta xét luật **S → NP VP**, có nghĩa rằng S (sentence) có thể được tạo ta bằng cách kết hợp 2 thành phần là NP (noun phrase) và VP (verb phrase). Tương tự, **NN → man** chỉ ra rằng *NN* có thể được tạo thành từ *man*, tuy nhiên đây là trường hợp đặc biệt nên ta cũng có thể xem đây là một cách gán nhãn từ loại *NN* cho từ *man*.

Mỗi luật $X \rightarrow Y_1 \ldots Y_n$ trong tập R đều được tạo thành từ một thành phần X thuộc tập N chỉ luật gốc (có thể gọi là luật bên trái xét theo ký hiệu →) và các Y_i với (i = 1…n) là các thành phần thuộc tập N hoặc ∑ được gọi là thành phần tạo luật (ta có thể gọi là luật phải). Những luật chỉ gồm một luật phải được gọi là luật đơn (unary rule), ví dụ:

$$NN \rightarrow man$$

$$S \rightarrow VP$$

Ta cũng có những luật gồm các luật phải là tổ hợp cả các thành phần thuộc tập N và tập ∑, ví dụ:

$$VP \rightarrow John\ Vt\ Mary$$

$$NP \rightarrow the\ NN$$

Dẫn xuất trái (Left-most Derivations)

Cho một văn phạm phi ngữ cảnh G, một dẫn xuất trái là một chuỗi các xâu $s_1 \ldots s_n$, trong đó:

s_1 = S, cụ thể s_1 chứa một thành phần đơn là ký tự bắt đầu.

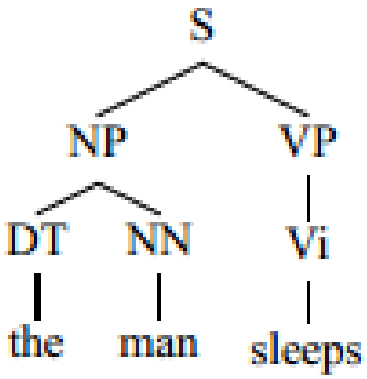
$s_n \in \sum^*$, s_n được tạo thành từ các phần tử kết thúc, cụ thể là các thành phần thuộc tập ∑ (viết $\sum^*$ để chỉ tập tất cả các xâu có thể được tạo thành từ các từ trong tập ∑).

Mỗi s_i (i = 2...n) là dẫn xuất từ s_{i-1} bằng cách lấy cách lấy các phần tử không kết thúc gần nhất bên trái X và thay thế chúng bằng các α trong đó α là tập luật phải được tạo ta từ X trong tập R, nói cách khác X → α.

Ta xét ví dụ CFG trong hình 2.2. Xét câu *"The man sleeps"* ta có dẫn xuất trái sau:

- s_1 = S.
- s_2 = NP VP (S → NP VP)
- s_3 = DT NN VP (NP → DT NN)
- s_4 = the NN VP (DT → the)
- s_5 = the man VP (NN → man)
- s_6 = the man Vi (VP → Vi)
- s_7 = the man sleeps (Vi → sleeps)

Ta dễ dàng có thể hiện một dẫn xuất như một cây cú pháp. Ví dụ, dẫn xuất trên có thể được thể hiện như một cây cú pháp như sau:



Hình 2.2. Cây cú pháp biểu diễn từ dẫn xuất

Theo cây cú pháp trên, ta có thể thấy rằng S là nút gốc, tương ứng với s_1 = S. Tương tự, các nhánh bên dưới tương ứng với từng bước tiến hành của dẫn xuất như NP → DT NN, VP → Vi...

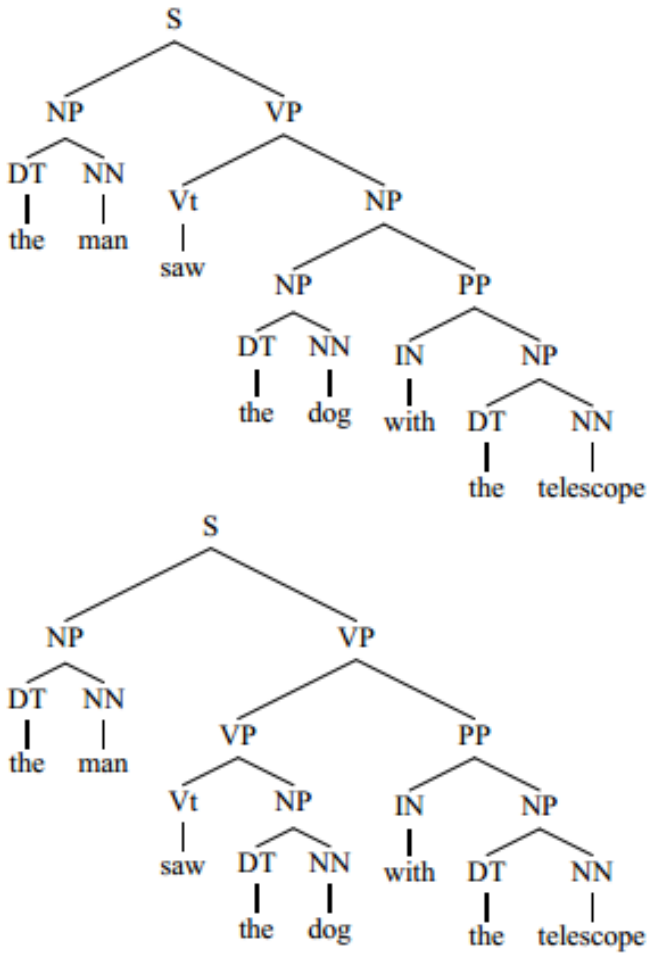
Một văn phạm phi ngữ cảnh G nói chung sẽ có 1 tập các dẫn xuất trái có thể xảy ra, mỗi dẫn xuất sẽ kết thúc tại một xâu $s_n \in \sum^*$, như ví dụ trên s_n là *The man sleeps*.

Một xâu $s \in \sum^*$ được gọi là thuộc ngôn ngữ được định nghĩa bởi CFG nếu có ít nhất một dẫn xuất có thể được xây dựng từ s.

2.1.2. Tính mập mờ trong phân tích cú pháp

Như đã đề cập ở trên, một xâu S có thể có nhiều hơn một dẫn xuất có thể thực hiện, trong trường hợp này, ta nói xâu s là mập mờ.

Ta xét ví dụ sau:



Hình 2.3. Tính mập mờ trong phân tích cây cú pháp[8]

Từ hình 2.3, ta có thể thấy xâu *the man saw the dog with the telescope* có thể được phân tích thành hai cây cú pháp khác nhau, cả hai cây cú pháp này đều thỏa mãn CFG. Sự mập mờ này đến từ sự không rõ ràng của cụm giới từ (PP - prepositional phrase), *PP with the telescope* có thể được đi cùng *dog* hoặc *saw the dog* đều khả thi. Ở cây cú pháp đầu tiên, PP đi cùng *the dog*, dẫn đến hình thành một NP là *the dog with the telescope*. Ở cây cú pháp thứ hai, PP lại đi cùng một VP *saw the dog*. Hai cây cú pháp này đều thể hiện ý nghĩa của câu khác nhau[8].

Tính mập mờ là một vấn đề rắc rối trong ngôn ngữ tự nhiên. Khi các nhà nghiên cứu lần đầu xây dựng một ngữ pháp đủ lớn phù hợp cho các ngôn ngữ như tiếng Anh, họ phát hiện rằng các câu văn thường có một lượng lớn cây cú pháp có thể xây dựng.

2.1.3. Văn phạm phi ngữ cảnh hướng thống kê PCFGs

2.1.3.1. Các khái niệm cơ bản

Cho G là một văn phạm phi ngữ cảnh, ta có cách khái niệm sau:

- T_G là tập hợp tất cả các cây cú pháp có thể xây dựng được trong G. Khi G rỗng ta có thể viết đơn giản tập hợp này là T.

- Với bất kỳ cây cú pháp t ∈ T_G, ta có yield(t) để mô tả xâu s ∈ ∑*, s cũng là xâu gồm chuỗi các từ được tạo ra từ t.

- Ta có câu s ∈ ∑*, ta định nghĩa T_G(s):

$$T_G(s) = \{t: t \in T_G, \text{yield}(t) = s\} \quad (2.1)$$

Nói cách khác, T_G(s) là tập tất cả các cây cú pháp tạo thành xâu s.

- Ta nói một câu s là mập mờ nếu: $|T_G(s)| > 1$.

- Ta nói một câu s là đúng cú pháp nếu: $|T_G(s)| > 0$.

Ý tưởng chính của PCFGs là mở rộng định nghĩa đã có để đưa ra một xác suất phân bố trên mỗi cây cú pháp. Ta sẽ tìm cách để định nghĩa một thông số phân bố trên mỗi cây cú pháp có thể được tạo ra, thông số đó được kí hiệu p(t), t ∈ T_G và thỏa mãn:

$$p(t) \geq 0 \text{ và } \sum_{t \in T_G} p(t) = 1 \quad (2.2)$$

Hướng tiếp cận này dường như rất khó bởi mỗi cây cú pháp đều có một cấu trúc phức tạp và T_G có thể là vô hạn. Tuy nhiên, ta sẽ xem xét một cách đơn giản để có thể định nghĩa hàm p(t) xác định thông số cho mỗi cây cú pháp t.

Sau khi đã có hàm p(t), ta sẽ tiến hành tính toán cho những cây cú pháp, sau đó sắp xếp chúng theo thông số giảm dần, từ đó ta có thể tìm được cây cú pháp có xác suất cao nhất như kết quả từ việc phân tích cú pháp văn bản. Nói cách khác, kết quả phân tích sau cùng sẽ là:

$$\arg\max_{t \in T_G(s)} p(t)$$

Hàm p(t) là một cách tiếp cận tốt cho xác suất phân bố khác nhau giữa các cây cú pháp, ta sẽ có một cách hiệu quả để giải quyết vấn đề mập mờ trong phân tích cú pháp.

2.1.3.2. PCFGs (Probabilistic Context-Free Grammars)

Một văn phạm phi ngữ cảnh hướng thống kê (PCFGs - Probabilistic Context-Free Grammars), còn được biết đến với tên SCFG (Stochastic Context-Free Grammar) được đề xuất lần đầu bởi Booth (1969) và được định nghĩa như sau:

Một PCFGs bao gồm:

- Một văn phạm phi ngữ cảnh G = (N,∑,S,R).

- Một tham số q(A →B) là xác suất xảy ra của luật A → B trong một dẫn xuất trái. Với X ∈ N, ta có:

$$\sum_{A \to B \in R, A = X} q(A \to B) = 1 \tag{2.3}$$

Hơn nữa q(A→B) ≥ 0 với mọi A → B ∈ R.

Cho một cây cú pháp t ∈ T_G chứa các luật $A_1 \to B_1$, $A_2 \to B_2$... $A_n \to B_n$, xác suất của t khi sử dụng PCFGs là:

$$p(t) = \prod_{i=1}^{n} q(A_i \to B_i) \tag{2.4}$$

Xét ví dụ sau:

N = {S, NP, VP, PP, DT, Vi, Vt, NN, IN}
S = S
Σ = {sleeps, saw, man, woman, dog, telescope, the, with, in}

R, q =

S	→	NP	VP	1.0
VP	→	Vi		0.3
VP	→	Vt	NP	0.5
VP	→	VP	PP	0.2
NP	→	DT	NN	0.8
NP	→	NP	PP	0.2
PP	→	IN	NP	1.0

Vi	→	sleeps	1.0
Vt	→	saw	1.0
NN	→	man	0.1
NN	→	woman	0.1
NN	→	telescope	0.3
NN	→	dog	0.5
DT	→	the	1.0
IN	→	with	0.6
IN	→	in	0.4

Hình 2.4. Một PCFGs đơn giản[8]

Ta thấy khác với CFG, mỗi luật X ∈ R đều có một tham số xác suất p(X) tương ứng, như p(VP → Vi) = 0.3 hay p(NN → man) = 0.1 và đều thõa mãn:

$$\sum_{A \to B \in R, A = X} q(A \to B) = 1$$

Cụ thể, ta có:

q(NP → DT NN) + q(NP → NP PP) = 0.8 +0.2 = 1.0, A = NP.

q(IN → with) + q(IN → in) = 0.6 + 0.4 = 1.0, A = IN.

Với PCFGs như hình 2.4, ta xét cây cú pháp t như hình 2.2:

Khi đó:

$$p(t) = q(S \rightarrow NP\ VP) * q(NP \rightarrow DT\ NN) *$$

$$q(DT \rightarrow the) * q(NN \rightarrow man) *$$

$$q(VP \rightarrow Vi) * q(Vi \rightarrow sleeps).$$

Một cách đơn giản, thực hiện xây dựng cây cú pháp dựa trên PCFGs theo các bước sau đây:

1. Khởi tạo $s_1 = S$, $i = 1$.

2. Trong khi s_i chứa ít nhất một kí tự chưa kết thúc:

- Tìm luật trái trong s_i, gọi là X.

- Chọn một luật có dạng $X \rightarrow A$ từ tập luật cùng $q(X \rightarrow A)$.

- Tạo s_i+1 bằng cách thay thế X trong s_i bằng A.

- Đặt $i = i + 1$ và lặp lại quá trình.

Bằng cách này, ta có thể dễ dàng tính toán xác suất của từng bước trong dẫn xuất trái. Kết quả cuối cùng của toàn bộ cây cú pháp là xác suất của bước cuối cùng, cũng là kết quả của những cây cú pháp độc lập theo từng cách chọn các luật riêng biệt.

2.1.3.3. Xây dựng PCFGs từ kho dữ liệu (Corpus)

Giả thiết rằng ta đã có một tập dữ liệu huấn luyện gồm các cây cú pháp t_1, t_2... t_m. Khi đó yield(t_i) chỉ câu được tạo ra từ cây cú pháp thứ i, cũng là câu thứ i trong kho dữ liệu.

Mỗi cây cú pháp t_i là gồm một tập các luật phi ngữ cảnh, giả sử tất cả các cây cú pháp trong kho dữ liệu đều có gốc là S, khi đó ta định nghĩa một PCFGs(N,∑,S,R,q) như sau:

- N là tập các phần tử không kết thúc trong các t_1, t_2,… t_m.

- ∑ là tập các từ trong các cây t_1, t_2,… t_m.

- S là ký hiệu bắt đầu.

- R là tập luật bao gồm tất cả các luật có dạng $A \rightarrow B$ trong t_1, t_2,… t_m.

- q là thông số xác suất của từng luật trong tập R, được tính theo công thức:

$$q(A \to B) = \frac{Count(A \to B)}{Count(A)} \quad (2.5)$$

Trong đó: Count(A→B) là số lần xuất hiện luật A→B trong kho dữ liệu, Count(A) là số lần xuất hiện của các luật có dạng A→X, X ∈ N trong kho dữ liệu.

Ví dụ, ta có luật VP → Vt NP xuất hiện 105 lần trong kho dữ liệu trong khi các luật có VP là luật trái xuất hiện 1000 lần, khi đó:

$$q(VP \to Vt\ NP) = \frac{105}{1000}$$

2.1.3.4. Xây dựng cây cú pháp với PCFGs

Luận văn sẽ tập trung tìm hiểu sử dụng thuật toán CKY để giải quyết bài toán xây dựng cây cú pháp với PCFGs. Thuật toán là dựa trên thuật toán CKY (Cocke-Kasami-Younger) hướng xác suất, được đưa ra lần đầu bởi Ney năm 1991.

Thuật toán CKY chỉ có thể áp dụng cho một loại PCFGs đặc thù, cụ thể là trong các PCFGs đó, các luật đều là ở chuẩn Chomsky (CNF - Chomsky Normal Form). Điều này có thể xem là một hạn chế, tuy nhiên ta có thể chuyển một PCFGs bất kỳ về dạng PCFGs thỏa mãn CNF.

Ngữ pháp chuẩn Chomsky (CNF)

Định nghĩa: Một văn phạm phi ngữ cảnh G = (N,∑,R,S) được gọi thỏa mãn chuẩn Chomsky nếu mỗi luật A→B ∈ R đều có một trong hai dạng sau:

- $X \to Y_1 Y_2$, $X \in N$, $Y_1 \in N$, $Y_2 \in N$.

- $X \to Y$, $X \in N$, $Y \in \sum$

Ví dụ:

N = {S, NP, VP, PP, DT, Vi, Vt, NN, IN}
S = S
Σ = {sleeps, saw, man, woman, dog, telescope, the, with, in}

R, q =

S	→	NP	VP	1.0
VP	→	Vt	NP	0.8
VP	→	VP	PP	0.2
NP	→	DT	NN	0.8
NP	→	NP	PP	0.2
PP	→	IN	NP	1.0

Vi	→	sleeps	1.0
Vt	→	saw	1.0
NN	→	man	0.1
NN	→	woman	0.1
NN	→	telescope	0.3
NN	→	dog	0.5
DT	→	the	1.0
IN	→	with	0.6
IN	→	in	0.4

Hình 2.5. Một PCFGs với CNF

Chuyển đổi về dạng CNF

Yêu cầu đặt ra là chuyển đổi một ngữ pháp PCFGs không theo chuẩn CNF về dạng CNF.

Ta có các trường hợp sau:

- Luật $X \in R$ có dạng $X \rightarrow Y_1\ Y_2\ Y_3$

Ta tiến hành biến đổi

$$Y_1\_Y_2 \rightarrow Y_1\ Y_2$$

$$X \rightarrow Y_1\_Y_2\ Y_3$$

Như vậy ta có thể biến đổi một luật từ không thuộc CNF về dạng CNF.

- Xét ví dụ:

$$A \rightarrow B$$

$$B \rightarrow \alpha$$

Trong đó A, B là các ký hiệu không kết thúc, α là ký hiệu kết thúc hay nói chính xác là một từ. Trong trường hợp này, ta trực tiếp biến đổi 2 luật này về luật có dạng:

$$A \rightarrow \alpha$$

Thuật toán CKY xây dựng cây cú pháp với văn phạm PCFGs

Trong phần này ta sẽ trình bày một thuật toán để phân tích cây cú pháp với văn phạm PCFGs có chuẩn CNF.

Dữ liệu đầu vào là một PCFGs $G = (N,\sum,S,Q,q)$ với chuẩn CNF, và một câu $s = x_1x_2\ldots x_n$ với x_i là từ thứ i trong câu.

Đầu ta của thuật toán là kết quả:

$$\arg\max_{t \in T_G(s)} p(t)$$

Thuật toán CKY là một thuật toán quy hoạch động. Ý tưởng chính của thuật toán như sau:

- Cho một câu có sẵn $x_1\ldots x_n$, ta định nghĩa $T(i,j,X)$ cho bất kỳ $X \in N$ và (i,j) thỏa mãn $1 \leq i \leq j \leq n$ là tập gồm tất cả cây cú pháp cho các từ $x_i\ldots x_j$ có gốc là X.

- Ta định nghĩa

$$\pi(i,j,X) = \max_{t \in T(i,j,X)} p(t)$$
$$(\pi(i,j,X) = 0 \ if \ T(i,j,X) = \varnothing) \quad (2.6)$$

Do đó $\pi(i,j,X)$ là điểm số cao nhất trong tất cả các cây cú pháp tạo thành từ các từ $x_i \ldots x_j$ và có X là gốc. Điểm số đó của cây t có được thông qua các điểm số của các luật mà nó chứa.

Ví dụ $A_1 \rightarrow B_1$, $A_2 \rightarrow B_2$,... $A_n \rightarrow B_n$, theo công thức (2.4) ta có

$$p(t) = \prod_{i=1}^{n} q(A_i \rightarrow B_i)$$

Do vậy

$$\pi(1,n,S) = \max_{t \in T_G(s)} p(t) \quad (2.7)$$

Đặc biệt, trong thuật toán CKY này, ta có thể sử dụng π như một hàm đệ quy để thực hiện các bước tính toán.

Đây là một thuật toán tiếp cận theo hướng "Bottom - Up", ta sẽ tiến hành thực hiện tính toán π(i,j,S) tại j = i đầu tiên, sau đó sẽ tiếp tục với j = i+1,…

Ta có, với mọi i = 1…n, X ∈ N

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) \ if \ X \rightarrow x_i \in R \\ 0 \qquad\qquad otherwise \end{cases} \quad (2.8)$$

Lại có với $1 \le i \le j \le n$, X ∈ N

$$\pi(i,j,X) = \max_{\substack{X \rightarrow Y\ Z \in R \\ s \in \{i \ldots (j-1)\}}} (q(X \rightarrow Y\ Z) \times \pi(i,s,Y) \times \pi(s+1,j,Z)) \quad (2.9)$$

- Thuật toán CKY

Đầu vào: câu $s = x_1 \ldots x_n$, văn phạm phi ngữ cảnh hướng thống kê PCFGs G = (N,∑,S,R,q)

Khởi tạo:

For all i ∈ {1…n}, for all X ∈ N,

$$\pi(i,i,X) = \begin{cases} q(X \rightarrow x_i) \ if \ X \rightarrow x_i \in R \\ 0 \qquad\qquad otherwise \end{cases}$$

Thuật toán:

- For k = 1…(n-1)

- For i = 1…(n-1)

+ Đặt j = i +k

+ For all X ∈ N

$$\pi(i, j, X) = \max_{\substack{X \to Y\ Z \in R \\ s \in \{i \ldots (j-1)\}}} (q(X \to Y\ Z) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

và

$$bp(i, j, X) = \arg \max_{\substack{X \to Y\ Z \in R \\ s \in \{i \ldots (j-1)\}}} (q(X \to Y\ Z) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

Đầu ra: $\pi(1, n, S) = \max_{t \in T_G(s)} p(t)$

2.1.3.5. Ưu điểm và hạn chế của hướng tiếp cận PCFGs

Ưu điểm

- PCFGs đưa ra hướng tiếp cận xây dựng một cây cú pháp có độ khả thi.

- Có thể loại bỏ những ngữ pháp không hợp lý và các lỗi ngữ pháp vì khi đó cây cú pháp có thông số, xác suất thấp.

- Giải quyết được vấn đề mập mờ do PCFGs sử dụng xác suất để lựa chọn cây cú pháp phù hợp nhất.

- Có thể phát triển mở rộng, số văn bản được phân tích càng nhiều, PCFGs càng thêm chính xác do xác suất từng luật cũng được điều chỉnh.

- PCFGs rất đơn giản và mô hình xác suất đơn giản đối với cấu trúc cây, mô hình toán học đơn giản, thuật toán không quá phúc tạp[6].

Nhược điểm

- PCFGs quan tâm đến cú pháp nhiều hơn là ngữ nghĩa, do vậy đôi khi cây cú pháp được chọn phù hợp về cú pháp nhưng lại không phù hợp về nghĩa.

- Do được tính toán thông qua từng cây con trong cây cú pháp, PCFGs có xu hướng tốt hơn với câu ngắn và giảm dần với các câu dài.

- Khi 2 cây cú pháp được tạo ra có cùng xác suất, PCFGs sẽ chọn cây cú pháp đầu tiên. Điều này đôi khi không chính xác.

- Với số lượng luật trong PCFGs tăng lên, công sức bỏ ra khi xây dựng cây cú pháp cũng tăng lên nhanh chóng.

2.2. Ngôn ngữ AIML

2.2.1. AIML là gì?

AIML (Artificial Intelligence Markup Language) là ngôn ngữ lập trình trí tuệ nhân tạo bắt nguồn từ XML (Extensible Mark-up Language), được sử dụng để phát triển các ứng dụng trí tuệ nhân tạo. AIML được phát triển bởi cộng đồng phần mềm miễn phí Alicebot trong những năm 1995 - 2000. Mỗi file AIML bắt đầu bằng thẻ <aiml> biểu thị phiên bản AIML đang được sử dụng, mỗi file này chứa các phần tử AIML gồm có các đối tượng dữ liệu được gọi là đối tượng AIML[13].

```
<aiml version = "1.0.1" encoding = "UTF-8"?>
   <category>
      <pattern> HELLO ALICE </pattern>

      <template>
         Hello User!
      </template>

   </category>
</aiml>
```

Hình 2.6. Ngôn ngữ AIML

2.2.2. Các Category và đặc tính của AIML

Category là một đơn vị cơ bản trong ngôn ngữ AIML, nó bao gồm một câu hỏi đầu vào, một câu trả lời làm đầu ra và một ngữ cảnh nhất định. Câu hỏi được đặt trong các thẻ <pattern> và nội dung trong thẻ <template> là câu trả lời tương ứng. Ngữ cảnh của một category được dựa vào 2 loại thẻ là <that> và <topic>.

AIML gồm các Category sau:

- Category nguyên tử: chứa các mẫu hội thoại nguyên tử, nghĩa là đây là những mẫu hội thoại nhỏ nhất không thể chia tách hay rút gọn được.

- Category mặc định: trong category này, mẫu đầu vào có thể chứa các ký hiệu thay thế "*" hoặc "_". Mẫu đầu vào này sẽ được chatbot rút gọn để tìm kiếm mẫu tương tự có sẵn trong cơ sở tri thức.

- Category đệ quy: bằng cách sử dụng các thẻ <srai> (Simply recursive artificial intelligence) và <sr> (Symbolic reduction) để ánh xạ đến các đầu vào khác hoặc chia tách thành nhiều đầu vào khác nhau.

Trong thẻ <pattern> có thể bao gồm word, khoảng cách (space) và ký tự đặc biệt (“*” và “_”). Độ ưu tiên lần lượt là:

1. “_”

2. word

3. “*”

“*” và “_” có thể tương ứng với một hoặc nhiều từ.

Độ ưu tiên của thông tin trong thẻ <pattern> được minh họa thông qua các ví dụ sau đây.

Ví dụ 1:

```
<category>
    <pattern>HELLO AI</pattern>
    <template>Nice to meet you</template>
</category>
<category>
    <pattern>HELLO *</pattern>
    <template>Hi <star> </template>
</category>
```

Hình 2.7. Ví dụ về độ ưu tiên thông tin (1)

Khi người sử dụng nhập vào thông tin để tương tác với ứng dụng trí tuệ nhân tạo thì sẽ nhận được kết quả tương ứng như sau:

- *Human:* *Hello AI*

  Robot: *Nice to meet you*

- *Human:* *Hello Alice*

  Robot: *Hi Alice*

Ví dụ 2:

```
<category>
    <pattern>HELLO AI</pattern>
    <template>Nice to meet you</template>
</category>
<category>
    <pattern>HELLO *</pattern>
    <template>Hi <star> </template>
</category>
<category>
    <pattern>HELLO _</pattern>
    <template>Hi. How old are you?</template>
</category>
```

Hình 2.8. Ví dụ về độ ưu tiên thông tin (2)

Trong ví dụ này, kết quả tương tác thu được như sau:

- *Human:* *Hello Anh*

  Robot: *Hi. How old are you?*

- *Human:* *Hello Alice*

  Robot: *Hi. How old are you?*

2.2.3. Một số thẻ thông dụng trong AIML

<aiml>	Định nghĩa bắt đầu và kết thúc của một tài liệu AIML
<category>	Định nghĩa một đơn vị tri thức gồm câu hỏi và câu trả lời
<pattern>	Định nghĩa một mẫu dữ liệu có thể so khớp với đầu vào từ người dùng
<template>	Định nghĩa phản hồi đến người dùng ứng với đầu vào phù hợp pattern
<star>	Được sử dụng để khớp với kí tự * trong dữ liệu từ thẻ <pattern>
<srai>	Được sử dụng để cho phép định nghĩa một phản hồi cho nhiều đầu vào có mục đích tương tự nhau
<random>	Được sử dụng để lấy ngẫu nhiên một phản hồi trong một tập các phản hồi được định nghĩa sẵn
<get>, <set>	Được sử dụng để làm việc với các biến trong AIML, các biến có thể được truy xuất dữ liệu trong qua các thẻ này
<that>	Được sử dụng để đưa ra các phản hồi tùy thuộc theo ngữ cảnh
<topic>	Được sử dụng để lưu trữ ngữ cảnh phục vụ cho việc các đoạn hội thoại sau có thể được diễn ra dựa trên ngữ cảnh đó
<think>	Được dùng để lưu trữ các biến mà không cần thông báo cho người dùng
<condition>	Được dùng để điều chỉnh các phản hồi tùy thuộc vào từng điều kiện rẽ nhánh thích hợp

2.2.3.1. Star/Srai/Sr

<star> là một lệnh giúp lấy các từ thay thế trong "*" hoặc "_".

Cú pháp: <star index = "x"/> (<star/> tương đương với index = 1)

```
<category>
    <pattern>HELLO *</pattern>
    <template>Hi <star/> </template>
</category>
<category>
    <pattern>* HELLO _. HOW ARE YOU?</pattern>
    <template>Hi <star index = "2"/>. How old are you?</template>
</category>
```

Hình 2.9. Ví dụ về thẻ <star>

- *Human:* *Hello Duc Anh*

Robot: *Hi Duc Anh*

- *Human:* *Hey Hello Duc Anh. How are you?*

Robot: *Hi Duc Anh. How old are you?*

<srai> có thể hiểu như một lời gọi "hàm". Giả sử như trong ví dụ ở Hình 2.10, <category> thứ nhất có "tên hàm" là "HELLO"; trong <category> thứ 2 ta gọi đến hàm "HELLO".

```
<category>
    <pattern>HELLO</pattern>
    <template>Hi. How are you?</template>
</category>
<category>
    <pattern>HELLO AI</pattern>
    <template><srai>HELLO</srai></template>
</category>
```

Hình 2.10. Ví dụ về thẻ <srai>

Lúc này, câu trả lời của <category> thứ 2 sẽ được đẩy sang cho <category> thứ nhất. Kết quả thu được sẽ là:

- *Human:* *Hello AI*

Robot: *Hi. How are you?*

Thẻ <sr> là sự kết hợp giữa thẻ <star> và thẻ <srai>

<sr> = <srai><star/></srai>

```
<category>
    <pattern>ANH DUC</pattern>
    <template>DUC ANH</template>
</category>
<category>
    <pattern>HELLO *</pattern>
    <template>HI <sr/> </template>
</category>
```

Hình 2.11. Ví dụ về thẻ <sr>

- *Human:* *Hello ANH DUC*

  Robot: *Hi DUC ANH*

2.2.3.2. Set/Get

2 thẻ <set> và <get> tương tự như 2 hàm setParameter và getParameter trong java. Cú pháp:

<set name= “nameOfParam”> Value </set>

<get name= “nameOfParam”/>

```
<category>
    <pattern>I'M *</pattern>
    <template>Hello <set name = "nameOfUser"> <star> </set> </template>
</category>

<category>
    <pattern>WHAT IS MY NAME?</pattern>
    <template>Your name is <get name = "nameOfUser"/> </template>
</category>
```

Hình 2.12. Ví dụ về thẻ <set>, <get>

- *Human:* *I’m Duc Anh*

  Robot: *Hello Duc Anh*

- *Human:* *What is my name?*

  Robot: *Your name is Duc Anh*

<category> thứ nhất đã gán giá trị “Duc Anh” cho biến “nameOfUser”. Do vậy ở <category> thứ 2, hàm <get> đã lấy ra giá trị “Duc Anh” từ biến “nameOfUser”.

2.2.3.3. That/Topic

<that> là một thẻ bên trong thẻ <category>, là nội dung của câu robot ngay trước đó. Trong category mà có thẻ <that> thì category này chỉ được thực thi khi thỏa mãn 2 điều kiện sau:

1. Câu hỏi trùng với nội dung trong thẻ <pattern>

2. Nội dung câu trả lời ngay phía trước trùng với nội dung trong thẻ <that>

```
<category>
    <pattern>HELLO</pattern>
    <template>Do you know him?</template>
</category>

<category>
    <pattern> * Yes * </pattern>
    <that>Do you know him?</that>
    <template>Great</template>
</category>

<category>
    <pattern> * No * </pattern>
    <that>Do you know him?</that>
    <template>He is a famous player</template>
</category>
```

Hình 2.13. Ví dụ về thẻ <that>

- *Human: Hello*

  Robot: Do you know him?

  Human: Yes, I know him

  Robot: Great

- *Human: Hello*

  Robot: Do you know him?

  Human: No, I'm sorry

  Robot: He is a famous player

Thẻ <topic> nằm bên ngoài và bao gồm nhiều thẻ <category>. Topic là một đặc tính để chuyển câu sang một chủ đề cụ thể. Nếu đang ở trong một chủ đề đã được định nghĩa thì những <category> bên trong chủ đề (topic) này sẽ có mức độ ưu tiên cao hơn so với các <category> khác.

```
<category>
    <pattern>What do you like?</pattern>
    <template>I like lots of things</template>
</category>

<category>
    <pattern>Animal Topic</pattern>
    <template>Yes.
        <think><set name = "topic"> Animal </set>
        </think>
    </template>
</category>

<topic name = "Animal">
    <category>
        <pattern>What do you like?</pattern>
        <template>I like dog</template>
    </category>
</topic>
```

Hình 2.14. Ví dụ về thẻ <topic>

- *Human:* *What do you like?*

  Robot: *I like lots of things*

  Human: *Animal*

  Robot: *Yes*

  Human: *What do you like?*

  Robot: *I like dog*

2.2.3.4. Condition

Thẻ <condition> thường nằm trong thẻ <template> được sử dụng như một câu lệnh kiểm tra điều kiện. Cú pháp:

<condition name = “nameOfParam”>

<li value = “value1”> answer1

<li value = “value2”> answer2

</condition>

```
<category>
    <pattern>I'm going shopping</pattern>
    <template>Yes. <think><set name="go">OK</set></think></template>
</category>

<category>
    <pattern>I'm at home</pattern>
    <template>Yes. <think><set name="go">NOT</set></think></template>
</category>

<category>
    <pattern>Where are you going?</pattern>
    <template>
        <condition name = "go">
            <li value = "OK"> I'm going shopping </li>
            <li value = "NOT">I'm at home</li>
        </condition>
    </template>
</category>
```

Hình 2.15. Ví dụ về thẻ <condition>

- *Human:* *I'm at home*

  Robot: *Yes*

- *Human:* *Where are you going?*

  Robot: *I'm at home*

2.2.3.5. Random/Think

<random> là một thẻ dùng để lựa chọn một câu trả lời bất kỳ trong <template>, <think> là một thẻ nhằm mục đích: tất cả những nội dung thể hiện trong thẻ <think> sẽ không được hiển thị ra ở câu trả lời.

```
<category>
    <pattern>HELLO</pattern>
    <template>
        <think><set name = "hello">Hello</set></think>
        <random>
            <li>Hello</li>
            <li>What's your name?</li>
            <li>How are you?</li>
        </random>
    </template>
</category>
```

Hình 2.16. Ví dụ về thẻ <random> và thẻ <think>

2.2.4. ProgramAB

Là một chương trình mã nguồn mở được phát triển bởi Richard Wallace và cài đặt trên ngôn ngữ Java, hoạt động với AIML 2.0 và hiện nay đang là nền tảng được ưu tiên với những tính năng mới[12].

Chương trình được xây dựng với cấu trúc để cho phép các lập trình viên dễ dàng mở rộng AIML với những thẻ có thể tự định nghĩa. ProgramAB có thể được sử dụng trong nhiều cách:

- Chạy ProgramAB để giao tiếp với một chatbot.

- Phân tích các file logs và phát triển nội dung của các bot.

- Sử dụng ProgramAB như một thư viện để phát triển ứng dụng trên Java và phát triển các tính năng khác.

- Lập trình các ứng dụng riêng với những thẻ AIML tự định nghĩa.

- Điều chỉnh và xây dựng ProgramAB theo yêu cầu cần thiết.

Với mỗi chương trình sử dụng ProgramAB, ta có thể tự điều chỉnh nội dung các cuộc hội thoại cũng như phát triển thêm các cuộc hội thoại thông qua điều chỉnh bots. Cụ thể ta có thể thêm các dữ liệu AIML tự định nghĩa vào các bot hoặc tích hợp nhiều bot vào tùy yêu cầu. Hiện nay, trong ProgramAB mặc định có sẵn dữ liệu của bot như Alice, tuy nhiên ta có thể tìm hiểu thêm một số bot tương tự như Anna, Charlie, Super…

2.3. Kết luận chương

Chương hai đã trình bày các nội dung về văn phạm phi ngữ cảnh, tính mập mờ trong phân tích cú pháp và đề xuất giải pháp sử dụng văn phạm phi ngữ cảnh hướng thống kê PCFGs kết hợp thuật toán CKY để xây dựng cây cú pháp và kiểm tra ngữ pháp; đồng thời nghiên cứu mã nguồn mở AIML phục vụ mục đích xây dựng chatbot.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ, CÀI ĐẶT ỨNG DỤNG

Trong chương 3, luận văn trình bày những nội dung cơ bản về thiết kế ứng dụng, xây dựng kho dữ liệu và cài đặt, đánh giá kết quả hoạt động của ứng dụng dựa trên một số mẫu kiểm thử.

3.1. Phân tích thiết kế

3.1.1. Xác định yêu cầu

3.1.1.1. Chức năng chính

Hội thoại giữa người và máy

Ứng dụng cho phép người dùng có thể giao tiếp, hội thoại với máy thông qua văn bản hoặc giọng nói.

Chi tiết chức năng:

- Người dùng có thể giao tiếp với máy thông qua các đoạn hội thoại, các đoạn hội thoại có thể được nhập vào băng văn bản hoặc bằng giọng nói. Với mỗi câu hội thoại của người dùng, máy sẽ tự động tìm câu trả lời thích hợp và đưa ra cho người dùng.

- Trong dữ liệu của máy sẽ có nhiều chủ đề, người dùng có thể chọn chủ đề để nói chuyện hoặc có thể thay đổi chủ đè bất kỳ khi nói chuyện.

Kiểm tra chính tả, ngữ pháp

Ứng dụng tập trung kiểm tra chính tả và ngữ pháp của các dữ liệu từ người dùng. Tuy nhiên, chức năng này mới chỉ hỗ trợ kiểm tra chính tả và kiểm tra ngữ pháp nghiêng về chia động từ trong câu, không bao gồm kiểm tra cú pháp câu.

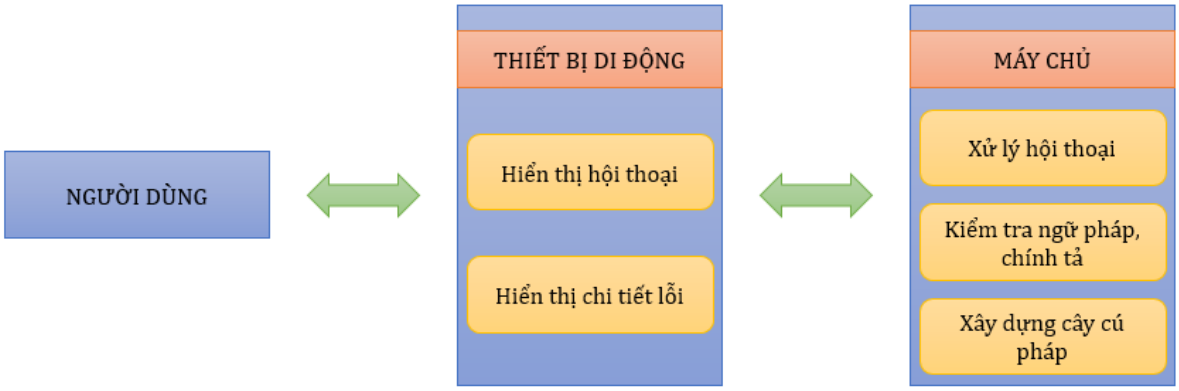
Chi tiết chức năng:

- Người dùng có thể tiến hành đưa dữ liệu vào ứng dụng để ứng dụng kiểm tra lỗi chính tả và ngữ pháp.

- Người dùng có thể chỉ xem lỗi của câu hoặc có thể xem chi tiết cách sửa lỗi từ ứng dụng.

- Dữ liệu đầu vào và đầu ra từ ứng dụng vẫn theo dạng hội thoại.

3.1.1.2. Mô hình hệ thống



Hình 3.1. Mô hình hệ thống AI English

Máy chủ (Server)

- Phụ trách xử lý dữ liệu từ client gửi lên và xử lý, trả về kết quả cho client.

- Các chức năng chính thực hiện trên server: Nhận hội thoại từ người dùng, tìm câu trả lời thích hợp và trả về cho người dùng, kiểm tra chính tả, ngữ pháp của câu người dùng gửi lên, từ câu người dùng gửi lên, xây dựng cây cú pháp và trả về cho người dùng.

Thiết bị di động (Device)

- Thiết bị di động thực hiện nhiệm vụ chính là hiển thị nội dung hội thoại, nhận yêu cầu người dùng gửi đi server và chờ xử lý. Các nội dung như lỗi chi tiết hay sửa lỗi, cây cú pháp cũng được hiển thị trên thiết bị di động.

- Các thiết bị di dộng cần sử dụng nền tảng Android 4.2 trở lên để cài đặt và sử dụng ứng dụng.

3.1.1.3. Chức năng người dùng

Người sử dụng sẽ có những chức năng sau:

- Đăng ký thông tin cá nhân

- Hội thoại bằng tin nhắn với chatbot

- Tự động kiểm tra chính tả

- Tự động kiểm tra ngữ pháp

- Dịch các tin nhắn sang tiếng Việt

- Nghe nội dung tin nhắn

- Tra cứu động từ bất quy tắc

- Tra cứu một số câu thông dụng

3.1.2. Xây dựng tập luật dựa trên tập dữ liệu có sẵn

Như đã đề cập tại chương 2, để xây dựng cây cú pháp của câu theo phương pháp PCFGs và thuật toán CKY, ta cần có tập luật R theo chuẩn CNF. Tại phần này, luận văn sẽ trình bày quá trình xây dựng tập luật CNF được sử dụng trong ứng dụng.

Quá trình xây dựng tập luật CNF trong PCFGs gồm 3 bước chính:

- Xây dựng kho dữ liệu câu tiếng Anh.

- Xử lý các câu trong kho dữ liệu và tạo các luật cơ bản theo chuẩn CNF.

- Xây dựng tập luật theo văn phạm PCFGs từ các luật cơ bản.

3.1.2.1. Xây dựng kho dữ liệu câu tiếng Anh

Nhiệm vụ của phần này là yêu cầu xây dựng một kho dữ liệu gồm nhiều câu tiếng Anh để phục vụ xây dựng các tập luật.

Để thực hiện nhiệm vụ này, ta sử dụng tập dữ liệu của Tatoeba[20], đây là một trang web bao gồm nhiều tập dữ liệu câu bằng nhiều ngôn ngữ khác nhau. Tuy nhiên, tập dữ liệu câu lấy về từ Tatoeba có chứa nhiều ngôn ngữ khác nhau. Cụ thể, một số câu trong dữ liệu có dạng như sau:

```
5200482 5539161 tur Neden bununla meşgul olmama izin vermiyorsun?
5200483 5539162 eng Tom won't play tennis today.
5200484 5539163 eng We really have nothing to lose.
5200485 5539164 eng Tom won't escape punishment.
5200486 5539165 eng We seem to agree on everything.
5200487 5539166 eng Tom won't call this evening.
5200488 5539167 eng We should find out pretty soon.
5200489 5539168 eng Tom will now be proud of me.
5200490 5539169 eng We should find out soon enough.
5200491 5539170 eng Tom will get well very soon.
5200492 5539171 eng We should never have come here.
5200493 5539172 rus В этих предложениях речь идёт о разных вещах.
5200494 5539173 rus Том должен сделать презентацию.
```

Hình 3.2. Dữ liệu trong Tatoeba

Ta thực hiện tách dữ liệu các câu tiếng Anh theo thẻ **“eng”** và thu được tập dữ liệu các câu tiếng Anh cần dùng có dạng như sau:

```
let's try something.
i have to go to sleep.
today is june 18th and it is muiriel's birthday!
muiriel is 20 now.
the password is "muiriel".
i will be back soon.
i'm at a loss for words.
this is never going to end.
i just don't know what to say.
that was an evil bunny.
i was in the mountains.
is it a recent picture
i don't know if i have the time.
education in this world disappoints me.
you're in better shape than i am.
you are in my way.
this will cost €30.
i make €100 a day.
i may give up soon and just nap instead.
```

Hình 3.3. Dữ liệu câu tiếng Anh

Bằng cách như vậy, ta có được tập dữ liệu gồm 885113 câu tiếng Anh từ tập dữ liệu ban đầu (số lượng câu tiếp tục được tăng lên). Tập dữ liệu này sẽ được sử dụng để xây dựng các tập luật cơ bản ở phần tiếp theo.

3.1.1.2. Xử lý các câu trong kho dữ liệu và tạo các luật cơ bản theo chuẩn CNF

Trong phần này, từ tập dữ liệu đã có ở phần 3.1.2.1, ta tiến hành phân tích và tạo ra các luật cú pháp CFG.

Để thực hiện điều này, ta sử dụng thư viện Stanford-parser[21] để xây dựng một cây cú pháp từ một câu bất kỳ, sau đó từ cây cú pháp có được, ta tách ra từng luật cú pháp và thêm vào tập dữ liệu cú pháp CFG. Trong quá trình tách các luật, ta cố gắng đưa các luật về chuẩn CNF. Xét ví dụ sau đây:

Xét câu:

I am good

Cây cú pháp:

(ROOT (S (NP (PRP I)) (VP (VBP am) (ADJP (JJ good)))))

Các luật sau tách được:

$$VBP \rightarrow am$$

$$ADJP \rightarrow good$$

$$NP \rightarrow I$$

$$VP \rightarrow VBP\ ADJP$$

$$S \rightarrow NP\ VP$$

Ta tiến hành quá trình này với tất cả các câu và thu được một tập dữ liệu các luật tuân theo CNF, tuy nhiên tất cả các luật này vẫn được lưu trữ theo văn phạm CFG.

3.1.1.3. Xây dựng tập luật theo văn phạm PCFGs

Từ tập luật đã có từ phần 3.1.2.2, ta tiến hành xây dựng tập luật để sử dụng trong văn phạm PCFGs. Để thực hiện điều này, tại mỗi luật ta tiến hành tính xác suất của chúng theo công thức (2.5) đã được đề cập như sau:

$$q(A \rightarrow B) = \frac{Count(A \rightarrow B)}{Count(A)}$$

Kết quả tập luật có được sau khi xây dựng có dạng như sau:

```
1  VB -> let  # 0.0088639836994832
2  NP -> us  # 0.0073448443323640O7
3  VB -> try  # 0.0022909567168723694
4  NP -> something  # 0.0027710653037581402
5  VP -> VB NP  # 0.17639995901341435
6  S -> NP VP  # 0.3985791989451129
7  ROOT -> VB S  # 0.013263434067662876
8  NP -> i  # 0.09174941050500558
9  VB -> have  # 0.05121961901504503
10 TO -> to  # 0.9999620548560298
11 VB -> go  # 0.012187848023124377
12 VP -> sleep  # 3.4330644391449916E-4
13 S -> TO VP  # 0.34544585462337235
14 VP -> VB S  # 0.06638425624673223
```

Hình 3.4. Tập luật trong PCFGs

Các luật sẽ được trình bày theo định dạng: S → A B # x hoặc S → A # x, trong đó S là luật gốc, A B là các luật cấu tạo thành luật S, x là xác suất của luật.

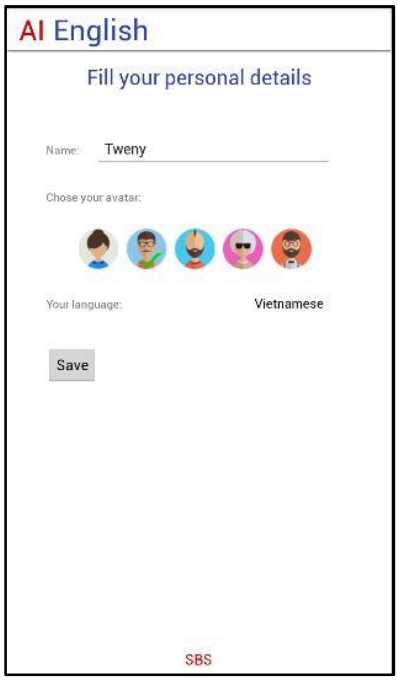
Khi tập dữ liệu các câu càng tăng, thời gian cần thiết để xử lý và xây dựng tập luật cũng tăng theo. Hơn nữa, kết quả cho thấy rằng đa số các luật được tạo thành là các luật gán nhãn, tức các luật có dạng S → A # x. Tập luật được xây dựng này sẽ được sử dụng để xây dựng chức năng phân tích cú pháp câu của ứng dụng AI English.

3.2. Cài đặt ứng dụng

Dựa trên lý thuyết đã nghiên cứu về PCFGs và AIML, với sự hỗ trợ của bạn bè và đồng nghiệp, tác giả đã xây dựng ứng dụng AI English. Đây là một ứng dụng giao tiếp với người dùng thông qua hội thoại, tương tự các ứng dụng chatbot. Ứng dụng có 2 chức năng chính: Hội thoại, kiểm tra ngữ pháp (chủ yếu là kiểm tra chính tả và chia động từ), cùng với đó là một số tính năng khác như: tra cứu từ điển, nghe nội dung hội thoại, tra cứu động từ bất quy tắc và các câu thông dụng.

Ứng dụng được thiết kế cài đặt trên các thiết bị di động hệ điều hành Android. Đây là một hệ điều hành mã nguồn mở, là nền tảng điền thoại thông minh phổ biến nhất trên thế giới, được nhiều công ty công nghệ và nhà phát triển lựa chọn khi cần một hệ thống không nặng nề, có khả năng tinh chỉnh, giá rẻ, chạy trên các thiết bị công nghệ cao[3].

Để sử dụng ứng dụng AI English, thiết bị yêu cầu cần có kết nối mạng.

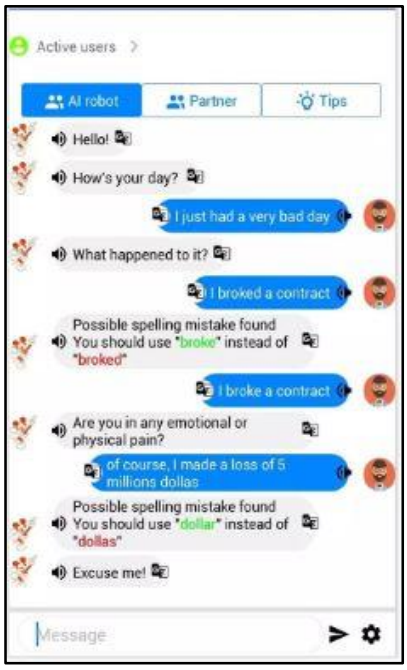


Hình 3.5. Giao diện khởi tạo của AI English

Người sử dụng lựa chọn các thông tin cá nhân sau đó click **Save**.

3.2.1. Giao diện chức năng hội thoại (Chatbot)

Giao diện ứng dụng khi thực hiện các đoạn hội thoại giữa người và máy sẽ hiển thị như sau:



Hình 3.6. Giao diện cuộc hội thoại ứng dụng AI English

Giao diện gồm các phần sau đây:

- Thanh nhập dữ liệu để người dùng nhập văn bản, một nút bấm để thực hiện gửi yêu cầu từ người dùng đến ứng dụng.

- Bên phải khung hinhg hiển thị dữ liệu người dùng nhập vào.

- Bên trái khung hình hiển thị trả lời từ ứng dụng.

- Các nút bấm hỗ trợ người dùng đọc dữ liệu bằng tiếng Anh và dịch dữ liệu sang tiếng Việt.

3.2.2. Giao diện chức năng tra cứu từ điển

Giao diện chức năng này được thực hiện trực tiếp trên các đoạn hội thoại, khi người sử dụng muốn tra cứu nghĩa của một từ nào đó.

Hình 3.7. Giao diện chức năng tra cứu từ điển

3.2.3. Giao diện chức năng kiểm tra chính tả, ngữ pháp

Giao diện chức năng này vẫn ở dạng đoạn hội thoại tương tự chức năng hội thoại, cụ thể:

Hình 3.8. Giao diện chức năng kiểm tra ngữ pháp

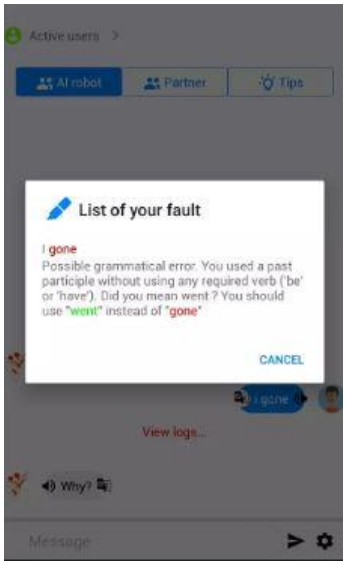
Giao diện chức năng này có một số sự khác biệt với chức năng hội thoại như sau:

- Bên trái khung hình hiển thị xử lý lỗi của câu từ người dùng.

- Phần chữ được in đỏ thể hiện lỗi sai của câu.

- Bên cạnh khung hình có một nút để người dùng có thể xem chi tiết lỗi và sửa lỗi.

Khi người dùng muốn hiển thị lỗi và sửa lỗi, ứng dụng sẽ có giao diện như sau:



Hình 3.9. Giao diện chi tiết lỗi và sửa lỗi

Cụ thể, giao diện chi tiết sẽ gồm 2 phần: Câu lỗi và câu sửa lỗi. Phần lỗi sẽ được tô đỏ và phần sửa lỗi sẽ được to xanh để phân biệt.

Ngoài ra, ứng dụng còn hỗ trợ người dùng với một bảng động từ bất quy tắc và danh sách, cách sử dụng các cụm từ thông dụng.

Hình 3.10. Bảng động từ bất quy tắc

Hình 3.11. Các cụm từ thông dụng

3.3. Đánh giá ứng dụng

Ứng dụng AI English đã cơ bản giải quyết được vấn đề kiểm tra ngữ pháp và phân tích cây cú pháp của câu trong tiếng Anh. Bằng cách sử dụng các đoạn hội thoại và giao diện đơn giản, ứng dụng tạo có thể giao tiếp với người dùng một cách dễ dàng và tiện lợi. Ứng dụng có thể giúp người dùng kiểm tra các câu đơn giản cũng như xem xét cú pháp câu để nâng cao khả năng về cách kết hợp các thành phần câu. Ngoài ra, ứng dụng cũng là một ứng dụng hội thoại có thể giúp người dùng trò chuyện để nâng cao vốn kiến thức.

Tuy nhiên, ứng dụng vẫn còn một số vấn đề như:

- Do kết hợp nhiều mã nguồn mở và phương pháp nên ứng dụng còn hạn chế về mặt xử lý cùng lúc nhiều yêu cầu như cả kiểm tra ngữ pháp, cả kiểm tra cú pháp và tích hợp cùng hội thoại.

- Thời gian xử lý cú pháp và ngữ pháp vẫn tương đối dài, đặc biệt đối với nhưng câu dài, gồm nhiều thành phần.

- Giao diện phần chi tiết cú pháp câu có thể gây khó hiểu với những người dùng không quen thuộc với các thành phần câu cũng như viết tắt của các thành phần câu đó.

- Với tính năng nhận dữ liệu đầu vào là âm thanh, ứng dụng yêu cầu phụ thuộc vào phần cứng điện thoại với chức năng voice tốt và điều kiện sử dụng là môi trường yên tĩnh.

- Đánh giá hiệu quả: với 100 câu tiếng Anh có lỗi là dữ liệu đầu vào thì ứng dụng đã phát hiện ra 81 câu bị lỗi, đạt 81%.

KẾT LUẬN

Đóng góp của luận văn

Trong thực tế, việc kiểm tra ngữ pháp và phân tích cú pháp câu tiếng Anh có thể được áp dụng trong nhiều ứng dụng như chia động từ, kiểm tra ngoại ngữ, học ngữ pháp trong tiếng Anh. Trong quá trình nghiên cứu về xây dựng ứng dụng hỗ trợ kiểm tra ngữ pháp tiếng Anh, luận văn đã đạt được một số nội dung sau:

- Luận văn đã tìm hiểu một số định nghĩa cơ bản về ngữ pháp trong tiếng Anh, một số hướng tiếp cận cơ bản bài toán kiểm tra ngữ pháp.

- Tìm hiểu cách tiếp cận PCFGs và áp dụng thuật toán CKY trong bài toán phân tích cú pháp câu.

- Giới thiệu một số mã nguồn mở hỗ trợ xây dựng Chatbot và kiểm tra ngữ pháp. Cài đặt ứng dụng đơn giản giải quyết bài toán kiểm tra ngữ pháp với tập luật tự huấn luyện.

Hạn chế của luận văn

Trong quá trình hoàn thành luận văn và ứng dụng, mặc dù đã đạt được một số kết quả nhất định trong bài toán phân tích ngữ pháp và xây dựng cú pháp tiếng Anh, vẫn có những hạn chế nhất định:

- Phương pháp tiếp cận phụ thuộc nhiều vào tập luật, do vậy khi tập luật có kích thương đủ lớn, thời gian xử lý của ứng dụng là lớn. Hơn nữa, độ chính xác của kết quả cũng phụ thuộc vào độ chính xác của tập luật.

- Kết quả của quá trình phân tích cú pháp chỉ có thể đưa ra cây cú pháp nếu văn bản phù hợp, nhưng chưa thể đưa ra lỗi cụ thể đối với nhưng văn bản lỗi.

- Tập dữ liệu hội thoại sử dụng trong Chatbot còn đơn giản.

- Ứng dụng mới chỉ tập trung vào mặt cú pháp, chưa thể hiện được về mặt ngữ nghĩa của câu.

Hướng phát triển

Với sự phong phú và cấp thiết của tiếng Anh, luận văn có nhiều hướng có thể phát triển tiếp tục:

- Tìm hiểu phương pháp tối ưu tập luật hiện tại, mở rộng tập luật, giảm thiểu thời gian với những tập luật gán nhãn, nâng cao hiệu quả phát hiễn lỗi của ứng dụng.

- Bổ sung tập dữ liệu hội thoại cho hệ thống Chatbot, phong phú về chủ đề.

- Phát triển các tình năng hỗ trợ học các ngữ pháp cụ thể như các dạng câu chủ động, bị động hay các thì trong tiếng Anh.

- Tìm hiểu phương pháp xác định lỗi cụ thể với bài toán phân tích cú pháp câu với những câu lỗi và đưa ra gợi ý sửa lỗi.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Đỗ Bá Lâm, Lê Thanh Hương, *Xây dựng hệ thống phân tích cú pháp tiếng Việt sử dụng văn phạm HPSG*, http://www.jaist.ac.jp/~bao/VLSP-text/ICTrda08/ICT08-VLSP-SP85-2.pdf.

2. Hoàng Đức Thịnh, *Ứng dụng mã nguồn mở AIML xây dựng hệ thống Chatbot trợ giúp phương pháp học tập cho sinh viên ngành kỹ thuật*, Tóm tắt luận văn thạc sĩ, Đại học Đà Nẵng, 2011.

3. Trần Võ Khôi Nguyên, Huỳnh Thái Dương, *Xây dựng ứng dụng hỗ trợ học tiếng Anh cho thiết bị di động chạy trên nền tảng Android*, Đồ án tốt nghiệp, ĐHQG TP. Hồ Chí Minh, ĐH Công nghệ thông tin, 2014.

4. Mai Ngọc Chừ, Vũ Đức Nghiệu, Hoàng Trọng Phiến, *Cơ sở ngôn ngữ học và Tiếng Việt*, NXB Giáo dục, 2002.

5. Phạm Thọ Hoàn, Phạm Thị Anh Lê, *Giáo trình Trí tuệ nhân tạo*, ĐH Sư phạm Hà Nội, 2011

6. Vương Hoài Thu, *Phân tích cú pháp tiếng Việt theo tiếp cận thống kê,* Khóa luận tốt nghiệp, Đại học Công nghệ - ĐHQG Hà Nội, 2009

7. Lê Anh Cường, *Xây dựng bộ phân tích cú pháp tiếng Anh trong hệ dịch tự động Anh Việt,* Luận văn Thạc sĩ, 2001

Tiếng Anh

8. Michael Collins, *ProbabilisticContext-Free Grammars (PCFGs)*, http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf.

9. Michael Collins, *Context Free Grammars*, http://aritter.github.io/courses/5525_slides/cfg.pdf

10. Daniel Naber, *A Rule-Based Style and Grammar Checker*, http://www.danielnaber.de/languagetool/download/style_and_grammar_checker.pdf.

Website tham khảo:

11. *Ngữ pháp và ngữ pháp học*, http://ngonngu.net/index.php?p=160.

12. http://alicebot.blogspot.com/2013/01/program-ab-aiml-20-reference.html

13. AIML Tutorial, https://www.tutorialspoint.com/aiml/

14. https://123doc.org//document/2761191-tri-tue-nhan-tao-mot-phuong - dien-cua-van-hoa-ung-dung.htm

15. http://hnue.tailieu.vn/doc/giao-trinh-tri-tue-nhan-tao-pham-tho-hoan - pham-thi-anh-dai-hoc-su-pham-ha-noi-283064.html

16. http://www.dostquangtri.gov.vn/Upload/Thongtinchuyende/20170808-10081156.pdf

17. http://www.dostquangtri.gov.vn/Upload/Thongtinchuyende/20170808-10082810.pdf

18. http://startup.vitv.vn/tin-chu/21-09-2016/tri-tue-nhan-tao-la-gi-nguon-goc-va-mot-so-ung-dung-cua-tri-tue-nhan-tao-trong-t/1321

19. Hồ Tú Bảo, Trí tuệ nhân tạo và chặng đường 50 năm www.jaist.ac.jp/~bao/Writings/AI50years.pdf

20. Bộ sưu tập câu và bản dịch, https://tatoeba.org/vie/

21. Bộ phân tích cú pháp của Stanford, http://nlp.stanford.edu:8080/parser/