

Đồ án tốt nghiệp: Xây dựng mô hình học sâu trong phát hiện và sửa lỗi chính tả trong tiếng Việt

Sinh viên thực hiện: Đỗ Đức Tiến

Ngày 10 tháng 1 2024

Mục lục

1	Giới thiệu và tổng quan đề tài	7
1.1	Tính cấp thiết đề tài	7
1.2	Tổng quan và mô tả đề tài	8
1.3	Mục đích đề tài	9
1.4	Phương pháp nghiên cứu	10
2	Các nghiên cứu liên quan đến tiếng Việt	13
3	Cơ sở lý thuyết tiếng Việt	14
3.1	Giới thiệu lý thuyết của tiếng Việt	14
3.1.1	Khái niệm về từ	14
3.1.2	Hình thái về từ tiếng Việt	14
3.1.3	Khái niệm văn bản	16
3.2	Chính tả tiếng Việt	16
3.2.1	Lỗi chính tả	16
3.2.2	Nguyên nhân gây lỗi chính tả	17
3.2.3	Phân loại lỗi chính tả	18
3.2.4	Phát hiện và sửa lỗi chính tả	18
3.3	Một số phương pháp kiểm thử	19
4	Cơ sở lý thuyết thuật toán	20
4.1	Tiền xử lý dữ liệu	20
4.1.1	Mô hình N-gram	20
4.1.2	Làm mịn (Smoothing)	24
4.1.3	Tách từ tiếng Việt	25
4.2	Phương pháp đánh giá	26
4.2.1	Accuracy - Độ chính xác	26
4.2.2	Confusion Matrix	26
4.2.3	Precision - Recall	27
4.2.4	F1-Score	28
4.2.5	ROC	28
4.3	Thuật toán Logic mờ (Fuzzy Logic)	30
4.3.1	Khái niệm	30
4.3.2	Công thức	30
4.3.3	Công thức Defuzzication	31
4.3.4	Ứng dụng thực tế	33
4.3.5	Ưu và nhược điểm của Fuzzy Logic	33
4.4	Thuật toán TF-IDF	34

4.4.1	Định nghĩa	34
4.4.2	Công thức và ví dụ	34
4.4.3	Ứng dụng thực tế	36
4.4.4	Ưu và nhược điểm của TF-IDF	36
4.5	Mô hình học sâu	37
4.5.1	Mạng thần kinh hồi quy (RNN)	37
4.5.2	Mô hình LSTM	45
5	Giải pháp đề xuất	51
5.1	Thu thập dữ liệu	51
5.2	Tiền xử lý dữ liệu	52
5.2.1	Thực hiện, thu thập và xử lý dữ liệu	52
5.2.2	Công cụ xử lý	54
5.3	Phương pháp đề xuất	54
6	Triển khai thực nghiệm	56
6.1	Môi trường thực nghiệm	56
6.2	Kết quả thực nghiệm	56
6.2.1	Đánh giá	56
6.2.2	Kiểm thử	58
7	Kết luận	62
7.1	Kết luận	62
7.2	Hướng phát triển	62

Lời nói đầu

- Ngày này trong thời đại công nghệ 4.0, mọi công nghệ đã được áp dụng trong đời sống rất nhiều, đặc biệt là trí tuệ nhân tạo. Nó không chỉ làm cuộc sống trở lên tốt mà còn xử lý được những công việc khó khăn tốn nhiều thời gian. Học máy là một lĩnh vực thuộc trí tuệ nhân tạo. Gần đây, học máy đang dẫn đầu xu thế và tạo nên những thay đổi vượt bậc trong trí tuệ nhân tạo nói chung và công nghệ thông tin nói riêng.
- Trong Đồ án tốt nghiệp này em đã thực hiện một dự án thực tế liên quan đến nghiên cứu một số mô hình học sâu phát hiện và sửa lỗi chính tả trong ngôn ngữ tiếng Việt có sử dụng học máy (Machine Learning) và học sâu (Deep Learning). Đây là một trong những chủ đề tương đối quan trọng trong xử lý ngôn ngữ tự nhiên.
- Em gửi lời cảm ơn chân thành đến cô **Roãn Thị Ngân**, giảng viên bộ môn Khoa học máy tính trường Đại học Xây dựng Hà Nội đã hướng dẫn, giúp đỡ em để em có thể hoàn thiện được Đồ án tốt nghiệp với đề tài nghiên cứu và xây dựng mô hình phát hiện lỗi chính tả, trong quá trình thực hiện làm Đồ án tốt nghiệp em do thời gian và khả năng có hạn nên không thể tránh khỏi được những sai sót, rất mong nhận em được sự góp ý của cô để Đồ án tốt nghiệp của em có thể hoàn thiện tốt hơn.
- Em xin bày tỏ lòng biết ơn sâu sắc đến trường Đại học Xây Dựng Hà Nội đã cung cấp một môi trường học tập tuyệt vời và các nguồn tài nguyên cần thiết để em thực hiện đồ án tốt nghiệp. Sự hỗ trợ và sự đánh giá cao của trường đã tạo điều kiện thuận lợi cho sự phát triển cá nhân và chuyên môn của em.
- Một lần nữa, em xin chân thành cảm ơn sự hỗ trợ và đóng góp của thầy, các bạn bè, gia đình và trường Đại học Xây Dựng Hà Nội. Sự giúp đỡ của mọi người đã giúp em hoàn thành đồ án tốt nghiệp một cách thành công và đạt được những thành tựu quan trọng.

Em xin chân thành cảm ơn!

Nhận xét người hướng dẫn

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Hà Nội, ngày, tháng, năm

Xác nhận giảng viên hướng dẫn

Nhận xét người phản biện

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Hà Nội, ngày, tháng, năm

Xác nhận giảng viên phản biện

Ghi chú - Chú thích

Tên ký tự	Tên tiếng Việt	Tên tiếng Anh
LSTM	Mạng bộ nhớ dài ngắn	Long-Short Term Memory
RNN	Mạng thần kinh hồi quy	Recurrent Neural Network
BiRNN	Mạng thần kinh hồi quy hai chiều	Bidirectional Recurrent Neural Network
BiLSTM	Mạng bộ nhớ dài ngắn hai chiều	Bidirectional Long-Short Term Memory
DL	Học sâu	Deep Learning
ML	Học máy	Machine Learning
NN	Mạng neuron	Neural Network
TF	Tần số xuất hiện của 1 từ trong 1 văn bản	Term Frequency
IDF	Tần số nghịch của 1 từ trong tập văn bản (corpus)	Inverse Document Frequency
TF-IDF	Thống kê số học phản ánh tầm quan trọng của một từ đối với một văn bản	Term Frequency–Inverse Document Frequency
OCR	Nhận dạng ký tự quang học	Optical Character Recognition
NER	Nhận dạng thực thể có tên	Named Entity Recognition
ASR	Nhận dạng giọng nói tự động	Automatic Speech Recognition
ROC	Đường cong đặc trưng hoạt động của bộ thu nhận	Receiver Operating Characteristic
MOM	Giá trị trung bình tối đa	Mean of Maximum Method
COA	Phương pháp khoảng trung tâm	Center of Area Method

Chương 1

Giới thiệu và tổng quan đề tài

1.1 Tính cấp thiết đề tài

- Ngày nay với sự phát triển vượt bậc của ngành Công nghệ thông tin, con người đã đạt được nhiều thành tựu to lớn trong việc giải quyết các bài toán thực tiễn. Chính sự phát triển của công nghệ thông tin đã đem lại cho thế giới những bộ mặt mới: Nền kinh tế tri thức, hợp tác toàn cầu, những công việc quản lý, vận hành những hoạt động từ vi mô tới vĩ mô của doanh nghiệp, tin học hóa quy trình hành chính, điều khiển tác nghiệp, giải trí, liên lạc, trợ giúp,... là những ứng dụng tiêu biểu của Công nghệ thông tin và truyền thông. Một trong các lĩnh vực khoa học ứng dụng thành tựu đó đang thu hút rất nhiều sự quan tâm của các nhà khoa học đó là Xử lý ngôn ngữ tự nhiên. Ứng dụng của xử lý ngôn ngữ tự nhiên trong rất nhiều lĩnh vực như: Dịch máy, điều khiển, nhận dạng, hệ hỗ trợ ra quyết định ..., đem lại lợi ích tối đa cho con người.
- Ngôn ngữ là một phần quan trọng của đời sống, là phương tiện truyền tải thông tin trong đời sống. Trong thời đại bùng nổ thông tin hiện nay thì ngôn ngữ đóng vai trò hết sức quan trọng, đặc biệt là ngôn ngữ viết. Khi viết, đôi khi ta mắc phải những lỗi sai chính tả. Chữ quốc ngữ là thứ ngữ ghi âm nên một số âm tiết rất dễ nhầm lẫn, khó phân biệt rõ ràng. Ngôn ngữ nói ở những vùng khác nhau thì lại có những đặc điểm khác nhau. Những điểm khác nhau này rất dễ gây ra những lỗi chính tả khi viết nếu người viết không để ý sử dụng tiếng Việt. Những thao tác chuyển thông tin ở dạng văn bản khác nhau cũng có thể gây ra lỗi chính tả. Khi ghi lại lời nói của người khác mà người đó sử dụng giọng địa phương cũng có thể dẫn đến những lỗi chính tả. Quét các văn bản giấy thành văn bản điện tử, sử dụng chương trình nhận dạng chữ, cũng có thể dẫn đến lỗi chính tả do chương trình nhận dạng nhầm lẫn ... Văn bản dễ bị sai chính tả do nhiều yếu tố khách quan. Để kiểm tra lỗi chính tả những văn bản này đòi hỏi nhiều công sức và thời gian, đặc biệt là khi khối lượng văn bản bùng nổ như hiện nay. Do đó cần có một công cụ hỗ trợ kiểm tra lỗi chính tả, giúp nhanh chóng phát hiện lỗi chính tả và đề nghị khắc phục.
- Trong thời đại 4.0, máy tính được tận dụng để giảm thiểu công sức của con người, đồng thời tăng tính hiệu quả. Tin học đã được áp dụng trong nhiều lĩnh vực khác nhau và chứng tỏ tính hiệu quả của nó. Những ứng dụng kiểm tra lỗi chính tả hiện có như VietRes, VietSpell, ... hiện vẫn còn khá đơn giản hoặc chưa hiệu quả, chưa đáp ứng được nhu cầu của thực tế. Trong bài toán này nhóm 06 muốn trình bày kết quả tìm hiểu về bài toán sửa lỗi chính tả tiếng việt trong văn bản.

1.2 Tổng quan và mô tả đề tài

- Bài toán sửa lỗi chính tả là một bài toán khá phức tạp, được không ít đơn vị nghiên cứu, phát triển và nó có tính ứng dụng cao, đặc biệt là trong các ứng dụng soạn thảo hay nhận dạng văn bản. Chương trình sửa lỗi chính tả cần có hai chức năng chính, cơ bản là chỉ ra lỗi sai và đưa ra gợi ý sửa lỗi. Tuy nhiên, các chức năng kiểm lỗi chính tả được tích hợp nhiều trong ứng dụng soạn thảo tiếng Việt hiện nay (Vietkey, Unikey, ...) không đưa ra gợi ý cho người dùng lựa chọn.
- Ngoài ra trong đề tài nghiên cứu có thể áp dụng được trong đời sống như là có thể tích hợp vào nhận dạng giọng nói tự động (ASR) hoặc là nhận dạng ký tự quang học (OCR), hoặc là nhận dạng thực thể có tên (NER) để tăng cường nhận dạng trong việc sửa lỗi chính tả tiếng Việt
- Để giải quyết bài toán này, hiện nay có một số phần mềm đã được nghiên cứu và phát hiện và sửa lỗi sai chính tả như sau:
 - **Google Docs:** là một ứng dụng dùng để hỗ trợ soạn thảo trực tuyến, được cung cấp bởi nhà phát hành Google. Google Docs đặc biệt ở chỗ cho phép người dùng kiểm tra chính tả một cách nhanh chóng ngay trên giao diện làm việc.
 - * **Các tính năng nổi bật:**
 - Sử dụng miễn phí ở mọi nơi khi có mạng Internet.
 - Cung cấp đầy đủ công cụ soạn thảo.
 - Tích hợp nhiều tiện ích và font chữ.
 - Hỗ trợ chia sẻ và sửa đổi cùng nhiều người.
 - Hỗ trợ chức năng nói để nhập dữ liệu.
 - * **Nhược điểm:**
 - Cần đăng ký tài khoản để sử dụng.
 - Chèn ảnh vào giao diện khó khăn dễ xảy ra lỗi.
 - Thiếu font chữ dành cho tiếng Việt.
 - **VSpell:** là một trang website đầu tiên kiểm tra chính tả tiếng Việt từ năm 1990. Bạn không cần phải tải về hay cài đặt mà vẫn có thể soát lỗi chính tả tiếng Việt chính xác.
 - * **Các tính năng nổi bật:**
 - Soát lỗi tệp tin Word, Text và HTML.
 - Soát lỗi trang web bằng URL.
 - Hệ thống tự động cập nhật từ điển.
 - * **Nhược điểm:**
 - Quét chính tả từ trên giao diện còn hạn chế.
 - **VCatSpell:** là phần mềm kiểm tra lỗi chính tả tiếng Việt đầu tiên và được ra mắt chính thức vào năm 1990. Cho đến hiện tại, phần mềm vẫn sở hữu nhiều điểm cộng được người dùng đánh giá cao.
 - * **Các tính năng nổi bật:**
 - Hỗ trợ trên mọi phiên bản của Microsoft Word.
 - Hỗ trợ tạo ra những đoạn văn bản hay báo cáo dài trang.

* **Nhược điểm:**

- Giao diện sơ sài, lỗi thời

– **Tummo Spell:** là phiên bản mới nhất của phần mềm kiểm tra lỗi chính tả văn bản Word, Excel nhanh chóng và hiệu quả. Đồng thời cũng giúp bạn sửa lỗi nhanh hơn.

* **Các tính năng nổi bật:**

- Tự động thêm vào giao diện Microsoft Word sau khi cài đặt.
- Nhanh chóng kiểm tra lỗi chính tả trong các file Word.
- Tìm kiếm, thay thế từ bị lỗi nhanh chóng.
- Cài đặt nhanh chóng, hoàn toàn miễn phí.

* **Nhược điểm:** Đôi lúc thông báo cả những từ đúng trong tập tin

– **Tiny Spell:** là một tiện ích nhỏ cho phép bạn kiểm tra và sửa lỗi chính tả dễ dàng và nhanh chóng trong bất kỳ ứng dụng Windows nào.

* **Các tính năng nổi bật:**

- Công cụ phát hiện lỗi chính tả nhanh, hiệu quả.
- Cung cấp các gợi ý phía dưới giúp người dùng tự sửa lỗi chính tả.
- Có thể kiểm tra lỗi chính tả ở bất kỳ ứng dụng Windows nào.
- Kèm theo bộ từ điển tiếng Anh Mỹ với hơn 110.000 từ.

* **Nhược điểm:** Giao diện rắc rối, khó sử dụng.

Bài toán sửa lỗi chính tả tiếng Việt bao gồm như sau:

- Đầu vào
 - Tập chuỗi các từ đầu vào $X = \{x_1, x_2, \dots, x_n\}$ với một vài từ sai chính tả x_i bất kì.
- Đầu ra
 - Tập chuỗi các từ đầu ra $Y = \{y_1, y_2, \dots, y_n\}$ với y_i là từ đúng chính tả được sửa.

Trong bài toán này, với mỗi từ sai chính tả x_i ta cần sửa thành một từ đúng y_i , tức là ta cần phải xây dựng một hàm $f: X \rightarrow Y$ thỏa mãn $f(x_i) = y_i$

1.3 Mục đích đề tài

Mục đích của đề tài nghiên cứu và phát hiện sửa lỗi chính tả trong tiếng Việt là cải thiện chất lượng văn bản tiếng Việt bằng cách tự động phát hiện và sửa các lỗi chính tả phổ biến, sau đây là một số mục đích

1. **Tăng cường chính xác ngôn ngữ:** Đề tài nhằm giúp người viết và người đọc hiểu và truyền đạt thông điệp một cách chính xác hơn. Bằng cách phát hiện và sửa lỗi chính tả, văn bản trở nên dễ đọc hơn và không gây hiểu lầm.
2. **Tiết kiệm thời gian và công sức:** Việc phát hiện và sửa lỗi chính tả thủ công có thể tốn nhiều thời gian và công sức. Đề tài này nhằm áp dụng các phương pháp và công nghệ tự động để tiết kiệm thời gian và giảm công sức trong việc sửa lỗi chính tả.

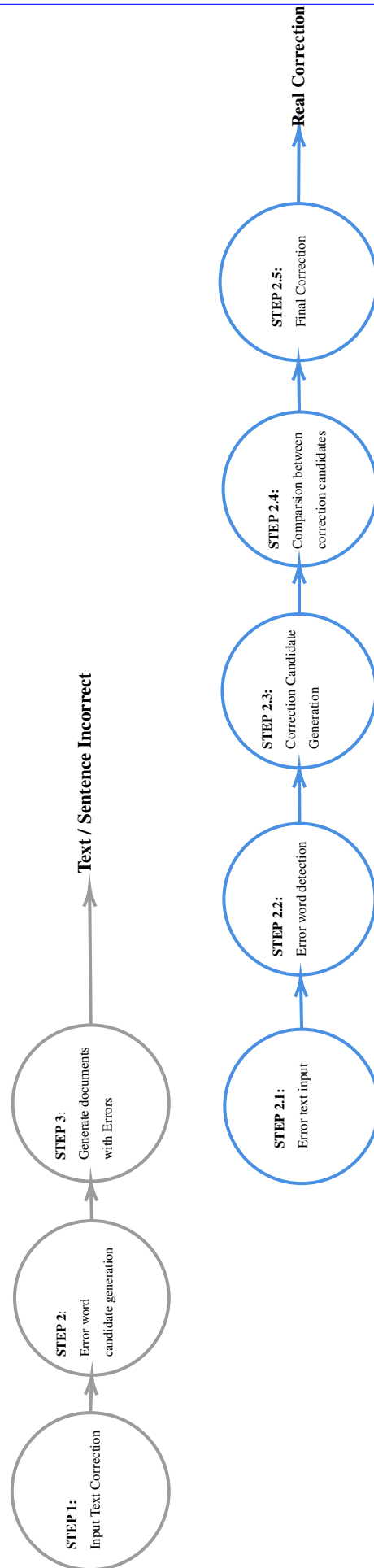
3. **Nâng cao hiệu quả viết:** Đề tài hướng đến việc cung cấp gợi ý và phản hồi tức thì cho người viết, giúp họ nhận ra và sửa lỗi chính tả một cách nhanh chóng. Điều này góp phần nâng cao hiệu quả viết của người sử dụng và cải thiện chất lượng văn bản.
4. **Hỗ trợ các ứng dụng xử lý ngôn ngữ tự nhiên:** Một hệ thống phát hiện và sửa lỗi chính tả có thể được tích hợp vào các ứng dụng xử lý ngôn ngữ tự nhiên khác như chatbot, máy dịch, hoặc công cụ soạn thảo văn bản. Điều này giúp cải thiện tính chính xác và trải nghiệm người dùng của các ứng dụng này.
5. **Cải thiện giao tiếp và ứng dụng trong kỹ thuật ngôn ngữ:** Việc phát hiện và sửa lỗi chính tả trong tiếng Việt giúp cải thiện giao tiếp chính xác và chuyên nghiệp, đặc biệt trong các lĩnh vực kỹ thuật, kỹ năng viết và gửi thư điện tử.
6. **Tạo ra tài liệu đáng tin cậy:** Bằng cách giảm thiểu lỗi chính tả, đề tài này giúp tạo ra tài liệu đáng tin cậy và chính xác hơn. Điều này đặc biệt quan trọng trong các tài liệu như sách giáo trình, báo cáo, văn bản học thuật và các nguồn thông tin trực tuyến.
7. **Tăng cường sự chính xác trong công việc:** Việc cung cấp một công cụ phát hiện và sửa lỗi chính tả tự động giúp người sử dụng làm việc một cách chính xác hơn. Điều này có thể áp dụng trong các lĩnh vực như viết bài, biên tập, dịch thuật và các công việc liên quan đến văn bản.
8. **Đáp ứng nhu cầu của người dùng:** Lỗi chính tả có thể làm giảm trải nghiệm đọc và gây khó khăn cho người sử dụng. Để đáp ứng nhu cầu của người dùng, đề tài này nhằm cung cấp một công cụ hỗ trợ tự động để phát hiện và sửa các lỗi chính tả, cải thiện trải nghiệm đọc và tăng cường sự hài lòng của người dùng.
9. **Bảo vệ danh tiếng và uy tín:** Lỗi chính tả trong văn bản có thể làm giảm độ tin cậy và uy tín của tác giả hoặc tổ chức. Đề tài này nhằm giúp ngăn chặn và sửa lỗi chính tả, đảm bảo rằng các văn bản được viết và công bố với mức độ chính xác cao, bảo vệ danh tiếng và uy tín của cá nhân và tổ chức.

Những mục đích này nhằm cải thiện chất lượng ngôn ngữ, tạo ra tài liệu đáng tin cậy, tăng cường sự chính xác và đáp ứng nhu cầu của người dùng trong việc phát hiện và sửa lỗi chính tả trong tiếng Việt.

1.4 Phương pháp nghiên cứu

- Trong đồ án sửa lỗi chính tả này, nhóm sẽ lần lượt thực hiện các bước sau: Đầu tiên nhóm sẽ thu thập bộ dữ liệu để huấn luyện. Tiếp theo, nhóm sẽ tiến hành quy trình tiền xử lý dữ liệu để làm sạch văn bản và đưa văn bản này thông qua một kênh nhiễu để tạo ra các lỗi chính tả một cách ngẫu nhiên.
- Mục tiêu của đề tài là với dữ liệu văn bản mới đầu vào, ta cũng thực hiện lần lượt bước tiền xử lý dữ liệu như với tập huấn luyện. Sau đó đưa tập văn bản mới vào mô hình để dự đoán sửa lỗi các từ sai chính tả.
- Các quy trình đó bao gồm (được giải thích dưới đây và dựa theo bài báo [2]):
 - Nhập vào một câu hoặc là một đoạn văn đúng chính tả

- Thiết lập các hàm tạo nhiễu để tạo các lỗi trong câu từ của đoạn văn đó, các lỗi hàm thì sẽ được đề cập rõ hơn ở chương 6
- Hiển thị kết quả của câu đó đã bị lỗi sai chính tả với các hàm đã được thiết lập
- Từ câu tạo nhiễu đó sẽ bắt đầu tiến hành nhận diện các loại lỗi trong chính tả đã thiết lập như ở các bước trên
- Đề xuất các từ đúng nhất so với các từ sai đó và sửa lại cho đúng
- Trả về kết quả là một từ đúng chính tả



Hình 1.1: Quy trình giải quyết bài toán

Chương 2

Các nghiên cứu liên quan đến tiếng Việt

Ngày nay, đã có nhiều đề tài nghiên cứu một số thuật toán học sâu và học máy trong việc nhận diện và sửa lỗi chính tả trong tiếng Việt, và các nhà nghiên cứu đã sử dụng học sâu (DL) và học máy (ML) để tiến hành sau đây là một số nghiên cứu điển hình

- [5] Việt Hoàng cùng với các cộng sự của mình đã tạo thêm nhiều bằng cách tạo nhiều với hàm bằng teen code và các lỗi bằng gõ phím, và sử dụng mô hình LSTM hai chiều (BiLSTM) để huấn luyện mô hình, mặc dù có độ chính xác đến 99% nhưng nhược điểm ở đây là hàm tạo nhiễu ở đây chỉ tạo được nhiễu ở mức nhỏ và sửa lỗi chính tả cũng ở mức nhỏ nên không tạo được hiệu quả khi gặp phải các lỗi khác.
- [6] Lê Văn Hoàng cũng với cộng sự của mình đã thiết lập một tập dữ liệu thông tin bằng tệp CSV, đồng thời tạo nhiễu cùng với các trường hợp khác nhau như: Xóa dấu câu, thay thế bằng bàn phím, xóa, thêm bớt các chữ cái. Có tạo lỗi cụ thể hơn, tuy nhiên hàm tạo lỗi được lập ra là hàm ngẫu nhiên và chỉ lặp trong 1 trường hợp nhất định nên sẽ không tạo được hiệu quả khi sửa lỗi
- [13] Dinh-Truong-Do và các cộng sự của mình đã sử dụng mô hình Transformer để huấn luyện mô hình để nhận diện và sửa lỗi chính tả với độ chính xác của nhận diện lên đến 93% và sửa lỗi có độ chính xác lên đến 76%
- [18] Hiếu Trần cùng với các cộng sự đã sử dụng mô hình phân cấp của Transformer cho encoders để huấn luyện với nguồn dữ liệu thu thập từ trang Wikipedia tiếng Việt cho độ chính xác khi nhận diện là 66.96% và sửa lỗi có độ chính xác khi sửa lỗi là 96.01%
-

Chương 3

Cơ sở lý thuyết tiếng Việt

3.1 Giới thiệu lý thuyết của tiếng Việt

3.1.1 Khái niệm về từ

- Trong quá trình học tập và sử dụng ngôn ngữ trong đời sống hằng ngày, mỗi chúng ta đều quen thuộc với khái niệm về ”từ”. Nhưng để định nghĩa được chính xác từ là gì hoàn toàn không phải là một vấn đề đơn giản. Trong ngành ngôn ngữ học, đã có hàng trăm định nghĩa về từ được đưa ra, nhưng hầu như chưa có một định nghĩa nào có thể bao quát hết được mọi vấn đề liên quan đến khái niệm ”từ”. Theo công trình của **PGS.TS. Đinh Điền**, có một số khái niệm tiêu biểu sau đây về từ:
 - Theo **L Bloomfield** thì: ”từ là một hình thái tự do nhỏ nhất”.
 - B.Golovin quan niệm: ”từ là đơn vị nhỏ nhất có nghĩa của ngôn ngữ, được vận dụng độc lập, tái hiện tự do trong lời nói để xây dựng nên câu”.
 - Còn Solncev thì lại quan niệm: ”Từ là đơn vị ngôn ngữ có tính hai mặt : âm và nghĩa. Từ có khả năng độc lập về cú pháp khi sử dụng trong lời”.
- Trong tiếng Việt, cũng có nhiều định nghĩa về từ như:
 - Theo nhà nghiên cứu Ngôn ngữ học Trương Văn Trinh và Nguyễn Hiến Lê thì: ”Từ là âm có nghĩa, dùng trong ngôn ngữ để diễn tả một ý đơn giản nhất, nghĩa là ý không thể phân tích ra được”.
 - PGS. Nguyễn Kim Thản thì định nghĩa: ”Từ là đơn vị cơ bản của ngôn ngữ, có thể tách khỏi các đơn vị khác của lời nói để vận dụng một cách độc lập và là một khối hoàn chỉnh về ý nghĩa (từ vựng hay ngữ pháp) và cấu tạo”.
 - Theo Hồ Lê, ”Từ là đơn vị ngôn ngữ có chức năng định danh pgu liên kết hiện thực, hoặc chức năng mô phỏng tiếng động, có khả năng kết hợp tự do, có tính vững chắc về cấu tạo và tính nhất thể về ý nghĩa”.

3.1.2 Hình thái về từ tiếng Việt

Như trình bày trong phần trên, có rất nhiều định nghĩa về từ nhưng các nhà ngôn ngữ học vẫn chưa thống nhất quyết định chọn theo lối định nghĩa nào. Điều này cũng xảy ra trong tiếng Việt của chúng ta. Do vậy, với mục đích phục vụ thuận tiện cho việc xử lý tự động ngôn ngữ bằng máy tính, nhưng vẫn phù hợp với các định nghĩa về từ trong ngôn ngữ học đại cương cũng như tính đặc thù của ngôn ngữ đơn lập như tiếng Việt.

Hình vị tiếng Việt

- Đầu tiên, chúng tôi sử dụng quan niệm của công trình của PGS.TS. Đinh Điền như sau: tiếng là đơn vị cơ bản trong tiếng Việt vì nó có thể nhận diện tương đối dễ dàng bởi người bản ngữ cũng như nhận diện một cách tự động bởi máy tính. Xét về mặt kỹ thuật trên máy tính, ta cũng có thể thực hiện được các thao tác lưu trữ, xử lý, tìm kiếm và sắp xếp các tiếng một cách dễ dàng do số lượng cũng như chiều dài của các tiếng này là nhỏ (Trong tiếng Việt, có khoảng 9270 tiếng các loại, và chiều dài của mỗi tiếng cũng được giới hạn là 7 ký tự - nghiêng là tiếng dài nhất với 7 ký tự).
- Ngoài ra, tiếng còn được xem là “từ chính tả”. Tuy nhiên, nếu xét trên các tiêu chí của ngôn ngữ học, thì tiếng không thể được xem là một từ thực sự. Thậm chí, tiếng cũng chưa hoàn toàn đủ tư cách để được xem là “hình vị thực sự” vì chưa thỏa tiêu chí về nội dung (phải có ý nghĩa hoàn chỉnh). Vì vậy, trong đồ án này, chúng tôi dựa theo quan điểm của PGS.TS. Đinh Điền trong công trình là xem tiếng chỉ là “hình vị tiếng Việt”:
- Hình vị tiếng Việt ở đây phải được hiểu là: bên cạnh khái niệm hình vị như trong ngôn ngữ học đại cương, còn phải xét đến yếu tố hình tổ, là yếu tố thuần túy hình thức biểu hiện những kiểu quan hệ bên trong giữa các thành tố trong từ. Ta có thể gọi đây là những “thanh hình vị” hay “á hình vị”. Như vậy, trong tiếng Việt sẽ có 3 loại hình vị như sau:
 - **Hình vị gốc:** là những nguyên tố, đơn vị nhỏ nhất, có nghĩa, chúng có thể là hình vị thực (là những từ vựng) hay hình vị hư (ngữ pháp), chúng có thể đứng độc lập hay bị ràng buộc.
 - **Tha hình vị:** vốn cũng là hình vị gốc, nhưng vì mối tương quan với các thành tố khác trong từ mà chúng biến đổi đi về âm, nghĩa,... Tha hình vị bao gồm:
 - * **Tha hình vị láy nghĩa:** trong các từ ghép bội nghĩa, như: giá cả, hỏi han, tuổi tác,...; nhà cửa, yêu thương, ngược xuôi,...
 - * **Tha hình vị láy âm:** chum chim, đo đỏ,...; lé dé, đủng đỉnh,...
 - * **Tha hình vị định tính:** là các yếu tố phụ để miêu tả thuộc tính, như: xanh lè, tối om, cười khẩy,....
 - * **Tha hình vị tựa phụ tố:** là đơn vị hoạt động giống như những phụ tố (affix) trong các ngôn ngữ biến hình, như: giáo viên, hiện đại hoá, tân tổng thống,...
 - **Á hình vị:** là những chiết đoạn ngữ âm được phân xuất một cách tiêu cực, thuần túy dựa vào hình thức, không rõ nghĩa, song có giá trị khu biệt, làm chức năng cấu tạo từ. Ví dụ: dưa hấu, dưa gang, bí ử, đậu nành, cà niễng,....

Từ tiếng Việt

- Trong đồ án này, chúng tôi sử dụng nghĩa từ theo “từ được cấu tạo bởi những hình vị”. Theo công trình này, thì “từ tiếng Việt được cấu tạo bởi những hình vị tiếng Việt”.
- Từ tiếng Việt ở đây bao gồm: từ đơn, từ ghép, từ láy và từ ngẫu hợp.
- Xuất phát từ nhu cầu xử lý tự động ngữ liệu tiếng Việt bằng máy tính, PGS.TS. Đinh Điền đã đề nghị cách thức hình thức hoá các quan niệm về hình vị tiếng Việt và từ tiếng Việt nói trên trong công trình như sau:

- Do “hình vị tiếng Việt” cũng chính là từ chính tả (từng chữ độc lập), nên việc hình thức hoá rất đơn giản, không cần đặt ra. Trong ngữ liệu tiếng Việt cũng như tiếng Anh, đơn vị cơ bản được lưu cũng chính là từ chính tả này. Tuy nhiên, nếu chỉ lưu trữ ở cấp độ hình vị như vậy, thì lượng thông tin trong kho ngữ liệu sẽ rất hạn chế và chúng ta sẽ không thể khai thác hiệu quả vốn có của nó được.
- Để lưu trữ thông tin về ranh giới từ tiếng Việt, chúng tôi sử dụng khái niệm từ từ điển học. Từ từ điển học ở đây được định nghĩa là “những đơn vị mà căn cứ vào đặc điểm ý nghĩa của nó phải xếp riêng trong từ điển và có đánh dấu đây là đơn vị từ của ngôn ngữ”. Việc chọn lựa những từ nào sẽ đưa vào từ điển là hoàn toàn do các nhà ngôn ngữ hay người xây dựng kho ngữ liệu quyết định, dựa theo quan điểm về từ đã nêu trên. Trong đồ án này chúng tôi sử dụng từ điển tiếng Việt của công trình của **GS Hoàng Phê**.
- Do có nhiều thuật ngữ về từ” khác nhau (từ chính tả, từ từ điển học,...), vì vậy, từ đây trở về sau, thuật ngữ “từ” được sử dụng trong luận văn được quy ước là để chỉ “từ từ điển”.

3.1.3 Khái niệm văn bản

- Trong ngôn ngữ (language), văn bản là 1 thuật ngữ rộng nói về 1 thứ gì đó mà chứa các từ ngữ diễn đạt 1 sự việc.
- Trong ngôn ngữ học (linguistic), văn bản là 1 hoạt động giao tiếp, thi hành 7 nguyên tắc cấu thành cơ bản và 3 nguyên tắc điều khiển của văn bản học. Cả tiếng nói, ngôn ngữ viết hay ngôn ngữ thông thường đều có thể xem như văn bản trong ngôn ngữ học.
- Trong lý thuyết văn học, văn bản là 1 đối tượng (object) được nghiên cứu, dù nó là 1 cuốn tiểu thuyết, 1 bài thơ, 1 vở phim, 1 mẫu quảng cáo hay bất cứ thứ gì có thành phần thuộc về ký hiệu. Cách dùng rộng rãi thuật ngữ này được bắt nguồn từ sự xuất hiện của ký hiệu những năm 1960 và được củng cố vững chắc bằng những nghiên cứu văn hoá sau đó trong những năm 1980.
- Trong truyền thông các thiết bị di động, văn bản (hay tin nhắn văn bản) là 1 đoạn tin nhắn số hoá ngắn giữa những thiết bị.
- Trong tin học, văn bản liên hệ đến dữ liệu ký tự (character data), hay đến 1 trong những thành phần của chương trình trong bộ nhớ.
- Trong học thuật, văn bản thường được dùng như 1 hình thức viết tắt của sách giáo khoa.

3.2 Chính tả tiếng Việt

3.2.1 Lỗi chính tả

Trước hết cần phải hiểu chính tả là gì. Chính tả được hiểu là “phép viết đúng” hoặc “lỗi viết hợp với chuẩn”. Nói cách khác, chính tả là việc tiêu chuẩn hóa chữ viết của một ngôn ngữ. Yêu cầu cơ bản của chính tả là phải thống nhất cách viết cụ thể trên phạm vi toàn quốc và trong tất cả các loại hình văn bản viết. Tiếp theo cần hiểu thế nào là lỗi chính tả: lỗi chính tả là lỗi viết sai chuẩn chính tả bao gồm các hiện tượng vi phạm các quy định chính tả về viết hoa, viết tắt, dùng số và biểu thị chữ số và hiện tượng vi phạm diện mạo ngữ âm của từ thể hiện trên chữ viết, tức

chữ viết ghi sai từ, hay còn gọi là lỗi âm vị. Lỗi âm vị trong tiếng Việt thường thể hiện qua các dạng: lỗi âm vị âm đoạn tính và lỗi âm vị siêu âm đoạn tính. Lỗi âm vị âm đoạn tính bao gồm, lỗi sai về phụ âm đầu, âm đệm, âm chính, âm cuối. Lỗi âm vị siêu đoạn tính chính là hiện tượng viết sai thanh điệu.

3.2.2 Nguyên nhân gây lỗi chính tả

Có rất nhiều nguyên nhân dẫn tới thực trạng này nhưng chúng ta có thể quy về một số nguyên nhân chính sau đây:

1. Thứ nhất là do không nắm vững chính tự. Ví dụ, lẽ ra phải viết là ngành thì lại viết là ngànhh. Điều này có nguyên nhân sâu xa từ một số bất hợp lí của chữ quốc ngữ. Sự bất hợp lí này thể hiện như sau: không đảm bảo sự tương ứng một đối một giữa âm và chữ. Chẳng hạn, âm [k] có 3 cách ghi là c, k, q; con chữ g ghi âm [z] và âm [ɣ]. Có những nhóm hai, ba con chữ để ghi một âm vị: ph, ngh. Điều này làm người nghe lúng túng vì tại sao cùng đọc là [k] nhưng lúc thì viết là c, lúc thì viết là k, lúc lại viết là q, cùng đọc là /ŋ/ mà lúc viết là ng lúc lại viết là ngh. Đã có nhiều ý kiến đề nghị khắc phục những bất hợp lí này nhưng cho đến nay vì nhiều nguyên nhân khác nhau nó vẫn tồn tại.
2. Thứ hai là do không hiểu nghĩa. Tuy chính tả tiếng Việt là chính tả ngữ âm nhưng trên thực tế, muốn viết đúng, nhiều trường hợp phải nắm được ngữ nghĩa. Ví dụ: lẽ ra phải viết là giành (với nghĩa là tranh) thì lại viết là dành (với nghĩa là giữ lại để sau này dùng hoặc để riêng cho ai, cho việc gì) và ngược lại; lẽ ra phải viết là tham quan (tham là tham gia, tham dự, tìm tòi, nghiên cứu, tìm hiểu, quan là nhìn trực tiếp một cách kĩ lưỡng, tỉ mỉ, sâu sắc) thì lại viết là thăm quan; lẽ ra phải viết là khúc chiết (có nghĩa là có từng đoạn, từng ý, rành mạch và gãy gọn) thì lại viết là khúc triết.
3. Thứ ba là do không cập nhật những quy định chính tả hiện hành. Chẳng hạn: trước đây do đề cao sự cân đối của chữ viết nên dấu thanh được đánh vào âm đứng giữa trong âm tiết. Ví dụ hoá được viết là hóa, thúy được viết là thúy. Nhưng hiện nay, với quy định dấu phải đánh vào âm chính thì cách viết như trên đã lạc hậu. Hoặc trước đây, tên cơ quan, tổ chức viết khác so với hiện nay. Ví dụ, trước đây viết là Trường đại học bách khoa Hà nội, còn hiện nay viết là Trường Đại học Bách khoa Hà Nội. Do không cập nhật điều đó nên nhiều người đã viết theo quy định cũ dẫn đến sai chính tả.
4. Thứ tư là do ảnh hưởng của cách phát âm địa phương. Ví dụ phương ngữ Bắc Bộ không có ba âm quặt lưỡi || ʃ|| vì thế nhiều người gặp khó khăn khi phải viết các từ có chứa những phụ âm đầu ch – tr, r – d – gi, s – x. Người nói phương ngữ Bắc Trung Bộ lại nhầm giữa dấu hỏi () và dấu ngã (). Vì thế họ rất lúng túng khi gặp những từ có dấu hỏi và dấu ngã. Họ sẽ không hiểu: viết là mâu thuẫn đúng hay mâu thuẫn đúng. Cũng như vậy, phương ngữ Nam Bộ lại có vấn đề khi viết các âm đầu là v hay z, âm cuối là n hay ng, c hay t, viết dấu hỏi hay dấu ngã. Một số người sẽ rất lúng túng khi gặp những từ có chứa những phụ âm đầu, phụ âm cuối và thanh điệu này.
5. Thứ năm là do sự cầu thả của người viết. Biểu hiện của loại lỗi do nguyên nhân này rất phong phú. Ví dụ, viết hoa không theo quy tắc nào (Nguyễn thị Kim Liên, Hải phòng). Hoặc đang viết bình thường lại viết chữ to hơn nên vô tình cũng mắc lỗi viết hoa bừa bãi (Đây là ngày thứ hai tôi ở Hà Nội.) Hoặc sau dấu chấm không viết hoa. Hoặc hường lại viết là hương.

6. Thứ sáu là do ảnh hưởng của ngôn ngữ mạng. Ngôn ngữ mạng phù hợp với nhu cầu muốn giao tiếp nhanh, muốn thể hiện cá tính và sự cập nhật về công nghệ hiện đại của một số người, phần lớn là giới trẻ. Tuy nhiên, trong giao tiếp có nghi thức việc sử dụng ngôn ngữ này không phù hợp và khi viết sử dụng ngôn ngữ mạng sẽ bị coi là mắc lỗi chính tả. Ví dụ cần phải viết: ạ thì lại viết ah, ừ thì lại viết uh, được thì lại viết đk, trong thì lại viết (.)

3.2.3 Phân loại lỗi chính tả

- Lỗi nhận thức: hay xảy ra khi sử dụng các từ đồng âm, gần âm. Ví dụ: đổ tổ trong khi đáng ra phải là giổ tổ.
- Lỗi viết tắt: vô ý tạo ra hay nhiều từ không có nghĩa. Ví dụ: “kh” tắt cho từ không, ”hc” tắt cho học, ”t” có thể là tôi, tao,....
- Lỗi dùng từ lóng: là một từ ngữ không chính thức của một ngôn ngữ, thường được sử dụng trong đời sống thường ngày. Ví dụ “maj”, “thj”, ...
- Lỗi khi gõ bàn phím: Khi chúng ta gõ bàn phím trên máy tính thì có trường hợp gõ nhanh và đôi khi xảy ra nhầm lẫn khi gõ các chữ cái trong bàn phím. Như đường gõ nhầm thành đường, được thành đượđ, mơ thành mow

3.2.4 Phát hiện và sửa lỗi chính tả

- Với thời đại phát triển công nghệ như hiện nay, trên các trang mạng đã và đang có rất nhiều ứng dụng, phần mềm phát hiện lỗi chính tả.
- Giáo dục ý thức viết đúng chính tả cho người dân. Cần phải làm cho mọi người hiểu rằng viết đúng chính tả không chỉ thể hiện trình độ văn hóa mà còn thể hiện ý thức tôn trọng cộng đồng, lòng yêu quý đối với tiếng Việt của người viết. Còn viết sai chính tả ảnh hưởng nghiêm trọng tới giao tiếp của từng người dân, của toàn xã hội và rất nhiều trường hợp ảnh hưởng tới quốc gia, dân tộc.
- Tuyên truyền, phổ cập các chuẩn mực chính tả rộng rãi trong cộng đồng sử dụng tiếng Việt bằng các con đường khác nhau như qua các phương tiện thông tin đại chúng, qua nhà trường.
- Duy trì các biện pháp giúp người sử dụng tiếng Việt viết đúng chính tả. Ví dụ: Văn bản của các cơ quan, tổ chức và văn bản trên các phương tiện thông tin đại chúng phải tuyệt đối tuân thủ các quy định về chính tả để người dân coi đó là các văn bản mẫu và làm theo; Duy trì mục dạn vườn trên đài truyền hình, báo chí giúp người dân nâng cao kỹ năng chính tả.
- Có chính sách phát triển ngôn ngữ phù hợp trong bối cảnh tiếng Việt có nhiều sự biến đổi trước những biến động của thế giới. (Chẳng hạn chọn cách ứng xử phù hợp với ngôn ngữ mạng. Trong giao tiếp có nghi thức như khi làm các văn bản giấy tờ, trong học tập... không được sử dụng ngôn ngữ mạng. Giáo dục cho cá nhân ý thức rõ khi nào có thể sử dụng ngôn ngữ mạng, khi nào không được sử dụng ngôn ngữ mạng).

3.3 Một số phương pháp kiểm thử

- Chúng ta có thể tạm chỉ ra hai phương pháp chính đó là dựa vào luật và dựa vào thống kê. Các phương pháp dựa theo luật có ưu điểm là không tốn quá nhiều tài nguyên của thiết bị, tuy nhiên các chương trình sử dụng phương pháp này không có khả năng học, và hiện tại thì kết quả cũng chưa cao đối với nhiều ngôn ngữ.
- Có khá nhiều phương pháp dựa vào thống kê khác nhau đã được đưa ra để kiểm lỗi chính tả tiếng Anh. Trong phạm vi giới hạn của đồ án, em xin chỉ liệt kê một vài phương pháp đánh giá là nổi bật.

Chương 4

Cơ sở lý thuyết thuật toán

4.1 Tiền xử lý dữ liệu

4.1.1 Mô hình N-gram

Giới thiệu

- Nhiệm vụ của mô hình ngôn ngữ là cho biết xác suất của một câu w_1, w_2, \dots, w_m là bao nhiêu.
- Theo công thức Bayes:

$$P(AB) = P(B|A) * P(A)$$

Thì:

$$P(w_1 w_2 \dots w_m) = P(w_1) * P(w_2|w_1) * P(w_3|w_1 w_2) * \dots * P(w_m|w_1 w_2 \dots w_{m-1})$$

- Theo công thức này, mô hình ngôn ngữ cần phải có một lượng bộ nhớ vô cùng lớn để có thể lưu hết xác suất của tất cả các chuỗi độ dài nhỏ hơn m . Rõ ràng, điều này là không thể khi m là độ dài của các văn bản ngôn ngữ tự nhiên (m có thể tiến tới vô cùng). Để có thể tính được xác suất của văn bản với lượng bộ nhớ chấp nhận được, ta sử dụng xấp xỉ Markov bậc n :

$$P(w_m|w_1, w_2, \dots, w_{m-1}) = P(w_m|w_{m-n}, \dots, w_{m-1})$$

- Nếu áp dụng xấp xỉ Markov, xác suất xuất hiện của một từ w_m được coi như chỉ phụ thuộc vào n từ đứng liền trước nó ($w_{m-n}, w_{m-n+1}, \dots, w_{m-1}$) chứ không phải phụ thuộc vào toàn bộ dãy từ đứng trước (w_1, w_2, \dots, w_{m-1}). Như vậy, công thức tính xác suất văn bản được tính lại theo công thức:
- Với công thức này, ta có thể xây dựng mô hình ngôn ngữ dựa trên việc thống kê các cụm có ít hơn $n+1$ từ. Mô hình ngôn ngữ này gọi là mô hình ngôn ngữ N-gram.
- Một cụm N-gram là 1 dãy con gồm n phần tử liên tiếp nhau của 1 dãy các phần tử cho trước.

Công thức tính Xác suất thô

- Gọi $C(w_{i-n+1}, \dots, w_{i-1}, w_i)$ là tần số xuất hiện của cụm $w_{i-n+1}, \dots, w_{i-1}, w_i$ trong tập văn bản huấn luyện.
- Gọi $P(w_i | w_{i-n+1}, \dots, w_{i-1})$ là xác suất w_i đi sau cụm $w_{i-n+1}, \dots, w_{i-1}$.
- Ta có công thức tính như sau:

$$P(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{C(W_{i-n+1}, \dots, W_{i-1}, W_i)}{\sum_w C(W_{i-n+1}, \dots, W_{i-1}, w)}$$

- Dễ thấy, $C(w_1, \dots, w_n)$ chính là tần số xuất hiện của cụm $w_1 \dots w_n$ trong văn bản huấn luyện. Do đó công thức trên viết lại thành:

$$P(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{C(W_{i-n+1}, \dots, W_{i-1}, W_i)}{C(W_{i-n+1}, \dots, W_{i-1})}$$

- Tỷ lệ ở vế phải còn gọi là tỷ lệ tần số. Cách tính xác suất dựa vào tỷ lệ tần số còn gọi là ước lượng xác suất cực đại. Cũng có thể gọi đây là công thức tính “xác suất thô” để phân biệt với các cách tính xác suất theo các thuật toán hiệu quả hơn.

Tính xác suất dựa trên N-grams

- Giả sử ta muốn dự đoán âm tiết tiếp theo s_5 dựa trên thông tin ngữ cảnh trước đó của các âm tiết s_1, s_2, s_3, s_4 . Chúng ta cần tính xác suất của sự kiện âm tiết tiếp theo là s_5 khi các âm tiết trước nó là s_1, s_2, s_3, s_4 , hay nói cách khác ta cần tìm xác suất:

$$P(s_1 \dots s_5)$$

- Tổng quan với s_n , ta tìm:

$$P(s_1 \dots s_n)$$

- Theo quy tắc dây chuyền (chain rule) ta có:

$$P(s_1 \dots s_n) = P(s_1) \cdot P(s_2 | s_1) \dots P(s_n | s_1, \dots, s_{n-1})$$

- Áp dụng công thức trên cho một câu có n âm tiết, ta sẽ tìm ra được tổ hợp âm tiết “tốt nhất” cho câu, và tổ hợp này được coi là đúng chính tả. Tuy nhiên, việc tính xác suất đó trong thực tế là điều gần như không thể, bởi chúng hiếm khi xảy ra và nếu có thể tính được thì cũng cần có một lượng dữ liệu rất lớn được dùng để tính toán.
- Để giải quyết vấn đề trên, chúng ta có thể xấp xỉ những xác suất đó bằng một giá trị n-gram nhất định với n cho trước.

N-Gram đơn giản

- Ở phần này, chúng ta sẽ nêu lên cách sử dụng n-gram vào việc dự đoán một từ trong tiếng Anh, từ đó sẽ suy ra cách áp dụng phương pháp cho âm tiết trong tiếng Việt.
- Giả sử chúng ta có thông tin lịch sử h là “its water is so transparent that” và chúng ta muốn tính xác suất từ w tiếp theo sẽ là “the”:

$$P(\text{the} | \text{its water is so transparent that})$$

- Để tính xác suất trên ta có thể dựa vào tần suất xuất hiện của w và h . Ví dụ, chúng ta có thể lấy một lượng dữ liệu đầu vào (corpus) rất lớn, đếm số lần xuất hiện của h và số lần xuất hiện w đi theo sau h . Việc này có thể trả lời được câu hỏi: “Trong tổng số lần xuất hiện h có bao nhiêu lần đứng tiếp sau nó là w ”:

$$P(\text{the} | \text{its water is so transparent that}) = C(\text{its water is so transparent that the}) / C(\text{its water is so transparent that})$$

- Với một corpus đủ lớn, như là internet thì chúng ta có thể đếm được các tần suất và ước lượng được xác suất nêu ở công thức trên. Tuy nhiên, chúng ta có thể dễ dàng nhận thấy rằng ngay cả dữ liệu trên internet cũng không đủ lớn để cho ta những ước lượng tốt nhất trong hầu hết các trường hợp thực tế. Bởi lẽ ngôn ngữ con người hết sức sáng tạo, nó biến đổi không ngừng và do đó mà không phải lúc nào chúng ta cũng có thể đếm được toàn bộ các câu nói. Thậm chí chỉ một sửa đổi nhỏ của một câu thôi cũng có thể khiến số lần xuất hiện của nó trên web là bằng không.

- Tương tự như vậy, nếu chúng ta muốn biết xác suất xuất hiện liên kết của toàn bộ chuỗi từ W như “its water is so transparent that”, chúng ta có thể đặt ra câu hỏi “trong toàn bộ tổ hợp khác nhau có thể có của chuỗi 5 từ đó, có bao nhiêu trong số chúng có thứ tự như trên?”. Chúng ta sẽ phải đếm số lần xuất hiện của “its water is so transparent that” và sau đó chia chúng cho tổng số lần xuất hiện của từng từ trong số 5 từ đó. Việc này đòi hỏi một lượng tính toán không hề nhỏ.

- Chính bởi lẽ đó, chúng ta cần có những phương pháp tốt hơn để ước lượng giá trị xác suất của từ w với thông tin lịch sử h cho trước, và xác suất của toàn bộ chuỗi từ W . Để dễ dàng hơn trong việc diễn đạt, chúng ta sẽ có một vài quy ước như sau: để biểu thị xác suất của một biến độc lập ngẫu nhiên X_t lấy giá trị là “the”, hay $P(X_t = \text{the})$, chúng ta sẽ rút gọn lại là $P(\text{the})$. Chuỗi N từ có thể biểu diễn dạng $w_1 \dots w_n$ hoặc w_{n_1} . Với xác suất liên kết của từng từ riêng biệt trong chuỗi: $P(X = w, Y = w_2, \dots)$ được thay bằng $P(w_1, w_2, \dots, w_n)$.
- Để tính xác suất $P(w_1, w_2, \dots, w_n)$ của các chuỗi từ, ta có thể phân tích xác suất này sử dụng quy tắc dây chuyền:
- Áp dụng cho từ ta có:
- Quy tắc dây chuyền cho ta thấy mối liên hệ giữa việc tính xác suất liên kết của một chuỗi với việc tính xác suất điều kiện của một từ với các từ cho trước. Công thức trên cho ta thấy ta có thể ước lượng xác suất liên kết của một chuỗi bằng cách nhân các xác suất điều kiện với nhau. Tuy nhiên, việc sử dụng quy tắc dây chuyền vẫn chưa giải quyết được vấn đề mà chúng ta đã đề cập ở trên. Ta không thể ước lượng xác suất bằng cách đếm số lần xuất hiện của chuỗi từ, bởi ngôn ngữ rất phong phú, từng ngữ cảnh riêng biệt đều có thể chưa bao giờ xuất hiện trước đó.

- Có thể hiểu việc sử dụng mô hình n-gram là thay vì tính xác suất của từ cho trước dựa trên toàn bộ thông tin lịch sử h , chúng ta xấp xỉ thông tin đó chỉ bằng một vài từ cuối gần nhất.
- $P(w_n|w_{n-1})$ Hay nói cách khác, thay vì tính xác suất:

$P(\text{the}|\text{Walden Pond's water is so transparent that})$

- Chúng ta tính xấp xỉ nó bằng xác suất:

$$P(\text{the}|\text{that})$$

- Ví dụ với mô hình bigram, xấp xỉ xác suất của một từ cho trước với toàn bộ từ trước đó $P(w_n|w_{n-1}^1)$ bằng xác suất điều kiện của từ đứng ngay trước nó.
- Việc này giả thiết rằng xác suất của một từ chỉ phụ thuộc vào từ trước đó, được gọi là một giả thuyết Markov (Markov assumption). Các mô hình Markov là một lớp các mô hình xác suất giả thiết rằng chúng ta có thể dự đoán được xác suất của một đơn vị (unit) nào đó trong tương lai mà không cần phải dựa quá nhiều vào thông tin trong quá khứ. Chúng ta có thể suy ra được bigram (lấy thông tin của một từ trước đó), trigram (lấy thông tin của hai từ trước đó) cho tới n-gram (lấy thông tin của $n - 1$ từ đứng trước).
- Suy ra công thức tổng quát xấp xỉ n-gram xác suất điều kiện của từ tiếp theo trong một chuỗi là:

$$P(W_n|W_1^{n-1}) \approx P(W_n|W_{n-N+1}^{n-1})$$

- Từ giả thuyết bigram cho xác suất của một từ độc lập, chúng ta có thể tính xác suất của một chuỗi đầy đủ bằng cách áp dụng công thức trên với $n =$ với công thức quy tắc dây chuyền:

$$P(W_1^n) \approx \prod_{k=1}^n P(W_k|W_{k-1})$$

- Vậy làm thế nào để ước lượng được các giá trị xác suất bigram hay n-gram này. Có một cách đơn giản và dễ dàng nhận thấy nhất để ước lượng xác suất đó là ước lượng hợp lý cực đại (Maximum likelihood estimation hay MLE). Chúng ta dùng MLE cho các tham số của một mô hình n-gram bằng cách đếm trọng corpus và chuẩn hóa chúng sao cho các giá trị của nó nằm giữa 0 và 1.
- Ví dụ, để tính một giá trị xác suất bigram của một từ y với từ cho trước x đứng trước nó, chúng ta sẽ tính toán giá trị đếm bigram $C(xy)$ và chuẩn hóa nó với tổng số tất cả các bigram khác có từ đứng trước là x :

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)}$$

- Chúng ta có thể đơn giản hóa công thức trên, khi tổng số lần xuất hiện của tất cả các bigram bắt đầu bởi từ w_{n-1} cho trước phải bằng với các giá trị unigram cho từ w_{n-1} đó:

$$P(W_n|W_{n-1}) = \frac{C(W_{n-1}W_n)}{C(W_{n-1})}$$

- Với trường hợp tổng quát sử dụng MLE chỉ n-gram ta có công thức:

$$P(W_n | W_{n-N+1}^{n-1}) = \frac{C(W_{n-N+1}^{n-1} W_n)}{C(W_{n-N+1}^{n-1})}$$

4.1.2 Làm mịn (Smoothing)

- Có một vấn đề gặp phải trong quá trình tính toán ước lượng hợp lý cực đại, đó là vấn đề dữ liệu thưa (sparse data) xảy ra do ước lượng hợp lý cực đại dựa trên một tập dữ liệu huấn luyện riêng biệt. Với một n-gram bất kỳ, đôi khi chúng ta có thể có được xấp xỉ tốt của xác suất. Nhưng do bất kỳ corpus nào cũng có giới hạn nhất định, một vài từ nào đó hoàn toàn chính xác nằm trong chuỗi lại không xuất hiện trong corpus. Lượng dữ liệu bị thiếu này cho thấy n-gram cho bất kỳ một corpus cho trước nào cũng có một số lượng lớn trường hợp xác suất n-gram bằng không cần có một xác suất nào đó khác không cho nó. Hơn nữa, phương pháp ước lượng hợp lý cực đại cũng sinh ra các xấp xỉ không tốt khi các giá trị đếm khác không nhưng lại quá nhỏ, dẫn đến giá trị ước lượng xấp xỉ bằng không.
- Chúng ta cần có một phương pháp giúp có được những ước lượng tốt hơn so với những tần suất thấp hay tần suất bằng không. Các giá trị đếm bằng không thậm chí còn gây ra một vấn đề nghiêm trọng hơn. Khi xử lý hiện tượng nhập nhằng cho một câu mà câu đó lại chứa một giá trị n-gram không hề xuất hiện trong huấn luyện thì ước lượng hợp lý cực đại xác suất của n-gram này cũng như toàn bộ câu sẽ có một giá trị bằng không. Điều này có nghĩa rằng để đánh giá được mô hình ngôn ngữ chúng ta cần phải thay đổi phương thức ước lượng hợp lý cực đại sao cho tất cả các giá trị n-gram đều phải khác không kể cả khi chúng không xuất hiện trong tập huấn luyện.
- Vì những lý do nêu trên, chúng ta cần sử dụng các phương pháp làm mịn để giải quyết, giảm thiểu các vấn đề gặp phải do lượng dữ liệu có hạn. ở đồ án này sẽ sử dụng phương pháp làm mịn Laplace (Laplace smoothing, hay còn gọi là add one smoothing – thêm một).
- Một cách làm mịn đơn giản là cộng thêm 1 vào tất cả các giá trị đếm trước khi chúng ta chuẩn hóa chúng thành giá trị xác suất. Giải thuật này được gọi là làm mịn laplace hay luật laplace
- Chúng ta sẽ áp dụng làm mịn laplace bắt đầu với các xác suất unigram. Ước lượng xấp xỉ cực đại của xác suất unigram cho từ w_i là giá trị đếm c_i được chuẩn hóa bởi tổng số từ vựng N :

$$P(W_i) = \frac{c_i}{N}$$

- Làm mịn laplace chỉ đơn giản là cộng một vào mỗi giá trị đếm. Và khi trong bảng từ vựng có V từ (hay v là tổng số từ có trong tập dữ liệu), mỗi từ được cộng thêm một, chúng ta cũng cần cộng thêm vào mẫu số một giá trị là V :

$$P_{Laplace}(W_i) = \frac{C_i + 1}{N + V}$$

- Tương tự như vậy, với các giá trị xác suất bigram ta có công thức làm mịn:

$$P_{Laplace}(W_n | W_{n-1}) = \frac{C(W_{n-1} W_n) + 1}{C(W_{n-1}) + V}$$

- Trong đó V là số loại từ (mỗi từ khác nhau được coi là một loại) có trong tập dữ liệu, hay V chính là số lượng các unigram. Và như vậy ta có thể phát triển tính tương tự cho các giá trị n -gram khác như 3-gram, 4-gram,...
- Có một giải pháp khá hiệu quả cho việc tính toán các giá trị xác suất n -gram trên máy tính đó là sử dụng dạng logarit hóa của các giá trị xác suất. Vì sau khi làm mịn như ở trên, tất cả các xác suất P đều có giá trị nằm trong khoảng $0 < P < 1$. Do đó, việc logarit hóa giúp cho tránh được vấn đề giới hạn dữ liệu do giá trị xác suất quá nhỏ, nằm ngoài khả năng biểu thị của kiểu dữ liệu (underflow), đặc biệt là khi ta nhân nhiều giá trị xác suất n -gram với nhau. Vì khi giá trị quá nhỏ, máy tính sẽ coi nó bằng không và đồng thời cũng làm tăng tốc độ tính toán thay vì nhân chia các giá trị với nhau thì với logarit, chúng trở thành các phép toán cộng - trừ, một công việc dễ dàng hơn cho bộ xử lý. Ví dụ

$$p_1 \cdot p_2 \cdot p_3 \cdot p_4 = \exp(\log p_1 + \log p_2 + \log p_3 + \log p_4)$$

4.1.3 Tách từ tiếng Việt

Xử lý nhập nhằng

- Nhập nhằng chia làm 2 loại:
 - Nhập nhằng chồng (Overlapping Ambiguity)
 - Nhập nhằng hợp (Combination Ambiguity)
- Ta gọi D là tập hợp các từ tiếng Việt (từ điển tiếng Việt). Các trường hợp nhập nhằng trên được mô tả hình thức như sau:
 - Chuỗi $\alpha\beta\gamma$ được gọi là nhập nhằng chồng nếu $\{\alpha\beta, \beta\gamma\} \subset D$
 - Chuỗi $\alpha\beta$ được gọi là nhập nhằng nếu $\{\alpha, \beta, \alpha\beta\} \subset D$
- Trong thực tế, loại nhập nhằng chồng xảy ra thường xuyên hơn loại nhập nhằng hợp, bởi vì hầu hết các tiếng của tiếng Việt đều có thể đóng vai trò là một từ đơn độc lập. Do đó, hầu hết các từ ghép đều có thể bị nhập nhằng hơn. Tuy nhiên, hầu như mọi trường hợp này đều được giải quyết tốt bằng giải thuật **Maximum Matching**. Vì thế mọi hệ thống nhận diện nhập nhằng hiện tại đều chỉ chú ý đến việc giải quyết loại nhập nhằng đầu tiên là nhập nhằng chồng.

Tách từ tiếng Việt dùng mô hình Maximum Matching

- **Maximum Matching (MM)** được xem như là phương pháp tách từ dựa trên từ điển đơn giản nhất. MM cố gắng so khớp với từ dài nhất có thể có trong từ điển. Đó là một thuật toán ăn tham. (Greedy Algorithms) nhưng bằng thực nghiệm đã chứng minh được rằng thuật toán này đạt được độ chính xác $> 90\%$ nếu từ điển đủ lớn. Tuy nhiên, nó không thể giải quyết vấn đề nhập nhằng và không thể nhận diện được các từ chưa biết bởi vì chỉ những từ tồn tại trong từ điển mới được phân đoạn đúng
- Giải quyết MM gồm hai giải thuật con: FMM (Forward Maximum Matching: so khớp cực đại theo chiều tiến) và BMM (Backward Maximum Matching: so khớp cực đại theo chiều lùi). Nếu chúng ta nhìn vào kết quả của FMM và BMM thì sự khác biệt này cho chúng ta biết nơi nào nhập nhằng xảy ra. Ngoài ra, MM là phương pháp tách từ hoàn toàn phụ thuộc vào từ điển, từ điển phải đủ lớn, đủ chính xác và độ tin cậy phải cao thì mới cho kết quả tách từ chấp nhận được. Đây cũng là nhược điểm rất lớn của phương pháp này.

- Ví dụ: Người nông dân ra sức cải tiến bộ công cụ lao động của mình.
- Đầu ra **FMM**: Người# nông dân # ra sức # cải tiến # bộ # công cụ # lao động # của # mình #.
- Đầu ra **BMM**: Người # nông dân # ra sức # cải # tiến bộ # công cụ # lao động # của # mình #.

4.2 Phương pháp đánh giá

4.2.1 Accuracy - Độ chính xác

- Accuracy (độ chính xác) chỉ đơn giản đánh giá mô hình thường xuyên dự đoán đúng đến mức nào. Độ chính xác là tỉ lệ giữa số điểm dữ liệu được dự đoán đúng và tổng số điểm dữ liệu. Công thức được tính như sau:

$$accuracy = \frac{correct\ predictions}{all\ predictions}$$

- Tuy nhiên, một mô hình có độ chính xác cao chưa hẳn đã tốt. Accuracy lộ rõ hạn chế khi được sử dụng trên bộ dữ liệu không cân bằng (imbalanced dataset).
- **Ví dụ:** Một ngân hàng muốn phát triển hệ thống phát hiện các giao dịch bất thường. Ngân hàng có thể cung cấp cho bạn bộ dataset gồm 1.000.000 giao dịch, trong đó có 1000 giao dịch bất thường. Dễ dàng thấy rằng, chỉ cần mô hình luôn dự đoán mọi giao dịch đều bình thường thì mô hình đã có độ chính xác 99.9%. Tuy nhiên, trên thực tế mô hình của bạn không thể phát hiện được các giao dịch bất thường. Ở đây, tập dữ liệu của chúng ta đang bị mất cân bằng (imbalance), nên việc dựa vào độ chính xác để đánh giá mô hình không mang lại nhiều kết quả tích cực.

4.2.2 Confusion Matrix

- Là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:
- True Positive (TP): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
- True Negative (TN): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
- False Positive (FP): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai) – Type I Error
- False Negative (FN): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai) – Type II Error

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I error	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{TP+FP}$	Negative Predictive Value $\frac{TN}{TN+FN}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

4.2.3 Precision - Recall

- Với những thông tin có được từ Confusion matrix, chúng ta có thể định lượng độ hiệu quả của mô hình qua nhiều thang đo khác nhau. Precision và Recall là hai thang đo quan trọng trong số đó.
- Precision trả lời cho câu hỏi: trong số các điểm dữ liệu được mô hình phân loại vào lớp Positive, có bao nhiêu điểm dữ liệu thực sự thuộc về lớp Positive. Mặt khác, Recall giúp chúng ta biết được có bao nhiêu điểm dữ liệu thực sự ở lớp Positive được mô hình phân loại đúng trong mọi điểm dữ liệu thực sự ở lớp Positive.
- Precision và Recall có giá trị trong $[0,1]$, hai giá trị này càng gần với 1 thì mô hình càng chính xác. Precision càng cao đồng nghĩa với các điểm được phân loại càng chính xác. Recall càng cao cho thể hiện cho việc ít bỏ sót các điểm dữ liệu đúng.
- Trở lại với ví dụ về xét nghiệm COVID-19, precision và recall của mô hình trong ví dụ này là:

$$Precision = \frac{TP}{TP+FP} = \frac{13}{13+17} = 0.43$$

$$Recall = \frac{TP}{TP+FN} = \frac{13}{13+20} = 0.39$$

- Có thể thấy rằng Precision và Recall của mô hình này còn thấp, tức độ chính xác của mô hình chưa cao.
- Trong số 30 trường hợp được chẩn đoán dương tính, chỉ có 13 trường hợp thực sự nhiễm COVID-19 (khoảng 43%, được thể hiện qua precision). Việc kết quả xét nghiệm chưa đủ tin cậy (trong trường hợp này là phân lớp sai – “báo động nhầm”) khiến các cơ sở y tế phải xét nghiệm bằng các phương pháp khác có độ chính xác cao hơn (như xét nghiệm RealTime-PCR), gây lãng phí thời gian và tiền bạc, đồng thời ảnh hưởng trực tiếp đến đời sống của các trường hợp có kết quả xét nghiệm chưa chính xác.
- Với giá trị recall = 0.39, ta có thể hiểu rằng trong số 33 trường hợp thật sự dương tính với COVID-19, mô hình chỉ phát hiện ra 13 trường hợp (khoảng 39%). Trong thực tế, những đối tượng nhiễm bệnh bị bỏ sót bởi mô hình luôn có khả năng trở thành nguồn lây nhiễm cho cộng đồng. Có thể dễ dàng nhận ra rằng việc cải thiện Recall là tối cần thiết.

4.2.4 F1-Score

- Một mô hình tốt khi cả Precision và Recall đều cao, thể hiện cho mô hình ít phân loại nhầm giữa các lớp cũng như tỉ lệ bỏ sót các đối tượng thuộc lớp cần quan tâm là thấp. Tuy nhiên, hai giá trị Precision và Recall thường không cân bằng với nhau (giá trị này tăng thì giá trị kia thường có xu hướng giảm). Để đánh giá cùng lúc cả Precision và Recall, ta sử dụng độ đo **F1-Score**

$$F_{\beta} = \left(1 + \beta^2\right) \frac{Precision \times Recall}{\beta^2 \times Precision + Recall}$$

- Tham số β quyết định mức độ coi trọng giữa Precision và Recall
 - $\beta > 1$: Recall được coi trọng hơn Precision
 - $\beta < 1$: Precision được coi trọng hơn Recall
 - $\beta = 1$: Precision và Recall được coi trọng như nhau
- Việc quyết định nên ưu tiên Precision hay Recall phụ thuộc vào từng bài toán. Ví dụ, với bài toán xác định một khu vực có bom mìn hay không, việc bỏ sót bom mìn cho hậu quả nghiêm trọng hơn so với khi báo động một khu vực an toàn là có bom, vì vậy cần ưu tiên Recall hơn Precision. Mặt khác, việc bỏ sót spam mail có vẻ không tệ nếu so sánh với khi phân loại nhầm một email quan trọng thành spam mail, do đó ở bài toán này, Precision nên được cân nhắc ưu tiên.
- Với những bài toán mà Precision và Recall được cân nhắc ngang nhau, ta chọn $\beta = 1$, khi đó ta đang sử dụng F1-Score. F1-Score là kỳ vọng harmonic (harmonic mean) của Precision và Recall.

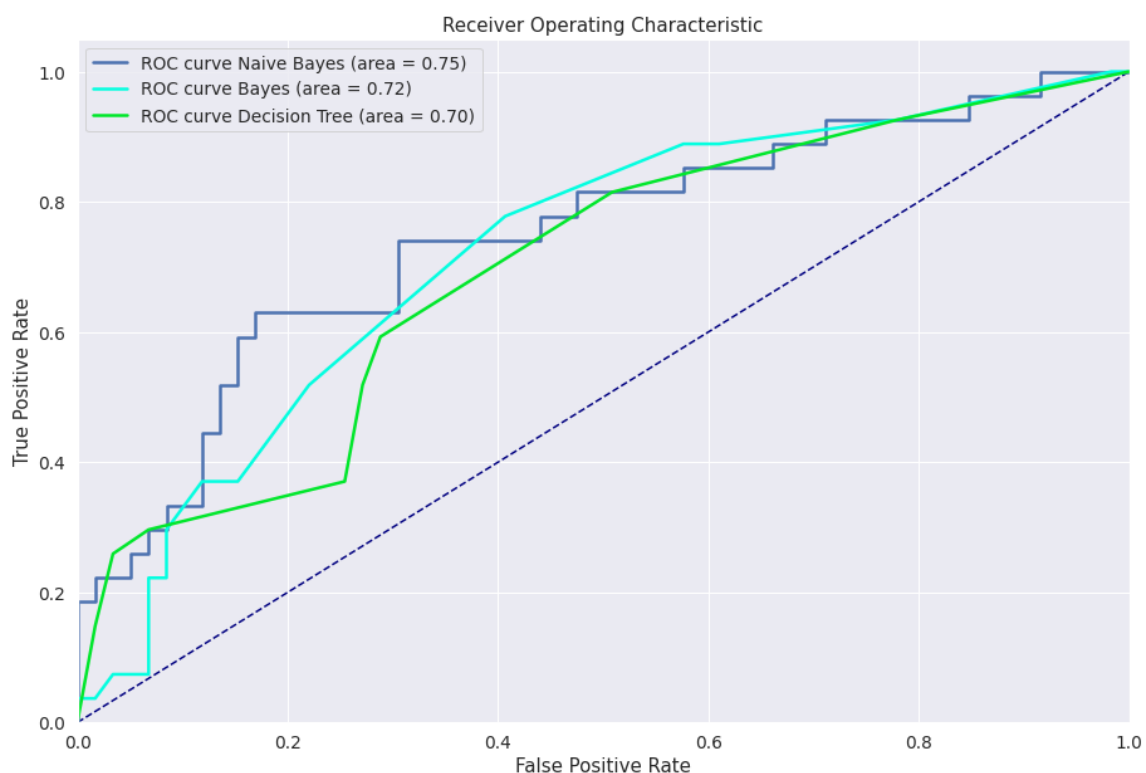
$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2.5 ROC

- Khái niệm:** ROC (Tiếng Anh: Receiver Operating Characteristic) là một đồ thị được sử dụng khá phổ biến trong validation các model phân loại nhị phân. Đường cong này được tạo ra bằng cách biểu diễn tỷ lệ dự báo true positive rate (TPR) dựa trên tỷ lệ dự báo false positive rate (FPR) tại các ngưỡng Threshold khác nhau.
- Trong các bài toán phân lớp, các thuật toán phân lớp thường dự đoán điểm số hay xác suất thuộc về một lớp của dữ liệu đầu vào. Điều này giúp ta biết được mức độ chắc chắn của mô hình khi phân lớp. Sau khi dự đoán xác suất hay điểm số, cần phải chuyển các giá trị đó về các nhãn của các lớp. Việc chuyển từ xác suất, điểm số sang các nhãn được quyết định bởi một “ngưỡng” (threshold).
- Hãy xem xét thuật toán Logistic Regression. Đầu ra của Logistic Regression là giá trị của hàm sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}}$$

- Giá trị của $S(x)$ nằm trong $[0,1]$, thể hiện xác suất điểm dữ liệu đầu vào thuộc về lớp positive. Để quy đổi xác suất về nhãn của lớp, ta cần xác định giá trị threshold. Giá trị threshold mặc định là 0.5, tức là:
 - Nếu $S(x) > \text{threshold}(0.5)$, đầu ra của mô hình bằng 1
 - Nếu $S(x) < \text{threshold}(0.5)$, đầu ra của mô hình bằng 0
- Vấn đề được đặt ra rằng đôi khi default threshold = 0.5 không là “ngưỡng” phân loại tốt nhất, điều này xảy ra khi các lớp của bài toán không cân bằng (dự đoán một căn bệnh hiếm gặp với xác suất xảy ra cực thấp), hay mức độ ưu tiên của một loại sai lầm cao hơn loại sai lầm còn lại, v.v...
- Vì vậy, đôi khi ta cần phải thay đổi threshold để mô hình đạt được kết quả mong muốn. ROC curve là một công cụ để chọn ra threshold phù hợp cho mô hình.



Hình 4.1: Hình ảnh ví dụ về ROC

- Với mỗi giá trị threshold, ta thu được hai giá trị được biểu diễn trên ROC curve:
 - True Positive Rate (hay Sensitivity – Recall): là độ nhạy của mô hình, cho biết mức độ dự đoán chính xác trong lớp positive. TPR là thương của số điểm dữ liệu được dự đoán đúng thuộc lớp positive với số điểm dữ liệu thuộc lớp positive.

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

- False Positive Rate: là xác suất mắc Type II Error

$$FPR = 1 - \text{Specificity}$$

- Trong đó, Specificity cho biết mức độ dự đoán chính xác trong lớp negative.

$$Specificity = \frac{TN}{TN + FP}$$

- Các điểm màu cam đại diện cho mỗi threshold, ứng với trục tung là giá trị TPR và trục hoành là giá trị FPR. Nối các điểm màu cam lại với nhau ta được ROC curve. Đường đứt đoạn màu xanh đại diện cho kết quả của “no skill model” – mô hình dự đoán bằng cách random kết quả. Cần lưu ý rằng, giá trị FPR càng thấp thì xác suất mắc Type II Error thấp, vì vậy các điểm ở bên trái nên được cân nhắc hơn nếu ta cần giảm thiểu False Negative (Type II Error). Mặt khác, các điểm nằm càng cao thì có TPR càng lớn, đồng nghĩa với việc mô hình có giá trị Recall lớn. Tùy thuộc vào bài toán để lựa chọn một điểm – ứng với một threshold phù hợp.

4.3 Thuật toán Logic mờ (Fuzzy Logic)

4.3.1 Khái niệm

- Logic mờ (tiếng Anh: Fuzzy logic) được phát triển từ lý thuyết tập mờ để thực hiện lập luận một cách xấp xỉ thay vì lập luận chính xác theo logic vị từ cổ điển. Logic mờ có thể được coi là một ứng dụng của lý thuyết tập mờ để xử lý các giá trị trong thế giới thực cho các bài toán phức tạp
- Logic mờ cho phép độ liên thuộc có giá trị trong khoảng đóng 0 và 1, và ở hình thức ngôn từ, các khái niệm không chính xác như “hơi hơi”, “gần như”, “khá là” và “rất”. Cụ thể, nó cho phép quan hệ thành viên không đầy đủ giữa thành viên và tập hợp.
- Tính chất này có liên quan đến tập mờ và lý thuyết xác suất.

4.3.2 Công thức

Thiết lập Fuzzy

- Một tập phổ quát X được định nghĩa trong vũ trụ diễn ngôn và nó bao gồm tất cả các yếu tố có thể có liên quan đến vấn đề nhất định. Nếu chúng ta xác định tập A trong tập phổ quát X , ta thấy các mối quan hệ sau:

$$A \subseteq X$$

- Trong trường hợp này, ta nói tập A được bao gồm trong tập phổ quát X . Nếu A không được bao gồm trong X , mối quan hệ này được biểu diễn như sau:

$$A \not\subseteq X$$

- Nếu phần tử x có trong tập hợp A thì phần tử này được gọi là thành viên của tập hợp và ký hiệu sau đây được sử dụng.

$$x \in A$$

- Nếu phần tử x không thuộc A thì sẽ sử dụng ký hiệu như sau:

$$x \notin A$$

- Nói chung, chúng ta biểu diễn một tập hợp bằng cách liệt kê các phần tử của nó. Ví dụ, các phần tử $a_1, a_2, a_3, \dots, a_n$ là các phần tử của tập hợp A , được biểu diễn dưới dạng như biểu thức dưới đây:

$$A = (a_1, a_2, \dots, a_n)$$

Mối quan hệ với Fuzzy

- Mối quan hệ của các Fuzzy thường sẽ phổ biến từ 0 đến 1 hoặc từ 0 đến một số n nào đó (Nhưng chủ yếu là dao động từ 0 đến 1) tùy vào mỗi ứng dụng:

$$\mu_A : X \rightarrow \{0, 1\}$$

$$\mu_A : X \rightarrow \{0, n\}$$

- Fuzzy sẽ thiết lập ranh giới mờ hồ dao động từ 0 đến 1 hoặc là từ 0 đến vị trí n .
- Do đó, Fuzzy Logic là ‘tập ranh giới mờ hồ’ so với tập rõ nét trong tập phần tử đó

Các phép toán với Fuzzy

Involution (Nghịch đảo)	$\overline{\overline{A}} = A$
Commutativity (Giao hoán)	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Associativity (Kết hợp)	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distributivity (Phân phối)	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Idempotency	$A \cup A = A$ $A \cap A = A$
Absorption (Phép hợp)	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$
Absorption by X and \emptyset	$A \cup X = X$
Identity	$A \cup \emptyset = A$ $A \cap X = A$
De Morgan's law (Định luật De Morgan)	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$
Absorption of complement	$A \cup (\overline{A} \cap B) = A \cup B$ $A \cap (\overline{A} \cup B) = A \cap B$
Law of contradiction (Luật mâu thuẫn)	$A \cap \overline{A} = \emptyset$
Law of excluded middle (Luật loại trừ giữa)	$A \cup \overline{A} = \emptyset$

4.3.3 Công thức Defuzzication

Công thức

- Giá trị trung bình tối đa (MOM - Mean of Maximum Method)

- Phương pháp MOM tạo ra một hành động kiểm soát đại diện cho giá trị trung bình giá trị của tất cả các hành động kiểm soát, có chức năng thành viên đạt đến tối đa

$$z_0 = \sum_{j=1}^k \frac{z_j}{k}$$

- Trong đó:
 - z_j : Điều khiển hoạt động của các phần tử đạt đến tối đa.
 - k : Số phần tử hoạt động
- Phương pháp khoảng trung tâm (COA - Center of Area Method): Phương pháp COA được sử dụng rộng rãi tạo ra trọng tâm của phân bố khả năng của một tập mờ C

$$z_0 = \frac{\sum_{j=1}^n \mu_c(z_j) \cdot z_j}{\sum_{j=1}^n \mu_c(z_j)}$$

Ví dụ

- Giả sử ta có một bảng điểm GPA của một sinh viên Đại học (dao động từ 0 đến 4) như sau: gồm số tín chỉ và điểm hệ 4 của môn đó:

Môn học	Số tín chỉ	Điểm hệ 4
Triết học Mác Lê nin	2	1
Kiến trúc máy tính	3	2.5
Lập trình Java	3	3

- Ta sẽ dùng công thức của Defuzzication ta sẽ tính GPA như sau:

$$GPA = \frac{\sum_{i=1}^n a_i * n_i}{\sum_{i=1}^n n_i}$$

- Trong đó:
 - n là số tín chỉ
 - a là điểm hệ 4

$$GPA = \frac{2 \times 1 + 3 \times 2.5 + 3 \times 3}{2 + 3 + 3} = 2.3125$$

4.3.4 Ứng dụng thực tế

- **Điều khiển nhiệt độ:** Fuzzy Logic thường được sử dụng trong các hệ thống HVAC (Hệ thống điều hòa không khí - Heating, Ventilation, and Air Conditioning) để điều khiển nhiệt độ. Bằng cách xác định các tập mờ cho các khoảng nhiệt độ như "lạnh", "ấm", và "nóng", một bộ điều khiển mờ có thể xác định mức độ nhiệt cần thiết phù hợp dựa trên nhiệt độ hiện tại và mức độ thoải mái mong muốn.
- **Điều khiển đèn giao thông:** Fuzzy Logic được sử dụng trong các hệ thống điều khiển đèn giao thông thông minh để điều chỉnh thời gian đèn tín hiệu dựa trên tình trạng giao thông thời gian thực. Bằng cách xem xét các yếu tố như lưu lượng giao thông, tắc nghẽn và hoạt động của người đi bộ, bộ điều khiển mờ có thể tối ưu hóa thời gian đèn tín hiệu và cải thiện hiệu suất giao thông tổng thể.
- **Điều khiển máy giặt:** Fuzzy Logic được sử dụng trong các máy giặt để xác định chu trình giặt và nhiệt độ nước phù hợp dựa trên loại và khối lượng quần áo. Bằng cách xem xét các biến như loại vải, mức độ vết bẩn và kích thước tải, bộ điều khiển mờ có thể điều chỉnh quá trình giặt để đạt hiệu suất làm sạch tối ưu.
- **Xe tự hành:** Fuzzy Logic được sử dụng trong quá trình ra quyết định của các xe tự hành. Bộ điều khiển mờ có thể xử lý các tình huống không chắc chắn và phức tạp trên đường bằng cách xử lý dữ liệu cảm biến như khoảng cách, tốc độ và điều kiện đường để đưa ra quyết định về gia tốc, phanh và lái.
- **Kiểm soát chất lượng:** Fuzzy Logic được áp dụng trong các hệ thống kiểm soát chất lượng để đánh giá chất lượng sản phẩm hoặc dịch vụ. Bằng cách xác định các tập mờ cho các mức chất lượng khác nhau, bộ điều khiển mờ có thể đánh giá mức độ phù hợp của một sản phẩm hoặc dịch vụ với các tiêu chuẩn đã được định nghĩa trước, ngay cả khi có sự không chắc chắn hoặc không chính xác trong việc đo lường.
- **Chẩn đoán y tế:** Fuzzy Logic được sử dụng trong các hệ thống chẩn đoán y tế để đánh giá khả năng một bệnh nhân mắc một bệnh cụ thể. Bằng cách xem xét các triệu chứng, tiền sử bệnh và kết quả xét nghiệm, bộ điều khiển mờ có thể cung cấp mFuzzy Logic (Logic Mờ) là một công cụ mạnh mẽ được sử dụng trong nhiều lĩnh vực thực tế. Dưới đây là một số ví dụ về ứng dụng của Fuzzy Logic:

4.3.5 Ưu và nhược điểm của Fuzzy Logic

Ưu điểm

- Trong hệ thống này, chúng ta có thể thu thập tất cả các thông tin đầu vào, bao gồm các thông tin không chính xác, bị bóp méo, nhiễu.
- Dễ xây dựng và dễ hiểu
- Là giải pháp cho những vấn đề phức tạp, chẳng hạn như nghiên cứu thuốc.
- Ngoài ra, chúng ta có thể liên hệ toán học theo khái niệm trong logic mờ. Những khái niệm này rất đơn giản.
- Do tính linh hoạt của logic mờ, chúng ta có thể thêm và xóa các quy tắc trong hệ thống này.

Nhược điểm

- Hiện nay, vẫn chưa có cách tiếp cận về thiết kế đối với logic mờ này.
- Nếu logic học quá đơn giản, thì một người bất kỳ nào cũng đều có thể hiểu được nó.
- Logic mờ chỉ thích hợp cho các bài toán không có độ chính xác cao.

4.4 Thuật toán TF-IDF

4.4.1 Định nghĩa

- **Khái niệm:** Trong truy hồi thông tin, tf-idf, TF*IDF, hay TFIDF, viết tắt từ cụm từ tiếng Anh: Term Frequency–Inverse Document Frequency, là một thống kê số học nhằm phản ánh tầm quan trọng của một từ đối với một văn bản trong một tập hợp hay một ngữ liệu văn bản. TF-IDF thường dùng dưới dạng là một trọng số trong tìm kiếm truy xuất thông tin, khai thác văn bản, và mô hình hóa người dùng.
- Giá trị TF-IDF tăng tỉ lệ thuận với số lần xuất hiện của một từ trong tài liệu và được bù đắp bởi số lượng tài liệu trong kho ngữ liệu có chứa từ, giúp điều chỉnh thực tế là một số từ xuất hiện nói chung thường xuyên hơn. TF-IDF là một trong những lược đồ (scheme) tính trọng số phổ biến nhất hiện nay. Một cuộc khảo sát được thực hiện vào năm 2015 cho thấy 83% các hệ thống khuyến nghị dựa trên văn bản (text-based recommender systems) trong các thư viện số sử dụng TF-IDF

4.4.2 Công thức và ví dụ

Công thức

- Ta có công thức của **TF**:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

- Trong đó:

- $n_{i,j}$ là số từ xuất hiện trong một đoạn văn bản
- $\sum_k n_{i,j}$ là tổng số từ trong một đoạn văn bản đó (bao gồm từ trùng được đếm liên tiếp được tính là một từ)

- Công thức của **IDF**:

$$IDF(w) = \log \left(\frac{N}{df_t} \right)$$

- Trong đó:

- N là tổng số văn bản trong một tập xác định
- df_t là số văn bản chứa từ nhất định, điều kiện là phải xuất hiện trong tập văn bản đó

- Từ đó ta có công thức đầy đủ của thuật toán **TF-IDF**:

$$W_{i,j} = TF_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

Ví dụ

- Ta có 2 đoạn văn bản sau đây:

Từ	Số lần xuất hiện từ
Nguyễn	1
Thùy	1
Trang	1
cục	2
trưởng	1
tình	1
báo	1
TocoToco	1

(a) Văn bản 1

Từ	Số lần xuất hiện từ
Nguyễn	1
Thùy	1
Trang	1
đã	1
điều	1
tra	1
vụ	2
án	2

(b) Văn bản 2

- Gọi hai văn bản 1 và 2 lần lượt là D_1 và D_2
- Áp dụng công thức và tính như sau:

- Trường hợp có từ ở 2 văn bản:

* Tính **TF**:

$$TF("Trang", d_1) = \frac{1}{9} = 0.1111$$

$$TF("Trang", d_2) = \frac{1}{10} = 0.1$$

* Tính **IDF**:

$$IDF("Trang", D) = \log\left(\frac{2}{1}\right) = 0$$

* Tính được **TF-IDF**:

$$TF_IDF("Trang", d_1, D) = TF("Trang", d_1) \times IDF("Trang", D) = 0.1111 \times 0 = 0$$

$$TF_IDF("Trang", d_2, D) = TF("Trang", d_2) \times IDF("Trang", D) = 0.1 \times 0 = 0$$

- Trường hợp có 1 từ xuất hiện ở 1 trong 2 đoạn văn bản

* Tính **TF**:

$$TF("TocoToco", d_1) = \frac{1}{9} = 0.111$$

$$TF("TocoToco", d_2) = \frac{0}{10} = 0$$

* Tính **IDF**:

$$IDF("TocoToco", D) = \log\left(\frac{2}{1}\right) = 0.301$$

* Tính **TF-IDF**

$$TF_IDF("TocoToco", d_1, D) = 0.111 \times 0.301 = 0.033411$$

$$TF_IDF("TocoToco", d_2, D) = 0 \times 0.301 = 0$$

4.4.3 Ứng dụng thực tế

- **Hệ thống tìm kiếm:** TF-IDF được sử dụng để xác định độ quan trọng của từ khóa trong một văn bản. Khi người dùng tìm kiếm, hệ thống tìm kiếm có thể sử dụng TF-IDF để xếp hạng các văn bản dựa trên sự phù hợp của từ khóa với nội dung.
- **Phân loại văn bản:** TF-IDF cũng có thể được sử dụng để phân loại các văn bản vào các danh mục khác nhau. Bằng cách tính toán trọng số của các từ trong văn bản, thuật toán TF-IDF giúp xác định đặc trưng và sự khác biệt giữa các văn bản, từ đó giúp phân loại chính xác.
- **Gợi ý tài liệu:** TF-IDF có thể được sử dụng để gợi ý các tài liệu liên quan dựa trên nội dung. Bằng cách so sánh độ tương đồng giữa các văn bản, thuật toán TF-IDF giúp xác định các tài liệu có chủ đề tương tự hoặc liên quan đến tài liệu hiện tại.
- **Xác định từ khóa quan trọng:** TF-IDF cung cấp thông tin về độ quan trọng của từ trong một văn bản. Điều này có thể được sử dụng để xác định các từ khóa quan trọng trong văn bản, giúp người dùng nắm bắt ý chính và nội dung chính của văn bản một cách nhanh chóng.

4.4.4 Ưu và nhược điểm của TF-IDF

Ưu điểm

- **Đơn giản và hiệu quả:** Thuật toán TF-IDF dễ hiểu và dễ triển khai. Nó không đòi hỏi nhiều tài nguyên tính toán và có thể được áp dụng cho các tập dữ liệu lớn.
- **Tính toán linh hoạt:** TF-IDF cho phép điều chỉnh trọng số của từng thuật ngữ trong một văn bản dựa trên tần suất xuất hiện của chúng trong văn bản và trong toàn bộ tập dữ liệu. Điều này cho phép thuật toán tạo ra một biểu diễn số học của văn bản, tạo điểm khác biệt giữa các từ khóa và từ thông thường.
- **Tính đại diện:** TF-IDF giúp định rõ đặc trưng và nội dung của mỗi văn bản. Nó tạo ra một biểu diễn số học cho mỗi văn bản dựa trên sự phân bố từ khóa quan trọng. Điều này giúp xác định độ tương đồng và sự khác biệt giữa các văn bản.
- **Tính khách quan:** TF-IDF không đòi hỏi sự can thiệp của con người trong việc đánh giá và xác định trọng số của từng thuật ngữ. Nó dựa trên tần suất xuất hiện và phân bố của từ khóa trong dữ liệu, sẽ đảm bảo tính khách quan trong xác định độ quan trọng của từng thuật ngữ.

Nhược điểm

- **Không xử lý được ngữ cảnh:** TF-IDF chỉ xem xét tần suất và độ quan trọng của từ trong một văn bản cụ thể, mà không xử lý được ngữ cảnh hoặc mối quan hệ giữa các từ. Điều này có thể dẫn đến việc bỏ qua các thông tin quan trọng trong văn bản.
- **Sự ảnh hưởng của từ phổ biến:** Nếu một từ xuất hiện quá phổ biến trong một tập văn bản, nó có thể có trọng số cao trong toàn bộ tập dữ liệu. Điều này có thể làm giảm hiệu quả của thuật toán TF-IDF trong việc phân loại và gợi ý tài liệu.

- **Sự thiếu cân nhắc về ngữ nghĩa:** TF-IDF không xem xét ngữ nghĩa của từ, chỉ xem xét tần suất và độ quan trọng. Điều này có thể dẫn đến việc không phân biệt được sự khác biệt giữa các từ đồng nghĩa hoặc từ trái nghĩa.

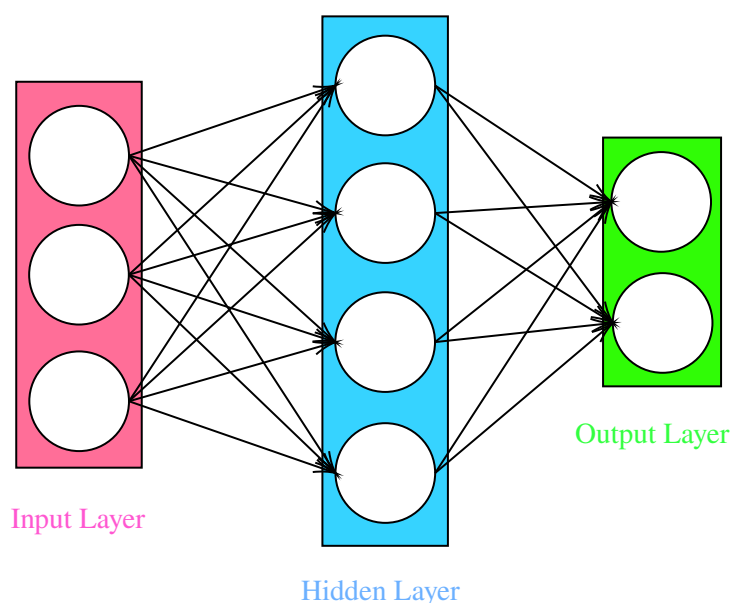
4.5 Mô hình học sâu

4.5.1 Mạng thần kinh hồi quy (RNN)

Về cơ bản nếu thấy sequence data hay time-series data mà muốn áp dụng Neural Network sẽ nghĩ ngay đến RNN. Tuy nhiên mạng RNN gặp phải một số hạn chế đó là phải thực hiện tuần tự, đạo hàm bị triệt tiêu (Vanishing gradient), bùng nổ đạo hàm (Exploding gradient). LSTM là một mạng cải tiến của RNN nhằm giải quyết phần nào các vấn đề mà mạng RNN gặp phải. Rất nhiều các bài toán học máy sử dụng LSTM đem lại kết quả rất đáng chú ý so với việc sử dụng các phương pháp khác. Dữ liệu văn bản mà ta thu thập được là một dạng dữ liệu kiểu chuỗi tuần tự vì vậy khá phù hợp khi sử dụng LSTM huấn luyện dữ liệu mà nhóm đã thu thập được.

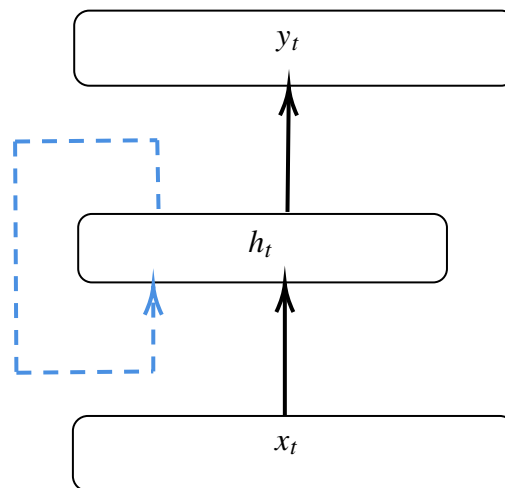
Giới thiệu lý thuyết mạng RNN

- Recurrent Neural Networks (RNN) - một cách dịch đại khái sang tiếng Việt là mạng thần kinh hồi quy. Sở dĩ, nó được gọi là Recurrent (hồi quy) vì nó thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng ghi nhớ các thông tin được tính toán trước đó (như là việc nhìn vào các bối cảnh phía trước rồi đưa ra các hành động tiếp theo). RNN là mô hình mạng thần kinh được thiết kế để nhận dữ liệu tuần tự hoặc kiểu dữ liệu time series (chuỗi thời gian).
- Đầu tiên, trước khi tìm hiểu về RNN, chúng ta hãy cùng xem cách mạng Neural thông thường lan truyền như hình dưới đây:



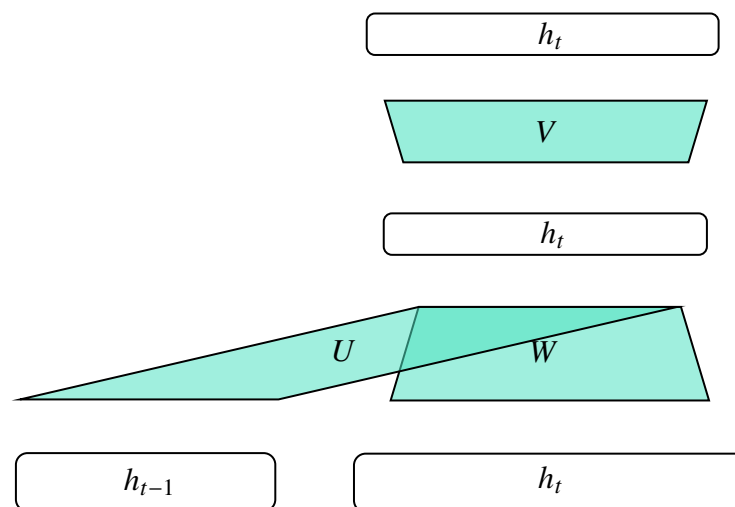
Hình 4.2: Kiến trúc mạng Neuron thông thường

- Với mạng Neural Network thông thường, các nút màu đỏ là giá trị đầu vào còn màu xanh lá cây là giá trị đầu ra. Ví dụ trong bài toán về phân loại hình ảnh dùng Convolution Neural Network đầu vào sẽ là các pixel và đầu ra là các ảnh cần phân loại. Theo cách này, ta có thể thiết kế một mạng Neural Network có độ dài đầu vào và đầu ra cố định.
- Tuy nhiên, các dữ liệu trong thực tế thường không có chiều dài cố định. Ví dụ: khi ta viết một Status trên Facebook, nó không cố định chiều dài số lượng từ mà ta viết. Chính vì vậy mà mạng RNN được thiết kế để xử lý các dữ liệu có đầu vào và đầu ra thay đổi.
- Về cơ bản, mô hình đơn giản của RNN có dạng như sau:



Hình 4.3: Cấu trúc đơn giản của mạng RNN

- Điểm khác biệt lớn nhất của một mạng Neuron thông thường với RNN chính là phần mũi tên màu xanh lặp lại ở trạng thái ẩn h_t . Đường mũi tên xanh này ngụ ý rằng tại thời điểm t bất kì, đầu vào của mạng sẽ là tổ hợp của trạng thái ẩn h_t trước đó và vector đầu vào x_t tại thời điểm hiện tại. Ta có thể coi trạng thái ẩn h_t là "bộ nhớ" của mạng. h_t sẽ lưu trữ các thông tin đã được tính toán ở phía trước. Đầu ra ở bước y_t chỉ được tính dựa trên bộ nhớ ở bước thứ t .



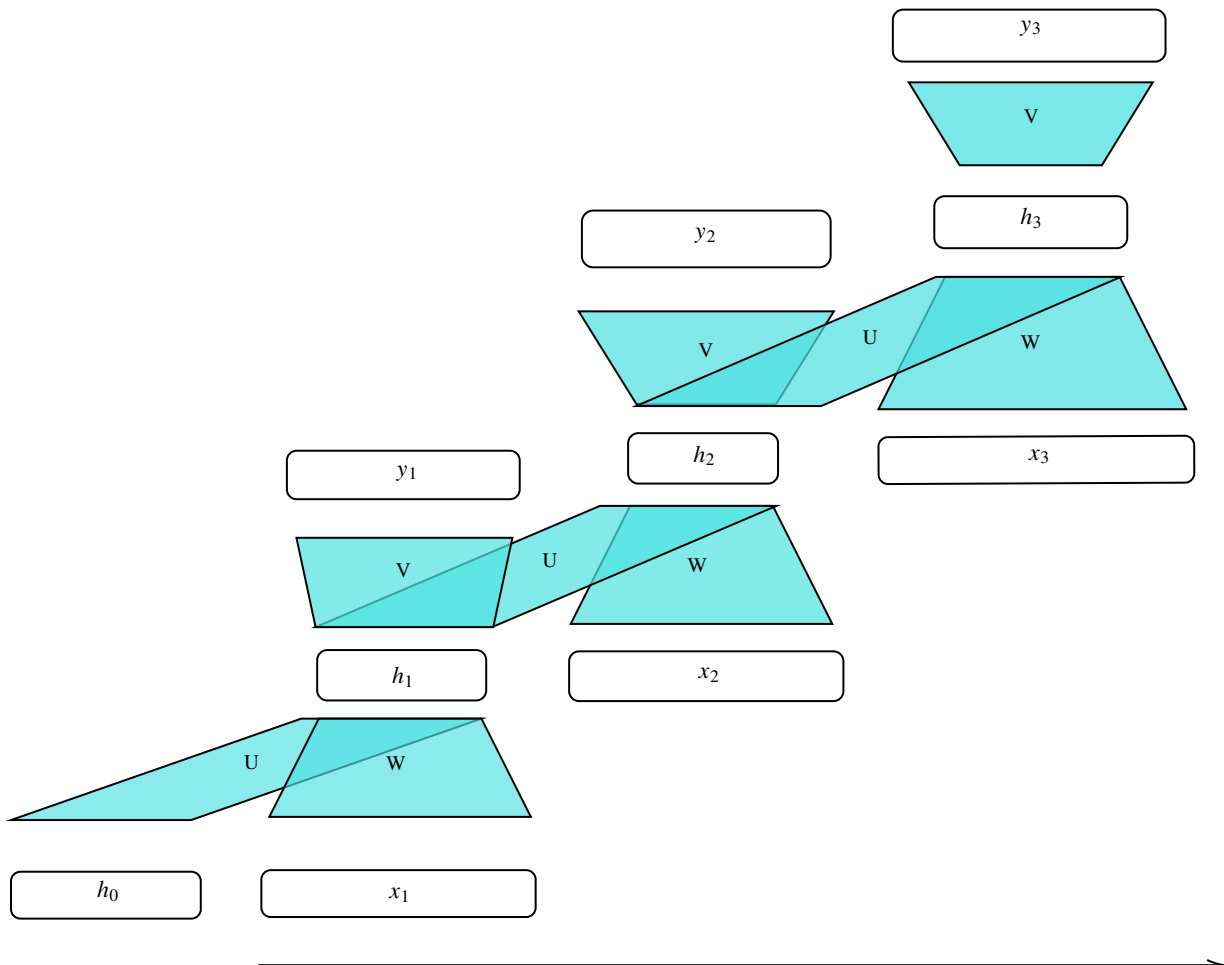
Hình 4.4: Mô tả RNN dưới dạng tham số

- Bản chất hình 4.2 và 4.3 là như nhau. Sự khác biệt duy nhất là ở hình 4.3, ta đã có thêm ma trận trọng số U , ma trận U có khả năng kết nối thông tin từ trạng thái trước đó h_{t-1} với trạng thái hiện tại h_t . Và đây chính là điểm mấu chốt của RNN, với U , RNN có thể lưu trữ được thông tin ngữ cảnh trước để làm đầu vào cho phép tính hiện tại.

Ứng dụng trong mạng RNN

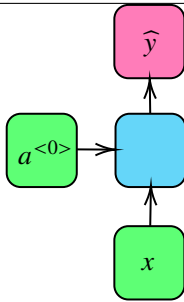
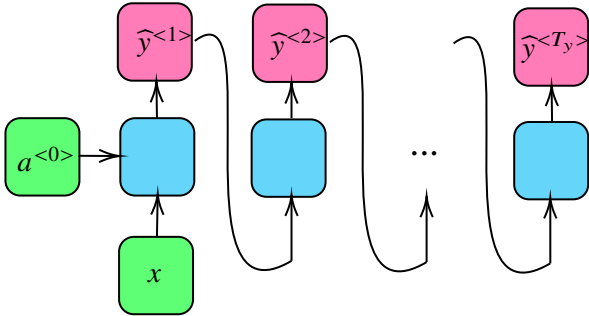
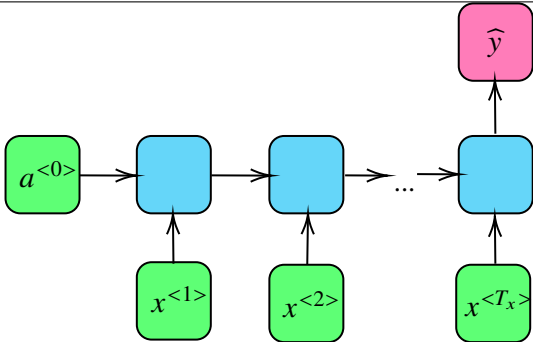
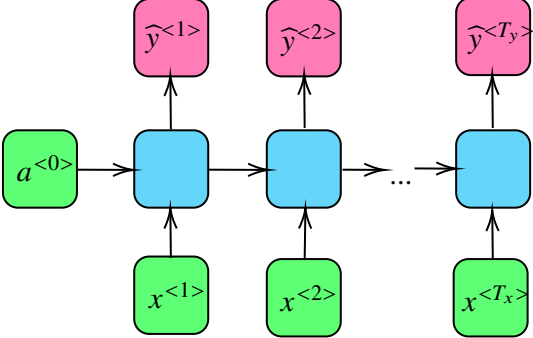
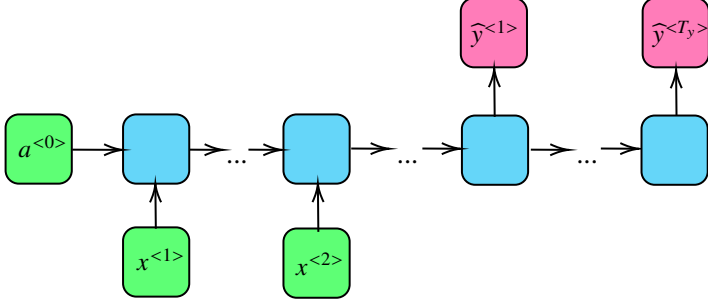
Lan truyền thuận trong mạng hồi quy RNN

Lan truyền thuận trong RNN đơn giản là phép ánh xạ từ mỗi chuỗi đầu vào đến một chuỗi đầu ra tương ứng (giống hết mạng Neuron thông thường).



Hình 4.5: Minh họa các lớp trong mạng RNN

- Để dễ hình dung, chúng ta sẽ khai triển (unfold) mạng RNN như trên. Ví dụ, ta có chuỗi đầu vào là “Tôi đi học” thì mạng RNN sẽ triển khai thành 3 layer-Neural Network. Trong đó, mỗi Layer tương ứng với 1 từ. Lúc này, việc tính toán bên trong RNN sẽ xảy ra như sau:
 - x_t là đầu vào của bước t . Ví dụ: x_2 là một one-hot vector tương ứng với từ thứ hai là “đi”.
 - h_t là trạng thái ẩn ở bước thứ t . Nó là “memory (bộ nhớ)” của mạng. h_t là một tổ hợp tuyến tính của trọng số U với bộ nhớ h_{t-1} ở bước trước đó và đầu vào x_t hiện tại

Type of RNN	Illustration	Example
One-to-one $T_x = T_y = 1$		Traditional Neural Network
One-to-many $T_x = 1, T_y > 1$		Music generation
Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation

với trọng số W . Hàm g thường là một hàm phi tuyến như \tanh hoặc $ReLU$. Đối với

bước đầu tiên, giá trị h_0 sẽ là một vector 0.

$$h_t = g(Uh_{t-1} + Wx_t)$$

- y_t là output (đầu ra) ở bước t . Ví dụ: nếu ta muốn dự đoán từ tiếp theo trong câu thì y_t là một vector xác suất các từ trong danh sách từ vựng.

$$y_t = \text{softmax}(Vh_t)$$

- Nếu ta gọi số chiều của lớp đầu vào, lớp ẩn và lớp đầu ra lần lượt là d_{in} , d_h và d_{out} . Bảng dưới đây mô tả số chiều của các tham số trong mạng RNN

		Số chiều
Input	x	d_m
Hidden State	h	d_h
Output	y	d_{out}
Tham số học	W	$d_h \times d_m$
	U	$d_h \times d_h$
	T	$d_{out} \times d_h$

- Mã giả dưới đây mô tả quy trình lan truyền thuận của mạng RNN.

Algorithm 1: RNN Pseudo code

```

1 function FORWARD RNN( $x, network$ ) return output sequence  $y$ 
2    $h_0 \leftarrow 0$  for  $i \leftarrow 1$  to LENGTH( $x$ ) do
3      $h_i \leftarrow g(Uh_{i-1} + Wx_i)$ 
4      $y_i \leftarrow f(Vh_i)$ 
5   return  $y$ 

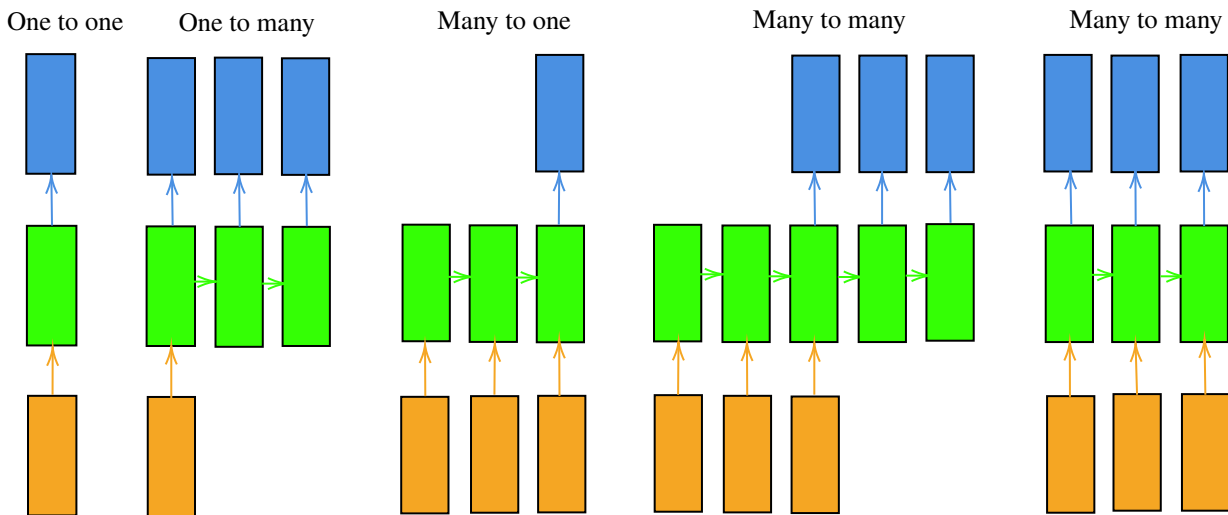
```

- Hàm FORWARDRNN nhận tham số câu đầu vào là vector x . Ban đầu lớp ẩn h được khởi tạo là 1 vector 0. Vòng lặp chạy với mỗi từ x_i ta tính được giá trị lớp ẩn h_i và y_i mới, trong khi bộ tham số là (U, V, W) là không đổi trong tất cả các bước. Điều này nói lên rằng ta thực hiện cùng 1 nhiệm vụ ở tất cả các bước. Và điều này làm giảm đáng kể số lượng tham số cần học trong mô hình.

Huấn luyện mạng hồi quy RNN cho bài toán sửa lỗi chính tả

Về cơ bản bài toán hồi quy RNN có thể phân loại như sau:

- Trong bài toán sửa lỗi chính tả nhóm thực hiện lần này, mô hình được sử dụng sẽ là many-to-many. Tức là ta có một chuỗi input và đầu ra sẽ là một chuỗi output tương ứng được dự đoán là kết quả sau khi sửa lỗi chính tả.
- Sau đây là mô tả quy trình huấn luyện mạng RNN cho bài toán sửa lỗi chính tả
 - Input: Chuỗi đầu vào $X = [x_1, x_2, \dots, x_n]$ với x_t là một bi-gram (chương 2.1). Các bigram được biểu bởi một one-hot vector có số chiều $|V| \times 1$, với $|V|$ là số từ trong tập từ điển.



Hình 4.6: Minh họa mô hình được sử dụng trong mạng RNN

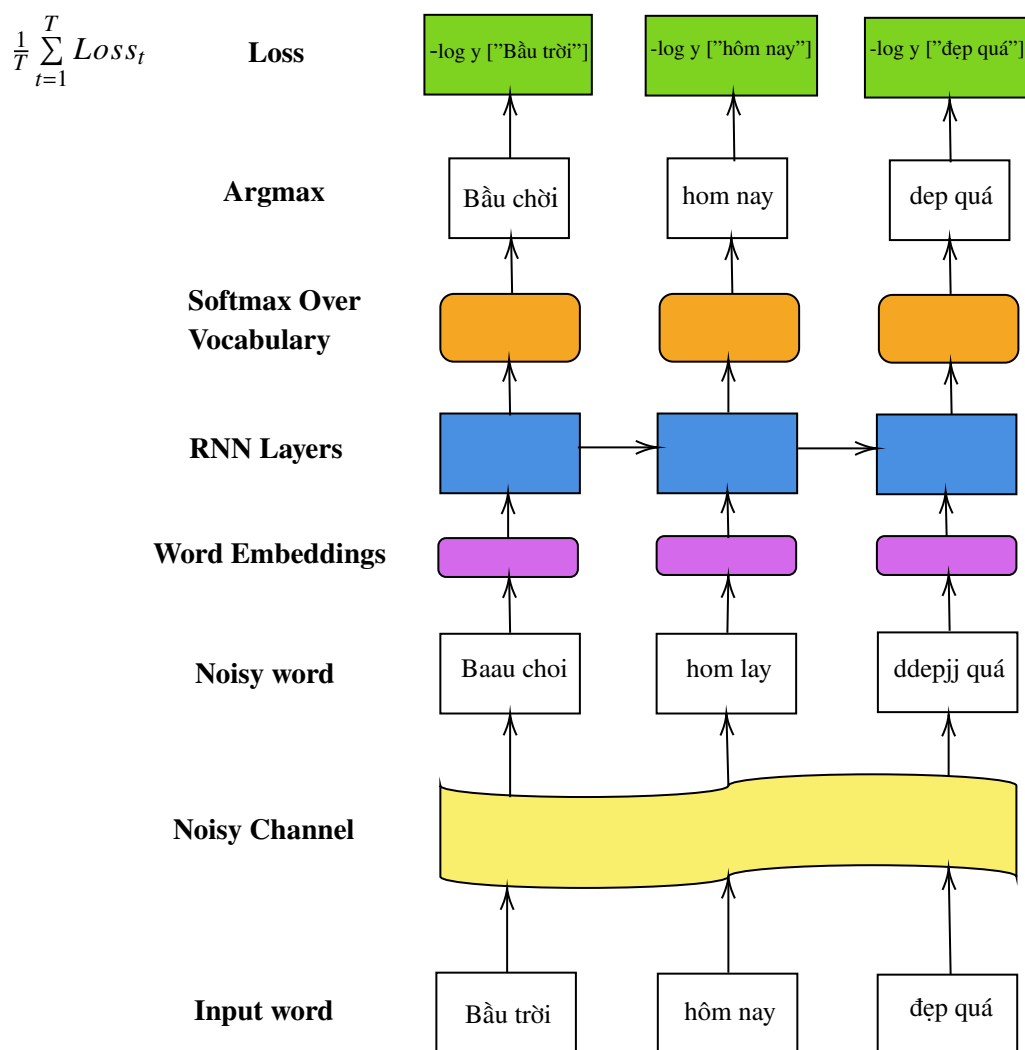
- Output: Chuỗi câu dự đoán vector $Y^* = [y_1^*, y_2^*, \dots, y_n^*]$ với y_t^* là y_t là một vector xác suất các từ trong danh sách từ vựng. y_t có số chiều $|V| \times 1$
- Lưu ý rằng, các bi-gram x_t từ văn bản gốc sẽ được đi qua một kênh làm nhiễu. Mục đích của việc này là ta muốn huấn luyện mô hình có khả năng dự đoán (predict) được những từ x_t gốc thực sự (target) sau khi các x_t gốc đầu vào (input) đã bị làm nhiễu (hay nói nôm na là những từ bị cố tình làm cho sai chính tả). Hình dưới đây sẽ mô tả chi tiết quá trình huấn luyện này.
- Ban đầu, các bi-gram được đưa qua kênh làm nhiễu (Noisy channel) và được ánh xạ thành các noisy word. Ta tiến hành mã hóa từng bi-gram thành các vector (Word Embeddings). Tiếp theo, tại thời điểm t , các vector bi-gram x_t này lần lượt được đưa vào lớp RNN (ô màu xanh trên hình). Các trạng thái ẩn h_t trong lớp RNN được tính theo công thức sau:

$$h_t = g(Uh_{t-1} + Wx_t)$$

Với:

- U : ma trận trọng số kết nối trạng thái ẩn h_{t-1} sang h_t
- W : ma trận trọng số kết nối từ đầu vào x_t đến lớp ẩn h_t
- Giả sử vector y_t có xác suất là $[0.8, 0.2, 0.2]$ ứng với các từ [Bầu chời, Bầu zời, Bầu trời]. Ta sẽ lấy giá trị max trong vector y_t , tức là 0.8 tương ứng với từ “Bầu chời” để làm đầu ra dự đoán cho mô hình tại thời điểm t .
- Và cuối cùng, ta cần một hàm mục tiêu để đánh giá độ tốt của mô hình sau mỗi lần dự đoán, cross-entropy sẽ là hàm mục tiêu được chúng tôi dùng ở đây. Nếu ta có một vector dự đoán xác suất đầu ra y_t^* và vector y_t chứa các giá trị nhãn thực sự (các nhãn này chính là mục tiêu để mô hình học), thì tại thời điểm t ta có hàm đánh giá $Loss_t$ như sau:

$$Loss_t = \sum_{w \in W} (y_t[w] \log y_t^*[w])$$



Hình 4.7: Quy trình huấn luyện mạng RNN cho bài toán sửa lỗi chính tả

- Hàm $Loss_t$ này được hiểu là là tổng của các tích giữa các phần tử w_t của vector y_t và các phần tử w_t của vector dự đoán y_t^* . Ví dụ, ta có vector dự đoán $y_t^* = [0.8, 0.2, 0.2]$ và vector chứa các giá trị nhãn thực sự $y_t = [1, 0, 0]$, thì ta có Loss tại thời điểm t như sau:

$$Loss_t = 1 * \log(0.8) + 0 * \log(0.2) + 0 * \log(0.2)$$

- Chúng tôi coi bài toán sửa lỗi chính tả thuộc bài toán dự đoán từ trong mô hình ngôn ngữ (2.3), tức là ta chỉ quan tâm tới từ hiện tại được dự đoán để sửa lỗi chính tả sau khi đã quan sát được các từ trước đó. Trong mô hình ngôn ngữ thì vector đầu ra y chứa các xác suất của từ được dự đoán. Chính vì vậy, ta có thể lược giản hàm $Loss_t$ như sau:

$$Loss_t = -\log y_t^* [w_t]$$

Với:

- w_t : từ đúng thật sự tại thời điểm t
- y_t^* : vector chứa xác suất các từ được dự đoán là sửa lỗi chính tả

- Và cuối cùng, hàm đánh giá mất mát của cả mô hình sẽ là trung bình của từng hàm Loss tại thời điểm t :

$$L = \frac{1}{T} \sum_{t=1}^T Loss_t$$

- Mục tiêu của ta là cực tiểu hóa hàm L bằng cách sử dụng cơ chế lan truyền ngược, trái ngược với cơ chế lan truyền thuận như ở 3.1.2.2. Do ngày nay, hầu như các công cụ hỗ trợ tính toán đã cung cấp khả năng xử lý cơ chế này, nên với giới hạn của đồ án, chúng tôi xin trích dẫn tham khảo chứng minh chi tiết cơ chế này tại đây [2].
- Trong thực tế, mạng RNN không thể lưu trữ được quá nhiều thông tin, và mạng LSTM ở 3.1.3 sẽ khắc phục được điều này.

Mạng RNN hai chiều

- Trong một mạng hồi quy đơn giản, trạng thái ẩn ở một thời điểm nhất định t đại diện cho mọi thứ mà mạng biết về trình tự cho đến thời điểm đó trong trình tự. Trạng thái ẩn tại thời điểm t là kết quả của một hàm của các đầu vào từ lúc khởi động cho đến khoảng thời gian t . Chúng ta có thể coi đây là bối cảnh của mạng ở phía bên trái thời điểm hiện tại.

$$h_t^f = RNN_{forward}(x_1^t)$$

- h_t^f tương ứng với trạng thái ẩn thông thường tại thời điểm t , và đại diện cho mọi thứ mà mạng đã thu thập được từ trình tự đến thời điểm đó. Trong nhiều ứng dụng, chúng ta có quyền truy cập vào toàn bộ chuỗi đầu vào cùng một lúc. Một câu hỏi được đặt ra là liệu chúng ta có thể tận dụng ngữ cảnh ở bên phải của đầu vào hiện tại? Một cách để khôi phục thông tin là huấn luyện RNN trên một trình tự đầu vào ngược lại, áp dụng đúng loại mạng mà chúng ta đang nghiên cứu. Với cách tiếp cận này, trạng thái ẩn tại thời điểm t hiện tại đại diện cho thông tin về trình tự ở bên phải của đầu vào hiện tại.

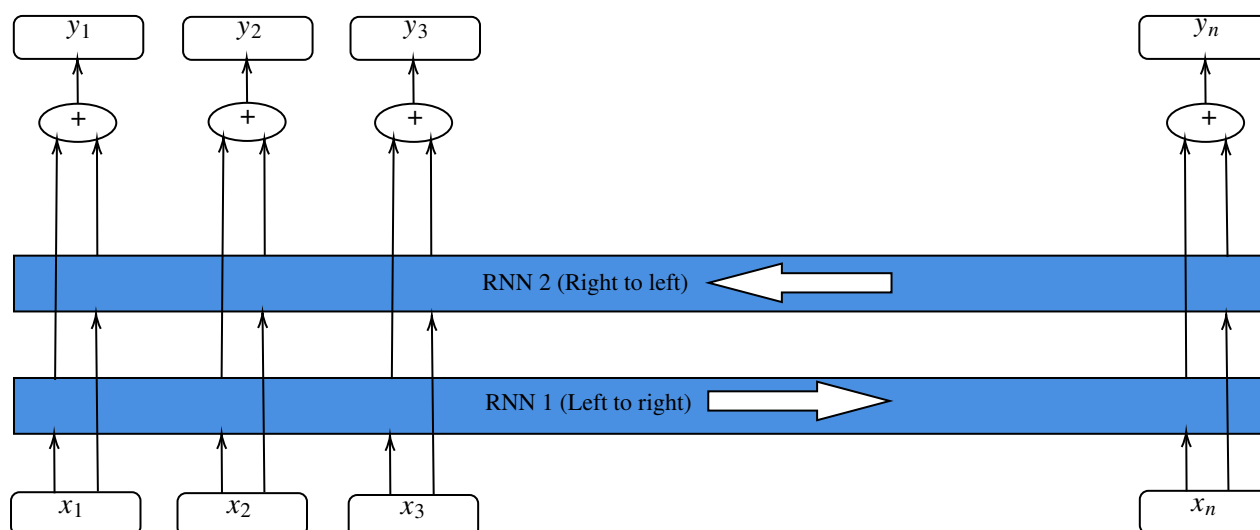
$$h_t^b = RNN_{forward}(x_t^n)$$

- Trạng thái ẩn h_t^b đại diện cho tất cả thông tin chúng ta đã xác minh về chuỗi từ t đến điểm cuối của chuỗi.
- Kết hợp giữa hai loại mạng: mạng tiến và mạng lùi dẫn đến mạng RNN hai chiều. Một Bi-RNN bao gồm hai RNN độc lập, trong đó một mạng đầu vào của nó sẽ được xử lý từ đầu đến cuối, và mạng còn lại sẽ được xử lý từ cuối đến đầu. Sau đó chúng ta kết hợp các đầu ra của hai mạng thành một biểu diễn duy nhất nắm bắt cả bối cảnh bên trái và bên phải của đầu vào tại mỗi thời điểm.

$$h_t = h_t^f \otimes h_t^b$$

- Hình dưới minh họa một mạng hai chiều trong đó các đầu ra của mạng tiến và mạng lùi được kết nối. Cách đơn giản khác để kết nối giữa ngữ cảnh tiến và lùi bao gồm phép cộng hoặc phép nhân thông thường. Đầu ra ở mỗi bước trong khoảng thời gian do đó nắm bắt thông tin ở bên trái và bên phải của đầu vào hiện tại. Trong các ứng dụng gắn nhãn theo trình tự, các đầu ra được nối này có thể đóng vai trò là cơ sở cho một quyết định gắn nhãn dữ liệu.

- RNN hai chiều cũng đã được chứng minh là khá hiệu quả trong bài toán phân loại chuỗi. Chúng ta sẽ chỉ cần kết hợp trạng thái ẩn từ chiều tiến và chiều lùi, sử dụng nó làm đầu vào để theo dõi quá trình xử lý.



Hình 4.8: Mô hình mô phỏng RNN hai chiều

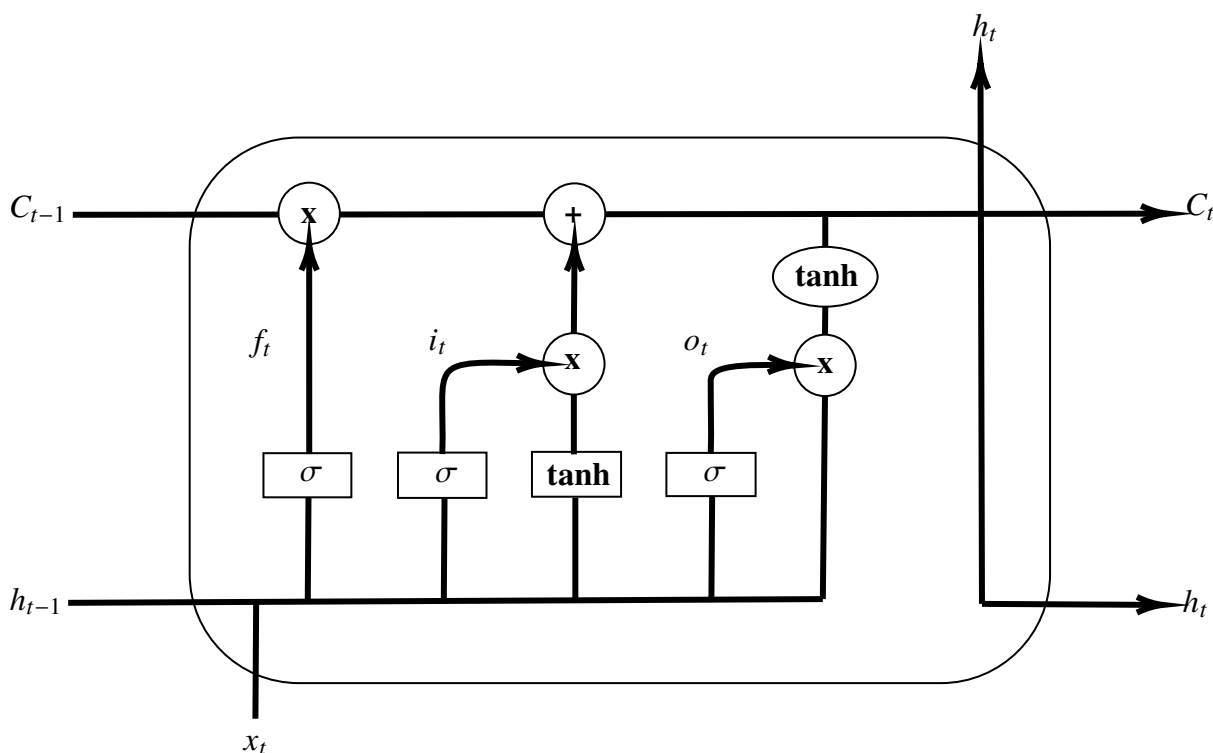
4.5.2 Mô hình LSTM

Khái niệm

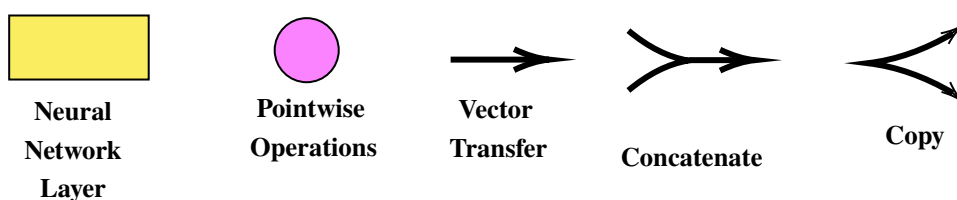
- Mạng trí nhớ ngắn hạn định hướng dài hạn còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN có khả năng học được sự phụ thuộc trong dài hạn (long-term dependencies) được giới thiệu bởi **Hochreiter** và **Schmidhuber** (1997). Kiến trúc này đã được phổ biến và sử dụng rộng rãi cho tới ngày nay. LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm. Tuy nhiên cấu trúc của chúng có phần phức tạp hơn mặc dù vẫn dựa trên tư tưởng chính của RNN là sự sao chép các kiến trúc theo dạng chuỗi.
- LSTM được thiết kế để tránh vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kỳ can thiệp nào.
- LSTM cũng có một chuỗi dạng như thế nhưng phần kiến trúc lặp lại có cấu trúc khác biệt hơn. Thay vì chỉ có một tầng đơn, chúng có tới 4 tầng ẩn (3 sigmoid và 1 tanh) tương tác với nhau theo một cấu trúc đặc biệt.
- Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một tầng tanh.

Ý tưởng

- Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ. Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy



Hình 4.9: Mô hình LSTM



Hình 4.10: Diễn giải các kí hiệu trong đồ thị mạng nơ ron (áp dụng chung cho toàn bộ bài)

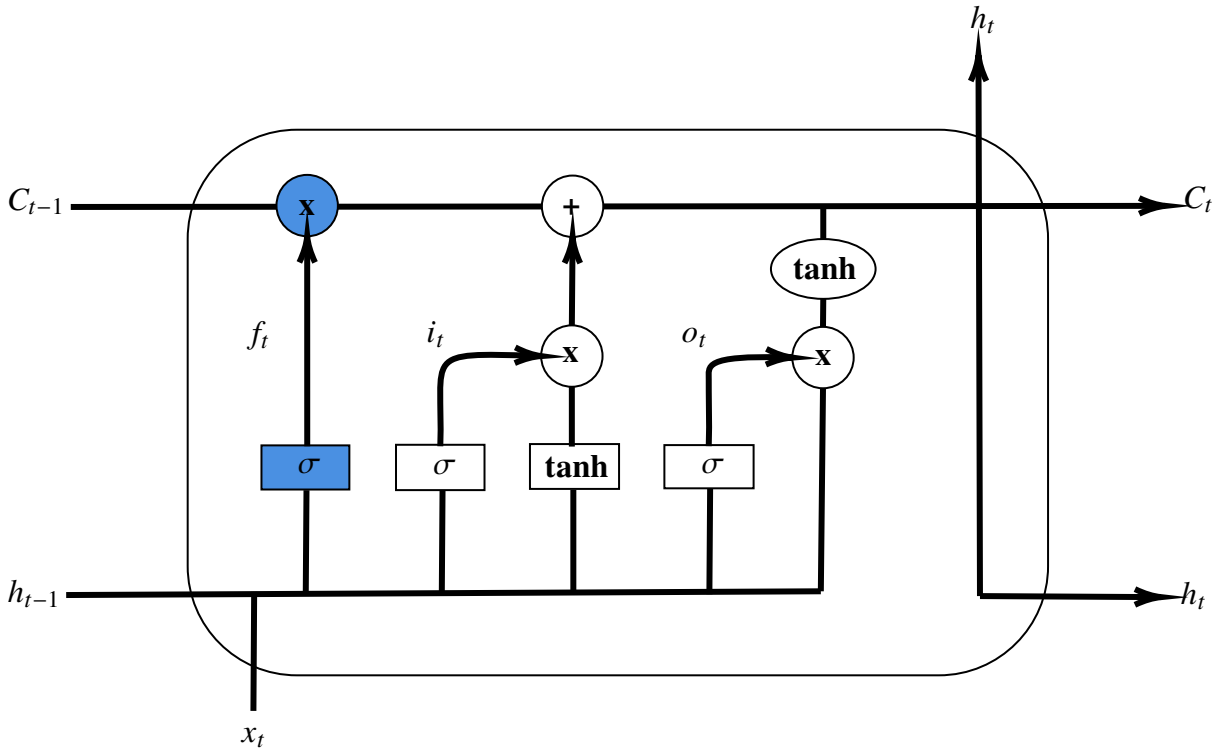
xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

- LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate). Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.
- Tầng sigmoid sẽ cho đầu ra là một số trong khoảng $[0, 1]$, mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Cấu trúc LSTM

- LSTM gồm 4 thành phần chính: **cell state**, **input gate**, **output gate** và **forget gate**. Các cell có nhiệm vụ nhớ các giá trị trong khoảng thời gian phụ thuộc và giá trị của cổng forget gate. Ba cổng có nhiệm vụ điều chỉnh luồng thông tin vào ra khỏi cell.

- Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là "tầng cổng quên" (forget gate layer). Nó sẽ lấy đầu vào là h_{t-1} và x_t rồi đưa ra kết quả là một số trong khoảng $[0, 1]$ cho mỗi số trong trạng thái tế bào C_{t-1} . Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.



Hình 4.11: Forget gate layer của LSTM

- Công thức của forget gate là:

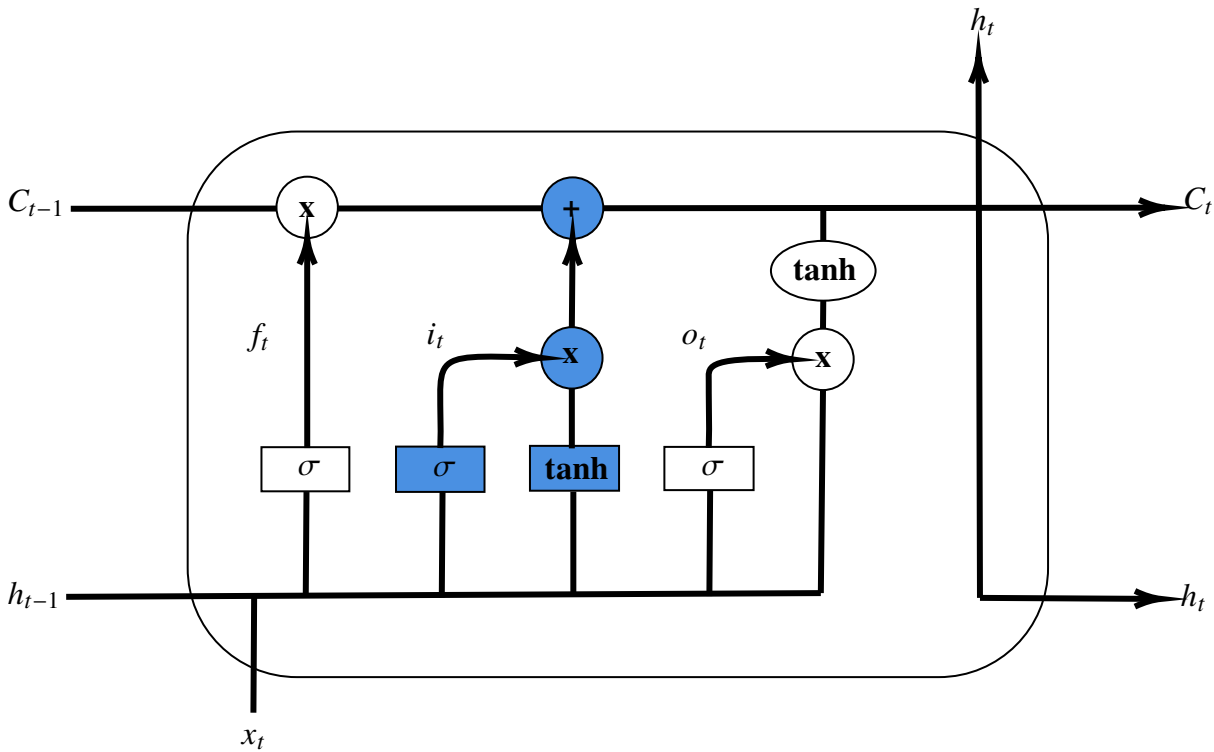
$$f_t = \sigma (W_f [h_{t-1}, x_t] + b_f)$$

- Trong đó W_f và b_f là ma trận trọng tham số bias
- Bước tiếp theo, input gate quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Nó bao gồm hai phần: hàm sigmoid quyết định các giá trị được cập nhật và hàm tanh tạo ra một giá trị mới cho trạng thái tế bào. Hàm tanh sử dụng dữ liệu vào x_t và trạng thái ẩn trước đó h_{t-1} và tạo ra một vector \tilde{C}_t có giá trị trong khoảng $[-1, 1]$. Hàm sigmoid i_t cũng giống như việc tính toán hàm f_t . Sau đó, ta sẽ đi nhân từng phần vector i_t với \tilde{C}_t và kết quả được thêm vào trạng thái tế bào trước đó để cập nhật trạng thái tế bào hiện tại. Giá trị đầu ra trong khoảng $[-1, 1]$ của hàm tanh sau khi nhân với hàm sigmoid quyết định mức độ quan trọng của dữ liệu đầu vào hiện tại phải quên đi hoặc giữ lại trong việc cập nhật trạng thái tế bào mới. Công thức của input gate là:

$$i_t = \sigma (W_i [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh (W_c [h_{t-1}, x_t] + b_c)$$

- Trong đó, W_i và W_c là ma trận trọng số ở input gate của hàm sigmoid và hàm tanh. Các tham số bias b_i và b_c cũng tương ứng của hàm sigmoid và hàm tanh.



Hình 4.12: Input gate trong LSTM

- Tiếp đến cập nhật trạng thái tế bào trước đó C_{t-1} thành trạng thái mới C_t . Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong. Ta sẽ nhân trạng thái trước đó C_{t-1} với f_t để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm kết quả của i_t nhân từng phần với \tilde{C}_t . Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao. Công thức được tính theo như sau:

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

- Cuối cùng, ở cổng output gate, ta tính toán hàm sigmoid o_t cũng tương đương các hàm f_t và i_t . Rồi ta sẽ lấy kết quả o_t và vector C_t để tạo ra trạng thái ẩn hiện tại h_t tương ứng công thức sau:

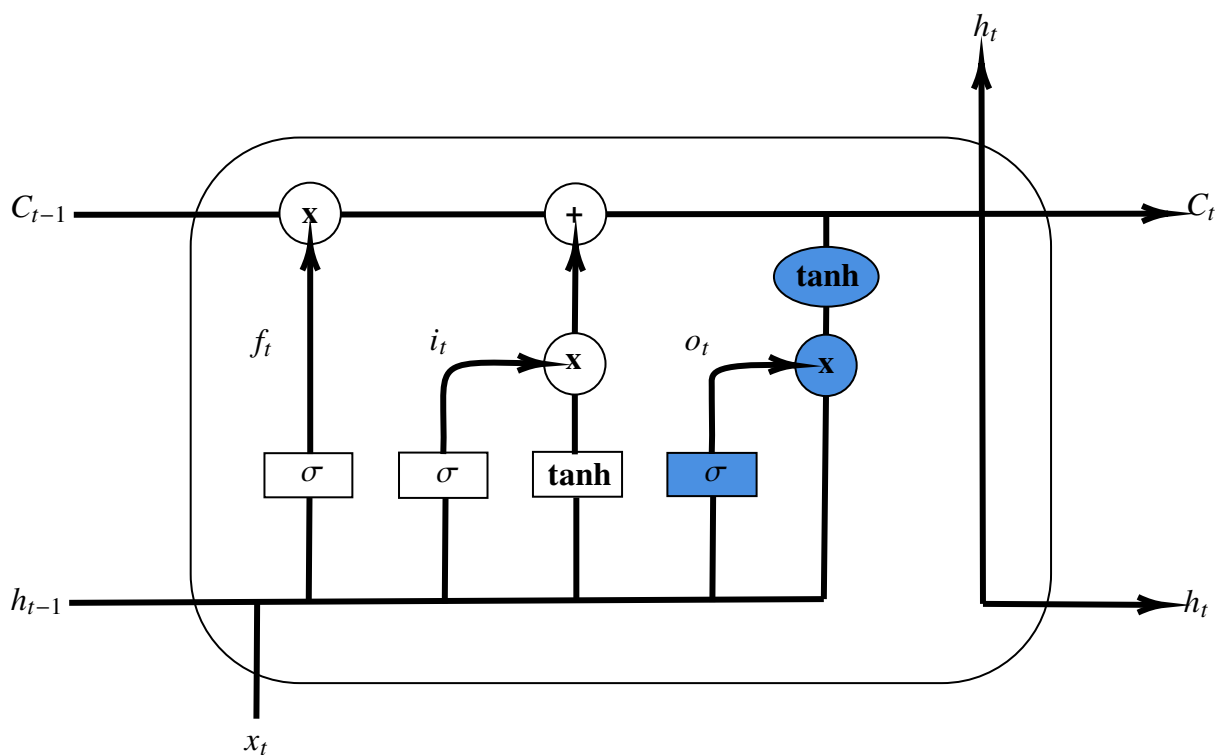
$$o_t = \sigma(W_0[h_{t-1}, x_t] + b_0)$$

$$h_t = o_t \otimes \tanh(C_t)$$

- Trong đó, W_0 và b_0 là ma trận trọng số và tham số bias

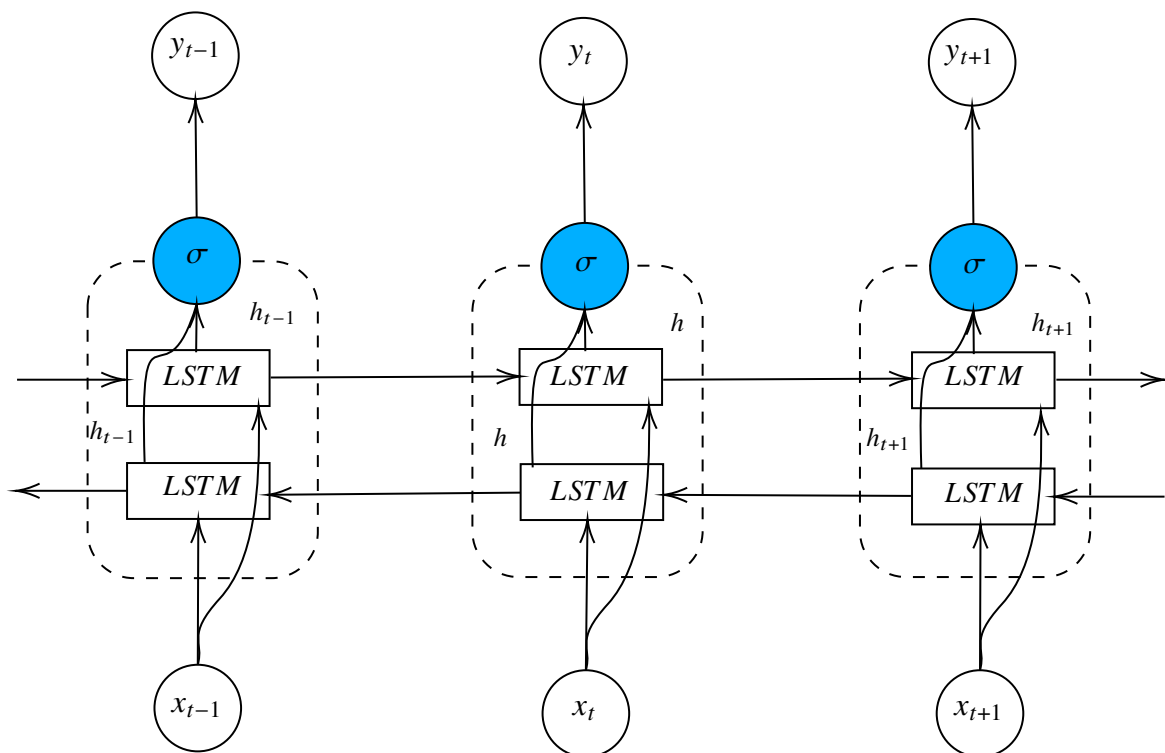
Mạng LSTM 2 chiều

- Như ở 3.1.2.4, ta có thể sử dụng mạng nơ ron hồi quy theo hai chiều ngược nhau để xử lý Một đơn vị RNN sẽ làm như thường lệ, tức là ta sẽ dùng nó để học các tín hiệu đầu vào từ thời điểm ban đầu tới thời điểm kết thúc (đi xuôi). Còn đơn vị RNN còn lại, ta sẽ đọc theo thứ tự thời điểm từ kết thúc trở lại ban đầu (đi ngược). Sau khi có cả hai kết quả, chúng sẽ được gom lại thành một để có thể dự đoán. Với ý tưởng như vậy, tại một thời điểm bất kỳ, mạng sẽ có được các thông tin trước và sau thời điểm t hiện tại.
- Do bản chất LSTM là cải tiến của RNN, cho nên ta có thể áp dụng nó và biến nó thành mạng nơ ron dài ngắn song song (BiLSTM). Mỗi LSTM sẽ vẫn có khả năng quên thông



Hình 4.13: Output gate layer trong LSTM

tin cũ (cổng quên), lọc thông tin mới (cổng đầu vào), hoặc giấu bớt kết quả (cổng đầu ra) như bình thường. Chính vì vậy, các thông tin từ quá khứ tới tương lai của mạng BiLSTM đều có thể tự học để tự điều chỉnh. Dẫn tới việc với các bài toán mà ta cần biết nhiều hơn về ngữ cảnh hiện tại của nó, thì mạng BiLSTM cho kết quả tốt hơn [4]. Hình dưới đây mô tả mạng BiLSTM:



Hình 4.14: Mô tả mô hình BiLSTM

Chương 5

Giải pháp đề xuất

Trong các nghiên cứu liên quan ở chương 2, với các bài báo ở trên thì em sẽ sử dụng các thuật toán để cải tiến lại hoặc là có thể cải tiến phần tạo nhiều và tiền xử lý dữ liệu

5.1 Thu thập dữ liệu

- Nguồn tập dữ liệu: [7]
- Dữ liệu sử dụng trong đề tài nghiên cứu lần này được tổng hợp từ nhiều trang báo điện tử khác nhau như vnexpress.net, tuoitre.vn, thanhnien.vn, nld.com.vn. Các bài báo từ rất nhiều lĩnh vực trong cuộc sống như chính trị xã hội, đời sống, khoa học, kinh doanh, pháp luật, sức khỏe, thể giới, thể thao, văn hoá, ... được chia thành các file nhỏ giúp việc huấn luyện mô hình trở nên dễ dàng và thuận tiện hơn. Bộ dữ liệu sẽ được chia thành hai tập: tập **train** và tập **test**.

– Tập train:

Topic	Topic ID	Files
Chính trị xã hội	XH	5219
Đời sống	DS	3159
Khoa học	KH	1820
Kinh doanh	KD	2252
Pháp luật	PL	3868
Sức khỏe	SK	3384
Thể giới	TG	2898
Thể thao	TT	5298
Văn hóa	VH	3080
Công nghệ thông tin	CNTT	2481
Tổng		33759

– Tập Test:

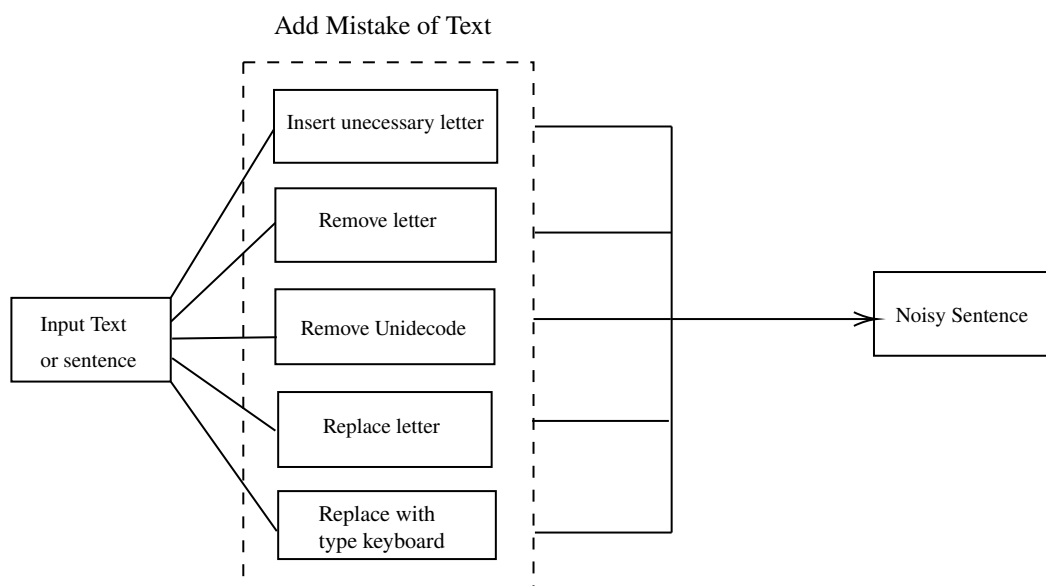
Topic	Topic ID	Files
Chính trị xã hội	XH	7567
Đời sống	DS	2036
Khoa học	KH	2096
Kinh doanh	KD	5266
Pháp luật	PL	3788
Sức khỏe	SK	5417
Thể giới	TG	6716
Thể thao	TT	6667
Văn hóa	VH	6250
Công nghệ thông tin	CNTT	4560
Tổng		50373

5.2 Tiền xử lý dữ liệu

5.2.1 Thực hiện, thu thập và xử lý dữ liệu

Trong phần xử lý dữ liệu thì sẽ được thao tác từ tập văn bản nhiều lần lượt đưa chuyển hóa sang bộ vector. Sau đó ta thực hiện xong xong quy trình huấn luyện mô hình học sâu và đánh giá quá trình huấn luyện với bộ vector này.

Tạo nhiễu văn bản

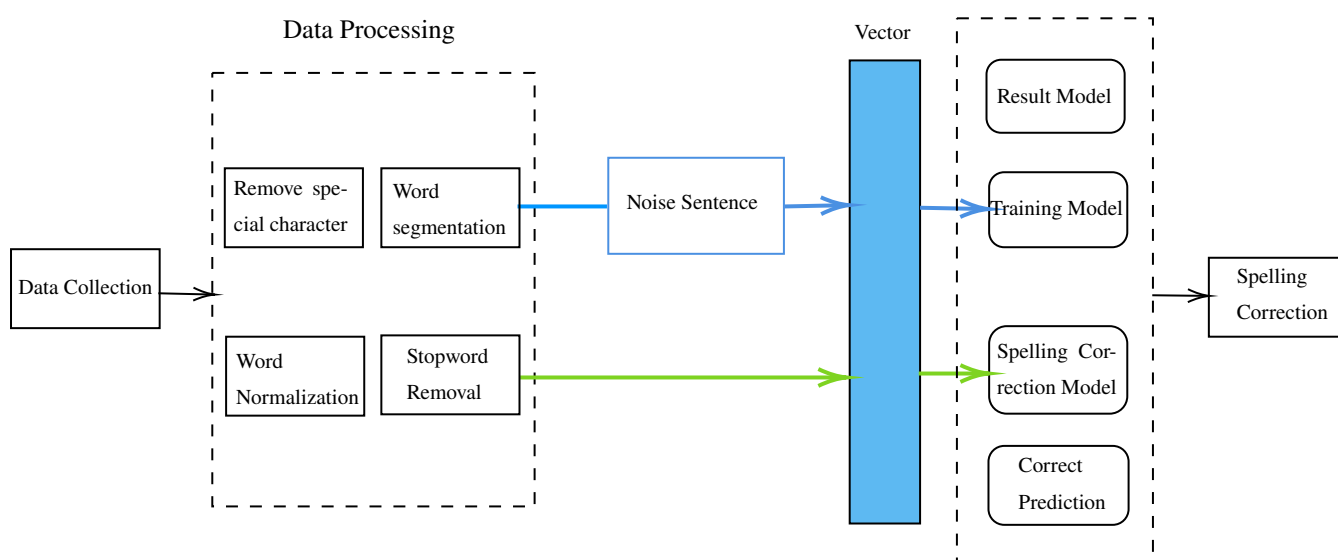


Hình 5.1: Tạo lỗi văn bản

- Tích hợp Fuzzy vào phần tạo nhiễu trong câu đã nhập: Tính tổng độ lỗi thông qua tích hợp Fuzzy với mức độ lỗi tạo nhiễu trong các câu dao động từ 0 đến 1, trong đó mỗi hàm tạo nhiễu thì có thể cho mức độ lỗi khác nhau (cũng dao động từ 0 đến 1) mà hàm tạo nhiễu này có thể thay đổi cả nội dung lẫn cấu trúc câu, sau đây một số hàm tạo nhiễu sẽ được đánh giá như sau:

- Với những lỗi như là xoá dấu câu hay là thay thế bằng các từ lóng hay từ có phát âm giống với từ gốc thì có thể tạo mức độ lỗi là 0.25
 - Với những lỗi như là lỗi bị mất, thêm hoặc thay thế chữ cái thì có thể tạo mức độ lỗi ảnh hưởng đến câu là 0.5
 - Với những lỗi như là lỗi gõ bàn phím như là lỗi gõ chữ cái có câu hay là cặp âm thì có thể tạo lỗi lớn hơn là 0.75
- Tuỳ mức độ mà chúng ta cho, chúng ta có thể hoàn toàn đánh giá được mức độ ảnh hưởng tạo nhiễu đến câu văn mà chúng ta đã nhập vào
 - Trong trường hợp mà câu mà không có hàm tạo nhiễu nào cho vào (tức là câu đó không có lỗi chính tả) thì độ lỗi của câu sẽ là bằng 0

Xử lý dữ liệu và xây dựng mô hình



Hình 5.2: Xử lý dữ liệu

Trong quy trình tạo nhiễu và xử lý dữ liệu thì cũng có một số bài báo đã xử lý theo quy trình và sơ đồ trên như bài báo [13]

- Xử lý và chất lọc những dữ liệu có các từ vựng và chữ cái đều sử dụng tiếng Việt. Loại bỏ các ký tự đặc biệt không cần thiết
- Có thể sử dụng các thuật toán như mô hình N Grams, có thể kết hợp với TF-IDF và N-Grams hoặc là loại bỏ Stopwords để dễ dàng huấn luyện mô hình và tiết kiệm tài nguyên, tránh lãng phí
- Đưa tập dữ liệu đã tách từ và chuẩn hoá vào thành vector rồi tiến hành thiết lập mô hình học sâu để huấn luyện dữ liệu

5.2.2 Công cụ xử lý

Trong đồ án này để thử nghiệm được mô hình chúng em đã kết hợp sử dụng các thư viện mã nguồn mở và các công cụ tự xây dựng để xử lý dữ liệu, huấn luyện mô hình và dự báo.

- **NLTK**: Công cụ xử lý ngôn ngữ tự nhiên mã nguồn mở dành riêng cho NLP và được tích hợp vào Python. Nó đang ngày càng hoàn thiện và tích hợp các công cụ mới bởi hàng ngàn lập trình viên và cộng tác viên trên khắp thế giới. NLTK bao gồm những thư viện hàm, các công cụ phân tích, các corpus, wordnet, ... giúp đơn giản hoá, tiết kiệm thời gian và công sức cho các lập trình viên.
- **Underthesea**: Do sự khác nhau về tiếng Anh và tiếng Việt, dựa trên sự phát triển của gói công cụ NLTK, nhóm phát triển người Việt đã tạo nên công cụ thuần Việt nhất, hỗ trợ xử lý các bài toán trong ngôn ngữ tiếng Việt một cách tối ưu nhất. Đây là một công cụ mã nguồn mở tích hợp cho Python, giải các bài toán tách từ, gán nhãn từ loại tiếng Việt một cách thuận tiện nhất.
- **Tensorflow**: Một khung làm việc mã nguồn mở do Google phát hành được sử dụng để xây dựng các mô hình học máy, tạo môi trường nghiên cứu, thực hiện các thử nghiệm một cách nhanh chóng và dễ dàng, đặc biệt là có khả năng chuyển đổi các bản thiết kế prototype tới các ứng dụng trong sản xuất.
- **Python**: Ngôn ngữ lập trình để xây dựng mô hình đối thoại tiếng Việt.
- **Google Colab**: Cho phép viết và thực thi Python trong trình duyệt với các lợi ích: Không yêu cầu cấu hình, sử dụng miễn phí GPU, chia sẻ dễ dàng

5.3 Phương pháp đề xuất

Xây dựng mô hình

- Trong đề tài này em đã đề xuất mạng LSTM hai chiều (BiLSTM) để cải tiến thêm cho việc huấn luyện của mô hình
- Mô hình BiLSTM với giá trị như sau:

Lớp	Tham số
Input Embedding	128 neuron
BiLSTM	128 neuron
Activation (Softmax)	128 neuron
BATCH SIZE	256

Đánh giá chính xác mô hình

- Dựa vào số dự đoán đúng chia cho tổng số mẫu để hiển thị các kết quả như f1-score, recall, precision
- Chuẩn hoá phần dự đoán mô hình về dạng nhị phân 0 và 1 (0 là phần dự đoán đúng chính tả còn 1 thì là dự đoán phần sai chính tả)
- Tuy nhiên nó có nhược điểm là sẽ có nhược điểm như sau:

- Mất cân bằng lớp: Nếu tập dữ liệu có sự mất cân bằng giữa các lớp hoặc giá trị mục tiêu, việc lấy tổng số mẫu để tính toán các phép đo này có thể làm mất đi sự cân nhắc giữa các lớp. Kết quả là, mô hình có thể có hiệu suất cao trên các lớp thiểu số nhưng thấp trên các lớp đa số, và điều này không được phản ánh trong phép đo tổng quát.
- False Positives và False Negatives: Các phép đo Precision, Recall và F1-Score không phản ánh mức độ của các sai dự đoán cụ thể như false positives (dự đoán sai positive) và false negatives (dự đoán sai negative). Việc lấy tổng số mẫu để tính toán các phép đo này không cho ta thông tin về sự phân loại sai cụ thể và có thể che đi một số vấn đề quan trọng.

Chương 6

Triển khai thực nghiệm

6.1 Môi trường thực nghiệm

- Môi trường thực nghiệm: Google Colab hoặc Visual Studio Code
- Ngôn ngữ lập trình: Python
- Cấu hình máy tính: Windows 11, RAM 8 GB
- Cấu hình Google Colab: Cap GPU, 51 GB RAM, thực hiện được với 100 phép tính và 12.5 GB RAM GPU

6.2 Kết quả thực nghiệm

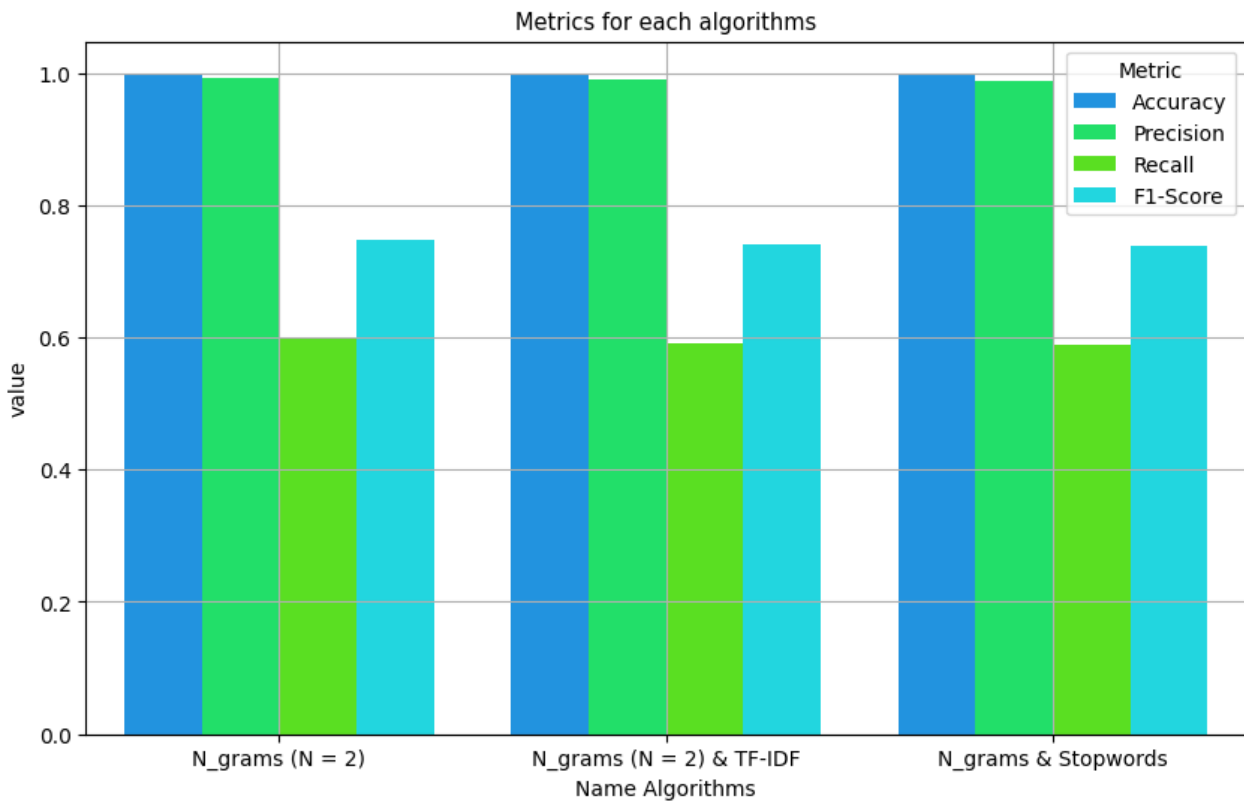
6.2.1 Đánh giá

- Tạo nhiễu trong một đoạn văn bản hoặc câu: Chạy 10 vòng với các hàm tạo nhiễu, ta có được các kết quả như sau, cho tỉ lệ lỗi là 0.18 (percent_error):
- Các lỗi ở đây bao gồm:
 - Lỗi mất chữ cái trong văn bản
 - Lỗi thêm chữ cái trong văn bản
 - Lỗi thay thế chữ cái
 - Lỗi mất dấu câu
 - Lỗi thay thế chữ cái trong từ
- Với các trường hợp tạo lỗi trên, mỗi one_hot_label sẽ là một nhãn và cũng sẽ là 1 từ vựng để tạo nhiễu vào các phần trong câu, các hàm sẽ tạo ngẫu nhiên để nổi hiển thị các lỗi như đề cập ở trên
- Câu được nhập vào như sau: "Đời loài người này ngắn lắm"
 - Epoch 1:
 - * Noise sentence: Đời loài người này ngắn lắm
 - * Total error of sentence: 0

- Epoch 2:
 - * Noise sentence: Đời loài người này ngắn lam
 - * Total error of sentence: 0.12499999999999999
 - Epoch 3:
 - * Noise sentence: Đời loài người này ngansw lắ
 - * Total error of sentence: 0.6649801992713752
 - Epoch 4:
 - * Noise sentence: Đời loài gười này ngắn lắ
 - * Total error of sentence: 0.24999999999999997
 - Epoch 5:
 - * Noise sentence: 0.4506775965978417
 - * Total error of sentence: Đời loài người này ngắn lắ
 - Epoch 6:
 - * Noise sentence: Dđời loài người này ngắn lắ
 - * Total error of sentence: 0.29999999999999993
 - Epoch 7:
 - * Noise sentence: Đời loài người này ngắn lắ
 - * Total error of sentence: 0.4377788686706765
 - Epoch 8:
 - * Noise sentence: Đời loàm người này ngắn lawsm
 - * Total error of sentence: 0.616717647347905
 - Epoch 9:
 - * Noise sentence: Đời loài người này ngắn lém
 - * Total error of sentence: 0.12499999999999999
 - Epoch 10:
 - * Noise sentence: 0.29999999999999993
 - * Total error of sentence: Đời loài người này ngắn lawsm
- Tuy nhiên đó là một câu nhập có độ dài ngắn nên hoặc là tỉ lệ độ tạo nhiễu nhỏ hơn nên chưa thể bao quát được các hàm tạo nhiễu trên, nếu muốn tạo nhiễu nhiều hơn thì có thể là nhập các câu có độ dài nhiều hơn và có độ tạo nhiễu nhiều hơn hoặc là tăng tỉ lệ độ nhiễu lên (precent_error), từ nào mà có nhiều ký tự thì có thể thấy rõ hơn
 - Đánh giá độ chính xác mô hình:

Name Model	Accuracy	Precision	F1-Score	Recall
N_grams	0.997969	0.991461	0.748326	0.600955
TF_IDF & N_grams	0.997918	0.990713	0.740360	0.591232
Stopwords & N_grams	0.997898	0.988741	0.73778	0.588426

- Như vậy là ta có thể thấy được rằng là mô hình N_grams cho kết quả F1-Score và Recall cao hơn so với 2 thuật toán còn lại (Mặc dù accuracy có độ chính xác lên đến 99%) nên ta có thể lấy phần có kết quả tốt nhất để kiểm thử và sửa lỗi một số câu và đoạn văn như ở phần kiểm thử dưới đây.
- Tuy nhiên nhưng thời gian huấn luyện với mô hình N_grams sẽ tốn nhiều thời gian hơn so với khi N_grams kết hợp với TF-IDF và Stopwords. Dưới đây là biểu đồ rõ hơn về đánh giá thuật toán



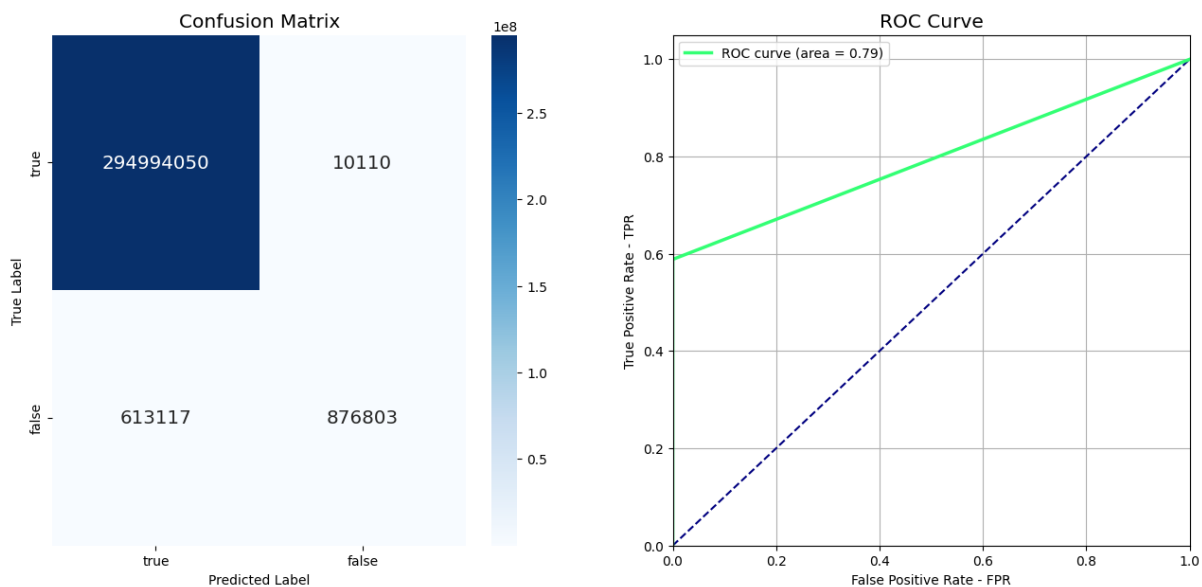
Hình 6.1: So sánh các mô hình với các thuật toán đề xuất

6.2.2 Kiểm thử

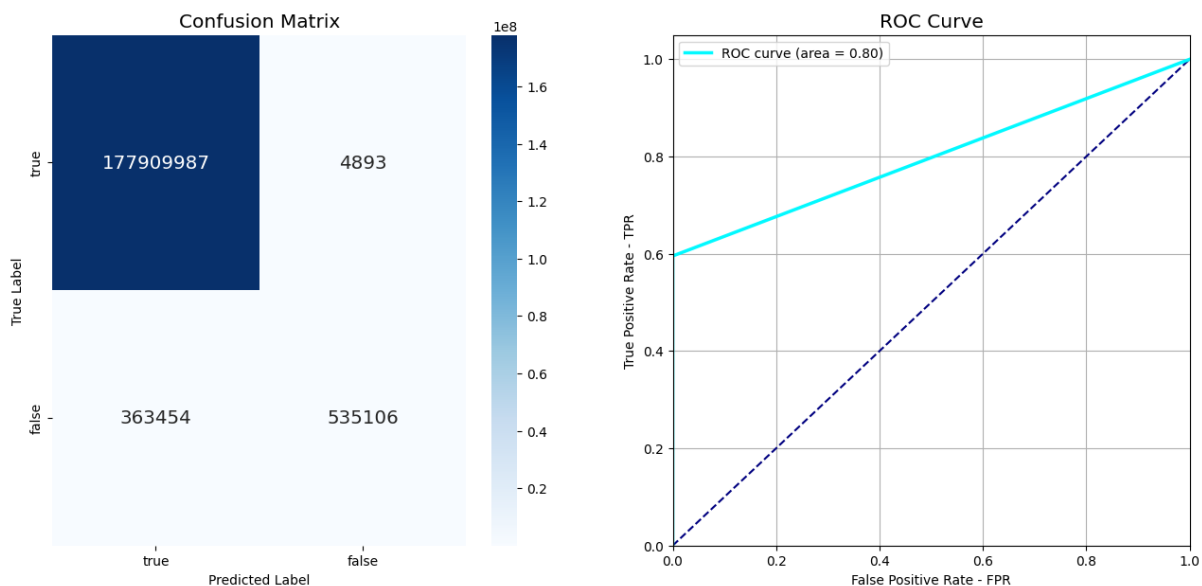
- Các câu bị sai lỗi chính tả

Câu văn bị lỗi chính tả	Câu văn đã sửa lỗi chính tả
Anh vaf em	Anh và em
T iu e	Tôi yêu em
Chuyền đajt kiến thức	Truyền đạt kiến thức
Toorng cunj tình báo TocoToco	Tổng cục tình báo TocoToco
Ngon lúi này rất ca	Ngon núi này rất cao
Phát thank dẫn chương chinh	Phát thanh dẫn chương trình

- Các đoạn văn bị sai lỗi chính tả



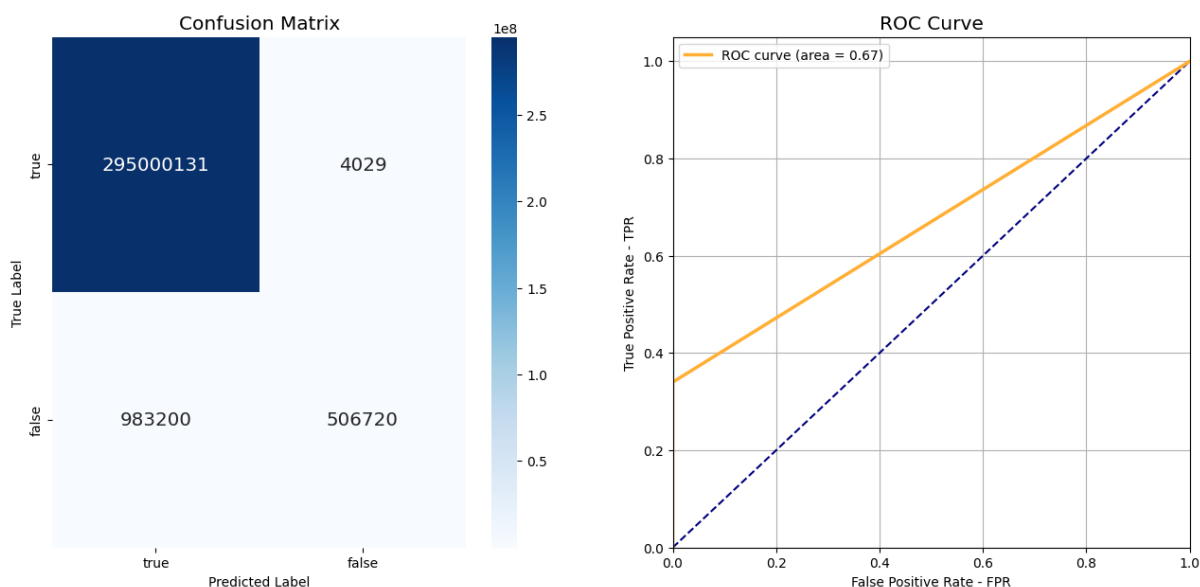
Hình 6.2: ROC của N_grams



Hình 6.3: ROC của N_grams với TF-IDF

– Câu 1:

- * **Câu sai:** Các phát **thanh** viên, **người** dẫn **chương** trình đều là **các chuyên gia** sử dụng ngôn ngữ. Thế **nhưng** đa phần họ **đều** cho rằng **mình** **không** có khiếu ăn **nhai** từ **nhỏ**, vậy **tại** sao họ vẫn **thành** công **nhờ** vào tài ăn nói của **mình**? Nguyên **nhân** **rất** đơn giản, đó là vì họ tự **nhận** thấy **mình** nói năng **không** được tốt, nên luôn cố gắng để nâng cao kỹ năng giao **tiếp**
- * **Câu đúng:** Các phát **thanh** viên **người** dẫn **chương** trình đều là **các chuyên gia** sử dụng ngôn ngữ. Thế **nhưng** đa phần họ **đều** cho rằng **mình** **không** có khiếu ăn **nói** từ **nhỏ** vậy **tại** sao họ vẫn **thành** công **nhờ** vào tài ăn nói của **mình**? Nguyên **nhân** **rất** đơn giản đó là vì họ tự **nhận** thấy **mình** nói năng **không** được tốt, nên luôn cố gắng để nâng cao kỹ năng giao **tiếp**



Hình 6.4: ROC của Stopwords

– Câu 2:

- * **Câu sai:** **Nhuwxng** con người có công và **giú** đỡ **t** khi tôi **gawjp** khó **khawn** thì tôi sẽ mãi mãi ghi nhớ, **t** sẽ trân trọng **hoj**, chứ **ko** phải cái đứa con gái nào từng làm ở TocoToco nó than: "Sầu **dowwfi** ai rủ **cuxng** tu", đúng là con **deeru** rồi lại **giar** dối đi **haji** người khác thế đâu, loại **nhuw** thế này, khứ đi cho xong!
- * **Câu đúng:** **Những** con người có công và **giúp** đỡ **tôi** khi tôi **gặp** khó khăn thì tôi sẽ mãi mãi ghi nhớ, **tôi** sẽ trân trọng **họ**, chứ **không** phải cái đứa con gái nào từng làm ở TocoToco nó than: "Sầu **đời** ai rủ **cũng** tu", đúng là con **đều** rồi lại **giả** dối đi **hại** người khác thế đâu, loại **như** thế này, khứ đi cho xong!

– Câu 3:

- * **Câu sai:** Các **phast** thanh viên **chuyền** hình, **laf** những **cno** **nguowfi** đã nổi **tiesng** rồi, **hoj** thường **noori** bật với cách ăn nói **truyeenf** tải đến **vowsi** khán giả **car** nước, nhiều **quys** khán **giar** đã rất ấn **tuownjg** với phong **cach** của họ, khán **giar** hâm mộ thường ấn tượng họ vì **hoj** đã **vieest** lên những câu **chuyeejn** để tạo nên **đuowjc** điểm nhấn mà **khasn** giả **ko** bao **gio** quên **đuowjc**, họ áp lực để làm **seo** mà giữ được hình **arnh** của mình **trướ** mắt của công chúng, chỉ là **nếk** có sự cố gì **xary** ra thôi là đi luôn **car** sự nghiệp và rất khó để **quayy** lại và được khán giả đón nhận như trước **kiak**
- * **Câu đúng:** Các **phát** thanh viên **truyền** hình là những **con người** đã nổi **tiếng** rồi, **họ** thường **nổi** bật với cách ăn nói **truyền** tải đến **với** khán giả **cả** nước, nhiều **quý** khán **giả** đã rất ấn **tượng** với phong **cách** của họ, khán **giả** hâm mộ thường ấn **tượng** họ vì **họ** đã **viết** lên những câu **chuyện** để tạo nên **được** điểm nhấn mà **khán** giả **không** bao **giờ** quên **được**, họ áp lực để làm **sao** mà giữ được hình **ảnh** của mình **trước** mắt của công chúng chỉ là **nếu** có sự cố gì xảy ra thôi là đi luôn **cả** sự nghiệp và rất khó để **quay** lại và được khán giả đón nhận như trước **kia**

– Câu 4:

- * **Câu sai:** **Bieen** tập **vien** Lê Quang Minh **nguowif** đàn ông **cura** chương **chình** **thowfi** sự 19 **giowf** dẫn trên đài **chuyên hìn** Việt Nam là **ngườ** đã gây **aasn** tượng

với khán giả **chuyền** hình với cách **dan** ấn tượng, đặc biệt là chương trình đón giao **thuwf** trên kênh sóng của VTV vào **toois** 30 Tết **Nguyene** Đán với một phong **cach** dẫn ấn tượng, **tajo** được sự tin **tuowrng** của khán giả khi anh **đuwa** tin tức trên màn **arnh** nhỏ.

- * **Câu đúng:** Biên tập viên Lê Quang Minh **người** dẫn ông **của** chương **trình** thời sự 19 **giờ** dẫn trên đài **truyền hình** Việt Nam là **người** đã gây **ấn** tượng với khán giả **truyền** hình với cách **dẫn** ấn tượng, đặc biệt là chương trình đón giao **thừa** trên kênh sóng **của** VTV vào **tối** 30 Tết **Nguyên** Đán với một phong **cách** ấn tượng, **tạo** được sự tin **tưởng** của khán giả khi anh **đưa** tin tức trên màn **ảnh** nhỏ

– Câu 5:

- * **Câu sai:** Văn **Sỹ** Hộ **sinl** trưởng trong một gia **đifnh** có **chuyền** thống thể thao, **casc** anh **e** của Văn **Sỹ** Hộ đều là những **cào** thủ xuất **sac** trong **độiy** hình của đội tuyển **Soong** Lam Nghệ An đặc **biệt** là người **ank car** Văn Sĩ Hùng người đã ghi nhiều bàn **thawsng** cho **đooji** tuyển Việt Nam ở Sea Games 19 và Tiger Cup 98 thử hỏi con người **tafi** cao bất **khuaast** và chí khí **lex** sống như Văn Sĩ Hộ **lafm** sao mà **ko** thoát khỏi được sự giằng xé **quawfn** quai trong nội tâm, **khum** đời thừa **seo đuwjc**.
- * **Câu đúng:** Văn **Sĩ** Hộ **sinh** trưởng trong một gia **đình** có **truyền** thống thể thao **các** anh **em** của Văn **Sĩ** Hộ đều là những cầu thủ xuất **sắc** trong **đội** hình của đội tuyển **Sông** Lam Nghệ An đặc **biệt** là người **anh cả** Văn Sĩ Hùng người đã ghi nhiều bàn **thắng** cho **đội** tuyển việt nam ở Sea Games 19 và Tiger Cup 98 thử hỏi con người **tài** cao bất **khuyết** và chí khí **lẽ** sống như Văn Sĩ Hộ **làm** sao mà **không** thoát khỏi được sự giằng xé **quần** quai trong nội tâm **không** đời thừa **sao** được.

Chương 7

Kết luận

7.1 Kết luận

- Đề án này đã đưa ra các lý thuyết và vấn đề trong quá trình thiết lập, huấn luyện và xây dựng một mô hình sửa lỗi chính tả cho tiếng Việt. Kết quả ban đầu đạt được là tiền đề để tạo ra các trợ lý ảo, xây dựng các ứng dụng thông minh có thể hiểu được ngôn ngữ tiếng Việt. Có khả năng áp dụng vào các bài toán thực tế, ví dụ như các gợi ý sửa lỗi chính tả trong tình soạn thảo, tự động sửa lỗi chính tả trong tìm kiếm, gợi ý từ tiếp theo trong soạn tin nhắn, ...
- Từ kết quả thực nghiệm của luận văn này, em có một số nhận xét:
 - Với các chuỗi câu dài thì mạng huấn luyện mất nhiều thời gian hơn.
 - Xây dựng hệ thống sửa lỗi chính tả dựa trên mô hình phân loại đầu vào theo hướng mạng nơ ron Bộ nhớ dài ngắn song song BiLSTM: Bộ dữ liệu đầu vào là các văn bản được thêm nhiễu và văn bản gốc. Quá trình huấn luyện được tiến hành dựa trên kỹ thuật mạng nơ ron sâu thông qua hàm softmax để thể hiện xác suất của lớp và Entropy chéo được định nghĩa để đánh giá mục tiêu của đầu ra để dự đoán các câu hỏi được đưa vào của người sử dụng. Phương pháp đánh giá dựa trên độ đo chính xác được sử dụng trong mô hình này nhằm đánh giá kết quả để đưa ra mô hình dự đoán tối ưu.
 - Với độ đo chính xác (Accuracy) đã giải quyết mặt hạn chế do dữ liệu thu thập đầu vào không được phong phú. Mô hình BiLSTM tốn nhiều thời gian huấn luyện hơn nhưng cho kết quả tốt hơn mô hình LSTM.

7.2 Hướng phát triển

- Tiếp tục kế thừa những nghiên cứu trước đây và phát triển mô hình sửa lỗi chính tả mới có khả năng sửa lỗi ở các loại khác ngoài lỗi phát âm, lỗi viết tắt, lỗi tiếng lóng, lỗi “quên bật Vietkey” còn có thể sửa lỗi theo ngữ cảnh.
- Để áp dụng vào thực tiễn, cần lấy dữ liệu thực tế thay vì tự tạo. Do tự tạo nhiều sẽ không thể bao quát được lỗi của tiếng Việt.
- Áp dụng các phương pháp học sâu khác để cải thiện độ chính xác của chương trình cao hơn như: Attention, Beam search, Federated Learning, Transfer Learning hoặc Transformer Learning

Tài liệu tham khảo

- [1] Viblo - Một số phương pháp làm mịn trong mô hình ở trong mô hình N-gram về Xử lý ngôn ngữ tự nhiên - NLP (**Nguyễn Phương Lan**):
<https://viblo.asia/p/mot-so-phuong-phap-lam-min-trong-mo-hinh-trong-mo-hinh-n-gram-aqkRnbMMRnA>
- [2] **Jung-Hun Lee, Minho Kim, Hyuk-Chu Kwon**: Deep Learning-Based Context-Sensitive Spelling Typing Error Correction - Publish in 2020
<https://ieeexplore.ieee.org/document/9160935>
- [3] **Daniel Jurafsky, James H. Martin**: Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition
- [4] Machine Learning cơ bản (Tác giả: **Vũ Hữu Tiệp**)
<https://machinelearningcoban.com>
- [5] **VietHoang1512**: Vietnamese Spelling Correction and Text classification:
<https://github.com/VietHoang1512/vietnamese-spell-correct-and-text-classify>
- [6] **HenryLee**: Vietnamese Spelling Correction:
https://github.com/henryle97/Spelling_Correction_Vietnamese/tree/master
- [7] <https://github.com/duyvuleo/VNTC/tree/master/Data/10Topics/Ver1.1>
- [8] **Kwang H. Lee** - First Course on Fuzzy Theory and Applications
<https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/First%20Course%20n%20Fuzzy%20Theory%20and%20Application.pdf>
- [9] Vietnamese spelling error detection and correction using BERT and N-gram language model - Dong Nguyen Tien, Tuoi Tran Thi Minh, Loi Le Vu and Tuan Dang Minh.
https://www.researchgate.net/publication/362085698_Vietnamese_Spelling_Error_Detection_and_Correction_Using_BERT_and_N-gram_Language_Model
- [10] Scoring, term weighting and the vector space model
<https://nlp.stanford.edu/IR-book/pdf/06vect.pdf>
- [11] Medium Blog (**Kunal Bhashkar**): Spelling Correction Using Deep Learning: How Bi-Directional LSTM with Attention Flow works in Spelling Correction.
<https://bhashkarkunal.medium.com/spelling-correction-using-deep-learning-how-bi-directional-lstm-with-attention-flow-works-in-366fabcc7a2f>

- [12] **Afshine Amidi** and **Shervine Amidi**: Recurrent Neural Networks cheatsheet - CS230
<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [13] VSEC: Transformer-based Model for Vietnamese Spelling Correction - Published in 2021
- Authors : Dinh-Truong Do, Ha Thanh Nguyen , Thang Ngoc Bui, Hieu Dinh Vo
https://www.researchgate.net/publication/355841643_VSEC_Transformer-based_Model_for_Vietnamese_Spelling_Correction
- [14] Demystify TF-IDF in Indexing and Ranking - Ted Mei on Medium Blog
<https://ted-mei.medium.com/demystify-tf-idf-in-indexing-and-ranking-5c3ae88c3fa0>
- [15] TF-IDF - Wikipedia:
<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>
- [16] V. C. D. Hoang, D. Dinh, N. le Nguyen and H. Q. Ngo, "A Comparative Study on Vietnamese Text Classification Methods,"2007 IEEE International Conference on Research, Innovation and Vision for the Future, 2007
<https://ieeexplore.ieee.org/document/4223084>
- [17] Fuzzy Logic - Wikipedia
https://en.wikipedia.org/wiki/Fuzzy_logic
- [18] Hierarchical Transformer Encoders for Vietnamese Spelling Correction
https://www.researchgate.net/publication/351990599_Hierarchical_Transformer_Encoders_for_Vietnamese_Spelling_Correction