TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP**

**WORD ADD-IN - KIỂM TRA**

**VÀ GỢI Ý SỬA LỖI CHÍNH TẢ**

**TIẾNG VIỆT CẢM NGỮ CẢNH**

*Người hướng dẫn*: : **PGS.TS. LÊ ANH CƯỜNG**

*Người thực hiện*: **LÊ NGỌC THÂN – 51303400**

**MAI ANH KIỆT – 51303318**

Lớp **: 13050303**

Khoá  **: 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP**

**WORD ADD-IN - KIỂM TRA**

**VÀ GỢI Ý SỬA LỖI CHÍNH TẢ**

**TIẾNG VIỆT CẢM NGỮ CẢNH**

*Người hướng dẫn*: : **PGS.TS. LÊ ANH CƯỜNG**

*Người thực hiện*: **LÊ NGỌC THÂN – 51303400**

**MAI ANH KIỆT – 51303318**

Lớp **: 13050303**

Khoá  **: 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

LỜI CẢM ƠN

Chúng tôi xin chân thành gửi lời cảm ơn đến thầy Lê Anh Cường người đã tận tình giúp đỡ chúng tôi thực hiện luận văn này trong học kì vừa qua. Thầy đã hướng dẫn chúng tôi bằng sự nhiệt thành của mình cũng như sẵn sàng trả lời bất kỳ thắc mắc nào của chúng tôi một cách đầy đủ nhất trong quá trình thực hiện đề tài này. Việc hoàn thành đề tài này là kết quả của sự nỗ lực, cố gắng không biết mệt mỏi của thầy và trò chúng tôi trong suốt thời gian qua.

Chúng tôi cũng xin gửi những lời cảm ơn chân thành nhất đến gia đình, người thân và bạn bè chúng tôi, những người đã tạo điều kiện, quan tâm, giúp đỡ, động viên chúng tôi trong suốt thời gian qua.

**LUẬN VĂN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm luận văn của riêng chúng tôi và được sự hướng dẫn của PGS. TS. Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Lê Ngọc Thân*

*Mai Anh Kiệt*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Một văn bản được soạn thảo đúng chính tả là một việc hết sức quan trọng nhất là trong các văn bản hành chính, pháp quy. Sai chính tả ở những văn bản hành chính như vậy sẽ gây hậu quả nghiệm trọng.Cùng với sự phát triển của công nghệ thông tin trong nhiều năm trở lại đây, hầu hết văn bản được soạn thảo trên các phần mềm máy tính. Nhưng hiện nay trong các trình soạn thảo văn bản phổ biến lại không tích hợp khả năng kiểm tra lỗi chính tả tiếng Việt một cách mạnh mẽ. Đối với những văn bản, tài liệu dài người soạn thảo sẽ phải mất rất nhiều công sức để kiểm tra lỗi chính tả. Do đó, một công cụ hổ trợ kiểm tra lỗi chính tả một cách tự động là một như cầu rất cần thiết.

Khóa luận này trình bày phương pháp kiểm lỗi chính tả tiếng Việt thông qua mô hình ngôn ngữ N-gram. Với phương pháp tiếp cận này kết hợp với sử dụng Add-in của MS Word để có thể xây dựng một ứng dụng kiểm lỗi chính tả có khả kiểm tra và sửa lỗi mạnh mẽ, có thể đáp ứng vào thực tế.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc483781769)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc483781770)

[TÓM TẮT iv](#_Toc483781771)

[MỤC LỤC 1](#_Toc483781772)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc483781773)

[CHƯƠNG 1 – GIỚI THIỆU 5](#_Toc483781774)

[1.1 Đặt vấn đề 5](#_Toc483781775)

[1.2 Phát biểu bài toán 6](#_Toc483781776)

[1.3 Mục tiêu và phạm vi của khóa luận 6](#_Toc483781777)

[1.4 Một số nghiên cứu liên quan 7](#_Toc483781778)

[1.5 Cấu trúc khóa luận 7](#_Toc483781779)

[CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT 9](#_Toc483781780)

[2.1 Đặc điểm của ngữ pháp tiếng Việt 9](#_Toc483781781)

[2.1.1 Đặc điểm của tiếng việt 9](#_Toc483781782)

[2.1.2 Các đơn vị của tiếng việt 9](#_Toc483781783)

[2.2 Chuẩn chính tả 11](#_Toc483781784)

[2.3 Nguyên nhân gây ra lỗi chính tả trong tiếng Việt 12](#_Toc483781785)

[2.3.1 Lỗi do âm đầu 12](#_Toc483781786)

[2.3.2 Lỗi do âm chính 13](#_Toc483781787)

[2.3.3 Lỗi do âm cuối 13](#_Toc483781788)

[2.3.4 Lỗi do thanh điệu 13](#_Toc483781789)

[2.4 Các trường hợp lỗi chính tả trong văn bản tiếng Việt 13](#_Toc483781790)

[2.4.1 Lỗi do đánh máy sai 14](#_Toc483781791)

[2.4.2 Lỗi do phát âm 14](#_Toc483781792)

[2.5 Giới thiệu Corpus 16](#_Toc483781793)

[2.5.1 Corpus là gì? 16](#_Toc483781794)

[2.5.2 Các bài toán cần xử lý trong quá trình tạo Corpus 16](#_Toc483781795)

[2.5.3 Các bước tạo Corpus cho chương trình 17](#_Toc483781796)

[2.6 Một sô phương pháp kiểm lỗi chính tả 18](#_Toc483781797)

[2.6.1 Kiểm tra lỗi chính tả đối với nước ngoài 18](#_Toc483781798)

[2.6.2 Kiểm lỗi chính tả đối với tiếng Việt 20](#_Toc483781799)

[CHƯƠNG 3 – MÔ HÌNH NGÔN NGỮ N-GRAM 22](#_Toc483781800)

[3.1 Mô hình ngôn ngữ 22](#_Toc483781801)

[3.2 Mô hình ngôn ngữ N-gram 22](#_Toc483781802)

[3.2.1 Tính xác suất bằng mô hình N-gram 23](#_Toc483781803)

[3.2.3 Các phương pháp làm mịn 24](#_Toc483781804)

[CHƯƠNG 4 – GIỚI THIỆU PHƯƠNG PHÁP 26](#_Toc483781805)

[4.1 Mô hình tổng quát 26](#_Toc483781806)

[4.1.1 Phần xử lý văn bản 27](#_Toc483781807)

[4.1.2 Phần kiểm tra lỗi 27](#_Toc483781808)

[4.1.3 Phần sửa lỗi 28](#_Toc483781809)

[4.2 Làm mịn 28](#_Toc483781810)

[4.3 Xây dựng Add-in vào Word 29](#_Toc483781811)

[4.3.1 Tạo Word VSTO Add-in 29](#_Toc483781812)

[4.3.2 Tạo Ribbon (Visual Designer) 30](#_Toc483781813)

[4.3.3 Tạo User Control 32](#_Toc483781814)

[4.3.4 Làm việc với namespace Microsoft.Office.Interop.Word 33](#_Toc483781815)

[CHƯƠNG 5 – CÀI ĐẶT, THỰC NGHIỆM VÀ ĐÁNH GIÁ 36](#_Toc483781816)

[5.1 Cài đặt 36](#_Toc483781817)

[5.1.1 Tạo Courpus thô 36](#_Toc483781818)

[5.1.2 Đếm tần suất N-gram và lưu trữ 36](#_Toc483781819)

[5.1.3 Viết chương trình 37](#_Toc483781820)

[5.2 Thực nghiêm 37](#_Toc483781821)

[5.3 Đánh giá 37](#_Toc483781822)

[CHƯƠNG 6 – TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 38](#_Toc483781823)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. Mô hình tổng quát phương pháp kiểm lỗi 6](#_Toc470152788)

[Hình 2. Ví dụ tìm lỗi và sửa lỗi 8](#_Toc470152789)

[Hình 3. Ví dụ thuật toán Viterbi 14](#_Toc470152790)

[Hình 4. Tạo Word Add-in 18](#_Toc470152791)

[Hình 5. Tạo Ribbon (Visual Designer) 19](#_Toc470152792)

[Hình 6. Thiết kế Ribbon 20](#_Toc470152793)

[Hình 7. Tạo User Control 21](#_Toc470152794)

[Hình 8. Thiết kế User Control 22](#_Toc470152795)

CHƯƠNG 1 – GIỚI THIỆU

1.1 Đặt vấn đề

Tiếng Việt là ngôn ngữ vốn phong phú và đa dạng. Trong việc sử dụng một ngôn ngữ, đảm bảo việc dùng chính xác từ ngữ là một điều rất quan trọng, nhất là trong các văn bản pháp luật, nhà nước. Nhưng hiện nay,tình trạng sai chính tả rất phổ biến. Sai lỗi chính tả xuất hiện ở mọi nơi, từ mọi cấp của các trường học hay thậm chí là giáo viên, từ những quyển sách, các tờ báo, cho tới những thông báo, văn bản của các cơ quan, tổ chức. Hậu quả của việc sai chính tả cũng tùy mức độ mà ảnh hưởng khác nhau, nhẹ thì có thể gây hiểu nhầm, nặng hơn có thể làm cả cộng đồng bị nhầm lẫn chỉ bới sai lầm của một cá nhân.

Nguyên nhân của việc sai lỗi chính tả rất nhiều, phần nhiều là do phương ngữ. Ở các cấp học của học sinh hiện giờ, điều chỉnh chính tả cho các em học sinh không được đảm bảo vì thậm chí nhiều giáo viên còn phát âm và viết sai chính tả do phương ngữ. Các báo hiện nay, nhất là báo điện tử, rất cẩu thả trong khâu kiểm duyệt, vô tình làm người đọc bị sai lỗi chính tả do nghĩ những từ ngữ từ báo đều đúng. Kể cả các biển báo, thông báo của các cơ quan, tổ chức cũng tắc trách và thiếu nghiêm túc trong việc đảm bảo độ chính xác về mặt chính tả. Ở giới trẻ hiện nay, ảnh hưởng từ internet rất lớn, nhất là các mạng xã hội, thường sẽ trao đổi với nhau bằng chữ viết, rất thường viết tắt, viết kiểu “teen-code”, lâu dần sẽ làm giảm khả năng viết đúng đúng chuẩn.

Vì vậy, để có thể khắc phục được việc sai chính tả trong khi soạn thảo văn bản, cần một chương trình hỗ trợ người dùng có thể kiểm tra lại văn bản mình để đảm bảo việc đúng, rõ ràng của văn bản. Trong khóa luận này tập trung vào việc tìm hiểu về tiếng việt, các lỗi thường gặp, nguyên nhân và tìm hiểu về add-in trên word và tích hợp chương trình kiểm lỗi này lên Microsoft Word.

1.2 Phát biểu bài toán

Bài toán kiếm tra lỗi chính tả là bài toán kiểm tra và phát hiện những từ sai lỗi chính tả trong văn bản. Lỗi chính tả được chia thành hai loại, đó là:

* Lỗi sai hoàn toàn âm tiết và không có trong từ điển tiếng Việt.

Ví dụ: iển ( từ đúng là: điển )

* Lỗi sai ngữ cảnh, âm tiết này có trong từ điển nhưng trong ngữ cảnh của câu hoặc đoạn văn đó thì không phù hợp.

Ví dụ: kiểm cha ( từ đúng là: kiểm tra )

Trong hệ thống kiểm tra lỗi chính tả sẽ có những chức năng chính sau:

* Xử lý văn bản: văn bản đầu vào sẽ được phân chia, lọc nhiều.
* Kiểm tra, phát hiện lỗi: tiểm kiếm những âm tiết bị sai lỗi chính tả trong văn bản.
* Đánh dấu và gợi ý: đánh dấu nhưng từ lỗi và gợi ý âm tiết đúng để sửa lỗi.

Tổng quan chương trình được thể hiện ở hình 1.1

Xử lý văn bản

**Input:**

Tập văn bản tiếng Việt

Kiểm tra, phát hiện lỗi

**Output:**

Đánh dấu từ sai và đưa ra gợi ý

1.3 Mục tiêu và phạm vi của khóa luận

Khóa luận này sẽ dùng phương pháp kiểm lỗi chính tả cảm ngữ cảnh. Thông thường, các phương pháp kiểm tra lỗi truyền thống xem các từ độc lập với nhau và dùng từ diểm để kiểm tra, nhưng khí đó sẽ không phát hiện được những từ sai ngữ cảnh. Bằng phương pháp kiểm tra lỗi chính tả cảm ngữ cảnh này sẽ giúp phát hiện được những lỗi sai về cách dùng, nghĩa là từ đó không sai chính tả, nhưng lại không phù hợp trong hoàn cảnh của câu văn đó thì việc sử dụng nó là điều không hợp lý. Bằng phương pháp này, sẽ giúp chương trình đạt hiệu quả cao hơn trong việc kiểm tra lỗi.

Mục tiêu của khóa luận này là một chương trình kiểm tra lỗi chính tả hoàn chỉnh, được tích hợp vào ứng dụng soạn thảo văn bản Microsoft Word.

Để đạt được mục tiêu trên, ngoài hiệu quả, thì còn cần một một mô hình ngôn ngữ không quá cầu kỳ và tốn nhiều không gian nhơ. Do đó mô hình ngôn ngữ N-gram được chọn làm hướng tiếp cận của khóa luận này.

Do ngôn ngữ là một lĩnh vực rộng và để phù hợp chương trình. Nên khóa luận này chỉ giới hạn ở các văn bản hành chính, các bài báo.

1.4 Một số nghiên cứu liên quan

Bài toán kiểm lỗi chính tả là bài toán nhận được sự quan tâm của rất nhiều nhà khoa học khắp thế giới từ rất lâu rồi. Các ngôn ngữ phổ biến như tiếng Anh, tiếng Pháp, tiếng Nhật,…đều có rất nhiều phương pháp kiểm lỗi, nhưng ở Việt Nam vẫn là một bài toán chưa được giải một cách tốt nhất.

Đối với các ngôn ngữ nước ngoài như tiếng Anh, Pháp, Hoa hay Nhật có một số phương pháp như là winnow [8], từ ngữ cảnh[9], lai Bayes[10] cho tiếng Anh, mô hình Nagata cho tiếng Nhật[13], mô hình CInsunspell cho tiếng Hoa[12] …

Đối với tiếng Việt có một số cách như là dựa vào từ điển, sử dụng mô hình lưới từ, sử dụng mô hình N-gram kết hợp với phương pháp ước lượng Bayes để tính xác suất N-gram. Ngoài ra còn có nhưng nghiên cứu có liên quan như phương pháp phân tích cú pháp câu[5], phương pháp mạng lưới từ [7] hoặc sử dụng mô hình N-gram kết hợp thuật toán Viterbi[6].

1.5 Cấu trúc khóa luận

Khóa luận này được chia làm 6 chương:

* Chương 1: Giới thiệu: Giới thiệu tổng quan về bài toán kiễm tra lỗi chính tả tiếng Việt, mục tiêu và phạm vi của khóa luận. Ngoài ra, phần này còn trình bày tình hình, một số phương pháp kiểm lỗi chính tả ở trong nước cũng như ngoài nước.
* Chương 2: Cơ sở lý thuyết: Giới thiệu tổng quan về các đơn vị cũng như đặc điểm của tiếng Việt, chuẩn tiếng Việt và tập ngữ liệu (Corpus).
* Chương 3: Mô hình ngôn ngữ N-gram: Trình bày chi tiết về mô hình ngôn ngữ N-gram.
* Chương 4: Giới thiệu phương pháp: Trình bày chi tiết phương pháp xây dựng chương trình.
* Chương 5: Cài đặt, thực nghiệm và đánh giá: Trình bày phương pháp cài đặt, thực nghiệm chương trình và đánh giá kết quả thực nghiệm độ hiệu quả của chương trình
* Chương 6: Tổng kết: Tổng hợp kết quả đã đạt được của khóa luận, đề xuất phương hướng phát triển trong tương lai.

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

2.1 Đặc điểm của ngữ pháp tiếng Việt

2.1.1 Đặc điểm của tiếng việt

Tiếng việt thuộc hệ ngôn ngữ Nam á và thuộc ngôn ngữ đơn lập, tức là mỗi một âm tiết được phát âm thanh tách rời nhau và được thể hiện bằng một chữ viết. Đặc điểm này thể hiện ở các mặt ngữ âm, từ vụng, ngữ pháp.

Đặc điểm ngữ âm: Trong tiếng Việt có một đơn vị được gọi là tiếng. Về mặt ngữ âm, mỗi tiếng được coi là một âm tiết. Hệ thống âm vị của tiếng Việt phong phú và có tính cân đối.

Đặc điểm từ vựng: Mỗi tiếng là một yếu tố có nghĩa. Tiếng là đơn vị cơ sở của hệ thống các đơn vị có nghĩa của tiếng Việt. Từ tiếng kết hợp hai phương thức ghép và láy, ta có thể tạo ra các đơn vị từ vựng khác để định danh sự vật, hiện tượng.

2.1.2 Các đơn vị của tiếng việt

*a. Chữ cái*

Chữ cái dùng để đại diện cho âm trong ngôn ngữ nói. Theo quy định của bộ Giáo dục và đào tạo, bảng chữ cái có 29 chữ cái bao gồm : A, Ă, Â, B, C, D, Đ, E, Ê, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, Y.

Ngoài ra còn có 11 phụ âm ghép là: CH, GI, GH, KH, NH, NG, NGH, PH, QU, TH, TR.

*b. Âm tiết*

Âm tiết là một đơn vị cấu tạo nên một sự phối hợp trong tiếng nói**.** Âm gồm có các thành phần là: nguyên âm, phụ âm, bán âm, âm đệm.

Nguyên bao gồm có 12 nguyên âm đơn là A, Ă, Â, E, Ê, I, Y, O, Ô, Ơ, U, Ư và 3 loại nguyên âm đôi đó là IÊ, YÊ, IA, YA / ƯƠ, ƯA / UÔ, UA.

Phụ âm gồm có 23 phụ âm: B, C / K / Q, CH, D, Đ, G / GH, GI, H, KH, L, M, N, NH, NG / NGH, P, PH, R, S, T, TH, TR, V, X.

Bán Âm có hai loại và là bán âm cuối. Loạị đầu tiên gồm hai chữ cái là I và Y. Loại thứ hau gồm hai chữ cái là O và U.

Âm đệm gồm có hai chữ cái O và U.

*c. Dấu Thanh*

Tiếng Việt có 6 thanh : ngang, sắc, huyền, hỏi, ngã, nặng và được biểu diễn bằng 5 dấu sắc, huyền, hỏi, ngã, nặng. Thanh ngang không có dấu ghi.

Sau đây là một số quy tắc đánh dấu thanh (Linh, 2013):

* Với những âm tiết chỉ có một chữ nguyên âm, thì dấu thanh được đặt vào chữ nguyên âm đó. Ví dụ: á à, ì ạch, ọ ẹ, ủ rũ, ọp ẹp, ục ịch, hà, lán, giá, giục, quả, quỹ, quỵt... (Chú ý: gi và qu được coi là phụ âm).
* Với những âm tiết, mà trong âm tiết đó chỉ cần có một chữ nguyên âm mang dấu phụ (Ă, Â, Ê, Ô, Ơ, Ư) và không kể kết thúc bằng chữ gì, thì dấu thanh bao giờ cũng đặt ở chữ đó (riêng ƯƠ, dấu đặt ở Ơ). Ví dụ: thuyền, trường…
* Với những âm tiết có hai chữ nguyên âm và kết thúc bằng một chữ phụ âm hoặc tổ hợp chữ phụ âm, thì dấu thanh được đặt vào chữ nguyên âm cuối cùng.
* Với những trường hợp còn lại thì dấu thanh được đặt vào con chữ nguyên âm áp chót.

*d. Tiếng*

Tiếng trong tiếng Việt còn được gọi là âm tiết. Tiếng là đơn vị nhỏ nhất có thể có nghĩa. Một tiếng khi được người nói phát ra bao giờ cũng phải đi với một thanh điệu. Trong chữ viết, mỗi tiếng được ghi thành một chữ.

Cấu tạo của một tiêng bao gồm: phụ âm đầu, nguyên âm, phụ âm cuối, dấu thanh.

|  |  |
| --- | --- |
| **Phụ âm đầu** | b c d đ g h k l m n q r s t v x ch gh gi kh ng nh ph qu th tr ngh |
| **Nguyên âm** | a â ă e ê i o ô ơ u ư y ai ao au ay âu ây eo êu ia iu iê oa oi oe oă oo ôi ơi ua uy ui uâ uô uê uơ ưa ưi ươ ưu yê iêu oai oao oay oeo uôi uây uyê ươi ươu uya uyu uêu yêu |
| **Phụ âm cuối** | c p t m n ch ng nh |
| **Dấu thanh** | ngang, huyền, hỏi, ngã, sắc, nặng |

Bảng 2.1: Bảng các thành phần âm tiết

*e. Hình vị*

Hình vị là đơn vị nhỏ nhất có ý nghĩa, tương tự với tiếng. Hình vị cũng có cấu tạo là một âm tiết. Trên chữ viết hình vị được viết thành một chữ.

Trong tiếng Việt hình vị có thể đóng vai trò là một từ hoặc cũng có thể làm thành tố cấu tạo nên một từ, nhưng nó chỉ được phân xuất ra nhờ phân tích bản thân các từ.

Trong giáo trình Xử lý ngôn ngữ tự nhiên của TS. Đinh Điền[2], bên cạnh đặc điểm như hình vị của ngôn ngữ học đại cương, còn phải có “hình tố”.

*f. Câu*

Câu là đơn vị ở bậc cao nhất. Được cấu tạo bởi nhiều từ theo những quy tắc nhất định.

2.2 Chuẩn chính tả

Theo [1], chuẩn chính tả bao gồm các chuẩn viết các âm (phụ âm, nguyên âm) và các thanh, chuẩn viết tên riêng (viết hoa), chuẩn viết phiên âm từ và các thuật ngữ vay mượn (tiếng nước ngoài).

Hiện nay, chuẩn viết các âm và thanh tiếng Việt đã được xác định theo hệ thống ngữ âm của tiếng Việt. Chuẩn viết hoa chưa được thống nhất nhưng theo xu hướng viết hoa chữ cái đầu của âm tiết thuộc tên riêng (Ví dụ: Việt Nam, Hồ Chí Minh…). Đối với chuẩn phiên âm từ vay mượn (tiếng nước ngoài) thì phức tạp hơn. Hiện nay đang tồn tại 2 cách viết phiên âm, đó là phiên âm âm tiết và có gạch nối giữa các tiếng (Ví dụ: Lê-nin, Các-Mác…) và viết liền âm tiết, tôn trọng âm tiết và chữ theo hệ Latinh (Ví dụ: Paris, London…).

Chính tả là cách viết chữ được xem là chuẩn, tức viết đúng âm đầu, đúng vần, đúng thanh, đúng quy định về viết hoa, viết tắt, viết thuật ngữ. Ngoài ra, việc bỏ dấu thanh phải tuân theo 4 quy cách sau:

* Dấu chỉ ghi trên hoặc dưới nguyên âm, không ghi trên hoặc dưới phụ âm. (Ví dụ đúng: láng, bệnh, mình)
* Dấu chỉ ghi trên hoặc dưới nguyên âm chính, không ghi trên hoặc dưới bán âm cuối. (Ví dụ đúng: sáu, máy – Ví dụ sai: saú, maí)
* Đối với nguyên âm đôi :
* Khi nguyên âm đôi đứng ở cuối từ, ta ghi dấu trên hoặc dưới nguyên âm thức nhất. Ví dụ: mía, táo
* Khi nguyên âm đôi đứng ở giữa từ, ta ghi dấu trên hoặc dưới nguyên âm thứ hai: liền, muốn, hướng.

Chuẩn chính tả còn quy định tiếng phải được viết đúng thanh điệu. Hiện nay cư dân ở thành phố không phân biệt được hai thanh hỏi ngã. Tuy chỉ có hai dấu nhưng số lượng lỗi này không ít mà rất phổ biến – kể cả ở những người có trình độ văn hóa cao.

Ngoài ra, chuẩn chính tả còn yêu cầu âm tiết viết đúng âm đầu, âm chính và âm cuối. Hiện nay, do việc phát âm của từng vùng miền khiến cho các âm này bị nhầm lẫn với nhau như ch-tr, l-n, an/ang…

2.3 Nguyên nhân gây ra lỗi chính tả trong tiếng Việt

Ở đây chỉ xét về lỗi do phát âm, bởi vì lỗi do đánh máy nguyên nhân là do người soạn thảo văn bản.

2.3.1 Lỗi do âm đầu

Lỗi này thường xảy ra đối với khu vực miền Bắc và miền Nam. Khu vực miền Bắc thường sai ở những âm như L\N, còn miền Nam thường sau các âm như S\X, Ch\Tr, D\Gi,…Ngoài ra, trong tiếng Việt có một số âm đọc giống nhau nhưng lại cố nhiều cách ghi. Ví dụ như âm được đọc là/k/ có các cách ghi C, K.

2.3.2 Lỗi do âm chính

Có 2 nguyên nhân gây nên sự nhầm lẫn âm chính. Nguyên nhân đầu tiên là do quy định ghi âm đặc biệt của chữ Quốc ngữ. Nguyên nhân thứ hai là cách phát âm lẫn lộn, đặc biệt là vùng Nam Bộ, đối với các âm chính trong hầu hết các vần. (Ví dụ: đạp 🡪 độp).

2.3.3 Lỗi do âm cuối

Lỗi này thường xảy ra ở khu vực miền Nam. Khu vực này Nam rất dễ bị nhầm lẫn giữa các âm cuối n/ng/nh và t/c/ch. Mặt hai bán âm cuối là /j/-/u/ được ghi bằng 4 chữ cái i-y, u-o rất dễ bị nhầm lẫn.

2.3.4 Lỗi do thanh điệu

Lỗi này thường xảy ra đối với khu vực miền Trung và miền Nam. Ở hai khu vực này không thể phân biệt được giữa dấu hỏi và dấu ngã.

2.4 Các trường hợp lỗi chính tả trong văn bản tiếng Việt

Có hai loại lỗi chính tả chính tả. Loại lỗi thứ nhất là lỗi do sai âm tiết và hoàn toàn không có trong từ điển. Loại lỗi thứ hai là âm tiết hoàn toàn đúng và có trong từ điển, nhưng đặt trong ngữ cảnh của câu văn thì nó lại không phù hợp.

Loại lỗi thứ nhất rất dễ để tìm ra được lỗi và sửa lỗi. Đơn giản chỉ là cần sử dụng từ điển để tìm từ sai. Nhưng trong các văn bản tiếng Việt có một điều làm cho việc tìm lỗi này tương đối phức tạp thêm là do trong các văn bản đó có sử dụng từ mượn từ tiếng nước ngoài. Do vậy cần một chuẩn để xác định tiếng nước ngoài.

Loại lỗi thứ hai rất khó để tìm ra được lỗi này. Bởi vì máy tính thì khó mà hiểu được nghĩa của một văn bản. Do đó cần phải xét các tập nhầm lẫn âm tiết để tìm lỗi. Tập nhầm lẫn âm tiết là tập hợp các âm tiết có thể bị nhầm lẫn với âm tiết đang xét. Ví dụ: Xét từ “tìm kiếm”, tập nhầm lẫn có thể có các trường hợp là: tiềm kiếm, tìm kiếm, tìm kiến…

Có 2 trường hợp lỗi chính khi soạn thảo văn bản đó là do đánh máy và phát âm.

2.4.1 Lỗi do đánh máy sai

Đây là lỗi chính tả phổ biến, hầu hết lỗi ở mức độ âm tiết. Lỗi sai chính tả này gây ra các lỗi sai chính tả đơn và sai chính tả phức. Có bốn loại lỗi chính tả đơn:

* Chèn: “tín” 🡪 “tính” (chèn dư thêm chứ “h”).
* Xóa: “khoảng” 🡪 “khoản” (xóa chữ “g”).
* Thay thế: “tre” 🡪 “che” (thay thế chữ “tr” thành chữ “ch”)
* Hoán vị: “chính” 🡪 “chíhn” (hoán vị chữ “nh” thành “hn”).

Lỗi chính tả phức là kết hợp các lỗi trên.

2.4.2 Lỗi do phát âm

Lỗi do phát âm xảy ra nguyên nhân là do có thói quen “đọc thế nào thì viết như thế đó”. Nên lỗi này dễ xảy ra ở cả văn bản viết tay và đánh máy, cụ thể 2 loại lỗi[3] như sau:

* Khi chữ viết phân biệt âm tiết mà phát âm theo một phương ngữ nào đó lại không phân biệt. Cho nên với những phương ngữ khác nhau có những vấn đề chính tả khác nhau. Trong khi với người nói phương ngữ miền Bắc có các vấn đề chính tả “viết ch- hay tr-”, “viết -iêu hay -ươu”, v.v. thì với người nói phương ngữ miền Nam lại có các vấn đề “viết -n hay -ng”, “viết dấu hỏi hay dấu ngã”, v.v.
* Khi chữ viết phân biệt âm tiết mà phát âm tiếng Việt ngày nay không còn phân biệt, đó là trường hợp “viết d- hay gi-”, một vấn đề chính tả chung cho mọi miền trong cả nước.

Ví dụ: Âm tiết “nóng” 🡪 Tập nhầm lẫn: “lóng”

Âm tiết “tre” 🡪 Tập nhầm lẫn: “che”

Âm tiết “sữa” 🡪 Tập nhầm lẫn: “sửa”, “xữa”

Âm tiết “siêu” 🡪 Tập nhầm lẫn: “siu”, “sươu”

Sau đây chúng tôi sẽ liệt kê một số lỗi phát âm thường gặp:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CH- | TR- | | | |
| D- | GI- | | | |
| D- | GI- | D- | | |
| D- | GI- | D- | | |
| D- | GI- | D- | | |
| HW- | NG- | HW- | | NG- |
| L- | N- | | | |
| S- | X- | | | |
| -C | -T | | | |
| -N | -NG | | | |
| -AI | -AY | | | |
| -EM | -ÊM | | | |
| -ÊCH | -ÊT | | | |
| -IÊM | -IM | | | |
| -IÊU | -IU | | | |
| -IÊU | -ƯƠU | | | |
| -OAI | -OI | | | |
| -OM | -ÔM | | -ƠM | |
| Hỏi | ngã | | | |
| Ngã | Nặng | | | |

Bảng 2.2. Liệt kê các lỗi phát âm thường gặp

2.5 Giới thiệu Corpus

Corpus là một tài nguyên quan trọng trong các bài toán xử lý ngôn ngữ tự nhiên bằng thống kê. Tùy vào mỗi bài toán mà có những loại Corpus khác nhau. Chất lượng của Corpus là điều quan trọng cần quan tâm, bởi vì nó ảnh hưởng rất lớn tới độ chính xác của hệ thống.

2.5.1 Corpus là gì?

Corpus hay còn được gọi là kho ngữ liệu, là tập hợp văn bản, ngôn ngữ được số hóa. Nó có thể ở dạng văn bản hoặc âm thanh. Trong khóa luận này loại Corpus sẽ được sử dụng là Corpus văn bản.

2.5.2 Các bài toán cần xử lý trong quá trình tạo Corpus

1. *Xử lý văn bản*

Do dữ liệu thu thập được phụ vào nguồn của Corpus nên dữ liệu gặp được có thể ở các dạng khác nhau. Do đó cần có công cụ phù hợp để trích tách văn bản thô từ dữ liệu ban đầu. Nguồn Corpus để thu thập thường sẽ là từ Internet, mà hầu hết các file chứa văn bản là HTML hoặc XML, khi đó sẽ cần chương trình đọc loại file này và lấy văn bản bên trong đó.

Sau khi được lấy được văn bản, tiếp đó sẽ chuyển những văn bản này thành một bộ mã thống nhất.

1. *Cắt câu*

Các câu sẽ được cắt dựa vào các dấu hiệu như “.”, “?”, “!”. Nhưng đối với một số ký tự có nhiều chức năng khác nhau, đặc biệt có thể thấy là dấu chấm, do dữ liệu thu thập được bao gồm rất nhiều thứ như những từ viết tắt (như Mr. A), địa chỉ Internet, … Vì vậy cần phải xác định chính xác ký hiệu nào là kết thúc câu để đảm bảo chính xác dữ liệu thu thập được.

1. *Phân tích từ tố*

Đây là quá trích tách các câu trong văn bản thành các đơn vị được gọi là từ tố ( mỗi từ tố có thể là từ, số,…).

Ví dụ:

Cho câu đầu vào: “Nước biển có màu xanh”.

Kết quả sau khi tách từ tố: “Nước”, “biển”, “có”, “màu”, “xanh”.

1. *Phân đoạn từ*

Trong tiếng Việt, ranh giới giữa các từ không rõ ràng. Trong mỗi câu, có rất nhiều khả năng tách từ khác nhau do sự đa dạng trong việc kết hợp với nhau giữa các từ. Vì thế bài toàn càng trở nên rất khó khăn.

Ví dụ: Cho câu: “Khả năng lãnh đạo tốt”.

Câu này có các khả năng tách từ như sau:

* Khả / năng / lãnh / đạo / tôt.
* Khả năng / lãnh / đạo / tôt.
* Khả / năng / lãnh đạo / tôt.
* Khả /năng / lãnh đạo / tôt.

2.5.3 Các bước tạo Corpus cho chương trình

Corpus là phần quan trọng để chương trình có thể đạt hiệu quả cao trong việc kiểm lỗi nên việc xây dựng một Corpus tốt là điều hết sức cần thiết. Các bước tạo Corpus bao gồm:

* + Tải file text từ Internet: internet là nguồn tài nguyên phong phú, thu thập Corpus từ các trang uy tín và đa dạng giúp Corpus đạt chất lượng cao hơn.
  + Lấy text: bước tiếp theo sẽ bóc tách phần text ra khỏi các thẻ html.
  + Chuẩn hóa dữ liệu: ở bước này, Corpus sẽ được loại bỏ những đoạn text kém chất lượng, xử lý nhiễu,…

2.6 Một sô phương pháp kiểm lỗi chính tả

2.6.1 Kiểm tra lỗi chính tả đối với nước ngoài

*a. Phương pháp từ ngữ cảnh*

Ngữ cảnh của một từ, hay một đoạn văn là những từ, những đoạn xung quanh nó. Để quyết định chọn một từ với những từ nhập nhằng với nó thì những từ xung quanh nó là một đầu mối quan trọng. Ví dụ, ta xét hai lựa chọn giữa desert và dessert, nếu có các từ lân cận là sand, arid, sun,…ta sẽ chọn desert, còn nếu ta gặp những từ như delicious, chocolate,…ta sẽ chọn dessert. Khi còn trong giải đoạn học, hệ thống sẽ lưu những từ lần cận của mỗi từ sai khi những từ này bị hệ thống phát hiện. Áp dụng thực tê, khi phát hiện từ sai, hệ thống sẽ kiểm tra những từ lân cận từ đó và chọn từ thích hợp để sửa lỗi.

*b. Mô hình CInsunSpell cho tiếng Hoa*

Mô hình CInsunSpell [12] được xây dựng bới Li Juanhua và Wang XiaoLong và được áp dụng trong sửa lỗi chính tả tiếng Hoa. Mô hình là sự kết hợp giữa N-gram, phương pháp ước lượng xác suất Bayes và phân phối trọng số tự động. Trong mô hình này, N-gram được sử dụng trong bước kiểm lỗi. CInsunSpell sử dụng trigram, xét xem độ liên kết của từ trung tâm và 4 tiếng lân cận nó. Tiếng trung tâm được coi như sai chính tả nếu như độ liên kết của nó với 4 tiếng lân cận nhỏ hơn một ngưỡng nào đó.

Tuy nhiên mô hình này cũng có nhược điểm va đó cũng là nhược điểm chung của mô hình N-gram, đó là dữ liệu quá thưa thớt, không đầy đủ, dẫn đến các xác suất N-gram sẽ có giá trị rất thấp (gần bằng 0). Để giải quyết vấn đề trên, mô hình CInsunSpell đã sử dụng thêm bigram kèm theo phương pháp làm mịn để nâng cao hiệu suất phát hiện lỗi.

*c. Phương pháp Nagata cho tiếng Nhật*

Phương pháp Nagata được đề xuất bởi Misaaki Nagata. Ban đầu mô hình được áp dụng cho các hệ thống nhận dạng chữ viết. Sau đó, mô hình này được áp dụng để sửa lỗi chính tả. Phương pháp gồm 2 bước:

* Bước 1: Câu đầu vào được chia thành nhiều chuỗi tiếng, mỗi chuỗi tiếng tạo thành một từ có trong từ điển hoặc gần giống với một từ nào đó trong từ điển.
* Bước 2: Bằng cách sử dụng phương pháp thống kê, hệ thống chọn ra những câu tốt nhất được chọn để sửa lỗi.

Ví dụ: Con người Việt Nam rất tôt bụng.

dất

Việt Nam

Con

người

ngùi

ngườ

Com

ngừ

Con người

rất

tốt

tố

tót

tốt bụng

bụng

Hình 1. Ví dụ cho mô hình Nagata

Hệ thống sử dụng hai phương pháp để chia chuỗi đầu vào thành các chuỗi con một cách hợp lý đó là mô hình ngôn ngữ thống kê và thuật toán Forward-DB Backward A\* để tách từ. Trong đó, ngôn ngữ thống kê dùng để tính xác suất kết hợp của chuỗi con và chuỗi nhãn từ loại và thuật toán Forward-DB Backward A\* [16] được dùng để đánh giá các xác suất kết hợp và tìm ra cách kết hợp có xác suất lớn nhất làm lời giải bài toán tách từ.

2.6.2 Kiểm lỗi chính tả đối với tiếng Việt

Kiểm lỗi chính tả là một bài toán khó đối với ngôn ngữ đơn lập như tiếng Việt. Phần này sẽ giới thiệu một số phương pháp kiểm lỗi chính tả phổ biến trong tiếng Việt.

*a. Kiểm lỗi chính tả dựa trên từ điển*

Đây là phương pháp kiểm lỗi chính tả đơn giản nhất. hai ý tưởng chính của phương pháp này như sau (Thái, 2003):

* Có khả năng một âm tiết là lỗi nếu tồn tại âm tiết trong tập nhầm lẫn của nó kết hợp với các âm tiết xung quanh tạo thành từ ghép.
* Nếu câu không chia được thành dãy từ thì có khả năng âm tiết ở vị trí bị ngắt là lỗi.

Ví dụ 1:

“Công ty đạt *tổn* doanh số rất cao trong năm nay”

Dễ dàng thấy được từ “tổn” không thuộc về một từ nào cả, do đó có khả năng đó là âm tiết lỗi.

Ví dụ 2:

“Máy *vu* tính ngày cang phổ biến trên thế giới”

Xét tập nhầm lẫn của âm tiết “vu”, ta thấy có âm tiết “vi” kết hợp với các âm tiết “máy” và “tính” tạo thành từ “máy vi tính”, vậy có khả năng “vu” là sai.

*b. Kiểm lỗi chính tả bằng mô hình lưới từ*

Mô hình này hoạt động dựa vào việc tách từ mờ, sau đó đánh giá các cách tách từ khác nhau để tìm ra cách tách từ tốt nhất để làm câu sửa lỗi (Duy & Điền, 2004). Trong lưới từ là một cấu trúc dữ liệu đặc biệt, giống như là một đồ thị có hướng và không chu trình xuất phát từ âm tiết đầu tiên đến âm tiết cuối cùng của câu. Nhờ có thể áp dụng những thuật toán tìm đường đi trong cấu trúc này làm cho việc tách từ mờ nhanh chóng hơn.

Về mặt lý thuyết thì lưới từ là một mô hình rất chặt chẽ và nếu được huấn luyện tốt thì nó có thể mang lại hiệu quả rất cao. Mặt khác, mô hình này cũng có nhược điểm đó là sự bùng nổ tổ hợp. Tiếng Việt khá đa dạng và phong phú nên việc tách từ cũng không hề đơn giản. Điều này làm cho việc tách từ tốn thời gian dẫn dến hệ thống của chương trình cũng trờ nên chậm chạp hơn.

Trong mô hình này, tác giả còn sử dụng N-gram để ước lượng những giá trị cần thiết. Nhưng trờ ngại do dữ liệu huấn luyện chưa đủ tốt nên việc ước lượng chưa cho được độ chính xác.

*c. Kiểm lỗi chính tả bằng MS Word*

Vào đầu năm 2005, hang Microsoft đã phát hành MS Word phiên bản tiếng Việt. Trong phiên bản này, Microsoft đã tích hợp tính năng kiểm tra lỗi chính tả cho tiếng Việt. Tuy nhiên, tính năng này chỉ có thể kiểm tra lỗi ở mức tiếng, một mức độ kiểm tra rất thấp.

CHƯƠNG 3 – MÔ HÌNH NGÔN NGỮ N-GRAM

3.1 Mô hình ngôn ngữ

Mô hình ngôn ngữ là một trong những phương pháp xử lý ngôn ngữ tự nhiên. Mô hình ngôn ngữ là một phần bố xác xuất trên các tập văn bản. Nói một cách đơn giản, mô hình ngôn ngữ có thể cho biết xác suất một câu (hoặc cụm từ) thuộc một ngôn ngữ là bao nhiêu.

Ví dụ: Khi ta áp dụng mô vào ngôn ngữ vào Tiếng Việt.

P[“hôm nay thật là vui”] = 0,001.

P[“vui là thật nay hôm”] = 0.

Mô hình ngôn ngữ được áp dụng trong rất nhiều lĩnh vực khác nhau của xử lý ngôn ngữ tự nhiên như: kiểm lỗi chính tả, phân đoạn từ…Vì vậy, nghiên cứu mô hình ngôn ngữ là tiền đề để nghiên cứu các lĩnh vực khác của xử lý ngôn ngữ tự nhiên.

Có rất nhiều hướng tiếp cận mô hình ngôn ngữ, nhưng chủ yếu là được xây dựng theo mô hình N-gram.

3.2 Mô hình ngôn ngữ N-gram

Mô hình N-gram là mô hình ngôn ngữ rất phổ biến và được sử dụng rất nhiều trong xử lý ngôn ngữ tự nhiên. Áp dụng tư tưởng “Dựa vào những dữ liệu cho trước, đã biết để đoán cái tiếp theo, chưa biết”. N-grams ước lượng xác suất xuất hiện của một từ khi đã biết N-1 từ ngay phía trước nó [11].

Một N-gram là một bộ các yếu tố (từ, tiếng, nhãn từ loại,…) liên tiếp trong ngữ liệu. Số yếu tố trong một N-gram được gọi là bậc của N-gram. Thông thường, người ta tính bậc của N-gram từ bậc 1 đến bậc 4. N-gram bậc 1 được gọi là unigram, bậc 2 được gọi là bigram, bậc 3 được gọi là trigram, bậc 4 được gọi là quadgram.

Một mô hình ngôn ngữ có nhiệm vụ là phải cho ta biết được xác suất của một câu là bao nhiêu.

Ví dụ, ta có một câu W1W2…Wn.

Để tính xác suất câu trên, ta sử dụng công thức Bayes:

P(A|B) = P(B|A) \* P(A).

Với công thức này ta sẽ được:

P(W1W2…Wn) = P(W1) \* P(W2|W1) \* P(W3| W1W2) \* … \* P(Wn|W1W2…Wn-1)

Nếu ta áp dụng như vậy, ta sẽ cần phải có một dung lượng bộ nhớ vô cùng lớn để có thể lưu trữ hết các xác suất của tất cả các chuỗi có độ dài bé hơn n. Điều là bất khả thi khi trong các văn bản ngôn ngữ tự nhiên, độ dài n có thể tiến tới vô cùng. Do đó, ta cần một cách khác để có thể tính xác suất của văn bản với lượng bộ nhớ chấp nhận được, ta mô hình Markov bậc m:

P(Wn| W1W2…Wn-1) = P(W­n|Wn-m Wn-m+1…Wn-1)

Khi dùng mô hình này, xác suât của một từ Wn thay vì phụ thuộc vào toàn bộ những từ nằm trước nó thì chỉ phụ thuộc vào một khoảng m từ đứng truốc nó. Công thức ban đầu sẽ được viết lại như sau:

P(W1W2…Wn) = P(W1) \* P(W2|W1) \* P(W3| W1W2) \* … \* P(Wn-1|Wn-m-1Wn-m…Wn-2)

\* P(Wn|Wn-mWn-m+1…Wn-1)

Mô hình N-gram là mô hình được xây dựng dựa trên công thức này. N-grams ước lượng xác suất xuất hiện của một từ khi đã biết N-1 ngay phía trước nó.

3.2.1 Tính xác suất bằng mô hình N-gram

Ta có công thức tổng quát để tính xác suất bằng N-gram như sau:

P(Wn| Wn-m+1 …Wn-1) =

Trong đó:

C(Wn-m+1 …Wn-1 Wn) là tần số xuất hiện của một câu Wn-m+1 …Wn-1 Wn.

C(Wn-m+1 …Wn-1) là tần số xuất hiện của một câu Wn-m+1 …Wn-1 Wn.

P(Wn| Wn-m+1 …Wn-1) là xác suất của một từ Wn với cụm từ nằm trước nó là Wn-m+1 …Wn-1

Một cách tổng quát hơn, ta có P(w|h) với w là từ ta cần tính xác xuất, h là dãy những từ nằm trước w.

Ta có:

Unigram: h rỗng.

Bigram: h là 1 từ nằm ngay trước w

Trigram: h là 2 từ nằm ngay trước w

N-gram: h là n-1 từ nằm ngay trước w

Ví dụ: ta có tập ngữ liệu mẫu nhỏ sau:

<s> Tôi đi học hằng ngày </s>

<s> Hằng ngày tôi đều phải đi học </s>

<s> Tôi học giỏi tiếng anh </s>

Ta sẽ tính bằng Bigram một số từ:

P(đi|Tôi) = 1/2 P(học|đi) = 2/3 P(học|Tôi) = 1/3

3.2.3 Các phương pháp làm mịn

Khi xác suất bằng không do ảnh hưởng của tập ngữ liệu. Ta cần dùng các phương pháp làm mịn để giải quyết vấn đề này.

*a. Phương pháp Add-one*

Theo phương pháp này, ta sẽ cộng cả tử và mẫu số với một giá trị thích hợp để đảm báo kết quả sẽ luôn khác 0. Ta sẽ cộng tử số thêm 1 và mẫu số với một giá trị V, với V là tổng số từ có trong từ điển.

Công thức sẽ được tính lại như sau:

P(Wn| Wn-m+1 …Wn-1) =

*b. Phương pháp cộng hệ số*

Trên thực tế, phương pháp cộng 1 cho kết quả không hợp lý bởi dưới một góc độ nào đó, giá trị 1 đôi khi là quá lớn và sai số của xác suất trở nên quá nhiều. Vì vậy, thay bởi cộng thêm 1 cho tử số, phương pháp cộng hệ số sẽ tăng một giá trị δ (0 < δ < 1). T

Theo đó, xác suất được tính lại như sau:

Image

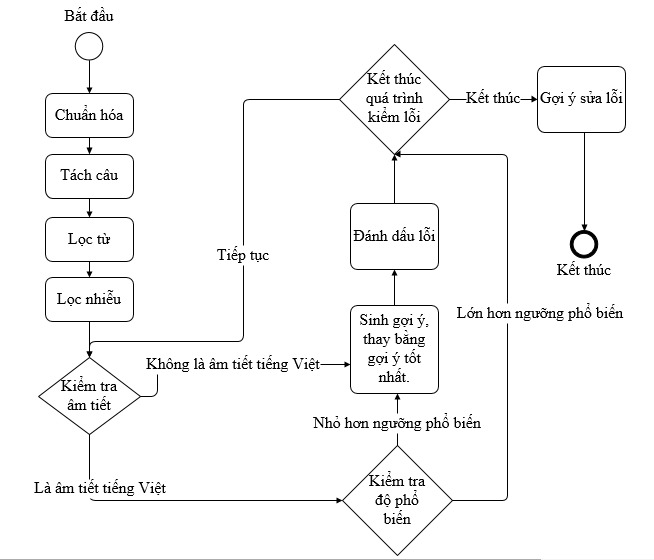
*c. Phương pháp GoodTuring*

Theo phương pháp này, với mỗi N-gram xuất hiện r lần, ra có thể xem như nó xuất hiện r\* lần, với r\* được tính như sau:

Image

CHƯƠNG 4 – GIỚI THIỆU PHƯƠNG PHÁP

4.1 Mô hình tổng quát



* Phần xử lý văn bản:
  + Chuẩn hóa lại văn bản đầu vào.
  + Tách văn bản thành từng câu dựa vào nhưng ký tự kết thúc.
  + Tách những câu ở bước trên thành từng từ riêng biệt với nhau.
  + Lọc những cụm từ tiếng Anh, tên riêng, các từ nước ngoài, viết tắt, số, các ký hiệu đặc biết, email, …
* Phần kiểm tra lỗi: tìm những từ có khả năng sai và đánh dấu để người dùng nhận biết trên Microsoft Word
* Phần sửa lỗi: đưa ra những từ gợi ý cho người dùng chọn để sửa lỗi.

4.1.1 Phần xử lý văn bản

*a. Chuẩn hóa văn bản*

Văn bản đầu vào sẽ bao gồm nhiều đối tượng khác nhau nhưng không phải là text như mục lục, link,... Nên đầu tiên cần phải loại bỏ những đối tượng này.

*b. Tách câu*

Sau khi chuẩn hóa, tiếp đến là bước tách văn bản thành từng câu. Bước này dùng làm tiền đề cho bước tách từ và dùng để xử lý riêng khi kiểm tra lỗi chính tả cho từng từ.

Tách thành từng câu dựa vào các ký tự kết thúc, việc này giúp giảm không gian xử lý của chương trình và không ảnh hưởng đến hiệu quả cũng như nâng hiệu năng hơn.

*c. Tách từ*

Những câu được tách ở bước trên sẽ tiếp tục tách thành các từ riêng biệt.

*d. Lọc nhiễu*

Bước cuối cùng trong phần tiền xử là lọc nhiễu. Trong các văn bản hiện nay thường xử dụng khá nhiều từ tiếng nước ngoài, đặc biệt là tiếng Anh, hoặc những từ mượn. Hệ thống sẽ phải lọc và xử lý. Ngoài những vấn đề về ngôn ngữ thì việc lọc còn phải xét những vấn đề khác như lọc ra những cụm số, tên riêng, địa chỉ email, địa chỉ website, các từ viết tắt,…Bước này là bước quan trọng để giúp tính toán N-gram được chính xác hơn.

4.1.2 Phần kiểm tra lỗi

Tiếp theo sẽ tiến hành tìm lỗi và đánh dấu những lỗi đó lên Word. Ý tưởng để kiểm tra lỗi là khi duyệt một từ, dùng từ điển tiếng Việt để xem từ đó có thuộc từ điển không, nếu không thì đánh dấu từ đó vào trường hợp sai hoàn toàn và đánh dấu trên Word. Trường hợp từ đó đúng thì tiếp đó sẽ sử dụng mô hình N-gram để đo độ liên kết của từ đó với các từ lân cận, nếu N-gram thấp thì từ đó sẽ được xem là lỗi chính tả.

4.1.3 Phần sửa lỗi

*a. Tạo tập từ gợi ý*

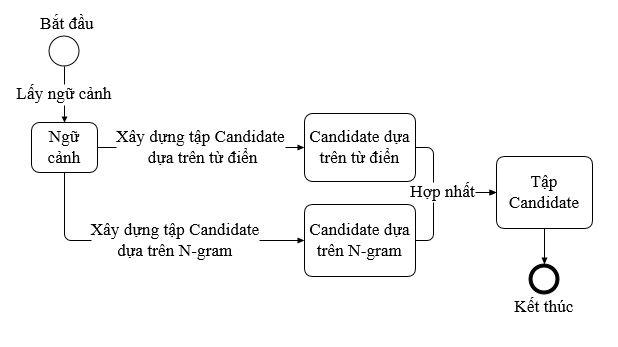
Việc xây dựng tập từ gợi ý cho mỗi âm tiết dựa trên sự liên kết giữa âm tiết đó với từ điển Tiếng Việt và từ điển N-gram. Với mỗi loại lỗi, hệ thống cho thấy những từ gợi ý tương ứng.

Quá trình xử lý tập hợp từ gợi ý xây dựng có ba bước, bao gồm:

• Bước 1: Lấy ngữ cảnh hiện tại xung quanh âm tiết: Gồm âm tiết, hai âm tiết liền trước và hai âm tiết liền sau.

• Bước 2: Xây dựng tập hợp từ gợi ý cho từng phần, bao gồm: dựa trên từ điển Tiếng Việt và từ điển N-gram.

• Bước 3: Hợp nhất hai tập hợp từ gợi ý để tạo ra một tập hợp gồm những từ gợi ý là duy nhất.



*b. Tính điểm và lựa chọn từ gợi ý*

Sau khi có tập gợi ý, những từ này sẽ được tính điểm để xác định đó có phải là từ gợi ý hợp lý hay không. Điểm này sẽ được so sánh với một chỉ số được gọi là “ngưỡng sửa lỗi”. Ngưỡng này được xác định dựa vào việc thống kệ nhiều trường hợp khác nhau dựa vào Corpus. Nếu điểm của từ nào lớn hơn hoặc bằng chỉ số này sẽ được xác định là hợp lý để gợi ý.

Điểm của những từ được chọn để gợi ý được dựa trên ba tiêu chí tính bằng công thức sau:

Trong đó:

* + là những tham số, có tổng được ước lượng dựa vào việc huấn luyện N-gram.
  + S (Similarity) có giá trị từ 0 đến 1 là điểm của từ gợi ý dựa vào độ tương tự đối với âm tiết hiện tại, điểm càng cao từ gợi ý càng giống với âm tiết hiện tại.
  + L (Model Language) có giá trị từ 0 đến 1 là điểm của từ gợi ý được tính dựa vào N-gram.
  + D (Dictionary) có giá trị từ 0 đến 1 là điểm của từ gợi ý được tính dựa vào từ điển từ ghép.

S thể hiện độ tương tự giữa âm tiết hiện tại và ứng cử viên. Trên thực tế, nhầm lẫn do lỗi đánh máy và do lỗi vùng miền thường xuyên xảy ra. Trong đó, lỗi vùng miền còn do nhầm lẫn trong mỗi thành phần của âm tiết. Vậy nên khi tính độ tương tự, phải cân nhắc tất cả những yếu tố kể trên. Qua học hỏi và nghiên cứu, chúng tôi tách phần dấu thanh của âm tiết và candidate để xét trường hợp nhầm lẫn về dấu thanh, sau đó sử dụng công thức bên dưới để tính độ tương tự giữa hai từ:

.

Với:

* S (Syllable): Âm tiết
* C (Candidate): Từ gợi ý
* WMS (WithoutMarkSyllable): Âm tiết sau khi loại bỏ dấu thanh
* WMC (WithoutMarkCandidate): Từ gợi ý sau khi loại bỏ dấu thanh
* MS (MarkSyllable): Dấu thanh của âm tiết
* MC (MarkCandidate): Dấu thanh của từ gợi ý
* DWM (DistanceWithoutMark): Phương thức tính khoảng cách giữa WMS và WMC được tính như sau:
* DM (DistanceMark): Phương thức tính khoảng cách giữa WS và WC

L (Model Language) thể hiện khả năng xuất hiện của từ gợi ý so với ngữ cảnh trong câu, hay hiểu rộng hơn là độ phổ biến của ngữ cảnh của từ gợi ý trong tập huấn luyện của N-gram. Dựa trên mô hình ngôn ngữ N-gram, hệ thống sẽ tính L dựa trên tần số của bigram, nếu từ gợi ý là hợp với ngữ cảnh của câu thì L sẽ rất cao, L được tính theo công thức bên dưới:

Với:

* V là kích thước của từ điển N-gram
* x, y: là tham số với tổng x + y = 1

Như đã trình bày ở trên, do việc Corpus không đủ lớn sẽ khiến cho một số xác suất quá nhỏ hoặc thậm chí bằng 0. Nên cần phải kết hợp sử dụng phương pháp làm mịn, chương trình sử dụng phương pháp Add-one để làm mịn. Phương pháp này áp dụng nhanh, đơn giản mà đạt hiệu quả khả quan.

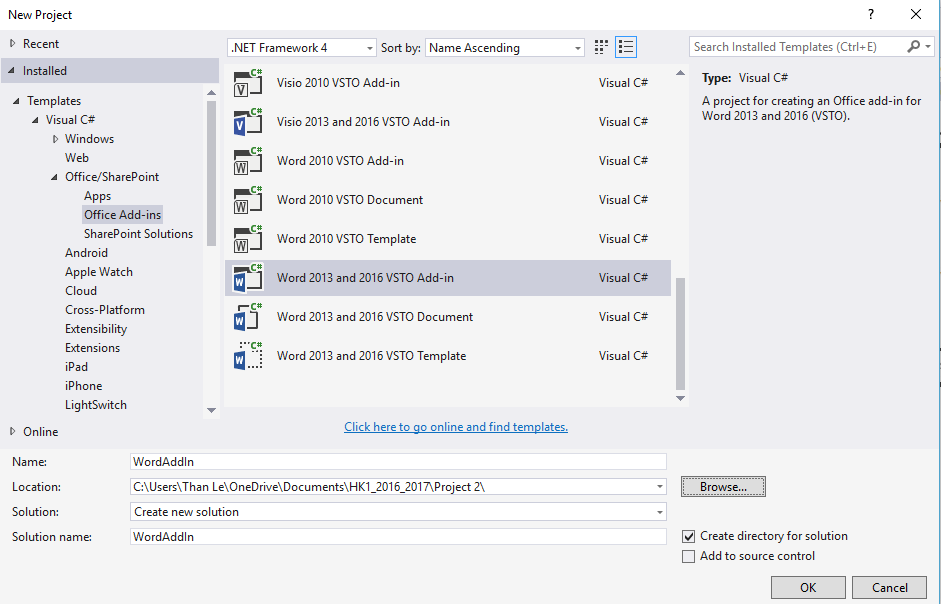
Trong thực tế, một số âm tiết dùng đúng ngữ cảnh nhưng có tần số quá thấp trong từ điển N-gram dẫn đến việc điểm L thấp, dẫn đến việc âm tiết đó có thể bị coi là sai chính tả. Trong trường hợp này nếu âm tiết đó cùng với ngữ cảnh xung quanh âm tiết là từ ghép thì tổng điểm Score sẽ được cải thiện dựa vào điểm D. D là điểm mà đánh giá từ gợi ý và ngữ cảnh xung quanh có thuộc từ điển từ ghép hay không. D được tính như sau:

4.2 Xây dựng Add-in vào Word

Add-in trong kỹ thuật máy tính là bộ phần mềm do hãng thứ ba tạo ra, nhằm thêm những tính năng cụ thể cho một phần mềm lớn hơn, cụ thể ở đây là MS Word.

4.2.1 Tạo Word VSTO Add-in

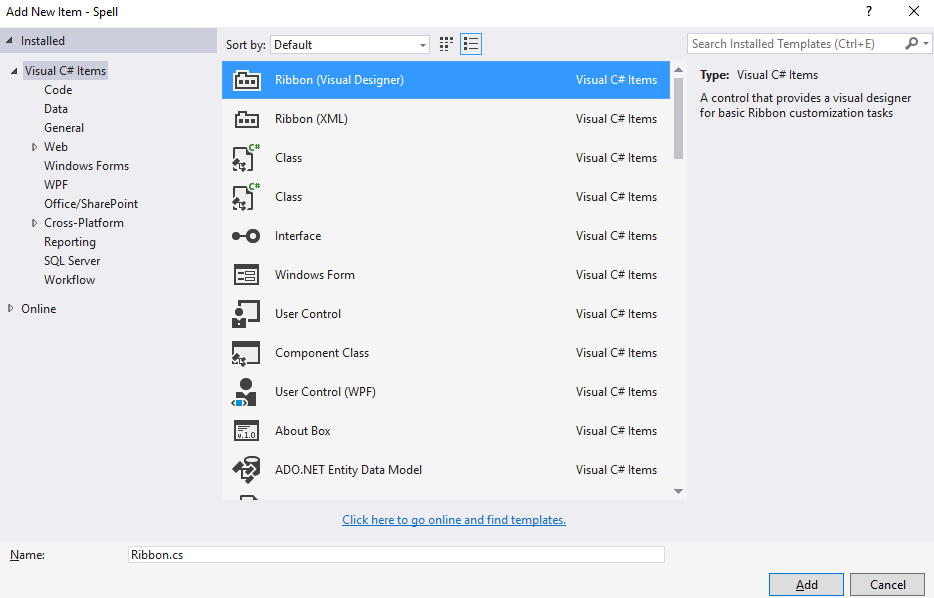
Trong phạm vi đồ án, chúng tôi dùng Microsoft Visual Studio 2015, và kể từ .Net Framework 4 trở lên, chúng ta đã có thể tạo được một Word Add-in bằng đường dẫn: New project → Installed → Templates → Visual C# → Office/ SharePoint → Office Add-ins và chọn Word 2013 and Word 2016 VSTO Add-in. Bằng cách này, chúng ta đã có thể tạo một Add-in trên Word 2013 và Word 2016.



Hình 4. Tạo Word Add-in

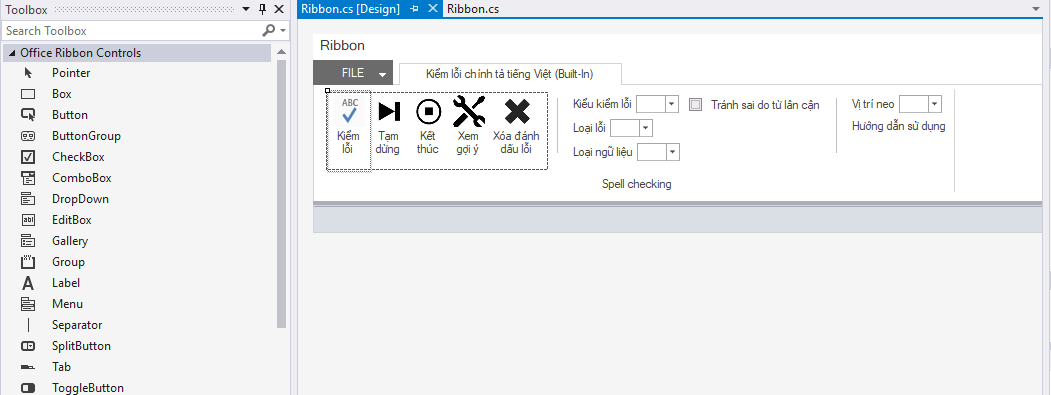
4.2.2 Tạo Ribbon (Visual Designer)

Sau khi tạo mới một Project, ta click chuột phải vào project chọn Add New Item. Tại đây, chọn Ribbon (Visual Desinger) theo đường dẫn Installed → Visual C# Items.



Hình 5. Tạo Ribbon (Visual Designer)

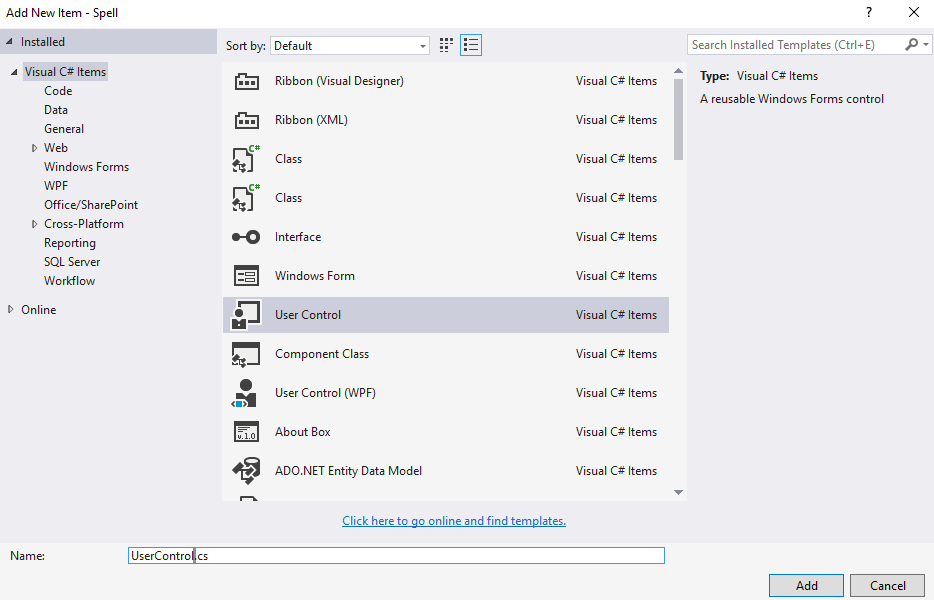
VSTO được tích hợp trên Visual Studio nên có thể sử dụng hầu như tất cả những tính năng sẵn có của bộ IDE mạnh mẽ này. Riêng với Ribbon trong VSTO, Visual Studio hỗ trợ những controls riêng trong Toolbox, được gọi là Office Ribbon Controls, tại đây, lập trình viên có thể kéo thả, và tùy chỉnh tương tự như làm việc trên Windows Form C#.



Hình 6. Thiết kế Ribbon

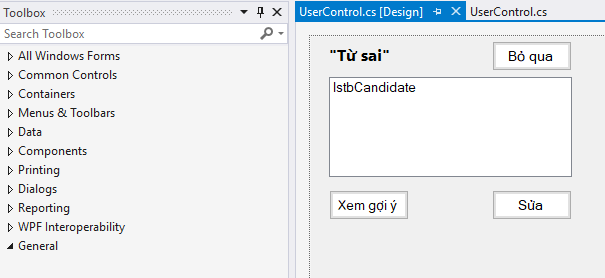
4.2.3 Tạo User Control

Sau khi có được Ribbon, ta lại chuột phải vào project và chọn Add New Item. Cũng trong đường dẫn Installed → Visual C# Items, ta chọn User Control, ta chọn User Control không có WPF.



Hình 7. Tạo User Control

Việc kéo thả để tạo giao diện trong User Control thực chất giống như khi làm việc với Windows Form C#, bởi chúng ta sử dụng Windows Form trong Toolbox.



Hình 8. Thiết kế User Control

4.2.4 Làm việc với namespace Microsoft.Office.Interop.Word

Để có thể tạo một add-in mà những đối tượng trong đó có thể tương tác được với văn bản trong MS Word, ta sử dụng namespace *Microsoft.Office.Interop.Word*. Để cho tiện trong coding, chúng tôi gán *Word = Microsoft.Office.Interop.Word* dùng từ khóa using

using Word = Microsoft.Office.Interop.Word;

*a. Microsoft.Office.Interop.Word.Range (Word.Range)*

Để có thể “get” và “set” được một từ, cụm từ, hay đoạn văn nào đó trong Word Document, ta cần biết từ, cụm từ hay đoạn văn đó nằm trong phạm vi nào, bắt đầu và kết thúc ở ký tự thứ bao nhiêu. Để làm được điều đó, ta cần sử dụng một đối tượng tên là Range trong namespace *Microsoft.Office.Interop.Word.*

Trong đối tượng Range, có rất nhiều properties cũng như methods. Trong phạm vi đồ án này, chúng tôi sử dụng chủ yếu 3 properties là Word.Range.Font và Word.Range.HighlightColorIndex và Word.Range.Text.

* Word.Range.Font: với property này, ta có thể quyết định “vùng” được chọn, format theo font gì, màu gì, có đậm, nghiêng, gạch chân hay không thông qua: Word.Range.Font.Color, Word.Range.Font.Bold, …
* Word.Range.HighlightColorIndex: HighlightColorIndex cho phép ta chọn màu highlight cho “vùng” được chọn.
* Word.Range.Text: property này cho phép ta “get” và “set” văn bản trong “vùng” được chọn, kiểu trả về là string.

*b. Microsoft.Office.Interop.Word.Words (Word.Words)*

Với Word.Words, ta thu về được một Words Collection. Mỗi phần tử trong Words Collection là một đối tượng Range, được đại diện cho một từ, không phải là đối tượng Word. Words(index) index là số chỉ mục, trả về một đối tượng đại diện cho một từ. Số chỉ mục thể hiện vị trí của từ trong Words Collection.

Thuộc tính Count trả về số lượng phần tử có trong Words Collection. Phương thức Add không có sẵn trong Words Collection. Thay vào đó, sử dụng phương pháp InsertAfter hoặc phương pháp InsertBefore để thêm văn bản vào một đối tượng Range.

*c. Microsoft.Office.Interop.Word.Sentences (Word.Sentences)*

Tương tự như Word.Words, với Word.Sentences, ta thu về được một Sentences Collection. Mỗi phần tử trong Sentences Collection là một đối tượng Range, được đại diện cho một câu, không phải là đối tượng Sentence. Sentences(index) index là số chỉ mục, trả về một đối tượng đại diện cho một câu. Số chỉ mục thể hiện vị trí của câu trong Sentences Collection.

Thuộc tính Count trả về số lượng phần tử có trong Words Collection. Phương thức Add không có sẵn trong Sentences Collection. Thay vào đó, sử dụng phương pháp InsertAfter hoặc phương pháp InsertBefore để thêm văn bản vào một đối tượng Range.

*d. Lấy văn bản từ MS Word*

Đoạn code sau giúp thu về Words Collection hay Sentences Collection hay Range:

Word.Words words = Globals.ThisAddIn.Application.Selection.Words;

Word.Words words = Globals.ThisAddIn.Application.ActiveDocument.Words;

Word.Sentences sentences = Globals.ThisAddIn.Application.Selection.Sentences;

Word.Sentences sentences = Globals.ThisAddIn.Application.ActiveDocument.Sentences;

Trong đó, ActiveDocument trả về một tập hợp toàn bộ văn bản có trong document, còn Selection trả về tập hợp văn bản trong khu vực người dùng chọn.

CHƯƠNG 5 – CÀI ĐẶT, THỰC NGHIỆM VÀ ĐÁNH GIÁ

5.1 Cài đặt

Dựa vào mô hình lý thuyết nêu trên, chương trình sẽ được xây dựng bằng ngôn ngữ C#. Ngôn ngữ này có thể đảm bảo được tốc độ và độ bảo mật của chương trình, ngoài ra còn giúp tích hợp chương trình vào Microsoft Word. Chương trình đã được thực nghiệm thành công trên môi trường Window 10.

Các bước cài đặt hệ thống:

* + Tạo Corpus.
  + Đếm tần suất N-gram và lưu trữ.
  + Viết chương trình
  + Tích hợp vào Word.

5.1.1 Xây dựng tập huấn luyên (Corpus)

Như đã trình bày các bước tạo Corpus ở chương 2. Do trong quá trình chuẩn hóa Corpus và xử lý nhiễu còn nhiều khó khăn, chúng tôi đã loại bỏ những uni-gram và bi-gram có tần số thấp với khả năng chúng là các trường hợp lỗi, và sẽ ít có khả năng xuất hiện trong các văn bản, nên số lương uni-gram và bi-gram đã giảm đáng kể. Đây là kết quả Corpus thô đã được thu thập

|  |  |
| --- | --- |
| Độ lớn của Corpus | 1.38 GB |
| Số lượng uni-gram khác nhau | 4815 |
| Số lượng bi-gram khác nhau | 387405 |

5.1.2 Xây dựng tập dữ liệu

Tương tự như quá trình xây dựng tập huấn luyện, dữ liệu được thu thập từ Internet bao gồm tin tức, giấy tờ, truyện ngắn, tiểu thuyết trên các trang, ... như dantri.com.vn, vnexpress.net. Những tài liệu này đã được viết theo một tiêu chuẩn. Các tài liệu này được chúng tôi kiểm tra cẩn thận để đảm bảo rằng không có lỗi chính tả trong các tài liệu này. Sau đó, chúng tôi tạo ra lỗi chính tả trên tập dữ liệu này bao gồm: sai chính tả và sai ngữ cảnh. Những lỗi này được chúng tôi phân phối trên tất cả các loại lỗi phổ biến. Ví dụ:

Câu lỗi: “Theo chũ trương được phê duyệt, bên cạh xây dựng khu siêu thị, nhà ở thơng mại và kinh danh nhà ở, công ty còn đầu tư khu chợ mớ để di dời trên 400 tiểu thương đang mua bán ở khu chợ củ đã xuống cấp.”

Câu đúng: “Theo chủ[chũ] trương được phê duyệt, bên cạnh[cạh] xây dựng khu siêu thị, nhà ở thương[thơng] mại và kinh doanh[danh] nhà ở, công ty còn đầu tư khu chợ mới[mớ] để di dời trên 400 tiểu thương đang mua bán ở khu chợ cũ[củ] đã xuống cấp.”

5.1.3 Viết chương trình

Bước cuối cùng của phần cài đặt là viết chương trình, trình tự và phương pháp thực hiện theo mô hình đã được nêu ở chương 4. Chương trình yêu cầu đảm bảo tốc độ xử lý, độ chính xác cao, đồng thời giao diện cần được xây dựng thân thiện, dễ sử dụng và có thể nâng cấp phát triển thêm.

5.2 Thực nghiệm

Để đánh giá hệ thống kiểm tra chính tả, chúng tôi sử dụng 4 tiêu chí sau:

Độ phát hiện chính xác (Detection precision): cho thấy hệ thống phát hiện hữu ích như thế nào, được xem là một thước đo chính xác.

Độ hồi tưởng (Detection recall): cho thấy mức độ hoàn thành việc phát hiện của hệ thống, được xem là một thước đo hoàn chỉnh.

Độ điều chỉnh chính xác (Correction precision): cho thấy một thước đo về chất lượng của sự sửa lỗi.

Độ đo F-score: là một thước đo của việc thử nghiệm tính chính xác trong các lỗi phát hiện.

5.2.1 Môi trường và dữ liệu thực nghiệm

*a. Môi trường thực nghiệm*

|  |  |
| --- | --- |
| Bộ xử lý | Intel® 2 Core™ i7-2620M CPU @ 2.70GHz (4 CPUs), ~ 2.7GHz |
| Bộ nhớ | 4.0 GB RAM |
| Hệ điều hành | Windows 10 Educatoin 64 bit |
| Công cụ lập trình | Java version "1.8.0\_65" |

*b. Dữ liệu thực nghiệm*

5.2.2 Kết quả thực nghiệm

* Kích thước dữ liệu thực nghiệm: 483kb
* Thời gian thực hiện: 20s
* Tổng số lỗi được sinh: 1570
* Tổng số lỗi phát hiện: 1539
* Tổng số lỗi phát hiện sai: 141
* Tổng số lỗi phát hiện đúng: 1398
* Tổng số lỗi sửa đúng: 1391
* Đô phát hiện chính xác: 90.8%
* Độ hồi tưởng: 89%
* Độ hiệu chỉnh chính xác:
* Độ đo F: 89.9%

5.3 Đánh giá

CHƯƠNG 6 – TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

**Tổng kết**

Dựa trên những tìm hiểu về N-gram trong kiểm lỗi chính tả tiếng việt trong đồ án một và việc xây dựng chương trình thành một Word Add-in trong đồ án 2 mà chúng tôi thực hiện, chúng tôi đã nâng cấp nhiều tính năng và xây dựng một chương trình có tính khả dụng trong cuộc sống. Người dùng có thể thực hiện việc soạn thảo văn bản song song với việc kiểm tra và đánh dấu những từ có khả năng là sai. Hoặc với các sử dụng thứ hai, người dùng có thể soạn thảo văn bản của mình, sau đó thực hiện việc kiểm tra lỗi và gợi ý sửa lỗi với những từ sai này. Việc hỗ trợ hai phương thức sử dụng sẽ giúp người dùng sử dụng một cách thoải mái và phù hợp với bản thân hơn.

Tuy nhiên, chương trình vẫn còn một số nhược điểm, đó là tỷ lệ lỗi tích cực vẫn còn khá cao (khoảng 9%). Mặc dù khi thu thập Corpus, chúng tôi đã cố gắng tạo ra một tập Corpus thật đầy đủ, nhưng đối với sự đa dạng của ngôn ngữ, điều đó thực sự là rất khó khăn. Do đó, với những thể loại xuất hiện ít trong Corpus, hiệu suất kiểm lỗi của những thể loại này là khá khiêm tốn và chưa thỏa mãn được yêu cầu của người dùng. Thêm vào đó, với sự đa dạng của trong cách tạo văn bản, việc kiểm lỗi cho chữ cái viết hoa trong chương trình vẫn chưa được xử lý thực sự tốt. Ngoài ra, giao diện chương trình chưa thực sự thân thiện và dễ sử dụng.

**Hướng phát triển**

Trong tương lai, chúng tôi dự định sẽ cải tiến phương pháp, tăng hiệu năng cũng như hiệu quả kiểm tra và sửa lỗi. Về mặt giao diện, chúng tôi sẽ tăng thêm một số chức năng phụ trợ cho người soạn thảo, giúp người dùng soạn thảo nhanh chóng, tiện lợi hơn.

TÀI LIỆU THAM KHẢO

|  |  |  |  |
| --- | --- | --- | --- |
| [1] | | Đặng Thị Lanh, Bùi Minh Toán, Lê Hữu Tỉnh, *Tiếng Việt,* Giáo trình chính thức đào tạo giáo viên tiểu học hệ CĐSP và SP 12+2. tập 1. NXB Giáo Dục, 1997. | |
| [2] | | Đinh Điền, *Giáo trình xử lý ngôn ngữ tự nhiên,* ĐH KHTN TPHCM, 2004. | |
| [3] | | Hoàng Phê. 1999. Chính tả tiếng Việt. Nhà xuất bản Đà Nẵng. | |
| [4] | | Nguyễn Đức Hải và Nguyễn Phạm Hạnh Nhi, *Phân tích cú pháp câu tiếng Việt và ứng dụng vào việc bắt lỗi chính tả,* luận văn cử nhân tin học, ĐH KHTN TPHCM, 1999 | |
| [5] | | Nguyễn Phương Thái, *Kiểm lỗi Chính tả Cảm Ngữ cảnh Tiếng Việt*, Luận văn thạc sĩ, Khoa Công nghệ, 2003. | |
| [6] | | Nguyễn Hưu Tiến Quan, *Kiểm lỗi chính tả Tiếng Việt sử dụng mô hình ngôn ngữ và phân doạn từ*, Luận văn cử nhân, ĐH Công Nghệ Hà Nội, 2012. | |
| [7] | | Nguyễn Thái Ngọc Duy, Đinh Điền, *Một cách tiếp cận trong kiểm lỗi chính tả tiếng Việt,* Luận văn cử nhân tin học, ĐH KHTN TPHCM, 2004. | |
| [8] | | Andrew R.Golding and Dan Roth, *A winnow-based approach to context-sensitive correction,*Machine Learning, Special issue on Machine Learning and Nature Language Processing, 1999. | |
| [9] | | Andrew R.Golding and Yves Schabes, *Combining trigram-based ang feature-based methods for context-sensitive spelling correction,*Proceedings of the 34th Annual Meeting of Association for Computational Linguistics, 1996. | |
| [10] | | Andrew R.Golding, *A Bayesian hybrid methods for context-sensitive correction,* Proceedings of the Third Workshop on Very Large Copora, 1995. | |
| [11] | | Jurafsky-Martin, *Speech and Language Processing 2ed,* 2006. | |
| [12] | | Li Jianhua and Wang XiaoLong, *Combine trigram and Automatic Weight Distribution in Chinese Spelling ErrorCorrection,* 2000. | |
| [13] | | Masaaki Nagata, *Context-Based Spelling Correction for Japanese OCR,* 1998. | |