**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Nguyễn Hữu Tiến Quang**

**KIỂM LỖI CHÍNH TẢ TIẾNG VIỆT SỬ DỤNG MÔ HÌNH NGÔN NGỮ VÀ PHÂN ĐOẠN TỪ**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành:** **Công Nghệ Thông Tin**

**Cán bộ hướng dẫn: Ts. Lê Anh Cường**

**HÀ NỘI - 2012**

**VIETNAM NATIONAL UNIVERSITY, HANOI**

**UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Quang Nguyen Huu Tien**

**VIENAMESE SPELLING CHECK USING LANGUAGE MODEL AND WORD SEGMENTATION**

**Major: Information Technology**

**Supervisor: Anh Cuong Le, Ph.D**

**HA NOI - 2012**

**KIỂM LỖI CHÍNH TẢ TIẾNG VIỆT SỬ DỤNG MÔ HÌNH NGÔN NGỮ VÀ PHÂN ĐOẠN TỪ**

**Nguyễn Hữu Tiến Quang**

*Khóa QH-2008-I/CQ, ngành Công Nghệ Thông Tin*

**Tóm tắt Khóa luận tốt nghiệp**

Trong vài thập kỷ gần đây, cùng với sự phát triển nhanh chóng của công nghệ thông tin, mạng lưới hành chính điện tử ngày càng được mở rộng. Đa số các văn bản hiện nay được soạn thảo bằng máy vi tính. Với những văn bản hành chính quan trọng, việc soạn thảo đúng chính tả là điều hết sức cần thiết, bởi lẽ việc sai chính tả có thể gây ra những hậu quả lớn trong công việc. Một văn bản sai chính tả có thể làm người đọc cảm thấy khó chịu, không tin tưởng vào trình độ của người tạo ra văn bản đó, và nghiêm trọng hơn nó có thể làm người đọc hiểu sai nội dung cần truyền đạt. Do đó, kiểm lỗi chính tả trên máy tính là nhu cầu cấp thiết, đặc biệt là đối với tiếng Việt chúng ta. Xuất phát từ thực tiễn đó, trong khóa luận tốt nghiệp này của chúng tôi sẽ khảo sát một số hệ thống kiểm lỗi chính tả đã có, đồng thời đề xuất xây dựng một hệ thống kiểm lỗi chính tả có độ chính xác cao, trong bước đầu thực nghiệm cho thấy những kết quả khả quan, hứa hẹn khả năng ứng dụng cao trong thực tế.

**VIETNAMESE SPELLING CHECK USING LANGUAGE MODEL AND WORD SEGMENTATION**

**Nguyen Huu Tien Quang**

*Khóa QH-2008-I/CQ, Information Technology Major*

**Abstract**

In recent decades, along with the rapid growth of information technology the electronic network for public administration and government in Vietnam is increasingly expanded. Recently, most of official document was edited and transmit using computer software. It is essential that important documents for publicity should has high level of precision. Misspelled document not only cause irritation to reader, lost reader integrity and truth to authors but also can cause serious consequences especially mis-understanding important contents. Therefore, a good spell checker software is really an urgent need, especially for Vietnamese language. Starting from that observation, we conducted researchs and summarized in our report examination results of some popular existing spell checker and at the same time propose a design of high performance spell checker system that can be easily applied into practice. In the first step of experiment, the system have shown a good result.

**LỜI CẢM ƠN**

Lời đầu tiên, Em xin tỏ lòng biết ơn sâu sắc tới thầy giáo TS. Lê Anh Cường, người đã trực tiếp hướng dẫn tận tình và đóng góp cho em những ý kiến quý báu trong suốt quá trình em thực hiện và hoàn thành khóa luận tốt nghiệp này.

Em xin chân thành cảm ơn các thầy cô giáo thuộc khoa Công Nghệ Thông Tin, trường Đại học Công Nghệ - ĐHQGHN. Các thầy cô đã tận tình dạy dỗ, truyền đạt cho em nhiều kiến thức quý báu trong suốt những năm Đại học để em có được ngày hôm nay.

Tôi xin gửi lời cảm ơn đến bạn bè của tôi, những người đã luôn bên cạnh, khuyến khích và động viên tôi trong học tập cũng như cuộc sống.

Cuối cùng, con xin gửi lời cảm ơn vô hạn đến ông bà, cha mẹ, người thân đã sinh thành và nuôi dưỡng con thành người.

Sinh viên

Nguyễn Hữu Tiến Quang

**LỜI CAM ĐOAN**

Tôi xin cam đoan khóa luận tốt nghiệp với đề tài “kiểm lỗi chính tả tiếng Việt sử dụng mô hình ngôn ngữ và phân đoạn từ” là khóa luận tốt nghiệp nghiên cứu do tôi thực hiện dưới sự hướng dẫn của thầy giáo TS. Lê Anh Cường.

Tất cả các số liệu, kết quả được trình bày trong khóa luận là hoàn toàn trung thực. Tất cả các bài báo, khóa luận, tài liệu, công cụ phần mềm của các tác giả được sử dụng trong khóa luận này đều được chỉ dẫn tường minh về tác giả và đều có trong danh sách tài liệu tham khảo.

Hà Nội, tháng 5 năm 2012

Sinh viên

Nguyễn Hữu Tiến Quang

MỤC LỤC

[MỞ ĐẦU 1](#_Toc326677117)

[CHƯƠNG 1. TỔNG QUAN 2](#_Toc326677118)

[1.1 Bài toán kiểm lỗi chính tả 2](#_Toc326677119)

[1.2 Tình hình kiểm lỗi chính tả trong và ngoài nước 3](#_Toc326677120)

[1.3 Hướng tiếp cận của chúng tôi 3](#_Toc326677121)

[1.4 Đóng góp của khóa luận 4](#_Toc326677122)

[CHƯƠNG 2. ĐẶC ĐIỂM CỦA NGỮ PHÁP TIẾNG VIỆT 6](#_Toc326677123)

[2.1 Đặc điểm của tiếng Việt 6](#_Toc326677124)

[2.2 Các đơn vị của tiếng Việt 6](#_Toc326677125)

[2.2.1 Âm 6](#_Toc326677126)

[2.2.2 Chữ cái 7](#_Toc326677127)

[2.2.3 Thanh và dấu 7](#_Toc326677128)

[2.2.4 Tiếng 7](#_Toc326677129)

[2.2.5 Hình vị 8](#_Toc326677130)

[2.2.6 Từ 8](#_Toc326677131)

[2.2.7 Câu 9](#_Toc326677132)

[2.3 Chuẩn chính tả 9](#_Toc326677133)

[2.4 Nguyên nhân gây lỗi chính tả tiếng Việt 10](#_Toc326677134)

[2.4.1 Lỗi thanh điệu 10](#_Toc326677135)

[2.4.2 Lỗi âm đầu 11](#_Toc326677136)

[2.4.3 Lỗi âm chính 11](#_Toc326677137)

[2.4.4 Lỗi âm cuối 11](#_Toc326677138)

[2.5 Tập nhầm lẫn âm tiết tiếng Việt 11](#_Toc326677139)

[2.5.1 Nhầm lẫn do đánh máy sai 12](#_Toc326677140)

[2.5.2 Nhầm lẫn do phát âm 13](#_Toc326677141)

[CHƯƠNG 3. CÁC PHƯƠNG PHÁP KIỂM LỖI CHÍNH TẢ 15](#_Toc326677142)

[3.1 Kiểm lỗi chính tả cho ngôn ngữ nước ngoài 15](#_Toc326677143)

[3.1.1 Kiểm lỗi chính tả cho các ngôn ngữ châu Âu 15](#_Toc326677144)

[3.1.2 Kiểm lỗi chính tả cho các ngôn ngữ châu Á 17](#_Toc326677145)

[3.1.3 Đánh giá chung về các mô hình trên 19](#_Toc326677146)

[3.2 Kiểm lỗi chính tả tiếng Việt 20](#_Toc326677147)

[3.2.1 Phương pháp kiểm lỗi chính tả dựa trên từ điển 20](#_Toc326677148)

[3.2.2 Kiểm lỗi chính tả dựa vào phân tích cú pháp 22](#_Toc326677149)

[3.2.3 Kiểm lỗi chính tả bằng mô hình mạng lưới từ 22](#_Toc326677150)

[3.2.4 Kiểm lỗi chính tả trong MS Word 23](#_Toc326677151)

[3.2.5 Những phần mềm kiểm lỗi chính tả khác 24](#_Toc326677152)

[CHƯƠNG 4. GIỚI THIỆU PHƯƠNG PHÁP 25](#_Toc326677153)

[4.1 Cơ sở lý thuyết 25](#_Toc326677154)

[4.1.1 Luật Bayes 25](#_Toc326677155)

[4.1.2 Mô hình N-gram tổng quát 26](#_Toc326677156)

[4.1.3 Giới thiệu Corpus 32](#_Toc326677157)

[4.1.4 Thuật toán Viterbi 36](#_Toc326677158)

[4.2 Giới thiệu phương pháp 38](#_Toc326677159)

[4.2.1 Mô hình tổng quát 39](#_Toc326677160)

[4.2.2 Hạn chế lỗi tích cực bằng Heuristic 43](#_Toc326677161)

[4.2.3 Làm mịn 44](#_Toc326677162)

[4.2.4 Học trong quá trình sử dụng 44](#_Toc326677163)

[CHƯƠNG 5. CÀI ĐẶT VÀ THỰC NGHIỆM 45](#_Toc326677164)

[5.1 Cài đặt 45](#_Toc326677165)

[5.1.1 Tạo Corpus thô 45](#_Toc326677166)

[5.1.2 Đếm tần suất N-gram và tổ chức lưu trữ 46](#_Toc326677167)

[5.1.3 Viết chương trình 46](#_Toc326677168)

[5.2 Thực nghiệm 47](#_Toc326677169)

[5.2.1 Môi trường và dữ liệu thực nghiệm 47](#_Toc326677170)

[5.2.2 Kết quả thực nghiệm 50](#_Toc326677171)

[5.2.3 Đánh giá 54](#_Toc326677172)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 55](#_Toc326677173)

[TÀI LIỆU THAM KHẢO 57](#_Toc326677174)

**Danh sách bảng biểu**

[Bảng 1: Bảng các thành phần âm tiết 8](#_Toc325109126)

[Bảng 2: Bảng liệt kê các lỗi phát âm thường gặp 14](#_Toc325109127)

[Bảng 3: Thông số trong Corpus 45](#_Toc325109128)

[Bảng 4: Bảng các giá trị trong các công thức tính độ đo 47](#_Toc325109129)

[Bảng 5: Môi trường thực nghiệm 48](#_Toc325109130)

[Bảng 6: Các âm tiết được gây lỗi trong dữ liệu thực nghiệm loại 1 49](#_Toc325109131)

[Bảng 7: Kết quả thực nghiệm loại 1 51](#_Toc325109132)

[Bảng 8: Kết quả thực nghiệm dữ liệu loại 2 52](#_Toc325109133)

[Bảng 9: Kết quả thực nghiệm dữ liệu loại 2 qua các độ đo 53](#_Toc325109134)

**Danh sách hình ảnh**

[Hình 1: Ví dụ về tập nhầm lẫn âm tiết tiếng Việt 12](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676477)

[Hình 2: Ví dụ kiểm lỗi chính tả bằng phương pháp Nagata 18](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676478)

[Hình 3: Ví dụ kiểm lỗi chính tả dựa trên từ điển 21](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676479)

[Hình 4: Sơ đồ các bước tạo corpus 36](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676480)

[Hình 5: Mô hình tổng quát của chương trình 39](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676481)

[Hình 6: Ví dụ kiểm lỗi bằng phương pháp đề xuất 42](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676482)

[Hình 7: Mô hình lưu trữ bigram 46](file:///D:\research\THESIS\Final_thesis.docx#_Toc326676483)

[Hình 8: Kết quả thực nghiệm dữ liệu loại 1 52](#_Toc326676484)

[Hình 9: So sánh kết quả thực nghiệm loại 2 53](#_Toc326676485)

[Hình 10: So sánh kết quả thực nghiệm loại 2 qua các độ đo 53](#_Toc326676486)

**Bảng các ký hiệu viết tắt**

|  |  |
| --- | --- |
| Từ hoặc cụm từ | Ký hiệu viết tắt |
| NLP | Natural Language Processing |
| MLE | Maximum Likelihood Estimation |
| VC++ | Visual C++ |

MỞ ĐẦU

Như chúng ta đã biết, việc soạn thảo văn bản đúng chính tả là điều hết sức quan trọng. Tuy nhiên, từ trước tới nay, việc kiểm lỗi chính tả hầu hết hoàn toàn dựa vào sức người. Sau khi hoàn thành một văn bản, người soạn thảo phải rà soát lại thật kỹ lưỡng để tìm ra những sai sót trong quá trình đánh máy. Điều này thực sự rất khó khăn, đặc biệt là đối với những văn bản, tài liệu dài, người kiểm lỗi phải bỏ ra rất nhiều công sức. Trên thực tế, có những lỗi chính tả mà ngay cả chính người soạn văn bản cũng không xác định được. Điều này có thể xuất phát từ nhiều yếu tố khác nhau như trình độ, vùng miền, phát âm,… Vì vậy, kiểm lỗi chính tả tiếng Việt tự động là một nhu cầu thiết thực đối với người dân Việt Nam chúng ta, đặc biệt là những người thường xuyên sử dụng máy tính.

Cùng với sự phát triển nhanh chóng của công nghệ thông tin, bài toán đặt ra đối với những người nghiên cứu NLP là làm sao máy tính có thể thay thế (một phần hoặc toàn bộ) cho con người trong việc kiểm lỗi chính tả, hay nói cách khác là việc làm sao xây dựng một chương trình bắt lỗi chính tả tự động trên máy tính.

Tuy nhiên, trong khi bài toán kiểm lỗi chính tả cho các ngôn ngữ châu Âu đã được giải quyết khá trọn vẹn và triệt để, thì đối với tiếng Việt chúng ta lại còn gặp rất nhiều khó khăn. Do đó chúng tôi lựa chọn kiểm lỗi chính tả tiếng Việt là đề tài nghiên cứu trong khóa luận tốt nghiệp của mình. Trong khóa luận này sẽ trình bày một hướng tiếp cận khác đó là áp dụng mô hình ngôn ngữ và phân đoạn từ vào kiểm lỗi chính tả tiếng Việt.

Bố cục của khóa luận gồm 5 chương chính như sau:

* Chương 1: Giới thiệu tổng quan về bài toán kiểm lỗi chính tả.
* Chương 2: Đặc điểm của ngữ pháp tiếng Việt.
* Chương 3: Các phương pháp kiểm lỗi chính tả trong và ngoài nước.
* Chương 4: Giới thiệu mô hình và phương pháp sử dụng trong khóa luận.
* Chương 5: Tóm tắt quá trình cài đặt, cùng với các thực nghiệm và so sánh.

Ngoài ra còn có phần tài liệu tham khảo và mục lục.

# TỔNG QUAN

Trong chương mở đầu của khóa luận, chúng tôi sẽ giới thiệu về chung về bài toán kiểm lỗi chính tả và vai trò, ý nghĩa của nó. Phần cuối của chương sẽ diễn đạt khái quát hướng tiếp cận của chúng tôi trong việc xây dựng hệ thống kiểm lỗi chính tả tiếng Việt.

## Bài toán kiểm lỗi chính tả

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) [2]. Bài toán kiểm lỗi chính tả là bài toán phát hiện ra những từ trong văn bản không viết theo đúng chuẩn chính tả. Có hai loại lỗi chính tả đó là:

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

Ví dụ: bogns (dạng đúng: bóng)

* Âm tiết có trong từ điển, nhưng đặt trong ngữ cảnh của câu văn hoặc đoạn văn chứa âm tiết đó thì lại không phù hợp.

Ví dụ: sử dung (dạng đúng: sử dụng)

Trong lĩnh vực công nghệ thông tin, hệ thống kiểm lỗi chính tả là một công cụ sử dụng để đánh dấu những âm tiết viết sai chính tả trong một văn bản. Một hệ thống kiểm lỗi chính tả thường có 2 chức năng chính:

* Phát hiện lỗi: Tìm kiếm những âm tiết bị lỗi chính tả trong văn bản.
* Sửa lỗi: Đưa ra những gợi ý sửa lỗi cho những âm tiết bị lỗi.

Có rất nhiều nguyên nhân khách quan cũng như chủ quan gây nên những lỗi chính tả trong các văn bản. Do đó, các chương trình soạn thảo và xử lý văn bản hiện nay rất cần tính năng soát lỗi chính tả tự động để giúp người sử dụng giảm thời gian và công sức bỏ ra. Các chương trình nhận dạng văn bản cũng cần tính năng này để nâng cao chất lượng nhận dạng. Ngoài ra, các ứng dụng lớn như trình quản lý thư điện tử, từ điển điện tử, hệ thống tìm kiếm,… cần tích hợp tính năng kiểm lỗi chính tả để tăng hiệu quả của ứng dụng. Vì vậy, bài toán kiểm lỗi chính tả đóng một vai trò quan trọng trong công nghệ thông tin, đặc biệt là trong lĩnh vực xử lý ngôn ngữ tự nhiên.

## Tình hình kiểm lỗi chính tả trong và ngoài nước

Như đã nói ở trên, vai trò kiểm lỗi chính tả tự động là rất lớn. Từ những năm 60 của thế kỷ XX, bài toán kiểm lỗi chính tả tự động bằng máy tính nhận được sự quan tâm của rất nhiều nhà khoa học. Cho đến nay, có rất nhiều những thuật toán, mô hình để giải quyết bài toán này được công bố. Tuy nhiên hầu hết chúng đều tập trung vào các ngôn ngữ biến cách của châu Âu như tiếng Anh, tiếng Pháp,… còn kiểm lỗi chính tả cho các ngôn ngữ đơn lập của châu Á như tiếng Hoa, tiếng Việt,… chỉ mới được bắt đầu nghiên cứu gần đây.

Đối với ngôn ngữ nước ngoài, đặc biệt là ngôn ngữ châu Âu như tiếng Anh, bài toán kiểm lỗi chính tả đã được xử lý một cách khá trọn vẹn và triệt để. Có thể kể đến đó là các mô hình từ ngữ cảnh [11], Lai Bayes [12], Winnow [10] cho tiếng Anh, Mô hình CInsunSpell [15] cho tiếng Trung, mô hình Nagata cho tiếng Nhật [17], …

Đối với tiếng Việt của chúng ta, bài toán kiểm lỗi chính tả khó khăn hơn rất nhiều. Khác với các ngôn ngữ châu Âu, ranh giới giữa các từ trong câu tiếng Việt không được phân biệt rõ ràng. Vì vậy, rất khó để áp dụng các mô hình kiểm lỗi chính tả của châu Âu vào kiểm lỗi tiếng Việt. Thêm vào đó, hiện nay trong nhiều văn bản khoa học của tiếng Việt thường có dùng thêm tiếng Anh để chú thích cho các thuật ngữ hoặc cho các từ không thể dịch sát nghĩa. Do đó, bài toán kiểm lỗi chính tả tiếng Việt lại càng khó khăn hơn. Trước khi chúng tôi thực hiện khóa luận tốt nghiệp này, đã có một số phương pháp kiểm lỗi chính tả được công bố. Có thể kể đến đó là phương pháp dựa trên từ điển [8], phương pháp dựa trên phân tích cú pháp câu [7] hay phương pháp mạng lưới từ [9]. Tuy nhiên, hiệu quả của các mô hình trên là chưa cao và khó có thể áp dụng vào thực tế.

## Hướng tiếp cận của chúng tôi

Trong luận văn thạc sỹ của Nguyễn Phương Thái [8] đã nêu lên vấn đề lớn nhất trong kiểm lỗi chính tả tiếng Việt là kiểm lỗi chính tả cảm ngữ cảnh. Khác với các phương pháp kiểm lỗi chính tả truyền thống, vốn xem các từ một cách độc lập với nhau và chủ yếu dựa vào từ điển, trong kiểm lỗi chính tả cảm ngữ cảnh, để quyết định một từ có bị sai chính tả hay không, hệ thống phải xét đến những đặc trưng xung quanh nó như lân cận, từ loại, trật tự từ… Do đó, phương pháp kiểm lỗi chính tả cảm ngữ cảnh có thể phát hiện được những lỗi sai về cách dùng từ, nghĩa là từ vốn không bị sai chính tả, nhưng trong hoàn cảnh của câu văn thì việc dùng nó là không hợp lý. Tuy nhiên, đi cùng với hiệu quả cao là sự khó khăn trong cách tiếp cận.

Mục tiêu sau cùng của khóa luận là một chương trình kiểm lỗi chính tả hoàn chỉnh, có thể sử dụng được ngay trong thực tế. Để đạt được điều đó, ngoài hiệu quả thực tế thì các mô hình ngôn ngữ được áp dụng ở đây không nên quá cầu kỳ, phức tạp hoặc đòi hỏi nhiều không gian nhớ. Tuy nhiên, những phương pháp kiểm lỗi chính tả hiện tại vẫn chưa đáp ứng được những yêu cầu trên. Đối với phương pháp dựa trên từ điển [8], phương pháp này có tốc độ xử lý cao, tuy nhiên độ chính xác còn rất hạn chế và còn phát hiện sai khá nhiều lỗi. Trong khi đó phương pháp dựa trên phân tích cú pháp câu [7] khá phức tạp, đòi hỏi thời gian xử lý lớn và rất khó có thể đạt hiệu quả cao. Mô hình mạng lưới từ [9] trên lý thuyết thì hiệu quả sẽ rất tốt, tuy nhiên lại gặp vấn đề ở thời gian xử lý do sự bùng nổ tổ hợp và khó có thể áp dụng vào thực tế.

Từ những yếu tố trên, chúng tôi quyết định chọn mô hình ngôn ngữ N-gram làm hướng tiếp cận chính. Mô hình N-gram tuy chưa phải là mô hình tối ưu nhất nhưng có một số ưu điểm đó là không quá phức tạp trong xử lý, nếu được tổ chức cấu trúc dữ liệu tốt thì lượng bộ nhớ chiếm dụng trong quá trình thực hiện là có thể chấp nhận được. Trong mô hình của khóa luận, N-gram được dùng làm phương tiện để tính toán các xác suất, các mối liên kết giữa các tiếng trong văn bản, dò tìm ra những âm tiết bị lỗi, và sau cùng đưa ra những phương án sửa lỗi hợp lý cho người dùng có thể lựa chọn.

## Đóng góp của khóa luận

Trong khóa luận tốt nghiệp này, chúng tôi nêu lên một cách giải quyết khác cho bài toán kiểm lỗi chính tả tiếng Việt, đó là phương pháp áp dụng mô hình ngôn ngữ và phân đoạn từ. Trong đó, mô hình N-gram sử dụng để đo độ gắn kết giữa các cụm tiếng, từ đó tìm ra được những lỗi chính tả trong văn bản đầu vào. Sau khi tìm được các âm tiết bị lỗi chính tả, mô hình sẽ đưa ra những âm tiết gợi ý mà theo mô hình là phù hợp nhất để sửa lỗi.

Tập ngữ liệu (Corpus) đóng một vai trò cực kỳ quan trọng tới độ chính xác của mô hình. Để mô hình đạt được hiệu quả cao, chúng ta cần có được một tập Corpus đủ lớn và thuộc nhiều thể loại. Trong khóa luận này sẽ nêu lên phương pháp để xây dựng một bộ Corpus thô cho mô hình.

Cuối cùng, sau khi áp dụng vào những lý thuyết đã nêu, chúng tôi thu thập được một bộ dữ liệu Corpus khá lớn với đầy đủ thể loại. Đồng thời chúng tôi cũng đã xây dựng một chương trình kiểm lỗi chính tả tiếng Việt khá hoàn thiện và có thể áp dụng vào thực tế dựa trên phương pháp áp dụng mô hình ngôn ngữ và phân đoạn từ đã nêu. Trong bước đầu thực nghiệm, chương trình đã cho thấy được những kết quả rất khả quan.

# ĐẶC ĐIỂM CỦA NGỮ PHÁP TIẾNG VIỆT

Nội dung chính trong chương 2 sẽ giới thiệu một số khái niệm và đặc điểm của ngữ pháp tiếng Việt. Đồng thời, cũng trong chương này, chúng tôi sẽ nêu lên những vấn đề thường gặp trong chính tả tiếng Việt.

## Đặc điểm của tiếng Việt

Tiếng Việt là thứ tiếng thuộc họ Nam Á và là một ngôn ngữ đơn lập. Trong tiếng Việt, quan hệ giữa các từ được biểu thị không phải bằng các phụ tố chứa trong bản thân từ mà bằng những phương tiện nằm ngoài từ như trật tự từ, hư từ. Đặc điểm này bao quát ngữ pháp tiếng Việt cả về ngữ âm, ngữ pháp và ngữ nghĩa [8].

## Các đơn vị của tiếng Việt

### Âm

Khi nói, luồng hơi phát ra từ phổi, làm rung các dây thanh ở họng, qua khoang miệng hoặc khoang mũi và tạo thành các âm. Âm tiếng việt bao gồm: nguyên âm, phụ âm, bán âm, âm đệm.

**Nguyên âm**

Nguyên âm được tạo do luồng hơi thoát ra tự do, không bị cản trở tại một điểm nào. Trong chính tả nguyên âm có thể mang dấu. Tiếng Việt 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**

Phụ âm được tạo ra do luồng hơi bị cản ở họng, lưỡi, răng hoặc môi. Trong chính tả, phụ âm không mang dấu. Trong tiếng Việt 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**

Tiếng Việt tồn tại 2 loại bán âm cuối. Loại bán âm cuối thứ nhất được thể hiện bằng 2 chữ cái i/y (Ví dụ: lại, lấy). Loại bán âm cuối thứ hai được thể hiện bằng 2 chữ cái o/u (Ví dụ: ao, sâu).

**Âm đệm**

Tiếng Việt thể hiện âm đệm bằng 2 chữ cái o và u (Ví dụ: hoa, quê). Âm đệm có tác dụng làm cho vần trở nên tròn môi hơn khi phát âm (Ví dụ so sánh: an 🡪 oan, ên 🡪 uên).

### Chữ cái

Chữ cái dùng để ghi âm. Theo quy định về chính tả trong sách giáo khoa của bộ Giáo dục đào tạo, bảng chữ cái tiếng Việt gồm có 33 con chữ xếp theo thứ tự sau: A, Ă, Â, B, C, D, Đ, E, Ê, (F), G, H, I ,(J), K, L, M, N, O, Ô, Ơ, P, Q, R, S, T, U, Ư, V, (W), X, Y, (Z). Trong số đó, các chữ cái trong ngoặc bao gồm (F), (J), (W), (Z) chỉ dùng để viết tên riêng nước ngoài và thuật ngữ gốc nước ngoài. Trường hợp viết tiếng Việt sử dụng các chữ cái trên là sai chính tả. Ví dụ: “fường”, “zá”…

### Thanh và dấu

Thanh là hiện tượng nâng cao hay hạ thấp trong một tiếng. Trong tiếng Việt, có 6 thanh (ngang, huyền, hỏi, ngã, sắc, nặng) và được thể hiện bằng 5 dấu (thanh ngang không có dấu ghi).

### Tiếng

* Giá trị ngữ âm: Tiếng là âm tiết. Khi nói, cứ phát âm một hơi là thành một âm tiết. Về chữ viết, mỗi âm tiết được ghi thành một chữ.
* Cấu tạo: Phụ âm đầu, vần, 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** | huyền, hỏi, ngã, sắc, nặng |

Bảng 1: Bảng các thành phần âm tiết

* Giá trị ngữ nghĩa: Tiếng là đơn vị nhỏ nhất có thể có nghĩa.
* Giá trị ngữ pháp: Ta đã biết ngữ pháp gồm những qui tắc cấu tạo từ, cấu tạo câu. Tiếng là đơn vị ngữ pháp dùng để cấu tạo từ.

### Hình vị

Trong ngữ pháp truyền thống hình vị là thành tố trực tiếp để tạo nên từ. Do đó, hình vị được coi như là đơn vị gốc, tế bào của ngôn ngữ và còn được gọi với tên khác là *“từ tố”*. Theo ngôn ngữ học đại cương, hình vị là đơn vị nhỏ nhất có nghĩa và có chức năng về mặt ngữ pháp.

Trong giáo trình Xử lý ngôn ngữ tự nhiên của TS. Đinh Điền [3], hình vị của tiếng Việt, bên cạnh đặc điểm như hình vị của ngôn ngữ học đại cương, còn phải có “hình tố”.

Hình vị tiếng Việt có khi gồm một tiếng, nhưng có nhiều hình vị được cấu tạo gồm nhiều tiếng (Ví dụ: lung linh, đỏng đảnh…)

### Từ

Theo [6] thì “từ” gồm một hoặc một tổ hợp tiếng có ý nghĩa hoàn chỉnh. Từ là đơn vị để đặt câu. Từ tiếng Việt có đặc trưng là đa âm tiết (trong tiếng Anh từ chỉ có một âm tiết). Xung quanh việc xác định thế nào là từ tiếng Việt còn nhiều tranh cãi. Tuy nhiên có các điểm sau thường được thống nhất [13]:

* Về mặt hình thức: từ là một khối thống nhất về cấu tạo (về chính tả, ngữ âm, v.v.).
* Về nội dung: từ có nghĩa hoàn chỉnh.
* Về khả năng: từ có khả năng hoạt động tự do và độc lập về mặt cú pháp.

Hai phương thức cấu tạo từ chủ yếu là *láy* và *ghép*:

Láy là sắp đặt các tiếng thành đôi, kế cận nhau và sao cho có phối hợp ngữ âm. Sự phối hợp ngữ âm này tạo nên nghĩa của *từ láy*.

Ví dụ: long lanh, chập chờn, ...

Ghép là sắp đặt các tiếng thành đôi, kế cận nhau và sao cho có phối hợp ngữ nghĩa. Sự phối hợp ngữ nghĩa này tạo nên nghĩa của *từ ghép*.

Ví dụ: nhà cửa, ăn ở, no ấm, xe đạp, ...

Đối với tiếng Anh thì bài toán phân đoạn từ khá đơn giản. Ranh giới ngăn cách các từ trong tiếng Anh chính là các ký tự trống. Tuy nhiên, đối với tiếng Việt thì bài toán phân đoạn từ khó khăn hơn rất nhiều bởi không có ranh giới cụ thể nào ngăn cách các từ với nhau.

### Câu

Trong ngôn ngữ, câu là đơn vị ở bậc cao nhất. Đặc điểm của câu đó là có nghĩa hoàn chỉnh và có cấu tạo rất đa dạng. Trong NLP, bài toán phân tích cú pháp câu là bài toán cơ bản. Tuy nhiên, đối với tiếng Việt thì bài toán vẫn chưa giải quyết được triệt để và đạt hiệu quả cao.

## Chuẩn chính tả

Theo [2], 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…

## Nguyên nhân gây lỗi chính tả tiếng Việt

Qua bảng điều tra lỗi chính tả trong nhà trường phổ thông và phương tiện truyền thông ở TP.HCM, Theo [6] kết luận các loại lỗi sau đây thường xảy ra phổ biến nhất (chỉ đề cập đến lỗi chính tả do phát âm sai).

### Lỗi thanh điệu

Theo những nhà ngữ âm học, khu vực từ miền Trung và Nam từ Thanh Hóa trở vào không phân biệt được hai thanh hỏi ngã. Chính xác hơn, ở trong khu vực này không tồn tại thanh ngã. Tuy nhiên, số lượng tiếng chứa thanh ngã là khá nhiều. Do đó, lỗi về thanh điệu thường xảy ra nhất nhiều ở các văn bản ở miền Trung và miền Nam.

### Lỗi âm đầu

Trong phương ngữ miền Bắc và Nam, thường xuyên xảy ra sự lẫn lộn giữa các chữ cái ghi âm đầu Ch/ Tr, S/X, D/Gi. Mặt khác, người miền Nam thường nhầm lẫn giữa V/D, Gi/W/Hw/Ngw/Qu. Ngoài ra, trong quy ước của tiếng việt còn những điểm rất dễ gây nhầm lẫn. Đó là với cùng một âm có tới nhiều cách ghi khác nhau (Ví dụ: /k/ - C,Qu,K). Tuy có quy định riêng cho cách ghi âm cho từng dạng. Tuy nhiên đối với một số người trình độ văn hóa chưa cao hoặc do sơ ý có thể gặp nhầm lẫn.

### Lỗi 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)

### Lỗi âm cuối

Ở đất nước Việt Nam, đặc biệt là khu vực miền 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. Do đó, lỗi âm cuối là một vấn đề rất khó khắc phục đối với người Việt Nam, đặc biệt là cư dân vùng Nam bộ như TP.HCM.

## Tập nhầm lẫn âm tiết tiếng Việt

Dưới góc độ của chương trình kiểm lỗi chính tả, có hai loại lỗi chính tả. Loại lỗi chính tả thứ nhất đó là lỗi âm tiết sai và hoàn toàn không có trong từ điển. Loại lỗi chính tả thứ hai là âm tiết có trong từ điển, nhưng đặt trong ngữ cảnh của câu văn hoặc đoạn văn chứa âm tiết đó thì lại không phù hợp.

Trong trường hợp âm tiết sai, trên lý thuyết việc phát hiện ra lỗi khá đơn giản. Chúng ta chỉ cần một từ điển thật chính xác và đầy đủ là có thể phát hiện tốt những loại lỗi âm tiết sai. Tuy nhiên, trong văn bản tiếng Việt hiện nay có dùng thêm một số thuật ngữ nước ngoài, do đó việc xác định loại lỗi này trở nên phức tạp hơn.

Đối với trường hợp âm tiết đúng nhưng sai trong cách dùng, việc phát hiện ra loại lỗi này khó khăn hơn rất nhiều. Máy tính rất khó có thể “hiểu” được ý nghĩa của văn bản. Do đó, để phát hiện ra lỗi chính tả ta cần xét các tập nhầm lẫn âm tiết. Tập nhầm lẫn ảnh hưởng rất lớn trong việc phát hiện và sửa lỗi của hệ thống. Tập nhầm lẫn của một âm tiết s là tập hợp các âm tiết mà mỗi âm tiết trong tập đó có thể bị nhầm lẫn thành s. Có rất nhiều nguyên nhân nhầm lẫn như đánh máy, phát âm (phương ngữ) ở người hay do đặc trưng của các hệ nhận dạng chữ. Một chương trình kiểm lỗi chính tả đi kèm với các hệ soạn thảo văn bản cần sử dụng các tập nhầm lẫn do đánh máy và phát âm hợp lại.

Phát âm

Thiếu dấu sắc

Thiếu chữ k

Thiếu chữ h

Hình 1: Ví dụ về tập nhầm lẫn âm tiết tiếng Việt

### 

### Nhầm lẫn do đánh máy sai

Đây là kiểu nhầm lẫn phổ biến. Trong tiếng Anh, có khoảng 80% lỗi chính tả trong văn bản đánh máy là lỗi do đánh máy sai. Thường có 4 loại lỗi nhầm lẫn do đánh máy như sau:

* Chèn: nhầm “run” thành “rung” (chèn thêm chữ cái ‘g’)
* Xóa: nhầm “chung” thành “cung” (xóa chữ ‘h’)
* Thay thế: nhầm “bán” thành “ván” (thay thế chữ ‘b’ thành chữ ‘v’)
* Hoán vị: nhầm “trung” thành “trugn” (hoán vị ‘ng’ thành ‘gn’)

Một âm tiết có thể bị biến đổi sau những lần biến đổi liên tiếp của 1 trong 4 phép biến đổi trên. Ví dụ: “nầm” thành “năm” (xóa dấu huyền và thanh thế ‘â’ thành ‘ă’). Trên thực tế, âm tiết chỉ thường nhầm lẫn sau 1 phép biến đổi.

### Nhầm lẫn do phát âm

Đối với người Việt Nam chúng ta, có một số lượng lớn âm tiết ai cũng viết đúng chính tả dễ dàng. Tuy nhiên, vẫn có những vấn đề chính tả thường gặp, được chia làm hai loại (Hoàng Phê, 1999 [5]) 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.

Bảng dưới đây liệt kê các loại lỗi phát âm thường gặp (Chính tả tiếng Việt [5]):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CH- | TR- | | | |
| D- | GI- | | | |
| D- | GI- | NH- | | |
| D- | GI- | R- | | |
| D- | GI- | V- | | |
| HW- | NG- | QU- | | W- |
| 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: Bảng liệt kê các lỗi phát âm thường gặp

*Ví dụ:*

Âm tiết: “dam” 🡪 Tập nhầm lẫn: “giam”

Âm tiết: “nhòm” 🡪 Tập nhầm lẫn: “nhồm”, “nhờm”

Âm tiết: “chương” 🡪Tập nhầm lẫn: “trương”

# CÁC PHƯƠNG PHÁP KIỂM LỖI CHÍNH TẢ

Theo [1], dựa vào loại hình ngôn ngữ (biến cách hay đơn lập), các phương pháp kiểm lỗi chính tả được chia thành hai “trường phái” chính: kiểm lỗi chính tả cho ngôn ngữ châu Âu và kiểm lỗi chính tả cho ngôn ngữ châu Á.

Trong chương này của khóa luận sẽ giới thiệu một vài phương pháp kiểm lỗi chính tả đã được công bố và đưa vào ứng dụng thành công trong thực tế cho các ngôn ngữ nước ngoài. Trong phần cuối của chương sẽ điểm qua một vài công trình nghiên cứu về kiểm lỗi chính tả tiếng Việt và đưa ra một số nhận xét về chúng.

## Kiểm lỗi chính tả cho ngôn ngữ nước ngoài

### Kiểm lỗi chính tả cho các ngôn ngữ châu Âu

Như đã nói ở trên, bài toán kiểm lỗi chính tả cho các ngôn ngữ châu Âu, nhất là tiếng Anh, đã được tìm hiểu từ rất sớm, với rất nhiều mô hình, phương pháp khác nhau như từ ngữ cảnh [11], Lai Bayes [12], Winnow [10],… Tuy nhiên, phương pháp kiểm lỗi chính tả cảm ngữ cảnh nhận được nhiều sự quan tâm của các nhà khoa học bởi nó có ưu điểm so với cách kiểm lỗi chính tả chỉ dựa vào từ điển.

#### Phương pháp từ ngữ cảnh

Một trong những đầu mối để phân biệt một từ với những từ nhập nhằng với nó là các từ xung quanh nó trong câu. Ví dụ, khi đang xem xét để chọn lựa giữa dessert và desert, nếu có các từ như arid, sand, sun,… ta sẽ chọn desert, còn nếu gặp các từ như chocolate, delicious,… gần đó, ta sẽ chọn từ dessert. Phương pháp Từ ngữ cảnh được Golding xây dựng dựa trên nhận xét này. Trong giai đoạn học, với mỗi từ thường bị viết sai chính tả, hệ thống sẽ ghi nhớ những từ thường xuất hiện xung quanh nó trong ngữ liệu. Khi chạy thực tế, với mỗi từ nghi ngờ, hệ thống sẽ xem xét các từ lân cận với nó, sau đó chọn ra từ thích hợp nhất để làm từ thay thế.

#### Phương pháp Đặc trưng lân cận

Phương pháp Từ ngữ cảnh rất tốt trong việc phát hiện các nguyên tắc chung nhất phụ thuộc vào những từ lân cận, nhưng không quan tâm đến trật tự của chúng. Trong trường hợp đó, các phương pháp Đặc trưng lân cận tỏ ra thích hợp để rút ra các quy luật về trật tự.

Một đặc trưng lân cận là một mẫu các yếu tố về cú pháp xung quanh từ đang xét. Có hai đặc trưng thường được sử dụng trong phương pháp này, đó là nhãn từ loại và từ lân cận.

Phương pháp Đặc trưng lân cận được thực hiện theo cách thức tương tự với phương pháp Từ ngữ cảnh. Ý tưởng chính ở đây là phân biệt những từ dễ bị dùng nhầm lẫn với nhau dựa vào các đặc trưng lân cận của chúng. Mỗi đặc trưng lân cận cung cấp vài dấu hiệu với khả năng quyết định khác nhau cho những từ trong tập nhầm lẫn. Những dấu hiệu này được kết hợp với nhau và tính toán theo luật quyết định (thường là Bayes). Sau cùng từ nào có độ hợp lý cao nhất sẽ được chọn.

Tuy nhiên, phương pháp Đặc trưng lân cận cũng có nhược điểm, đó là khi một từ có các đặc trưng mâu thuẫn hoặc hàm chứa nhau, gọi là xung đột. Để phương pháp này hoạt động tốt, các đặc trưng xung đột với nhau cần được phát hiện và loại bỏ. Hiện nay, ngoài heuristic, vẫn chưa có một phương pháp hoàn chỉnh để giải quyết vấn đề này.

#### Phương pháp danh sách quyết định

Danh sách quyết định là sự kết hợp của Từ ngữ cảnh và Đặc trưng lân cận để tận dụng ưu điểm của cả hai phương pháp này: Từ ngữ cảnh sẽ rút ra các nguyên tắc tốt nhất theo cách độc lập với trật tự từ, còn Đặc trưng lân cận được rút ra các nguyên tắc phụ thuộc vào trật tự.

Phương pháp này đòi hỏi phải xây dựng một danh sách rất lớn các đặc trưng - ở đây là các từ ngữ cảnh và các đặc trưng lân cận. Những đặc trưng được sắp xếp theo khả năng quyết định giảm dần, trong đó khả năng quyết định của mỗi đặc trưng phản ánh độ tin cậy của nó trong việc ra quyết định. Một từ nhập nhằng sẽ được phân loại bằng cách duyệt qua danh sách và so khớp với từng đặc trưng ngữ cảnh. Đặc trưng phù hợp đầu tiên sẽ được dùng để phân loại từ đang xét.

Vì danh sách quyết định đưa ra lời giải chỉ dựa vào một đặc trưng mạnh nhất, hiệu suất của nó phần lớn tùy thuộc vào cách xác định khả năng quyết định cho những đặc trưng. Đây chính là nhược điểm của phương pháp này, bởi vì việc tìm ra một công thức đánh giá tổng quát là rất khó.

### Kiểm lỗi chính tả cho các ngôn ngữ châu Á

Gần đây, kiểm lỗi chính tả cho các ngôn ngữ châu Á đã được quan tâm và đạt được một số thành tựu. Khác với ngôn ngữ châu Âu, việc kiểm lỗi chính tả cho các ngôn ngữ đơn lập của châu Á gặp nhiều khó khăn. Nguyên nhân chính là do các ngôn ngữ của châu Á như tiếng Hoa, tiếng Nhật, tiếng Hàn,… không có ranh giới từ rõ ràng. Để áp dụng được mô hình kiểm lỗi chính tả cảm ngữ cảnh của ngôn ngữ châu Âu, trước hết văn bản châu Á phải xác định được ranh giới từ. Đây là một điều không hề đơn giản. Đặc biệt là khi văn bản đầu vào sai chính tả, việc xác định ranh giới từ càng khó khăn hơn. Do đó, các nhà nghiên cứu đã đề ra một số mô hình mới trong kiểm lỗi chính tả cho các ngôn ngữ châu Á. Phần này sẽ điểm qua một số mô hình trong số đó.

#### Mô hình CInsunSpell cho tiếng Hoa

Mô hình CInsunSpell [15] là mô hình kiểm lỗi được đề xuất bởi Li Juanhua và Wang XiaoLong. Mô hình là sự kết hợp giữa N-gram (trigram) tiếng, phương pháp ước lượng xác suất Bayes và phân phối trọng số tự động. Mô hình được áp dụng trong sửa lỗi chính tả tiếng Hoa. Trong đó, N-gram (trigram) được sử dụng trong bước kiểm lỗi. Nó 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.

Nhược điểm của mô hình trên cũng là nhược điểm chung của mô hình N-gram, đó là dữ liệu quá thưa thớt, 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. Theo thử nghiệm của tác giả, mô hình CInsunSpell có tỉ lệ phát hiện lỗi là khoảng 60%. Giá trị này có thể thay đổi tùy thuộc vào các ngưỡng được sử dụng trong hệ thống.

#### Phương pháp của Nagata cho tiếng Nhật

Phương pháp Nagata [17] được áp dụng đầu tiên cho các hệ thống nhận dạng chữ viết. Tuy nhiên, nó vẫn có thể áp dụng để sửa lỗi chính tả do con người gây ra. Phương pháp gồm 2 giai đoạn như sau:

* Giai đoạn 1: Câu đầu vào được chia thành nhiều chuỗi tiếng, với điều kiện là mỗi chuỗi tiếng tạo thành một từ có trong từ điển hoặc gần giống với một từ nào đó trong từ điển.
* Giai đoạn 2: Bằng phương pháp thống kê chọn ra những câu tốt nhất được chọn làm ứng viên sửa lỗi.

*Ví dụ:*

Ngươi dan Việt Nam rất tôt bụng

Người da rát tót

dai tôi

đau tốt

dân

rát

Việt Nam

Người

dai

da

đau

Ngươi

dan

Người dân

rất

tôt

tôi bun

tót

tốt bụng

bụng

Hình 2: Ví dụ kiểm lỗi chính tả bằng phương pháp 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.

### Đánh giá chung về các mô hình trên

Các phương pháp kiểm lỗi cho ngôn ngữ châu Âu như Từ ngữ cảnh, Đặc trưng lân cận và Danh sách quyết định ứng dụng rất thành công cho ngôn ngữ châu Âu. Nhưng để áp dụng các phương pháp đó cho tiếng Việt chúng ta thì điều kiện cần thiết là câu đầu vào phải được tách từ mờ. Mô hình tách từ mờ không tốt sẽ ảnh hưởng rất lớn đến hiệu suất của hệ thống. Tuy nhiên, bài toán tách từ mờ vẫn chưa được giải quyết triệt để và đạt hiệu quả cao. Bên cạnh đó, các phương pháp trên đòi hỏi phải có kho dữ liệu được gán nhãn từ loại. Đối với tiếng Việt chúng ta rất khó để tìm ra được kho ngữ liệu phù hợp với yêu cầu này. Ngoài ra, những phương pháp trên đòi hỏi hệ thống phải thực hiện rất nhiều thao tác để có thể tìm ra được từ sai chính tả. Do đó rất khó áp dụng vào thực tế, đặc biệt là với tiếng Việt đa dạng và phong phú của chúng ta.

Tương tự với những phương pháp kiểm lỗi cho ngôn ngữ châu Âu, mặc dù trên lý thuyết phương pháp Nagata có thể áp dụng ngay cho tiếng Việt, tuy nhiên nó lại đòi hỏi kho ngữ liệu phải được gán nhãn, đó là một điều hết sức khó khăn. Chưa hết, trong ma trận tiếng nhầm lẫn của Nagata, nếu áp dụng cho tiếng Việt sẽ gặp một vấn đề rất khó giải quyết, đó là sự bùng nổ tổ hợp.

Đối với hệ thống kiểm lỗi chính tả CInsunspell cho tiếng Hoa, các tác giả đã sử dụng kết hợp giữa bigram và trigram ở mức tiếng để dò tìm lỗi chính tả. Đây là cách tiếp cận rất phù hợp cho tiếng Việt của chúng ta bởi nó không đòi hỏi những yêu cầu quá khó như tách từ mờ hay kho ngữ liệu được gán nhãn. Ngoài ra, mô hình còn có những ưu điểm nổi trội khác. Ưu điểm đầu tiên đó là sự đơn giản trong cài đặt và xử lý. Ưu điểm thứ hai đó là mô hình chỉ cần một kho dữ liệu đúng chính tả. Với những đặc điểm đó, việc áp dụng mô hình CInsunspell cho tiếng Việt là hết sức khả quan.

## Kiểm lỗi chính tả tiếng Việt

Kiểm lỗi chính tả là một bài toán rất khó, đặc biệt là đối với ngôn ngữ đơn lập như tiếng Hoa hay tiếng Việt. Trước đây, đã có rất nhiều người đi trước đã công bố một số công trình nghiên cứu về kiểm lỗi chính tả tiếng Việt với các hướng tiếp cận khác nhau. Mỗi cách tiếp cận sẽ có những ưu và nhược điểm riêng. Đã có những chương trình kiểm lỗi chính tả tiếng Việt được áp dụng vào thực tế như VietSpell Lưu Hà Xuyên hay gần đây hơn đó là CopCon của Mai Tuấn Khôi, tuy nhiên kết quả đạt được vẫn chưa thực sự cao. Kể từ năm 2005 cho đến nay, Microsoft đã tích hợp kiểm lỗi chính tả tiếng Việt cho MS Word, nhưng mô hình kiểm lỗi được sử dụng cho tiếng Việt trong đó vẫn còn rất thô sơ.

### Phương pháp kiểm lỗi chính tả dựa trên từ điển

Đây là phương pháp kiểm lỗi từ đơn giản nhất. Theo [8], hai ý tưởng chính của phương pháp này như sau:

* 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:*

“Sử dụng thành thạo máy *vu* tính là điều hết sức quan trọng”

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.

*Ví dụ 2:*

“Tính đến cuối năm, công ty đạt *tổn* doanh số rất cao”

Ta dễ nhận thấy “*tổn*” không thuộc về một từ nào cả, cho nên có khả năng đó là âm tiết lỗi (câu không phân đoạn được tại âm tiết này).

Hình 3: Ví dụ kiểm lỗi chính tả dựa trên từ điển

Theo phương pháp này, câu đầu vào sẽ được phân đoạn thành một dãy các từ. Sau đó, với mỗi từ trong wi trong dãy từ nhận được, nếu wi là từ ghép thì xét tiếp từ tiếp theo. Ngược lại, nếu wi là từ đơn thì hệ thống sẽ khởi tạo tập nhầm lẫn âm tiết tương ứng với nó. Sau khi đã tạo được tập nhầm lẫn âm tiết, hệ thống sẽ phân tích từ ghép tạo bởi nó (hoặc trong tập nhầm lẫn) với các từ đơn (hoặc trong tập nhầm lẫn) kề sau nó để xét xem có thể tạo thành từ ghép hay không. Nếu tìm được từ ghép *cwi* thì đánh dấu *wi* lỗi, ghi lại *cwi* là từ gợi ý và tiếp tục xét từ ở cách *wi* độ dài từ ghép *cwi*.

Ta có thể dễ dàng nhận thấy rằng với phương pháp kiểm lỗi chính tả bằng từ điển, chúng ta không thể bắt được những lỗi từ đơn. Đồng thời theo [8], khi thực nghiệm chương trình VietSpell2000 của Lưu Hà Xuyên được cài đặt theo phương pháp kiểm lỗi chính tả bằng từ điển với mức sửa lỗi bằng 4, chương trình rất dễ gặp các lỗi tiêu cực sau:

Câu vào: Sản phẩm được làm bằng tay.

Báo lỗi: bằng tay 🡪 bàn tay.

Chúng tôi sẽ trình bày kết quả thực nghiệm về hiệu quả của phương pháp này ở chương thực nghiệm của khóa luận.

### Kiểm lỗi chính tả dựa vào phân tích cú pháp

Năm 1999, trong luận văn tốt nghiệp đại học, Nguyễn Đức Hải và Nguyễn Phạm Hạnh Nhi [7] đã đề xuất mô hình kiểm lỗi chính tả tiếng Việt bằng cách phân tích cú pháp. Đây được xem như cách tiếp cận dựa trên Luật.

Theo mô hình này, câu đầu vào sẽ được hệ thống tiến hành phân tích cú pháp theo thuật toán Earley. Những chỗ không phân tích cú pháp được sẽ được xem như là bị lỗi chính tả. Earley là một thuật toán mạnh cho bài toán phân tích cú pháp. Tuy nhiên, nó lại gặp vấn đề trở ngại ở độ phức tạp của thuật toán. Độ phức tạp của thuật toán Earley là O(n3) trong đó n là số tiếng của câu. Do đó, thuật toán Earley phải được cài đặt thật tốt và hiệu quả để đạt được tốc độ cao.

Theo mô hình được đề xuất, để áp dụng thuật toán Earley cho tiếng Việt, câu đầu vào cần phải tách từ mờ trong điều kiện nó có thể không đúng chính tả. Đó là một điểm khó khăn của mô hình. Khó khăn thứ hai đó là các luật của ngôn ngữ tự nhiên rất phức tạp. Tiếng Việt có khoảng 3000 luật, việc thu thập được đầy đủ số luật là một điều hết sức khó khăn. Điểm khó khăn nữa của mô hình đó là trong tiếng Việt, việc phối hợp các tiếng để tạo thành một câu “chấp nhận được” là khá nhiều. Do đó, rất nhiều trường hợp câu bị sai chính tả những vẫn phân tích cú pháp được bình thường. Đây chính là nhược điểm chính của phương pháp này.

### Kiểm lỗi chính tả bằng mô hình mạng lưới từ

Trong năm 2004, Nguyễn Thái Ngọc Duy [9] đã đề xuất một mô hình kiểm lỗi chính tả mới trong luận văn tốt nghiệp của mình, đó là mô hình lưới từ cho kiểm lỗi chính tả tiếng Việt. Mô hình này hoạt động dựa vào việc tách từ mờ, sau đó tìm ra cách tách từ tốt nhất để làm câu sửa lỗi. Trong lưới từ thực chất là một cấu trúc đặc biệt, giống như là một đồ thị có hướng và không có chu trình xuất phát từ âm tiết đầu tiên đến âm tiết cuối cùng của câu. Sử dụng cấu trúc này giúp cho công việc tách từ mờ nhanh chóng hơn bởi nó có thể áp dụng những thuật toán tìm đường đi trong đồ thị.

Trên 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. Tuy nhiên, mô hình này có một nhược điểm rất khó khắc phục và xử lý đó là sự bùng nổ tổ hợp. Với sự đa dạng và phong phú của tiếng Việt thì việc tách từ cũng không hề đơn giản. Với một câu, có thể tìm ra những cách tách từ khác nhau. Thêm vào đó, với câu đầu vào sai chính tả thì việc tách từ mờ cũng trở nên khó khăn bởi mỗi tiếng thì hệ thống phải sinh ra những tiếng gần giống với nó, rồi xem xét cách phối hợp giữa nó và những tiếng lân cận nó trong câu để tìm ra được cách tách từ tốt nhất. Điều đó khiến cho chương trình trở nên chậm chạp, mất nhiều thời gian để phát hiện và sửa lỗi chính tả.

Trong mô hình của lưới từ này, tác giả còn sử dụng N-gram để ước lượng các giá trị cần thiết. Nhưng do ngữ liệu chưa được huấn luyện tốt nên việc ước lượng này còn chưa được chính xác. Ngoài ra, do chưa được tổ chức lưu trữ tốt nên chương trình phải tốn rất nhiều bộ nhớ cho lưu trữ N-gram. Do đó, chương trình càng trở nên khó khăn hơn trong việc áp dụng vào thực thế.

### Kiểm lỗi chính tả trong MS Word

Đầu năm 2005, Microsoft đã phát hành phần mềm MS Word 2003 phiên bản tiếng Việt dành riêng cho thị trường Việt Nam. Trong phiên bản này, Microsoft đã tích hợp chức năng kiểm lỗi chính tả cho văn bản tiếng Việt. Tuy nhiên, trái với mong đợi của nhiều người, chức năng kiểm lỗi chính tả trong MS Word 2003 chỉ được hoàn thành ở mức độ rất thấp là kiểm lỗi chính tả ở mức tiếng.

Hiện nay, việc xử lý kiểm lỗi chính tả cho tiếng Anh trong MS Word đã được xử lý khá trọn vẹn. Tuy nhiên, đối với tiếng Việt, nhiều tạp chí đánh giá chức năng kiểm lỗi chính tả tiếng Việt trong MS Word còn khá “ngô nghê”. Hầu hết chức năng chỉ phát hiện được những lỗi gõ sai như “khôgn, “dtôi”,… Với những lỗi chính tả tiếng Việt khó như d-gi, ch-tr thì gần như là không phát hiện được. Thêm vào đó, đôi khi chức năng kiểm lỗi chính tả tiếng Việt trong Word còn đưa ra những đề nghị sửa lỗi không hợp lý do từ điển chưa được đầy đủ.

### Những phần mềm kiểm lỗi chính tả khác

Cho tới nay, đã có một số chương trình đã được thương mại hóa và áp dụng vào thực tế. Có thể kể đến đó là VietSpell của Lưu Hà Xuyên, Cú Mèo Pro của Lê Anh Cường, hay CopCon của Mai Tuấn Khôi. Nhìn chung, các chương trình kiểm lỗi được tích hợp khá tốt cho các phần mềm soạn thảo văn bản và rất tiện lợi đối với người dùng. Tuy nhiên, hiệu quả của các phần mềm kiểm lỗi này chưa thực sự cao và chưa đáp ứng được đầy đủ những yêu cầu của người dùng.

# GIỚI THIỆU PHƯƠNG PHÁP

Nội dung chính của chương 4 sẽ trình bày chi tiết về mô hình ngôn ngữ N-gram và phương pháp áp dụng mô hình N-gram và phân đoạn từ vào kiểm lỗi chính tả tiếng Việt. Đồng thời, cũng trong chương này, chúng tôi sẽ nêu lên các bước để tạo một tập Corpus thô cho mô hình.

## Cơ sở lý thuyết

### Luật Bayes

Luật Bayes thường được sử dụng để tính xác suất có điều kiện.

**Công thức Bayes đảo**

Công thức Bayes cho hai biến cố A và B xảy ra đồng thời có dạng sau:

Công thức trên thường sử dụng khi một trong hai xác suất khó xác định.

**Công thức Bayes**

Gọi B1, B2, B3,… Bn là một hệ đầy đủ các biến cố nếu chúng từng đôi một xung khắc nhau (BiBj = Φ với i ≠ j) và hợp của chúng có một biến cố chắc chắn (biến cố có xác suất bằng 1), khi đó với mỗi biến cố A, ta có công thức sau:

Với A là biến cố có P(A) > 0, mỗi giá trị k (k=1,2,3…n) ta có công thức Bayes tổng quát sau:

Trong đó, P(B) được gọi là xác suất biết trước hay xác suất tiên nghiệm (prior probability), P(A) được gọi là xác suất cần tìm hay xác suất hậu nghiệm (posterior probability).

### Mô hình N-gram tổng quát

Mô hình N-gram là một mô hình ngôn ngữ rất phổ biến và 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 ngay phía trước nó [14]. Mô hình N-gram là những mô hình xác suất (tính toán dựa trên xác suất N-gram) được áp dụng trong các bài toán của xử lý ngôn ngữ tự nhiên.

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,… Mô hình N-gram được áp dụng cho các hệ thống tách từ, gán nhãn từ loại, kiểm lỗi chính tả, phát sinh câu ngẫu nhiên…

#### Ước lượng bằng N-gram

* Cho một câu đầu vào S gồm n từ ,, ,….
* Biểu diễn **N** từ như: ,, ,… bằng

Theo lý thuyết xác suất, xác suất P(S) của câu S sẽ được tính theo công thức Chain rule [14]:

Áp dụng công thức Chain rule cho dãy này ta có công thức tính xác suất của dãy. Công thức Chain rule chỉ ra mối liên kết giữa việc tính xác suất của một dãy và việc tính xác suất có điều kiện của một từ dựa vào dãy từ phía trước nó. Nó cũng chỉ ra việc tính xác suất kết hợp của một dãy từ bằng cách nhân các xác suất có điều kiện với nhau.

Tuy nhiên, phương pháp này rất khó áp dụng trong thực tế, bởi vì tính những xác suất dây chuyền như vậy đòi hỏi thông tin huấn luyện cực lớn và tiêu tốn rất nhiều thời gian xử lý.

Ví dụ: cho trước **h** = *Tôi đi*, **w** = *học.* Tính xác suất để từ *học* xuất hiện sau cụm từ *Tôi đi-* P(*học*|*Tôi đi*) => ***Làm sao có thể tính được xác suất này?***

Một cách là ước lượng, đánh giá nó dựa vào các tần suất liên quan. Ví dụ, chúng ta có 1 Corpus rất rộng, đếm số lần xuất hiện của *Tôi đi*, và số lần xuất hiện của *Tôi đi học.*

Khi đó:

P(*học*|*Tôi đi*) = C(*Tôi đi học*)/C(*Tôi đi*)

Với một Corpus đủ lớn, như web, chúng ta có thể tính số lần xuất hiện trên và ước lượng ra xác suất cần tính. Nhưng vấn đề nảy sinh là trong hầu hết các trường hợp chúng ta không có Corpus đủ lớn để giúp việc đánh giá xác suất này. Bởi vì, Ngôn ngữ là sự sáng tạo, các câu mới luôn được tạo ra, và chúng ta sẽ không thể nào đếm hết tổng số lượng các câu. Xét câu “*Tôi đi học*” các câu tương tự có thể sinh ra là:

* Hôm nay, tôi đi học.
* Hôm qua, tôi đi học.
* Ngày 1/11/2011, tôi đi học
* Ngày x tháng y năm z tôi đi học.
* Tôi đi học NLP
* Tôi đi học nhạc
* vv…vv…

Do những nguyên nhân trên, chúng ta cần được giới thiệu một phương pháp thông minh hơn giúp cho việc đánh giá **P(***w***|***h***)** hoặc xác suất của một dãy W.

***N-grams model thay bằng tính P(w|h), chúng ta xấp xỉ h bới một dãy ít số từ hơn.***

Trong bigram, thay vì sử dụng một dãy cho trước **h**, ta sử dụng một từ cuối cùng trong h (*đứng ngay trước w*) để ước lượng xác suất.

Ví dụ:

**P**(*học*|*Tôi đi*) ≈ **P**(*học*|*đi*)

Sự xấp xỉ xác suất của một từ chỉ phụ thuộc vào một từ ngày trước nó được gọi là xác suất Markov, Markov model là một lớp trong mô hình xác suất (probabilistic models) với ý tưởng có thể đoán từ mà không dựa vào quá nhiều từ phía trước nó [14].

Ta có:

* Unigram: h là rỗng
* Bigram: h là 1 từ ngay trước w
* Trigram: h là 2 từ ngay trước w
* N-grams: h là N-1từ ngay trước w

Xấp xỉ cho bigram [18] ta được:

Vậy với N < n, ta xấp xỉ:

Áp dụng xấp xỉ bigram ta có công thức tổng quát:

*Làm thế nào chúng ta có thể ước lượng, đánh giá xác suất bigram, N-gram?*

Cách đơn giản và trực quan nhất là phương pháp đánh giá xác xuất Maximum Likelihood Estimation (MLE) [14].

Với Bigram có công thức:

Ví dụ: sử dụng 1 mini-corpus với 3 câu. Trong đó mỗi câu bắt đầu bằng <s> và kết thúc bằng </s>

<s> Tôi thích ăn kem </s>

<s> Kem mà tôi thích là Sô-cô-la </s>

<s> Tôi là sinh viên </s>

P(Tôi|<s>)=2/3 P(kem|<s>) = 1/3 P(thích|Tôi)=2/3

P(</s>|kem)=1/2 P(kem|thích)=0/2 P(là|Tôi)=1/3

Trong trường hợp tổng quát ướng lượng MLE [14] cho N-gram có công thức là:

Phương pháp MLE sử dụng các tần suất có liên quan relative-frequency, mà kết quả của nó là cực đại hóa khả năng tập train T được cho bởi mô hình M (Ví dụ: P(T|M))



Ví dụ từ Vietnamese xuất hiện 500 lần trong 1 Corpus có 1000000 từ như bộ Corpus Brown. Câu hỏi đặt ra tính xác xuất khi lựa chọn ngẫu nhiên một từ trong một bộ dữ liệu text bất kỳ là từ Vietnamese? Phương pháp MLE ước lượng xác suất đó là 500/1000000 = 0.0005. Có thể 0.0005 không phải là ước lượng tốt nhất của xác suất xuất hiện từ Vietnamese trong tất cả các ngữ cảnh, ước lượng này có thể khác nhau ứng với mỗi corpus, nhưng xác suất này chỉ ra một cách gần đúng nhất cứ trong một bộ Corpus 1000000 từ thì từ Vietnamese xuất hiện 500 lần.

Ví dụ về tính xác suất theo bigram:

**P**(<s>Tôi thích ăn kem</s>)=**P**(Tôi|<s>)**P**(thích|Tôi)**P**(ăn|thích)**P**(kem|ăn)**P**(</s>|kem)

#### Tập train và tập test

Trong mô mình xác suất của NLP, tập Corpus được chia thành tập **train** và tập **test**.

Tập **train** dùng để “*huấn luyện*” cho mô hình xác suất, sau đó tập **test** dùng để “*kiểm tra mức độ chính xác”* của mô hình xác suất đã được huấn luyện.

Tập train và tập test có thể được dùng để đánh giá cho nhiều kiến trúc N-grams khác nhau. Giả sử chúng ta muốn so sánh các mô hình ngôn ngữ khác nhau (những mô hình dựa trên N-gram mà có thứ tự sắp xếp khác nhau hay sử dụng những giải thuật xấp xỉ khác nhau - smoothing algorithms) để từ đó chọn ra mô hình có mức độ chính xác cao hơn.

Việc đánh giá độ chính xác của mô hình dựa trên tập test, vì vậy điều quan trọng là không được đặt những câu thuộc tập test vào tập train. Nếu không mô hình sẽ mắc lỗi “***training on the test set***” [14]. Lỗi này làm cho những câu xuất hiện trong cả hai tập train và test luôn có xác suất cao, gây ra sự đánh giá không chính xác cho mô hình.

Để bổ sung cho tập train và tập test, một phần dữ liệu khác cũng cần được thêm vào. Nhiều khi, chúng ta cần thêm dữ liệu để bổ sung cho tập train, tập dữ liệu này gọi là tập **held-out**. Mặt khác, khi chúng ta cần có nhiều tập test (phục vụ cho các mục đích test khác nhau) chúng ta cũng cần làm mới tập test, thì phần dữ liệu làm mới này phải chưa được dùng trước đó. Phần dữ liệu dự trữ cho việc bổ sung này gọi là tập *development* hay được viết gọn là *devset* [14].

Như vậy vấn đề đặt ra là phải chia tập dữ liệu thành các tập *train*, *dev* và *test* như thế nào? Câu trả lời là cần có sự thỏa hiệp giữa các mục tiêu. Một mặt chúng ta cần tập test lớn (càng lớn thì mô hình càng được test nhiều), mặt khác chúng ta lại muốn tập train lớn nhất có thể (càng lớn thì mô hình càng học được nhiều). Trong thực hành người ta thường chia tập dữ liệu thành: ***80% cho tập train, 10% cho tập development và 10% cho tập test.***

Tuy nhiên, việc chia tập dữ liệu ra thành các phần trên phải đảm bảo mỗi tập con không mất đi tính chất của tập cha, mỗi tập con đều có tính chất ***“đại diện”*** cho tập cha. Trên thực tế, cần phải thỏa hiệp giữa các mục tiêu về lực lượng cũng như tính chất của mỗi tập con.

#### Ảnh hưởng của tập train tới các mô hình N-grams

Không thể phủ nhận tầm ảnh hưởng của training Corpus đối với mô hình N-gram. Về mặt lý thuyết, tập train có lực lượng dữ liệu càng lớn, chất lượng càng phong phú, đa dạng, sát với mục tiêu của mô hình ngôn ngữ thì mô hình càng “học” được nhiều, do đó tính chính xác của mô hình càng được nâng cao. Vì vậy, khi xây dựng tập train, cần đảm bảo về mặt **chất lượng** (“*tính đại diện*”) cũng như **số lượng** (*“lực lượng”*) cho tập dữ liệu.

#### Làm mịn

Do hoàn toàn dựa vào ngữ liệu nên phương pháp N-gram gặp một nhược điểm đó là khi dữ điệu không đủ độ dày cần thiết, các xác suất thu được sẽ rất nhỏ (thậm chí bằng 0) làm cho việc tính toán không còn chính xác.

Xét với trường hợp bigram trong công thức:

Trường hợp tồi tệ nhất xảy ra khi = 0 hoặc thậm chí .

Để khắc phục được nhược điểm này, một số phương pháp làm mịn đã được đề xuất. Dưới đây chúng tôi xin chỉ ra một số phương pháp làm mịn [14].

##### Phương pháp Add-one

Theo phương pháp này, tử số và mẫu số của được cộng thêm các giá trị thích hợp để đảm bảo chúng luôn khác 0. Trong phương pháp đề xuất tử sẽ được cộng thêm 1, mẫu được cộng thêm giá trị V, với V là tổng số từ có trong từ điển.

Như vậy, xác suất sẽ được tính lại như sau:

##### 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). Theo đó, xác suất được tính lại như sau:

##### 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:

Trong đó là tổng số N-gram xuất hiện r lần trong ngữ liệu

### Giới thiệu Corpus

Không chỉ trong mô hình N-gram mà trong rất nhiều bài toán xử lý ngôn ngữ tự nhiên bằng thống kê, Corpus là một tài nguyên không thể thiếu. Tùy vào từng bài toán và phương pháp giải quyết mà yêu cầu một loại Corpus thích hợp. Hiện nay, trên thế giới có những tổ chức chuyên cung cấp Corpus cho mục đích nghiên cứu hay phát triển phần mềm thương mại như LDC, OTA, v.v… nhưng hầu hết các ngôn ngữ có trong danh sách là những ngôn ngữ phổ biến như tiếng Anh, tiếng Pháp hay tiếng Hoa, còn đối với tiếng Việt thì xuất hiện rất ít. Như đã nói trong phần 1 của chương, chất lượng của tập train (80% của Corpus) ảnh hưởng rất lớn đến độ chính xác của hệ thống. Do đó việc xây dựng một Corpus chất lượng là điều hết sức quan trọng.

#### Corpus là gì?

Corpus là kho dữ liệu về ngôn ngữ, có thể ở dạng văn bản (bài báo, tin tức, tác phẩm văn học, v.v…) hay âm thanh (cuộc điện đàm, bài hát, bài báo tiếng, v.v…) [8]. Trong chương này sẽ đề cập đến loại Corpus được sử dụng nhiều nhất, đó là corpus văn bản.

#### Các bài toán trong quá trình tạo Corpus

##### Xử lý văn bản ở định dạng mức thấp

Ta có thể gặp các dạng dữ liệu khác nhau phụ thuộc vào nguồn Corpus. Do đó cần có công cụ thích hợp để lấy ra văn bản thô từ dữ liệu ban đầu. Ví dụ nguồn Corpus là Internet mà hầu hết các file chứa văn bản là HTML hoặc XML, khi đó ta sẽ cần chương trình đọc hai loại file này và lấy ra văn bản bên trong.

Sau khi đã lấy được văn bản, ta cần chuyển chúng thành một bộ mã thống nhất. Hiện tại, một trở ngại cho việc xử lý văn bản tiếng Việt là có quá nhiều bảng mã tiếng Việt. Mặc dù đã có Unicode, nhưng tiêu chuẩn này vẫn chưa được áp dụng triệt để và có tới hai bộ mã Unicode cho tiếng Việt là tổ hợp và dựng sẵn (TCVN 6909).

**Cắt câu**

Ta thường ngắt câu căn cứ vào các dấu hiệu như ‘.’, ‘?’, ‘!’. Tuy nhiên cách này chỉ cho kết quả gần đúng vì một số ký tự có nhiều chức năng, ví dụ như trong tiếng Anh, dấu chấm có thể là chấm câu, cũng có thể thuộc một chữ viết tắt (như Mr. John), ngoài ra còn nhiều loại từ tố chứa dấu chấm như địa chỉ Internet, tên file,... Do đó, bài toán cắt câu trong thu thập Corpus là không hề đơn giản. Chú ta cần xác định chính xác được những ký hiệu nào là ký hiệu kết thúc câu.

**Phân tích từ tố**

Phân tích từ tố là quá trình chia văn bản thành các đơn vị được gọi là từ tố (mỗi từ tố có thể là từ, dấu câu, số,…)

*Ví dụ:*

Câu vào: “Máy tính tôi cài window XP”

Kết quả: “máy”, “tính”, “tôi”, “cài”, “window”, “XP”

**Phân đoạn từ**

So với tiếng Anh thì bài toán phân đoạn từ cho tiếng Việt khó khăn hơn rất nhiều. Trong tiếng Anh, ranh giới giữa các từ được phân cách rõ ràng bằng các ký tự trống. Tuy nhiên, đối với tiếng Việt, không có ranh giới rõ ràng giữa các từ. Trong một câu, có rất nhiều khả năng tách từ khác nhau do sự đa dạng trong việc kết hợp giữa các từ của tiếng Việt. Do đó, bài toán phân đoạn từ trở nên khó khăn hơn rất nhiều.

Ví dụ: “Tốc độ truyền thông tin.”

Ta có các khả năng tách từ như sau:

+ Tốc / độ / truyền / thông / tin.

+ Tốc / độ / truyền / thông tin.

+ Tốc / độ / truyền thông / tin.

+ Tốc độ / truyền / thông / tin.

+ Tốc độ / truyền thông / tin.

*+* ***Tốc độ / truyền / thông tin.***

**Các bài toán khác**

Trong tiếng Anh, khi xử lý văn bản thô cũng cần phân tích hình thái từ. Tiếng Việt là ngôn ngữ đơn lập nên ta có thể bỏ qua vấn đề này.

Ngoài ra, bài toán phân tích danh từ riêng và từ viết tắt là bài toán khó. Danh từ riêng tiếng Việt có đặc điểm là viết hoa chữ cái đầu nên dễ nhận biết hơn so với danh từ riêng trong các ngôn ngữ tượng hình như tiếng Trung Quốc, Nhật, v.v…

*Ví dụ:*

“Theo số liệu *Bộ Giáo Dục Đào Tạo…* ”

“Ông Nguyễn Thanh Bình Chủ tịch *Ủy ban Nhân dân* tỉnhđến các thăm hộ nghèo”

Trong ví dụ trên các cụm danh từ riêng được viết nghiêng.

#### Các bước tạo Corpus cho hệ thống

1. **Download file text từ Internet**

Internet là một kho tài nguyên rất phong phú. Đó là nơi cung cấp nguồn tri thức tiếng Việt cực kỳ chất lượng. Nếu khai thác tốt, nó sẽ trở nên hữu ích cho công việc nghiên cứu xử lý ngôn ngữ tự nhiên. Có rất nhiều các công cụ giúp downoad các file định dạng html từ internet về máy tính. Trong quá trình thu thập dữ liệu, chúng tôi sử dụng 2 công cụ để thu thập các file định dạng html là teleport và internet download manager. Những file html được thu thập từ những nguồn tiếng Việt uy tín trên internet và phải đạt yêu cầu chính xác và đầy đủ thể loại: văn hóa, xã hội, khoa học,… để đảm bảo chất lượng cho Corpus của mình.

1. **Lấy text**

Sau khi thu thập được các file html, bước tiếp theo cần làm là bóc tách các thẻ html để lấy được nội dung của nó. Trong bước này, có thể sử dụng biểu thức chính quy để bắt các đoạn văn bản có trong file html.

1. **Chuẩn hóa**

Công việc chuẩn hóa bao gồm các giai đoạn:

* Chuyển mã tiếng Việt
* Lọc bỏ những đoạn text kém chất lượng
* Xử lý nhiễu trong Corpus
* Chuẩn hóa thanh, dấu
* Chuẩn hóa các địa chỉ email, cụm số, địa chỉ web để phù hợp với việc tính toán.

Bước này là bước hết sức quan trọng. Bởi xử lý được tốt bước này, Corpus của chúng ta sẽ trở nên chất lượng hơn và dễ dàng sử dụng hơn.

Download các trang web trên Internet

Website trên đĩa cứng

Lấy text

Chuẩn hóa

Corpus thô

File text sạch

Các file text

Công cụ: idm, teleport

Công cụ: Biểu thức chính quy, xpath

Hình 4: Sơ đồ các bước tạo corpus

### 

### Thuật toán Viterbi

Thuật toán Viterbi mang tên tác giả Andrew Viterbi [4], là một thuật toán quy hoạch động nhằm tìm dãy tương tự nhất của các trạng thái ẩn, được gọi là đường đi Viterbi – cho ra kết quả là dãy các sự kiện gắn kết. Thuật toán được ứng dụng rộng rãi trong các phương pháp máy trạng thái hữu hạn, đặc biệt với bài toán trích chọn thông tin. Thuật toán tiến “forward” là thuật toán đi kèm với thuật toán Viterbi thực hiện việc tính toán xác suất của một dãy các sự kiện gắn kết.

Thuật toán Viterbi được đánh giá như một sơ đồ chỉnh sửa lỗi trong các đường truyền thông số. Thuật toán cũng rất phổ dụng trong lý thuyết thông tin, nhận dạng tiếng nói, ngôn ngữ tính toán, tin – sinh học,… Thuật toán cung cấp một cách thức hiệu quả để tìm ra dãy trạng thái giống nhất theo nghĩa làm cực đại xác suất hậu nghiệm từ một quá trình được giả thiết là một quá trình Markov thời gian rời rạc trạng thái hữu hạn (finite-state discrete-time Markov). Những quá trình như vậy được gộp vào khung thống kê của ***lý thuyết quyết định hỗn hợp***.[4]

**Lý thuyết quyết định hỗn hợp**

Giả sử ta có một văn bản gồm n ký tự. Mỗi ký tự mang tới một vector đặc trưng zi, i = 1,2,3…,n.

Ký hiệu P(Z|C) là hàm mật độ xác suất của dãy con vector Z = z1, z2, z3,… zn phụ thuộc có điều kiện dãy các định danh C = c1, c2, c3,… cn trong đó zk là vector đặc trung đối với ký tuej thứ k, còn ck lấy giá trị trên M giá trị ( số các chữ cái trong bảng chữ) với k = 1,2,3…,n.

Ký hiệu P(C) là xác suất tiền định của dãy các giá trị C, nói khác đi, P(C) là phân bố xác suất tiền định của tất cả các dãy có n ký tự. Xác suất phân lớp chính xác văn bản được cực đại hóa nhờ chọn dãy ký tự mà có xác suất hậu nghiệm cực đại hay còn được gọi là cực đại hậu nghiệm (MAP) xác suất P(C|Z).

Từ quy tắc xác suất có điều kiện Bayes, ta có:

P(C|Z) = (1)

Do P(Z) là độc lập đối với dãy C (P(Z) chỉ là một thừa số vô hướng), nên chỉ cần làm cực đại hàm phân biệt:

gc = P(Z|C)P(C) (2)

Dung lượng lưu trữ đối với các xác suất này là khổng lồ về thực tế, vì vậy cần có các giả thiết nhằm rút gọn bài toán tới kích thước có thể quản lý được. Các giả thiết sau đây được đặt ra:

* Độ dài của dãy các quan sát không lớn. Gọi n là độ dài của một từ khóa. Do đó, P(C) là tần suất xuất hiện của các từ khóa.
* Tính độc lập có điều kiện giữa các vector đặc trưng. Hình dạng của ký tự sinh ra vector đặc trưng đã cho là độc lập với hình dạng của các ký tự kề cận, và như vậy, chỉ phụ thuộc ký tự trong truy vấn.

Do các giả thiết này và các điều đã được giải thích, công thức (2) được rút gọn thành:

gc(Z) = logP(c1…cn)

## Giới thiệu phương pháp

Mục tiêu cuối cùng của khóa luận là một chương trình kiểm lỗi chính tả có thể áp dụng ngay trong thực tế, đáp ứng nhu cầu soát lỗi chính tả của người sử dụng. Để đạt được mục tiêu đó, hệ thống phải đáp ứng những yêu cầu đó là sử dụng một mô hình đơn giản, xử lý nhanh, sử dụng ít bộ nhớ và điều quan trọng là hiệu suất phải chấp nhận được khi thử nghiệm với nhiều loại văn bản khác nhau.

Sau khi xem xét những yêu cầu trên và đánh giá tính khả thi, chúng tôi quyết định chọn N-gram tiếng làm hướng tiếp cận chính cho hệ thống kiểm lỗi chính tả của mình. Như đã nói, tuy N-gram tiếng không hẳn là một mô hình tối ưu nhất cho kiểm lỗi chính tả tiếng Việt, nhưng nó đáp ứng được những yêu cầu hiện tại đó là không đòi hỏi một kho ngữ liệu phải gán nhãn, đồng thời cũng không cần quá nhiều những xử lý phức tạp và đòi hỏi không gian nhớ lớn. Hiệu suất của mô hình sau khi được cải tiến bằng một số thuật toán theo những thử nghiệm ban đầu của chúng tôi thì hoàn toàn có thể chấp nhận được.

### Mô hình tổng quát

Toàn bộ hệ thống kiểm lỗi chính tả của chúng tôi gồm 2 phần: khối tiền xử lý, khối xử lý chính và hậu xử lý.

Hình 5: Mô hình tổng quát của chương trình

Chuẩn hóa

Tách tiếng

Tách câu

Lọc nhiễu

Tạo tập nhầm lẫn âm tiết

Tìm lỗi và sửa lỗi

So sánh và lấy kết quả

So sánh và lấy kết quả

* Khối tiền xử lý
  + Chuẩn hóa văn bản đầu vào.
  + Tách văn bản đầu vào thành những câu riêng biệt, là những chuỗi văn bản nằm giữa những ký tự kết thúc câu.
  + Tách những câu thành từng tiếng riêng biệt.
  + Lọc ra các cụm tiếng Anh, tên riêng, các từ nước ngoài, tiếng viết tắt, cụm số, ký hiệu đặc biệt … và đánh dấu để xử lý riêng.
* Khối xử lý chính
  + Xây dựng tập nhầm lẫn cho các âm tiết không được đánh dấu (không phải là tiếng Anh, tên riêng, cụm số ...).
  + Tìm ra dãy âm tiết thích hợp nhất.
* Khối hậu xử lý
  + Kết hợp câu đầu vào và dãy âm tiết thích hợp tìm được để đưa ra kết quả.

#### Khối tiền xử lý

1. **Chuẩn hóa văn bản**

Văn bản đầu vào có thể là một tệp tin có thể có những đối tượng “lạ” không phải là text như ảnh, link, mục lục,... Do đó, việc đầu tiên là phải loại bỏ những đối tượng trên và chỉ giữ lại mỗi text.

1. **Tách câu**

Sau khi chuẩn hóa được văn bản đầu vào, bước tiếp theo trong giai đoạn tiền xử lý là tách văn bản thành những câu riêng biệt. Bước này nhằm 2 mục đích: tạo tiền đề cho bước tách tiếng phía sau và quan trọng hơn là để tính các xác suất N-gram. Do các tiếng bị ngăn cách bởi các dấu câu xem như không có mối quan hệ gì với nhau nên có thể giảm được không gian xử lý mà không ảnh hưởng đến hiệu quả của quá trình kiểm lỗi.

1. **Tách tiếng**

Các câu sẽ được tách thành các tiếng riêng biệt để xử lý. Trong bước này, câu sẽ được chuẩn hóa các dấu cách và các ký tự trống trước khi tách để đảm bảo việc tách tiếng chính xác hơn.

1. **Lọc nhiễu**

Sau khi đã phân tách được các tiếng riêng biệt, bước tiếp theo là đánh dấu các tiếng nhiễu. Trong các văn bản hiện nay thường sử dụng một số từ tiếng Anh. Hệ thống phải lọc ra và đánh dấu những nhập nhằng này để xử lý tính toán N-gram. Ngoài ra, hệ thống còn phải lọc ra những cụm số, những tên riêng, địa chỉ email, địa chỉ web, các từ viết tắt… Bước xử lý này cũng rất quan trọng bởi nó có thể ảnh hưởng đến tính toán xác suất của N-gram. Ngoài ra, trong bước này người dùng có thể đánh dấu những từ ngữ riêng của mình để đảm bảo rằng từ đó là đúng chính tả và không cần sửa lỗi.

#### Khối xử lý chính

1. **Xây dựng tập nhầm lẫn**

Dựa vào đặc trưng ngữ pháp tiếng việt và những điều tra về chính tả tiếng Việt, hệ thống sẽ xây dựng tập nhầm lẫn âm tiết tương ứng với các âm tiết không bị đánh dấu trong câu. Các âm tiết trong tập nhầm lẫn là được sinh theo 2 dạng lỗi: do phát âm và do đánh máy. Trong giai đoạn này, hệ thống sẽ chọn ra những âm tiết dễ bị nhầm lẫn nhất và có tần suất kết hợp với các tiếng xung quanh cao hơn so với âm tiết gốc và gán trọng số cho từng âm tiết đó. Với những âm tiết dễ bị nhầm lẫn, trọng số sẽ được đánh số cao hơn so với những âm tiết khó bị nhầm lẫn. Ngoài ra, mỗi người dùng sẽ có những loại lỗi riêng tùy vùng miền và tùy từng thói quen. Do đó hệ thống có thể giúp người dùng xây dựng tập lỗi của riêng mình tùy vào loại lỗi họ thường gặp.

Các bước thực hiện bao gồm:

* Tách tiếng ra thành các phần : phụ âm, nguyên âm, dấu.
* Xây dựng tập nhầm lẫn cho các thành phần.
* Xây dựng tập nhầm lẫn do đánh máy.
* Lựa chọn những âm tiết trong các tập nhầm lẫn đã xây dựng có trong từ điển âm tiết tiếng Việt và có khả năng nhầm lẫn cao nhất đưa vào tập nhầm lẫn của tiếng và đánh trọng số cho các âm tiết đó.

1. **Tìm lỗi và sửa lỗi**

Ý tưởng để phát hiện lỗi ở đây là sử dụng mô hình N-gram và phân đoạn từ để đo độ liên kết của một tiếng với các tiếng khác lân cận. Tiếng nào có độ liên kết lỏng lẻo sẽ được xem là bị lỗi chính tả. Cách làm này rất phù hợp với tư duy của con người. Bởi khi đọc một câu để kiểm lỗi, con người thường xem những vị trí “khác thường” không liên quan đến với các phần khác ngay cạnh nó. Sau đó, sẽ tiến hành sinh âm tiết khác để sửa lỗi cho câu nghe có vẻ hợp lý hơn.

Sau khi đã có được tập nhầm lẫn âm tiết của các tiếng, ta áp dụng thuật toán Viterbi để tìm ra dãy tiếng có xác suất lớn nhất làm câu sửa lỗi. Các xác suất ở đây được ước lượng và so sánh bằng phương pháp N-gram kết hợp phân đoạn từ. Ở trong bước này, phải sử dụng một số chỉ số gọi là *“ngưỡng sửa lỗi*”. Âm tiết trong tập nhầm lẫn sẽ được tính toán theo trọng số và chỉ được lựa chọn để thay thế khi nó làm cho xác suất của câu lớn hơn một *“ngưỡng sửa lỗi”* nào đó. *“Ngưỡng sửa lỗi”* được chúng tôi lựa chọn sau khi xem xét và đánh giá dựa trên tập development. Ở đây có điểm khác với thuật toán gốc CInsunSpell đó là khi tìm được ứng viên thay thế có xác suất tốt hơn, hệ thống chưa vội kết luận ngay âm tiết gốc bị sai chính tả. Bởi trong điều kiện thực tế, ngữ liệu có thể không đủ lớn để bao trùm hết tất cả các N-gram thường có. Do đó sẽ xảy ra trường hợp âm C có xác suất nhỏ hơn âm tiết C’ trong tập nhầm lẫn nhưng vẫn đúng chính tả.

Hình 6: Ví dụ kiểm lỗi bằng phương pháp đề xuất

Về lý thuyết, ước lượng xác suất trigram sẽ tốt hơn hẳn bigram. Tuy nhiên, khi dùng trigram, với lượng ngữ liệu không đủ, các xác suất sẽ rất thấp nên ảnh hưởng rất lớn đến việc đánh giá. Qua thử nghiệm của chúng tôi, mô hình ước lượng bằng cách kết hợp giữa bigram và trigram cho kết quả rất khả quan. Theo mô hình đó, xác suất sẽ được ước lượng bằng công thức sau:

Trong đó k và m là các hệ số và k + m =1.

Xác suất của câu sẽ được ước lượng bằng công thức:

Với công thức ước lượng xác suất trên sẽ đảm bảo được độ chính xác của mô hình trigram trong trường hợp các xác suất quá nhỏ.

#### Khối hậu xử lý

Sau khi đã có được dãy âm tiết tốt nhất, ở trong khối hậu xử lý này sẽ so sánh với câu đầu vào, đánh giá và chỉ ra những âm tiết bị sai chính tả cũng như đưa ra gợi ý sửa lỗi cho những âm tiết đó.

### Hạn chế lỗi tích cực bằng Heuristic

#### Gán trọng số cho tập nhầm lẫn âm tiết

Các âm tiết trong tập nhầm lẫn được gán trọng số theo khả năng nhầm lẫn của âm tiết đó trong thực tế. Tập lỗi của người dùng được gán trọng số lớn nhất. Tiếp theo đó là các loại lỗi phát âm (ch/tr,s/x…), dấu (hỏi/ngã) và cuối cùng là các loại lỗi đánh máy sai. Theo cách đánh trọng số đó, những loại lỗi thường gặp trong thực tế sẽ được ưu tiên thay thế hơn. Điều này cũng hợp với tâm lý người kiểm lỗi, đó là ưu tiên những trường hợp sai hay gặp nhất để sửa lỗi trước.

#### Gán trọng số dựa trên từ điển

Ngoài việc sử dụng ước lượng xác suất bằng trigram và bigram để đo độ gắn kết, chúng tôi còn sử dụng heuristic “bám theo từ điển” để hạn chế lỗi tích cực. Nguyên nhân vẫn là việc Corpus chưa đủ lớn. Qua khảo sát cho thấy, một số từ ghép, đặc biệt là các thuật ngữ chuyên ngành, không xuất hiện đủ số lượng cần thiết trong Corpus. Điều này dẫn đến việc mô hình sẽ đánh giá rằng từ ghép có độ liên kết yếu và bị sai chính tả mặc dù nó hoàn toàn đúng và có sự kết hợp chặt chẽ với các tiếng còn lại. Nhưng ngược lại, nếu chúng ta bỏ qua tất cả các từ ghép thì sẽ xuất hiện những hiện tượng sai từ ghép như: “xum xuê” 🡪 “sum suê”, “điều hành” 🡪 “diễu hành”,…

Để giải quyết vấn đề trên, trong khi phân đoạn từ và ước lượng, chúng tôi tăng trọng số của những cụm tiếng có trong từ điển tiếng Việt lên một ngưỡng nào đó. Điều này sẽ làm tăng độ gắn kết của các từ ghép lên nhưng vẫn không bỏ qua nếu độ gắn kết đó chưa đạt được ngưỡng cho phép.

### Làm mịn

Như đã nói ở trên, việc Corpus không đủ rộng khiến cho một số xác suất trở nên quá nhỏ (thậm chí bằng 0) và rất khó để tính toán. Trong việc ước lượng xác suất bằng kết hợp mô hình trigram và bigram, chúng tôi sử dụng phương pháp làm mịn cộng hệ số. Nó chưa hẳn là phương pháp làm mịn tối ưu nhất. Tuy nhiên, phương pháp đó lại áp dụng rất nhanh trong hệ thống mà vẫn cho được kết quả khả quan.

### Học trong quá trình sử dụng

Trong quá trình sử dụng, người dùng sẽ *“dạy”*cho hệ thống những từ nào thuộc từ điển riêng của người dùng và những lỗi người dùng thường gặp. Hệ thống sẽ *“học”* và lưu lại để những lần sử dụng sau, người dùng không cần phải *“dạy”* lại cho hệ thống những trường hợp cũ.

# CÀI ĐẶT VÀ THỰC NGHIỆM

Trong chương này sẽ trình bày khái quát về quá trình cài đặt chương trình kiểm lỗi chính tả theo mô hình đề xuất ở chương 4. Cuối chương là phần thực nghiệm, so sánh kết quả chương trình của chúng tôi với một số chương trình kiểm lỗi chính tả khác.

## Cài đặt

Từ mô hình lý thuyết trên, chúng tôi đã tiến hành cài đặt chương trình kiểm lỗi chính tả. Chương trình được viết trên ngôn ngữ Visual C++. Ngôn ngữ VC++ có ưu điểm là tốc độ thực hiện rất tốt và đảm bảo được tính bảo mật cao. Chương trình đã được thực nghiệm thành công trên môi trường Window XP, Window 7, Window Vista.

Các bước để cài đặt hệ thống gồm:

* Tạo Corpus thô.
* Đếm tần suất N-gram và tổ chức lưu trữ.
* Viết chương trình.

### Tạo Corpus thô

Như đã trình bày các bước tạo Corpus thô cho hệ thống ở chương 4. Chúng tôi đã tiến hành thu thập và tạo Corpus. Dưới đây là kết quả của việc thu thập corpus:

|  |  |
| --- | --- |
| Độ lớn của Corpus | 1Gb |
| Số từ của Corpus | 144777805 |
| Số lượng bi-gram khác nhau | 861575 |
| Số lượng tri-gram khác nhau | 2720488 |

Bảng 3: Thông số trong Corpus

### Đếm tần suất N-gram và tổ chức lưu trữ

Sau khi thu thập được Corpus thô, chúng tôi tiến hành đếm tần suất các bigram, trigram và lưu trữ.

Một bigram sẽ được lưu trữ bằng 2 số kiểu integer. Số đầu tiên lưu vị trí các âm tiết cấu tạo thành nên bigram theo thứ tự: 16 bit đầu lưu trữ vị trí của âm tiết thứ 2 trong từ điển unigram, 16 bit tiếp theo lưu trữ vị trí của âm tiết đầu tiên của bigram trong từ điển unigram. Số thứ 2 lưu trữ tần suất của bigram đó.

Hình 7: Mô hình lưu trữ bigram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 31 | … | 16 | 15 | … | 0 |

|  |  |
| --- | --- |
| Tiếng thứ nhất | Tiếng thứ hai |

Bit

Nội dung

Một trigram sẽ được chia thành 2 phần: phần đầu bao gồm 2 âm đầu, phần thứ 2 là âm tiết cuối. Phần đầu được tổ chức lưu trữ như bigram và âm tiết cuối được lưu trữ bằng một số kiểu short. Cuối cùng là một số kiểu Integer để lưu trữ tần suất của trigram.

Cách thức tổ chức lưu trữ trên sẽ làm giảm đáng kể bộ nhớ sử dụng để lưu trữ dữ liệu của chương trình, đồng thời làm tăng tốc độ tìm kiếm của một bigram và trigram trong dữ liệu.

Công việc cuối cùng trong bước này là mã hóa dữ liệu, nhằm đảm bảo được tính bảo mật và độ an toàn cho dữ liệu, tránh việc sửa chữa, chọc phá khiến cho dữ liệu bị sai dẫn đến lỗi cho chương trình.

### Viết chương trình

Bước cuối cùng trong việc cài đặt là viết chương trình, trình tự thực hiện theo mô hình đã nêu ở chương 4. Chương trình yêu cầu phải viết tối ưu bộ nhớ, các bước xử lý, đồng thời phải xây dựng được giao diện thân thiện, dễ dàng sử dụng và có khả năng phát triển.

## Thực nghiệm

Sau khi cài đặt xong chương trình, chúng tôi đã tiến hành thực nghiệm cho chương trình trên các dữ liệu thực nghiệm thu thập được. Dưới đây là các kết quả thực nghiệm và so sánh kết quả của chương trình của chúng tôi với chương trình kiểm lỗi chính tả CopCon của Mai Tuấn Khôi và VietSpel2000 của Lưu Hà Xuyên.

Hệ thống được đánh giá thông qua ba độ đo: độ chính xác (precision), độ hồi tưởng (recall) và độ đo F (F-measure). Ba độ đo này được tính toán theo công thức sau:

Trong đó:

|  |  |
| --- | --- |
| **Giá trị** | **Ý nghĩa** |
|  | Số lỗi phát hiện đúng |
|  | Số lỗi phát hiện sai |
|  | Số lỗi bỏ qua |
|  | Tổng số lỗi |

Bảng 4: Bảng các giá trị trong các công thức tính độ đo

### Môi trường và dữ liệu thực nghiệm

#### Môi trường thực nghiệm

|  |  |
| --- | --- |
| Bộ xử lý | Intel® 2 Core™ CPU P7350 2.0 GHz |
| RAM | 3.0 GB |
| Hệ điều hành | Microsoft Windows 7 |
| Công cụ lập trình | Visual C++ |

Bảng 5: Môi trường thực nghiệm

#### Xây dựng dữ liệu thực nghiệm

#### Dữ liệu thực nghiệm loại 1

Bước đầu tiên của xây dựng dữ liệu thực nghiệm loại 1 là thu thập các đoạn văn từ trên các trang báo uy tín của Việt Nam như dantri.com.vn, chinhphu.vn, tuoitre.vn, nhandan.vn,… Ở loại dữ liệu này, yêu cầu đặt ra là phải đa thể loại và được viết đúng chuẩn chính tả. Sau khi thu thập được dữ liệu (khoảng 1000 đoạn văn, kích thước 500kb), tiến hành gây lỗi nhân tạo và gán nhãn lỗi trên tập dữ liệu thu thập với tất cả các loại lỗi thường gặp để tạo thành dữ liệu thực nghiệm loại 2.

Dưới đây là bảng các âm tiết bị gây lỗi:

|  |  |
| --- | --- |
| **Âm tiết** | **Dạng sai** |
| Sách | xách, ách |
| Trong | chong, trông |
| cập | vập |
| mạng | mãng, mang |
| Năm | nam, lăm |
| thường | thưởng, thương |
| lớn | lờn |
| lợi | lơi, lời |
| Doanh | danh |
| chương | trương |
| Công | cong |
| Chúng | trúng, chung |
| sắp | xắp |
| Đã | đa, đả |
| Đang | dang |
| Bán | ban, ván |
| Khách | khác, hách |
| biến | biên, viên |
| Nhiên | nhin, hiên |
| nhiễm | nhĩm |
| tầng | từng, từn |
| Tranh | chanh, tran |

Bảng 6: Các âm tiết được gây lỗi trong dữ liệu thực nghiệm loại 1

1. **Dữ liệu thực nghiệm loại 2**

Ở dữ liệu thực nghiệm loại 2, chúng tôi thu thập một số đoạn văn trên các trang báo. Sau đó, kiểm tra thủ công xem phát hiện những âm tiết nào mắc lỗi và gán nhãn lỗi cho âm tiết đó.

VD: Cao điểm vào khoảng 10h sáng có đến vài trăm người dân cùng phương tiện kéo đến *tại* ra cảnh “vỡ chợ” tại cây xăng.

### Kết quả thực nghiệm

Sau khi cài đặt xong chương trình, chúng tôi đã tiến hành thực nghiệm cho chương trình trên các dữ liệu thực nghiệm thu thập được. Dưới đây là kết quả của quá trình thực nghiệm.

1. **Dữ liệu thực nghiệm loại 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Âm tiết** | **Số lỗi** | **Số lỗi phát hiện** | **Sửa lỗi đúng** |
| Sách | 68 | 68 (100%) | 66 (97.1%) |
| Trong | 414 | 394 (95.2%) | 394 (100%) |
| Cập | 3 | 3 (100%) | 3 (100%) |
| Năm | 200 | 165 (82.5%) | 165 (100%) |
| Thường | 42 | 31 (73.8%) | 30 (96.7%) |
| Lớn | 27 | 27 (100%) | 27 (100%) |
| Lợi | 40 | 38 (95%) | 37 (97.4%) |
| Doanh | 40 | 39 (97.5%) | 39 (100%) |
| Chương | 15 | 15 (100%) | 15 (100%) |
| Công | 110 | 108 (98.2%) | 108 (100%) |
| Chúng | 48 | 34 (70.8%) | 34 (100%) |
| Sắp | 1 | 1 (100%) | 1 (100%) |
| Đã | 323 | 255 (79%) | 254 (99.6%) |
| Đang | 39 | 35 (89.7%) | 35 (100%) |
| Bán | 41 | 36 (87.8%) | 34 (94.4%) |
| Khách | 37 | 30 (81.1%) | 30 (100%) |
| Biến | 15 | 15 (100%) | 15 (100%) |
| Nhiên | 23 | 23 (100%) | 23 (100%) |
| Nhiễm | 3 | 3 (100%) | 3 (100%) |
| Tầng | 24 | 21 (87.5%) | 21 (100%) |
| Tranh | 42 | 42 (100%) | 42 (100%) |
| **Total** | 1570 | 1398 (89%) | 1391(99.5%) |

Bảng 7: Kết quả thực nghiệm loại 1

* 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 chính xác (pre): 90.8%
* Độ hồi tưởng (rec): 89%
* Độ đo F: 89.9%

Hình 8: Kết quả thực nghiệm dữ liệu loại 1

1. **Dữ liệu thực nghiệm loại 2**

Đối với dữ liệu thực nghiệm loại 2, chúng tôi thu thập những đoạn văn ở các báo khác nhau để làm dữ liệu thực nghiệm. Sau khi tiến hành thực nghiệm và đem so sánh với phần mềm kiểm lỗi Copcon v5.3 (Mai Tuấn Khôi) và VietSpell2000 v2.51 (Lưu Hà Xuyên). Kết quả đạt được như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Số lỗi** | **Số lỗi phát hiện** | **Số lỗi phát hiện đúng** | **Số lỗi sửa lỗi đúng** | **Số lỗi phát hiện sai** |
| Chương trình của chúng tôi | 22 | 21 | 20 | 19 | 1 |
| Copcon v5.3 | 22 | 16 | 14 | 14 | 2 |
| Vietspell2000 | 22 | 27 | 15 | 14 | 12 |

Bảng 8: Kết quả thực nghiệm dữ liệu loại 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Độ chính xác (pre)** | **Độ hồi tưởng (rec)** | **Độ đo F** |
| Chương trình của chúng tôi | 95.2% | 90.9% | 93% |
| Copcon v5.3 | 77.8% | 63.6% | 70% |
| Vietspell2000 | 55.6% | 68.2% | 61.3% |

Bảng 9: Kết quả thực nghiệm dữ liệu loại 2 qua các độ đo

Hình 9: So sánh kết quả thực nghiệm loại 2

Hình 10: So sánh kết quả thực nghiệm loại 2 qua các độ đo

Ví dụ kiểm lỗi của chương trình kiểm lỗi chính tả của khóa luận trong dữ liệu thực nghiệm loại 2:

Cao điểm vào khoảng 10h sáng có đến vài trăm người dân cùng phương tiện kéo đến tại<tạo> ra cảnh “ vỡ chợ ” tại cây xăng .

Đề án kiểm soát tải trọng xe được bộ GTVT gửi các địa phương và bộ ngành để lấy ý kiến trước khi trinh<trình> chính phủ.

### Đánh giá

Quá trình thực nghiệm loại 1 cho thấy chương trình kiểm lỗi chính tả của khóa luận đạt được kết quả kiểm lỗi khá cao (khoảng 90%). Sau khi so sánh kết quả thực nghiệm loại 2 với các chương trình kiểm lỗi VietSpell200 và Copcon qua các độ đo, chương trình kiểm lỗi của chúng tôi cho thấy được kết quả tốt hơn hẳn. Tuy rằng đây chỉ là những thực nghiệm đầu tiên của chương trình, nhưng nó hứa hẹn chương trình sẽ đạt được thành công khi áp dụng vào thực tế.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**Kết luận**

Xuất phát từ nhu cầu thực tế tại Việt Nam, trong bài nghiên cứu khoa học này, chúng tôi đã tiến hành nghiên cứu về lĩnh vực kiểm lỗi chính tả tiếng Việt, tìm hiểu và đánh giá các mô hình kiểm lỗi chính tả đã có. Đồng thời giới thiệu phương pháp sử dụng mô hình N-gram vào kiểm lỗi chính tả dựa trên những đặc trưng cơ bản của tiếng Việt.

Vì mục tiêu chính của khóa luận là tính thực tế, do đó mô hình ở đây không được sử dụng quá nhiều bộ nhớ. Thêm vào đó, chương trình phải xử lý trong khoảng thời gian ngắn mà vẫn phải cho kết quả tốt. So sánh với các tiêu chí trên, chương trình kiểm lỗi chính tả của chúng tôi áp dụng mô hình N-gram không quá phức tạp, không đòi hỏi quá nhiều không gian lưu trữ, cho kết quả tốt trong một khoảng thời gian chấp nhận được. Ngoài ra, chương trình của chúng tôi cũng đã được xây dựng thành một ứng dụng khá hoàn thiện, giúp cho người dùng có thể kiểm lỗi một đoạn văn hoặc một tập tin văn bản tiếng Việt.

Trong bước đầu cài đặt và thực nghiệm, kết quả đạt được của chương trình kiểm lỗi chính tả của chúng tôi áp dụng mô hình N-gram vào kiểm lỗi chính tả là rất khả quan. Chương trình soát được lỗi với hiệu suất khá tốt với bộ nhớ sử dụng và thời gian thực hiện là chấp nhận được đối với tập dữ liệu thực nghiệm khách quan và thuộc nhiều thể loại. Kết quả của quá trình kiểm lỗi đối với dữ liệu thực nghiệm là tốt hơn đối với một số chương trình kiểm lỗi khác.

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**

Như đã trình bày ở trên, do việc áp dụng mô hình N-gram tiếng và việc thiếu ngữ liệu, hiệu suất của chương trình vẫn chưa thực sự tốt đối với với những văn bản “lạ”. Vì vậy, hướng phát triển đầu tiên của chúng tôi đó là xây dựng các bộ ngữ liệu ứng với từng chủ đề khác nhau (hành chính, khoa học, văn hóa,…). Với mỗi văn bản đầu vào tương ứng, chương trình sẽ đánh giá và lựa chọn một bộ ngữ liệu phù hợp với văn bản đề tăng hiệu suất của chương trình lên.

Hướng phát triển tiếp theo trong tương lai, chúng tôi dự định sẽ cải tiến phương pháp, kết hợp một số phương pháp khác để đạt được độ chính xác cao hơn và tối ưu được bộ nhớ sử dụng hơn. Đồng thời, nghiên cứu nhúng hệ thống kiểm lỗi chính tả vào các bộ gõ (unikey, vietkey,…) và chương trình soạn thảo văn bản để có thể được áp dụng rộng rãi và dễ dàng sử dụng.

TÀI LIỆU THAM KHẢO

**Tài liệu tiếng Việt**

|  |  |
| --- | --- |
| [1] | Châu Hải Duy, *Cách tiếp cận dựa trên ngữ liệu cho kiểm lỗi chính tả tiếng Việt,* khóa luận tốt nghiệp, ĐH KHTN TPHCM, 2005. |
| [2] | 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. |
| [3] | Đinh Điền, *Giáo trình xử lý ngôn ngữ tự nhiên,* ĐH KHTN TPHCM, 2004. |
| [4] | Hà Quang Thụy, Phan Xuân Hiếu, Đoàn Sơn, Nguyễn Trí Thành, Nguyễn Thu Trang, Nguyễn Cẩm Tú, *Giáo trình khai phá dữ liệu Web,* NXB Giáo Dục, 2009. |
| [5] | Hoàng Phê. 1999. Chính tả tiếng Việt. Nhà xuất bản Đà Nẵng. |
| [6] | Lê Trung Hoa, *Lỗi chính tả và cách khắc phục,* NXB Khoa học xã hội, 2002. |
| [7] | 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 |
| [8] | Nguyễn Phương Thái, *Kiểm lỗi Chính tả Cảm Ngữ cảnh Tiếng Việt*, Luận văn thạc sĩ, Khoa Công nghệ, 2003. |
| [9] | 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. |

**Tài liệu tiếng Anh**

|  |  |
| --- | --- |
| [10] | 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. |
| [11] | 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. |
| [12] | Andrew R.Golding, *A Bayesian hybrid methods for context-sensitive correction,* Proceedings of the Third Workshop on Very Large Copora, 1995. |
| [13] | Dinh Dien, Hoang Kiem, Nguyen Van Toan, *Vietnamese word segmentation*, NLPRS, 2001. |
| [14] | Jurafsky-Martin, *Speech and Language Processing 2ed,* 2006. |
| [15] | Li Jianhua and Wang XiaoLong, *Combine trigram and Automatic Weight Distribution in Chinese Spelling ErrorCorrection,* 2000. |
| [16] | Masaaki Nagata, *A Stochastic Japanese Morphological Analyzer Using a Forward-BD Backward-A\*N-best search Algorithm,* In Proceedings of COLING-94, 1994. |
| [17] | Masaaki Nagata, *Context-Based Spelling Correction for Japanese OCR,* 1998. |
| [18] | Raymond J. Mooney, *Natural Language Processing: NGram Language Models*, University of Texas at Austin. |

**Công cụ sử dụng**

[Regexbuddy] *Công cụ biểu thức chính qui,* Just Great Software Co. Ltd.

[IDM v5.18] *Phần mềm Download trang web,* Internet Download Manager.

[Teleport] *Phần mềm Download trang web,* Tennyson Maxwell Information Systems.

[Copcon v5.3] *Phần mềm kiểm lỗi chính tả,* [www.Chinhta.vn](http://www.Chinhta.vn).

[VietSpell v2.51] *Phần mềm kiểm lỗi chính tả,* Lưu Hà Xuyên.