

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC**

HOÀNG CÔNG DUY VŨ - NGUYỄN LÊ NGUYỄN

**TÌM KIẾM VĂN BẢN TIẾNG VIỆT
THEO CHỦ ĐỀ**

KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT

TP. HCM, 2006

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ TRI THỨC**

**HOÀNG CÔNG DUY VŨ - 0212384
NGUYỄN LÊ NGUYÊN - 0212203**

**TÌM KIẾM VĂN BẢN TIẾNG VIỆT
THEO CHỦ ĐỀ**

KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN

**T.S Nguyễn Đình Thúc
Văn Chí Nam**

KHÓA 2002 - 2006

LỜI CẢM ƠN

Sau một thời gian làm việc cật lực, đến nay, mọi công việc liên quan đến luận văn tốt nghiệp đã hoàn tất. Trong suốt thời gian này, chúng tôi đã nhận được rất nhiều sự giúp đỡ. Ở phần đầu tiên của luận văn, cho phép chúng tôi có đôi điều gửi đến những người chúng tôi vô cùng biết ơn.

Đầu tiên, xin gửi lời cảm ơn chân thành nhất đến Thầy Nguyễn Đình Thúc, Thầy Văn Chí Nam, Thầy Đinh Diên những người đã tận tình hướng dẫn, động viên, và giúp đỡ chúng em trong suốt thời gian qua. Nếu không có những lời chỉ dẫn, những tài liệu, ngữ liệu, những lời động viên khích lệ của các Thầy thì luận văn này khó lòng hoàn thiện được.

Chúng con cũng xin cảm ơn cha mẹ, những người đã luôn dành cho chúng con những tình cảm thương yêu nhất, những người đã luôn hỗ trợ, dõi theo những bước đi của chúng con trong tất cả các năm học vừa qua. Xin cảm ơn các anh chị trong gia đình đã động viên chúng em trong suốt thời gian làm luận văn.

Chúng em cũng xin gửi lời cảm ơn tất cả các Thầy Cô trong khoa Công Nghệ Thông Tin, những người dày công dạy dỗ, truyền cho em rất nhiều tri thức quý báu suốt 4 năm theo học.

Cảm ơn các bạn, các anh chị trong nhóm **VCL¹** vì những đóng góp của các bạn, các anh cho luận văn này. Đặc biệt xin gửi lời cảm ơn chân thành đến với anh Ngô Quốc Hưng cho những công cụ phục vụ luận văn và những góp ý cho chương trình.

Và cuối cùng xin cảm ơn tất cả bạn bè của chúng tôi, những người đã sát cánh cùng vui những niềm vui, cùng chia sẻ những khó khăn của chúng tôi, cùng chúng tôi giải quyết nhiều khó khăn suốt 4 năm học đại học... Xin chân thành cảm ơn!

Hoàng Công Duy Vũ - Nguyễn Lê Nguyên

¹ Vietnamese Computational Linguistics

MỤC LỤC

MỤC LỤC	<i>i</i>
DANH MỤC HÌNH	<i>vii</i>
DANH MỤC BẢNG	<i>ix</i>
DANH MỤC BIỂU ĐỒ	<i>xi</i>
Chương 1: MỞ ĐẦU	1
1.1 <i>Đặt vấn đề</i>	1
1.2 <i>Giới thiệu đê tài</i>	2
1.3 <i>Giới thiệu sự ra đời của hệ thống tổng hợp thông tin từ báo chí cho mục đích an toàn sinh học cộng đồng</i>	2
1.4 <i>Mô hình tổng quan của dự án</i>	5
1.5 <i>Vai trò của hệ thống khai thác thông tin văn bản (text mining system) trong dự án</i>	5
1.6 <i>Mục tiêu thực hiện của luận văn</i>	8
1.6.1 <i>Tìm hiểu các thuật toán phân loại văn bản</i>	8
1.6.2 <i>Xây dựng ứng dụng tìm kiếm văn bản theo chủ đề</i>	8
1.7 <i>Đóng góp của luận văn</i>	9
1.8 <i>Bố cục của luận văn</i>	9
Chương 2: TỔNG QUAN	11
2.1 <i>Bài toán tách từ</i>	11
2.1.1 <i>Các vấn đề trong bài toán tách từ</i>	11
2.1.1.1 <i>Xử lý nhập nhằng [11]</i>	11
2.1.1.2 <i>Nhận diện từ chưa biết</i>	12
2.1.2 <i>Các hướng tiếp cận chính cho bài toán tách từ</i>	12
2.1.3 <i>Tách từ tiếng Việt dùng mô hình WFST</i>	12
2.1.3.1 <i>Mô hình WFST</i>	13

2.1.3.2	<i>Biểu diễn từ điển</i>	14
2.1.3.3	<i>Phân tích hình thái</i>	15
2.1.3.4	<i>Mô hình mạng Neuron</i>	16
2.1.4	<i>Tách từ tiếng Việt dùng mô hình Maximum Matching</i>	18
2.1.5	<i>Tách từ tiếng Việt dùng mô hình MMSeg</i>	19
2.1.5.1	<i>Thuật toán MM và các biến thể của nó</i>	19
2.1.5.2	<i>Các luật khử nhầm nhằng (Ambiguity Resolution Rules)</i>	20
2.1.6	<i>Tách từ tiếng Việt dùng mô hình Maximum Entropy</i>	21
2.1.6.1	<i>Ý tưởng của phương pháp</i>	21
2.1.6.2	<i>Mô hình thuật toán [11]</i>	22
2.1.6.3	<i>Ước lượng bộ tham số cho mô hình [11]</i>	25
2.2	<i>Bài toán phân loại văn bản</i>	27
2.2.1	<i>Một số khái niệm cơ bản</i>	27
2.2.2	<i>Tổng quan về bài toán phân loại văn bản tự động trên tiếng Anh</i>	28
2.2.3	<i>Các phương pháp tiếp cận cho bài toán</i>	30
2.2.4	<i>Phân loại văn bản tiếp cận theo hướng dãy các từ (Bag of words – BOW based Approach) [25]</i>	30
2.2.4.1	<i>Phương pháp xác suất Naïve Bayes</i>	30
2.2.4.2	<i>Phương pháp phân loại k người láng giềng gần nhất</i>	32
2.2.4.3	<i>Phương pháp sử dụng mạng nơron</i>	35
2.2.4.4	<i>Phương pháp phân loại văn bản bằng cây quyết định</i>	39
2.2.4.5	<i>Phân loại văn bản bằng phương pháp hồi quy</i>	40
2.2.4.6	<i>Phân loại văn bản sử dụng Support Vector Machines – SVM</i>	42
2.2.4.7	<i>Những phương pháp khác</i>	48
2.2.5	<i>Phân loại văn bản tiếp cận theo hướng mô hình ngôn ngữ thống kê N-Gram (Statistical N-Gram Language modeling based Approach) [26]</i>	48
2.2.6	<i>Tiếp cận theo hướng kết hợp 2 loại trên (Combining approach) [27]</i>	49
2.2.7	<i>Tổng quan về bài toán phân loại văn bản trên tiếng Việt</i>	51
2.2.7.1	<i>Phân loại văn bản tiếng Việt bằng phương pháp Naïve Bayes [21]</i>	52

2.2.7.2	<i>SVM - Ứng dụng lọc email [22]</i>	53
2.2.7.3	<i>Ứng dụng lý thuyết tập thô trong bài toán phân loại văn bản [23]</i>	56
2.2.7.4	<i>Phân tích các ưu khuyết điểm trong bài toán phân loại văn bản tiếng Việt</i>	58
Chương 3: CƠ SỞ LÝ THUYẾT	59
3.1	<i>Lý thuyết ngôn ngữ cho bài toán tách từ tiếng Việt [11]</i>	59
3.1.1	<i>Khái niệm về từ</i>	59
3.1.2	<i>Hình thái từ tiếng Việt</i>	60
3.1.2.1	<i>Hình vị tiếng Việt</i>	60
3.1.2.2	<i>Từ tiếng Việt</i>	61
3.2	<i>Cơ sở lý thuyết về văn bản, phân loại văn bản</i>	62
3.2.1	<i>Khái niệm văn bản</i>	62
3.2.2	<i>Khái niệm phân lớp</i>	63
3.2.3	<i>Khái niệm phân loại văn bản</i>	63
3.2.3.1	<i>Phân loại văn bản đơn nhã và đa nhã</i>	64
3.2.3.2	<i>Phân loại văn bản phụ thuộc lớp/loại văn bản so với phụ thuộc tài liệu</i>	65
3.2.3.3	<i>Phân loại văn bản “cứng” so với “mềm”</i>	65
3.2.3.4	<i>Các ứng dụng của phân loại văn bản [2]</i>	66
Chương 4: MÔ HÌNH – THIẾT KẾ – CÀI ĐẶT	67
4.1	<i>Chuẩn bị ngữ liệu</i>	67
4.2	<i>Kiến trúc tổng quát của hệ thống (The General Architecture)</i>	70
4.2.1	<i>Các module chính của hệ thống</i>	71
4.2.2	<i>Các chức năng chính của hệ thống</i>	71
4.3	<i>Module phân loại tài liệu (Vietnamese Document Classification Module)</i> ..	71
4.3.1	<i>Mô hình tổng quát (General Model)</i>	71
4.3.2	<i>Cách tiếp cận dựa trên dãy các từ (The BOW-based Approach)</i>	73

4.3.2.1	<i>Bài toán tách từ tiếng Việt</i>	73
4.3.2.2	<i>Tiền xử lý văn bản tiếng Việt</i>	79
4.3.2.3	<i>Chọn lựa đặc trưng</i>	81
4.3.2.4	<i>Xây dựng bộ phân lớp</i>	87
4.3.3	<i>Tiếp cận theo hướng mô hình ngôn ngữ thống kê</i>	87
4.3.3.1	<i>Tiền xử lý văn bản tiếng Việt</i>	87
4.3.3.2	<i>Xây dựng mô hình ngôn ngữ</i>	88
4.3.3.3	<i>Sử dụng mô hình Naïve Bayes kết hợp với mô hình ngôn ngữ thống kê n-gram</i>	91
4.3.3.4	<i>Các lợi ích của mô hình ngôn ngữ</i>	91
4.3.4	<i>Lọc và tìm kiếm tài liệu</i>	92
4.4	<i>Thiết kế cài đặt</i>	93
4.4.1	<i>Thiết kế cài đặt thư viện tách từ</i>	93
4.4.1.1	<i>Sơ đồ lớp</i>	93
4.4.1.2	<i>Cài đặt</i>	93
4.4.2	<i>Thiết kế cài đặt module phân loại văn bản</i>	95
4.4.2.1	<i>Sơ đồ lớp</i>	95
4.4.2.2	<i>Cài đặt</i>	95
4.4.3	<i>Thiết kế cài đặt ứng dụng tìm kiếm văn bản tiếng Việt theo chủ đề</i> ...	100
4.4.3.1	<i>Sơ đồ lớp</i>	100
4.4.3.2	<i>Giao diện ứng dụng</i>	101
4.4.3.3	<i>Cài đặt</i>	103
Chương 5: KẾT QUẢ THỰC NGHIỆM	104
5.1	<i>Bài toán tách từ tiếng Việt</i>	104
5.1.1	<i>Thí nghiệm</i>	104
5.1.2	<i>Đánh giá</i>	105
5.1.3	<i>Kết quả</i>	106
5.1.4	<i>Nhận xét</i>	107
5.2	<i>Bài toán phân loại văn bản tiếng Việt</i>	107

5.2.1	<i>Thí nghiệm</i>	107
5.2.1.1	<i>Định nghĩa các giá trị độ đo</i>	108
5.2.1.2	<i>Các mô hình dùng trong bài toán phân loại văn bản:</i>	109
5.2.2	<i>Dữ liệu thô (10 chủ đề)</i>	109
5.2.2.1	<i>So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 3 phương pháp chọn đặc trưng OCFS, CHI, GSS</i>	109
5.2.2.2	<i>So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm tron” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell</i>	110
5.2.2.3	<i>So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram</i>	112
5.2.2.4	<i>So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng chọn lựa với mô hình SVM-Multi</i>	113
5.2.2.5	<i>Mô hình N-gram với phương pháp “làm tron” (discounting smoothing method) Good Turing</i>	113
5.2.3	<i>Dữ liệu mịn (27 chủ đề)</i>	114
5.2.3.1	<i>So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 6 phương pháp chọn đặc trưng OCFS, CHI, GSS, IG, OR, MI</i>	114
5.2.3.2	<i>So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm tron” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell</i>	116
5.2.3.3	<i>So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram</i>	118
5.2.3.4	<i>So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng chọn lựa với mô hình SVM-Multi</i>	119
5.2.3.5	<i>Mô hình N-gram với phương pháp “làm tron” (discounting smoothing method) Good Turing</i>	120
5.2.3.6	<i>So sánh kết quả kiểm nghiệm giữa hướng tiếp cận của chúng tôi và hướng tiếp cận Naïve Bayes</i>	120
5.2.3.7	<i>Nhận xét</i>	121

Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	124
6.1 <i>Kết luận.....</i>	<i>124</i>
6.2 <i>Hướng phát triển</i>	<i>125</i>
TÀI LIỆU THAM KHẢO.....	126
<i>Phụ lục 1. Bảng kết quả thử nghiệm trên dữ liệu thô 10 chủ đề</i>	<i>134</i>
<i>Phụ lục 2. Bảng kết quả thử nghiệm trên dữ liệu mìn 27 chủ đề</i>	<i>139</i>

DANH MỤC HÌNH

Hình 1: Mô hình tổng quan của dự án BioCaster	5
Hình 2: Kiến trúc của dự án BioCaster.....	6
Hình 3: Mô hình chi tiết các module	7
Hình 4: Qui trình của mô hình WFST	13
Hình 5: Đồ thị trọng số biểu diễn cung từ điển	15
Hình 6: Đồ thị WFST biểu diễn từ điển.....	16
Hình 7: Mô hình mạng noron dùng cho khử nhập nhằng câu	17
Hình 8: Mô hình thuật toán Maximum Entropy	22
Hình 9: Quy trình của bài toán phân loại văn bản	29
Hình 10: Mô hình phân loại văn bản Naive Bayes.....	32
Hình 11: Mạng Perception	36
Hình 12: Minh họa phân lớp	36
Hình 13: Không gian mặt phẳng với x_p	43
Hình 14: Ý tưởng mặt phân cách.....	43
Hình 15: Minh họa SVs	44
Hình 16: Quá trình học	54
Hình 17: Quá trình phân lớp	55
Hình 18: Hình ảnh minh họa cách lấy thông tin từ web	67
Hình 19: Kiến trúc của hệ thống tìm kiếm trong luận văn	70
Hình 20: Mô hình thuật toán	72
Hình 21: Kiến trúc của mô hình kết hợp	74
Hình 22: Sơ đồ lớp ứng dụng tách từ	93
Hình 23: Sơ đồ lớp cài đặt ứng dụng phân loại văn bản	95
Hình 24: Sơ đồ lớp ứng dụng tìm kiếm văn bản tiếng Việt theo chủ đề	100
Hình 25: Màn hình Splash.....	101
Hình 26: Giao diện màn hình chính ứng dụng	101
Hình 27: Giao diện màn hình chức năng Offline	102
Hình 28: Giao diện màn hình chức năng online	102

DANH MỤC BẢNG

Bảng 1: Ví dụ về nhãn vị trí tiếng trong từ.....	24
Bảng 2: Kết quả thử nghiệm trên tập huấn luyện.....	55
Bảng 3: Kết quả thử nghiệm trên tập kiểm nghiệm.....	55
Bảng 4: Mô tả ngũ liệu.....	56
Bảng 5: Kết quả đánh giá.....	57
Bảng 6: Mô tả ngũ liệu dạng thô.....	68
Bảng 7: Mô tả ngũ liệu dạng mịn.....	69
Bảng 8: Thông tin mô tả vị trí của tiếng trong 1 từ.....	74
Bảng 9: Thống kê ngũ liệu CADASA.....	75
Bảng 10: Khung đặc trưng sử dụng cho mô hình SVM.....	77
Bảng 11: Quá trình sửa sai	78
Bảng 12: Tóm tắt các phương pháp chọn lựa đặc trưng.....	83
Bảng 13: Thống kê trên ngũ liệu QTAG.....	104
Bảng 14: Kết quả thực nghiệm so sánh các mô hình tách từ	106
Bảng 15: Kết quả thực nghiệm so sánh với các phương pháp trước	106
Bảng 16: Kết quả trung bình 10 chủ đề với 2500 terms.....	109
Bảng 17: Kết quả trung bình 10 chủ đề với 5000 terms.....	110
Bảng 18: Kết quả trung bình 10 chủ đề với $N = 2$	110
Bảng 19: Kết quả trung bình 10 chủ đề với $N = 3$	111
Bảng 20: Kết quả trung bình 10 chủ đề với $N = 4$	111
Bảng 21: Kết quả trung bình 10 chủ đề với 4 mô hình (2500 terms, $N = 2$).....	112
Bảng 22: Kết quả trung bình 10 chủ đề với 4 mô hình (5000 terms, $N = 2$).....	112
Bảng 23: Kết quả trung bình 10 chủ đề với số lượng đặc trưng khác nhau	113
Bảng 24: Kết quả trung bình 10 chủ đề với N khác nhau	113
Bảng 25: Kết quả trung bình 27 chủ đề với 2500 terms.....	114
Bảng 26: Kết quả trung bình 27 chủ đề với 5000 terms.....	115
Bảng 27: Kết quả trung bình 27 chủ đề với 7500 terms.....	115
Bảng 28: Kết quả trung bình 27 chủ đề với $N = 2$	116
Bảng 29: Kết quả trung bình 27 chủ đề với $N = 3$	116

Bảng 30: Kết quả trung bình 27 chủ đề với $N = 4$	117
Bảng 31: Kết quả trung bình 27 chủ đề với 4 mô hình (2500 terms, $N = 2$).....	118
Bảng 32: Kết quả trung bình 27 chủ đề với 4 mô hình (5000 terms, $N = 2$).....	118
Bảng 33: Kết quả trung bình 27 chủ đề với 4 mô hình (7500 terms, $N = 2$).....	119
Bảng 34: Kết quả trung bình 27 chủ đề với số lượng đặc trưng khác nhau	119
Bảng 35: Kết quả trung bình 27 chủ đề với N khác nhau	120
Bảng 36: Kết quả chi tiết 4 chủ đề giữa Bayes và SVM	121
Bảng 37: Kết quả trung bình 4 chủ đề giữa Bayes và SVM	121
Bảng 38: Kết quả chi tiết 10 chủ đề với 2500 terms	134
Bảng 39: Kết quả chi tiết 10 chủ đề với 5000 terms	134
Bảng 40: Kết quả chi tiết 10 chủ đề với $N = 2$	135
Bảng 41: Kết quả chi tiết 10 chủ đề với $N = 3$	135
Bảng 42: Kết quả chi tiết 10 chủ đề với $N = 4$	136
Bảng 43: Kết quả chi tiết 10 chủ đề với 4 mô hình (2500 terms, $N = 2$)	136
Bảng 44: Kết quả chi tiết 10 chủ đề với 4 mô hình (5000 terms, $N = 2$)	137
Bảng 45: Kết quả chi tiết 10 chủ đề với số lượng đặc trưng khác nhau	137
Bảng 46: Kết quả chi tiết 10 chủ đề với N khác nhau	138
Bảng 47: Kết quả chi tiết 27 chủ đề với 2500 terms	139
Bảng 48: Kết quả chi tiết 27 chủ đề với 5000 terms	140
Bảng 49: Kết quả chi tiết 27 chủ đề với 7500 terms	141
Bảng 50: Kết quả chi tiết 27 chủ đề với $N = 2$	142
Bảng 51: Kết quả chi tiết 27 chủ đề với $N = 3$	143
Bảng 52: Kết quả chi tiết 27 chủ đề với $N = 4$	144
Bảng 53: Kết quả chi tiết 27 chủ đề với 4 mô hình (2500 terms, $N = 2$)	145
Bảng 54: Kết quả chi tiết 27 chủ đề với 4 mô hình (5000 terms, $N = 2$)	146
Bảng 55: Kết quả chi tiết 27 chủ đề với 4 mô hình (7500 terms, $N = 2$)	147
Bảng 56: Kết quả chi tiết 27 chủ đề với số lượng đặc trưng khác nhau	148
Bảng 57: Kết quả chi tiết 27 chủ đề với N khác nhau	149

DANH MỤC BIỂU ĐỒ

<i>Biểu đồ 1: So sánh hệ thống với các mô hình tách từ</i>	106
<i>Biểu đồ 2: So sánh hệ thống với các phương pháp trước</i>	107
<i>Biểu đồ 3: So sánh kết quả trung bình 10 chủ đề với 2500 terms</i>	109
<i>Biểu đồ 4: So sánh kết quả trung bình 10 chủ đề với 5000 terms</i>	110
<i>Biểu đồ 5: So sánh kết quả trung bình 10 chủ đề với N = 2</i>	111
<i>Biểu đồ 6: So sánh kết quả trung bình 10 chủ đề với N = 3</i>	111
<i>Biểu đồ 7: So sánh kết quả trung bình 10 chủ đề với N = 4</i>	112
<i>Biểu đồ 8: So sánh kết quả trung bình 10 chủ đề 4 mô hình (2500 terms, N=2)</i> ..	112
<i>Biểu đồ 9: So sánh kết quả trung bình 10 chủ đề 4 mô hình (5000 terms, N= 2)</i> ..	113
<i>Biểu đồ 10: So sánh kết quả trung bình 10 chủ đề với số đặc trưng khác nhau</i>	113
<i>Biểu đồ 11: So sánh kết quả trung bình 10 chủ đề với N khác nhau</i>	114
<i>Biểu đồ 12:So sánh kết quả trung bình 27 chủ đề với 2500 terms</i>	114
<i>Biểu đồ 13: So sánh kết quả trung bình 27 chủ đề với 5000 terms</i>	115
<i>Biểu đồ 14: So sánh kết quả trung bình 27 chủ đề với 7500 terms</i>	115
<i>Biểu đồ 15: So sánh kết quả 27 chủ đề với N = 2</i>	116
<i>Biểu đồ 16: So sánh kết quả trung bình 27 chủ đề với N = 3</i>	117
<i>Biểu đồ 17: So sánh kết quả trung bình 27 chủ đề với N = 4</i>	117
<i>Biểu đồ 18: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (2500 terms, N = 2)</i>	118
<i>Biểu đồ 19: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (5000 terms, N = 2)</i>	119
<i>Biểu đồ 20: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (7500 terms, N = 2)</i>	119
<i>Biểu đồ 21: So sánh kết quả trung bình 27 chủ đề với số đặc trưng khác nhau</i>	120
<i>Biểu đồ 22: So sánh kết quả trung bình 27 chủ đề với N khác nhau</i>	120
<i>Biểu đồ 23: So sánh kết quả trung bình 4 chủ đề giữa Bayes và SVM</i>	121

Chương 1: MỞ ĐẦU

Nội dung

Chương này sẽ giới thiệu khái quát về dự án **BioCaster**² và vai trò của bài toán “tìm kiếm văn bản tiếng Việt theo chủ đề” trong dự án này. Từ đó, nêu lên mục tiêu và ý nghĩa của đề tài đối với bài toán tìm kiếm thông tin tiếng Việt nói riêng và các bài toán khác trong ngành xử lý ngôn ngữ tự nhiên nói chung.

1.1 Đặt vấn đề

Ngày nay, sự phát triển vượt bậc của công nghệ thông tin đặc biệt là sự bùng nổ của mạng internet, lượng thông tin được số hóa và đưa lên mạng ngày càng nhiều. Internet trở thành một kho kiến thức khổng lồ về mọi lĩnh vực. Do đó, số lượng văn bản xuất hiện trên mạng internet cũng tăng theo với tốc độ chóng mặt. Hiện tại, số lượng trang web mà Google đã chỉ mục lên đến hơn 3 tỉ trang [1], đó là chưa kể đến các văn bản được lưu trữ trên đó. Ngoài ra, các nghiên cứu [1] cũng chỉ ra rằng các trang web thay đổi rất nhanh, tăng gấp đôi sau 9-12 tháng.

Với lượng thông tin khổng lồ như vậy, một vấn đề nảy sinh là làm thế nào để tổ chức thông tin và tìm kiếm đạt hiệu quả cao nhất. Hơn nữa, với nhu cầu thực tế của người sử dụng, tìm kiếm thông tin theo những chủ đề chỉ định là một thách thức thật sự. Từ đó, bài toán “tìm kiếm văn bản theo chủ đề” trở thành một giải pháp hợp lý cho nhu cầu trên.

Đứng trước tình hình đó, cụ thể là dự án lớn **BioCaster** do chính phủ Nhật tài trợ, mong muốn của họ là xây dựng 1 ứng dụng có thể khai thác thông tin đặc biệt là tiếng Việt từ một số lượng lớn thông tin thu thập được từ web, chẳng hạn: thông tin về tình hình cúm gia cầm, thông tin về an toàn giao thông, thông tin về tình hình chiến tranh Với số lượng thông tin lớn như vậy, nhu cầu cần thiết là 1 hệ thống có thể tự động tìm kiếm thông tin theo những chủ đề đã chọn. Các thông tin tìm kiếm được trở thành đầu vào cho các nhu cầu xử lý tiếp theo.

² Dự án lớn do chính phủ Nhật tài trợ <http://www.nii.ac.jp/openhouse/abstract/outline.shtml>

Chúng em quyết định chọn đề tài “Tìm kiếm văn bản tiếng Việt theo chủ đề”, với mục tiêu xây dựng một ứng dụng nhằm giải quyết bài toán tìm kiếm nói trên. Ứng dụng này sẽ giúp giảm thời gian và công sức của con người hơn là tìm kiếm thủ công. Đây cũng là vấn đề đầu tiên mà dự án **BioCaster** nói trên cần phải thực hiện.

1.2 **Giới thiệu đề tài**

Đề tài “Tìm kiếm văn bản tiếng Việt theo chủ đề” được chúng em xây dựng dựa trên nền tảng tốc độ xử lý ưu việt của máy tính so với tốc độ tìm kiếm thủ công của con người. Bằng cách cho máy tính học một số tri thức về ngôn ngữ của con người, máy tính sẽ trở thành công cụ hữu hiệu trong việc tìm kiếm các văn bản theo những chủ đề đã được lựa chọn.

Tuy nhiên để máy tính có thể học được một số tri thức về ngôn ngữ của con người không phải là điều đơn giản. Để tìm kiếm được văn bản theo chủ đề, máy tính cần phải có những tri thức có đê cập đến những thông tin của chủ đề mong muốn hay không. Các tri thức này được rút trích từ các văn bản biết trước của các chủ đề muốn tìm kiếm. Nhưng với số lượng văn bản lớn, làm thế nào để rút trích được những tri thức cần thiết, và với các tri thức đó làm thế nào để chúng ta phân loại thông tin theo chủ đề, đó cũng là các vấn đề chính của luận văn này.

1.3 **Giới thiệu sự ra đời của hệ thống tổng hợp thông tin từ báo chí cho mục đích an toàn sinh học cộng đồng**

Tổ chức y tế thế giới **WHO** dự đoán sẽ có trên **7.4** triệu người trên thế giới thiệt mạng vì đại dịch cúm gia cầm trong tương lai. Thông tin này được cảnh báo nhằm để tránh xảy ra một trận đại dịch tương tự đại dịch **SARS**, một căn bệnh truyền nhiễm theo đường hô hấp có khả năng lây từ người sang người, đã xảy ra trên nhiều nước thông qua đường hàng không.

Trong lịch sử thế giới, năm 1918 dịch cúm tại Tây Ban Nha đã lây lan thành đại dịch làm 20 triệu người chết, gây ảnh hưởng đến 20 đến 40% dân số thế giới. Năm 1957 tại châu Á và năm 1968 tại Hồng Kông, dịch cúm đã quay trở lại. Tuy

nhiên hậu quả đã được giảm thiểu nhờ sự theo dõi cẩn thận và tiêm phòng vaccine miễn dịch. Tuy vậy hằng năm những dịch nhỏ hơn vẫn xảy ra đã gây ra nhiều lo ngại và thiệt hại cho nhiều nước. Trong khi đó tại những nơi mà thời gian là vàng trong việc ngăn chặn cái chết và sự lây lan rộng của dịch bệnh, yêu cầu tiên quyết nhất là các cơ quan đầu não cần phải đưa ra quyết định chính xác chỉ trong thời gian ngắn. Tuy vậy điều quan trọng đầu tiên để có thể ngăn chặn sự lây lan là phải dự báo được tình hình bệnh dịch và điều kiện cần thiết đầu tiên cho công việc dự báo là phải được cập nhật thông tin chính xác, kịp thời.

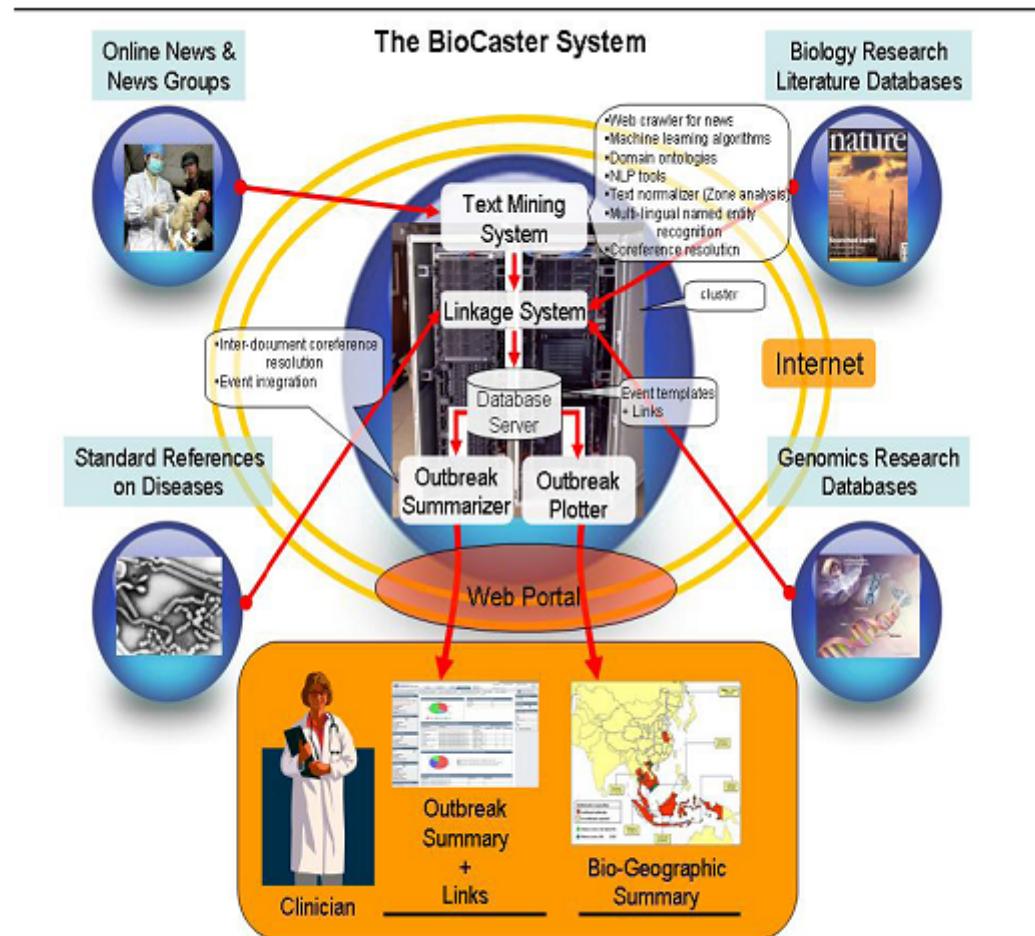
Có hai khó khăn tồn tại trong cách thu thập thông tin truyền thống: **thứ nhất** đó chính là số lượng thông tin quá đồ sộ trên Internet mà các thông tin có thể trùng lắp làm cho các cơ quan lãnh đạo cũng như các tổ chức chăm sóc sức khỏe cộng đồng khó xây dựng nên được một bức tranh toàn cảnh về sự bùng nổ của đại dịch; **thứ hai** là lượng thông tin thu thập từ các phương pháp tìm kiếm truyền thống, tìm kiếm theo từ khóa thường rất ít. Tiếp đó, những thông tin được công bố thường thông qua các phương tiện truyền thông nên ngôn ngữ sử dụng không có sự “giao thoa”, giữa các nước khác nhau không có một bản đồ thảo luận cụ thể. Chính vì lẽ đó, dự án xây dựng nhằm tạo nên một hệ thống có thể ghi nhận thông tin bùng nổ đại dịch chung cho nhiều nước tại châu Á.

Hiện nay, trên thế giới đang có xu hướng sử dụng phương pháp khai thác thông tin văn bản xây dựng các hệ thống lớn để thu thập các tài nguyên tri thức đặc biệt như các thông tin đặc điểm, các cơ sở dữ liệu,... Ví dụ như hệ thống rút trích kết quả từ các tạp chí sinh học (dự án **GENIA**) hay thông tin bệnh án từ các bản khai của người bệnh (dự án **MedLEE**). Khi đại dịch **SARS** diễn ra, nhiều chuyên gia của **WHO** đã đồng ý rằng việc sớm phát triển sự bùng nổ của dịch là nhân tố chính giúp chống lây lan dịch và giảm thiểu tử vong. Chính vì lẽ đó, nhiều nhóm nghiên cứu ở Bắc Mỹ đã bắt đầu phát triển hệ thống dự báo điện tử nhằm kiểm soát dữ liệu về sức khỏe cộng đồng bằng cách thống kê lượng thuốc tiêu thụ và từ các báo cáo của các bác sĩ điều trị. **Ví dụ** ở Canada có hệ thống **GPHIN** hay hệ thống **MiTAP** từ **MITRE**.

Cùng xu thế đó, mục đích chính của dự án là áp dụng những thí nghiệm về khai thác thông tin văn bản (**text mining**) để phát triển một hệ thống thông tin thông minh gọi là **BioCaster** nhằm dự báo và bảo vệ sức khỏe cộng đồng trước những đại dịch có nguy cơ bùng phát. Hệ thống sẽ liên tục ghi nhận một số lượng lớn tin tức của các vùng miền trên các quốc gia thuộc khu vực châu Á. Sau đó các tin tức này sẽ được tổng hợp và dịch sang tiếng Anh để có thể chia sẻ thông tin cho nhiều tổ chức. Từ hệ thống khai thác thông tin này, những thông tin quan trọng liên quan đến đại dịch như: triệu chứng, người bệnh, địa điểm, thời gian diễn ra, những loại thuốc... sẽ được người sử dụng nhanh chóng nắm bắt thông qua các bản tin tức trong thời gian nhanh nhất. Những ngôn ngữ được sử dụng sẽ là tiếng Anh, tiếng Nhật, tiếng Thái và tiếng Việt. Hệ thống sẽ được xây dựng và phát triển với mục đích tận dụng sức mạnh máy tính trong việc xử lý một số lượng đồ sộ thông tin thu thập.

Kết quả cuối cùng của dự án đặt ra là sẽ xây dựng một cổng thông tin **BioCaster** trên nền tảng Web dựa vào hệ thống khai thác thông tin từ văn bản và các ứng dụng liên quan đến xử lý ngôn ngữ tự nhiên. Những chủ đề quan tâm trên các báo điện tử sử dụng **RSS** sẽ được tập hợp thành một danh sách và sẽ được chuyển từ dạng Web thành file lưu trữ trên hệ thống đĩa cứng. Tiếp đó hệ thống khai thác thông tin văn bản sẽ phân tích những file này theo thời gian thực để đưa ra các thông tin liên quan đến việc phát hiện dịch như các hiện tượng đơn lẻ hay các hiện tượng có dấu hiệu nổ ra dịch lớn. Trong dự án này, hệ thống cũng sẽ xây dựng một định dạng thống nhất chung cho những tin tức mà không thể thu thập thông qua **RSS**. Và một trong những kết quả cuối cùng sẽ là một bản đồ điện tử mô tả thông tin dịch bệnh được cập nhật theo thời gian thực được đưa đến người sử dụng

1.4 Mô hình tổng quan của dự án



Hình 1: Mô hình tổng quan của dự án BioCaster

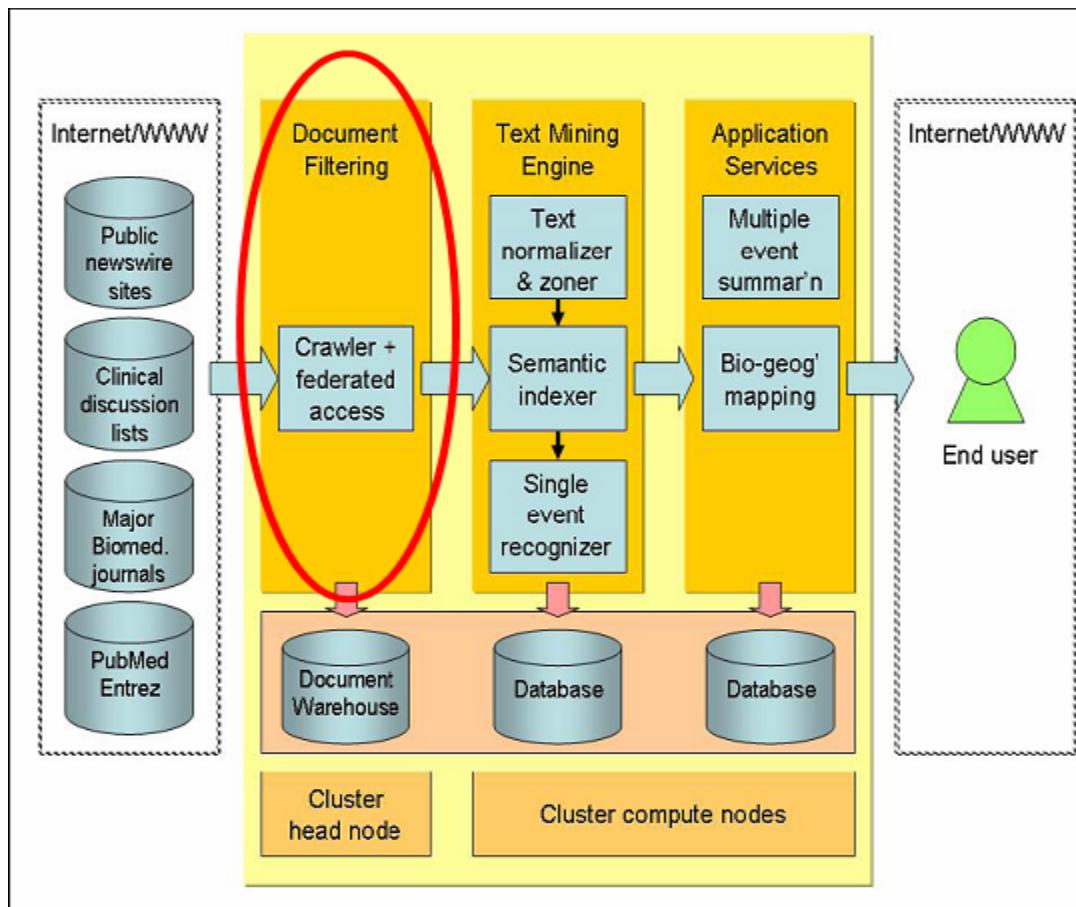
1.5 Vai trò của hệ thống khai thác thông tin văn bản (text mining system) trong dự án

Trong dự án này, phần module khai thác thông tin văn bản là module đầu tiên và được đánh giá là có vai trò quan trọng nhất. Vì tính chất của hệ thống là xử lý theo thời gian thực, do vậy tốc độ luôn xử lý luôn được quan tâm hàng đầu. Module khai thác thông tin văn bản sẽ thực hiện các công việc chính như sau:

- Tìm kiếm các tin tức lấy được từ **RSS** và nhiều nguồn khác nhau theo những chủ đề đang quan tâm để giảm thiểu thời gian xử lý.

- Phân tích văn bản có được từ đầu ra của phần trên để lấy ra các thông tin cần thiết như tên người, tên địa danh, các số liệu về thời gian, địa điểm xuất hiện dịch bệnh với quy mô đơn lẻ hay trở thành ô dịch lớn...

Kiến trúc của mô hình:



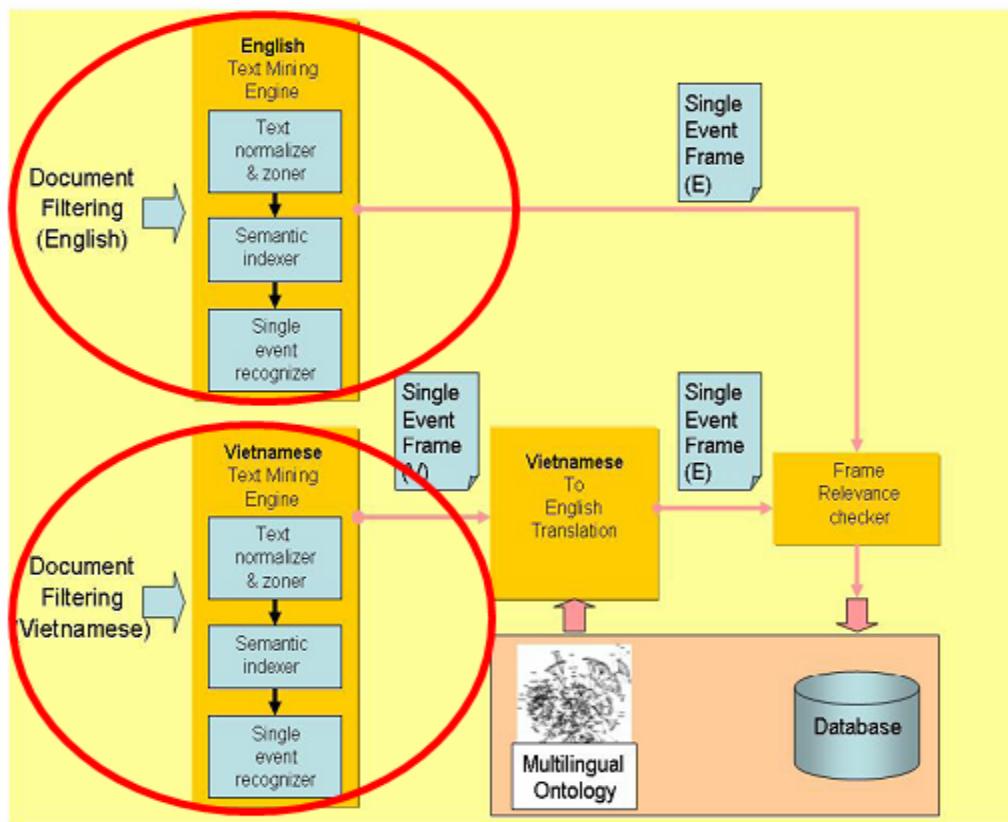
Hình 2: Kiến trúc của dự án BioCaster

Trong kiến trúc của mô hình, giai đoạn tìm kiếm văn bản theo chủ đề được mô tả như một bộ lọc văn bản, có nhiệm vụ tìm kiếm những văn bản theo chủ đề chọn lựa. Hiệu quả tìm kiếm của giai đoạn này càng cao sẽ có hai tác dụng chính:

- Độ chính xác các văn bản cần khai thác càng cao sẽ giúp nâng cao tính chính xác của thông tin thu thập.
- Độ chính xác các văn bản cần khai thác càng cao sẽ giúp giảm thiểu thời gian xử lý cho các giai đoạn sau, tránh phải xử lý các văn bản không liên quan đến các thông tin cần thiết. Như vậy tốc độ của hệ thống sẽ được nâng cao đáng kể.

Vì tính chất của hệ thống **BioCaster**: xây dựng bản đồ điện tử đa ngôn ngữ nên trong hệ thống, mỗi ngôn ngữ đóng một vai trò nhất định. Mỗi ngôn ngữ sẽ là một module riêng biệt có nhiệm vụ tương tự nhau trong giai đoạn thu thập văn bản, khai thác thông tin văn bản. Sau đó các thông tin thu thập được sẽ được chuyển thành một ngôn ngữ thống nhất. Như vậy, tổng quan mỗi module trên mỗi ngôn ngữ đều có các phần chính sau:

- Thu thập văn bản.
- Tìm kiếm các văn bản theo chủ đề.
- Khai thác thông tin từ văn bản thu thập.



Hình 3: Mô hình chi tiết các module

Trong chương tiếp theo, chúng tôi sẽ đưa ra những thông tin tổng quan về bài toán tìm kiếm văn bản theo chủ đề trên hai ngôn ngữ chính: tiếng Anh và tiếng Việt

1.6 Mục tiêu thực hiện của luận văn

Bài toán “Tìm kiếm văn bản tiếng Việt theo chủ đề” được chúng em chia thành hai giai đoạn xử lý chính: giai đoạn phân loại văn bản và giai đoạn tìm kiếm văn bản theo chủ đề định trước.

1.6.1 Tìm hiểu các thuật toán phân loại văn bản

Trong khuôn khổ luận văn, chúng em sẽ tìm hiểu lý thuyết của các phương pháp học được áp dụng cho bài toán phân loại văn bản hiện: phương pháp máy học sử dụng véc tơ hỗ trợ (**Support Vector Machines - SVM**), phương pháp K người láng giềng gần nhất (**k – Nearest Neighbours**), mô hình ngôn ngữ thông kê **N-gram**... Sau đó chúng em tiến hành cài đặt thực nghiệm để chứng minh cho mô hình phân loại văn bản tiếng Việt tốt nhất sử dụng cho hệ thống tìm kiếm nói trên.

1.6.2 Xây dựng ứng dụng tìm kiếm văn bản theo chủ đề

Để ứng dụng kết quả đạt được trong bài toán phân loại văn bản tiếng Việt theo chủ đề, chúng em xây dựng một ứng dụng cụ thể với hai chức năng chính:

- Tìm kiếm văn bản **offline** dựa trên tài nguyên đã có sẵn trên máy tính theo những chủ đề cho trước. Ngoài ra ứng dụng còn có thể giúp người sử dụng phân loại một văn bản được gõ vào trực tiếp từ một trình soạn thảo.
- Cập nhật và thu thập tin tức điện tử online qua **RSS (Really Simple Syndication)** cụ thể là từ 4 loại báo điện tử lớn và đầy đủ nhất Việt Nam hiện nay: VnExpress (www.vnexpress.net), TuoiTreOnline (www.tuoitre.com.vn), ThanhNienOnline (www.thanhnien.com.vn), NguoiLaoDong (www.nld.com.vn) theo từng chủ đề được người dùng chọn lựa.
- Chức năng mở rộng: tìm kiếm trực tuyến (**online**) từ 4 nguồn báo nói trên.

Ngoài ra ứng dụng cao nhất của luận văn chính là bước cơ bản trong việc thu thập các văn bản, tin tức liên quan đến chủ đề “Cúm gia cầm” được tích hợp vào dự

án **BioCaster** do chính phủ Nhật tài trợ nhằm xây dựng bản đồ điện tử thể hiện dịch cúm gia cầm đang xảy ra tại Việt Nam và trên thế giới.

1.7 **Đóng góp của luận văn**

Luận văn đã đưa ra cái nhìn tổng quát cho bài toán phân loại văn bản với hai cách tiếp cận: dựa trên nền tảng bài toán tách từ và không dựa trên bài toán tách từ. Luận văn cũng đã mở ra một hướng tiếp cận mới giúp giải quyết các bài toán xử lý ngôn ngữ tự nhiên trong điều kiện chúng ta chưa có một công cụ tách từ cho độ chính xác tuyệt đối. Mô hình ngôn ngữ thống kê **N-Gram (Statistical N-Gram Language Modeling)** là cách tiếp cận đơn giản nhưng cho hiệu quả tương đương.

Bên cạnh đó, việc cài đặt thành công bài toán phân loại và tìm kiếm văn bản theo chủ đề sẽ có thể được áp dụng vào nhiều ứng dụng cụ thể trong đời sống, đặc biệt là dự án **BioCaster** nói trên, góp phần giảm thiểu sự tiêu tốn về thời gian và công sức con người, đồng thời cũng khuyến khích nhiều hướng nghiên cứu cho việc xử lý và ứng dụng văn bản điện tử trên máy tính.

1.8 **Bố cục của luận văn**

Nội dung của luận văn được trình bày bao gồm **6 chương**:

- ✓ **Chương 1. Mở đầu:** trình bày 1 cách khái quát về luận văn cũng như các phương pháp tiếp cận giải quyết vấn đề đặt ra.
- ✓ **Chương 2. Tổng quan:** tình hình trong và ngoài nước về đề tài này, ưu và khuyết điểm của đề tài, các vấn đề cần giải quyết trong đề tài.
- ✓ **Chương 3. Cơ sở lý thuyết:** chương này được chia thành 5 phần:
 - Các cơ sở lý thuyết về văn bản
 - Các cơ sở lý thuyết về tách từ
 - Các cơ sở lý thuyết về phân loại văn bản
 - Các phương pháp máy học
 - Cơ sở mô hình ngôn ngữ thống kê.

- ✓ **Chương 4. Mô hình – Thiết kế - Cài đặt:** phương pháp giải quyết vấn đề cho bài toán tách từ, bài toán phân loại văn bản, các cách tiếp cận, những khó khăn và thuận lợi, các thiết kế cài đặt thuật toán và chương trình ứng dụng.
- ✓ **Chương 5. Kết quả thực nghiệm**
- ✓ **Chương 6. Kết luận và Hướng phát triển**

Chương 2: TỔNG QUAN

Nội dung

Chương này nhằm giới thiệu tổng quan về bài toán tìm kiếm văn bản theo chủ đề. Trong đó, chúng tôi sẽ phân tích các vấn đề liên quan đến bài toán này như: bài toán tách từ, bài toán phân loại văn bản. Song song đó chúng tôi sẽ đề cập đến các phương pháp tiếp cận cho bài toán tách từ, bài toán phân loại văn bản tiếng Anh và một số công trình áp dụng cho các vấn đề liên quan trong tiếng Việt.

2.1 Bài toán tách từ

Trong phần này, chúng tôi đề cập sơ lược 4 công trình tách từ đã áp dụng thành công trên tiếng Việt (một số bài báo đã được đăng ở các hội nghị trong và ngoài nước). 4 công trình này đều do các tác giả trong nhóm **VCL** thực hiện và bước đầu có được một số kết quả tuy vẫn còn khá hạn chế, từ đó phân tích ưu khuyết điểm để đề nghị mô hình tách từ mới cho tiếng Việt (sẽ được trình bày trong chương 4). Lưu ý, để phục vụ cho bài toán phân loại văn bản tiếng Việt, chúng tôi chỉ đề cập đến các mô hình tách từ tiếng Việt dùng cho đơn ngữ.

2.1.1 Các vấn đề trong bài toán tách từ

2.1.1.1 Xử lý nhập nhằng [11]

Nhập nhằng trong tách từ được phân thành hai loại:

- Nhập nhằng chồng (Overlapping Ambiguity)
- Nhập nhằng hợp (Combination Ambiguity)

Ta gọi D là tập hợp các từ tiếng Việt (từ điển tiếng Việt). Các trường hợp nhập nhằng trên được mô tả hình thức như sau:

- Chuỗi $\alpha\beta\gamma$ được gọi là nhập nhằng chồng nếu $\{\alpha\beta, \beta\gamma\} \subset D$.
- Chuỗi $\alpha\beta$ được gọi là nhập nhằng hợp nếu $\{\alpha, \beta, \alpha\beta\} \subset D$

Trong thực tế, loại nhập nhằng chồng xảy ra thường xuyên hơn loại nhập nhằng hợp, bởi vì hầu hết các tiếng của tiếng Việt đều có thể đóng vai trò là một

từ đơn độc lập³. Do đó, hầu hết các từ ghép đều có thể bị nhập nhằng hợp. Tuy nhiên, hầu như mọi trường hợp này đều được giải quyết tốt bằng giải thuật **Maximum Matching**. Vì thế, mọi hệ thống nhận diện nhập nhằng hiện tại đều chỉ chú ý đến việc giải quyết loại nhập nhằng đầu tiên là nhập nhằng chòng [15].

2.1.1.2 Nhận diện từ chưa biết

Trong văn bản không chỉ có sự tồn tại của từ thuần túy có trong từ điển, mà còn có các đơn vị thông tin khác nữa. Do không nắm được các thông tin này, nên việc tách từ sẽ bị ảnh hưởng.

Từ chưa biết bao gồm các từ tên riêng tiếng Việt hoặc tiếng nước ngoài và các **factoids**⁴.

2.1.2 Các hướng tiếp cận chính cho bài toán tách từ

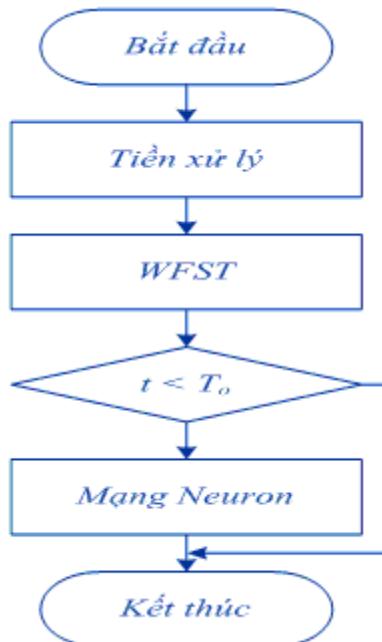
- Tiếp cận dựa vào từ điển cố định.
- Tiếp cận dựa vào thông kê thuần túy.
- Tiếp cận dựa trên cả hai phương pháp trên.

2.1.3 Tách từ tiếng Việt dùng mô hình WFST

Đây có thể được xem là mô hình tách từ đầu tiên dành cho tiếng Việt [16]. Mô hình này là một cải tiến của mô hình **WFST** (Weighted Finite State Transducer) của Richard Sproat [17] để phù hợp hơn với tiếng Việt. Mô hình được đề xuất để giải quyết vấn đề tách từ trong tiếng Việt là một mô hình lai như sau:

³ Trong các từ điển được sử dụng, có 6979 từ đơn trong từ điển từ, và tổng cộng 7457 tiếng trong từ điển tiếng.

⁴ Theo định nghĩa của WordNet thì Factoid là một đối tượng biểu diễn những thông tin đặc biệt. Trong luận văn này chúng tôi chỉ xét các loại thông tin sau: ngày tháng, thời gian, phần trăm, tiền tệ, số, độ đo, địa chỉ email, số điện thoại, trang web



Hình 4: Qui trình của mô hình WFST

Đầu tiên cho câu đi qua phần tiền xử lý, giai đoạn này loại bỏ các lỗi về cách trình bày một câu. Tuy nhiên điều quan trọng hơn là trong công đoạn này còn chuẩn hoá về cách bỏ dấu, cách viết các ký tự y,i... trong tiếng Việt. (*Do còn chưa chuẩn hóa tiếng Việt có một số âm tiết khi viết thì khác nhau nhưng nghĩa và cách đọc thì như nhau. Ví dụ: thời kỳ = thời kì, hoà = hòa v..v...).*

Sau đó câu được đưa vào một mô hình WFST. Giai đoạn này sẽ tự động nhận diện các từ láy, danh từ riêng (do đặc điểm tiếng Việt: danh từ riêng phải viết hoa chữ cái đầu tiên của mỗi tiếng), tên riêng người Việt (Theo luật sinh), tên riêng nước ngoài,... và gán cho chúng một trọng số thích hợp. Mô hình WFST sẽ căn cứ trên các trọng số này để chọn ra một cách tách từ thích hợp.

Nếu trong giai đoạn trên, câu cần tách vẫn còn nhập nhằng (điều này được xác định thông qua một giá trị ngưỡng nào đó) mô hình sẽ tự động gọi mô hình mạng Neural để khử các nhập nhằng đó và chọn ra trường hợp tách từ phù hợp.

2.1.3.1 Mô hình WFST

Tách từ tiếng Việt có thể được xem như là một vấn đề thống kê chuyển đổi trạng thái (Stochastic Transduction Problem). Tác giả miêu tả từ điển D là một đồ

thị biến đổi trạng thái hữu hạn có trọng số (**WFST - Weighted Finite State Transducer**).

Giả sử

H : là tập tiếng.

P : là từ loại của từ (**POS – Part Of Speech**).

Mỗi cung của D hoặc là bắt đầu từ một phần tử của H tới một phần tử của H hoặc từ ε , ký hiệu kết thúc từ, tới một phần tử của P .

Nói cách khác, mỗi từ được miêu tả trong từ điển là một dãy tuân tự các cung, bắt đầu bằng một trạng thái ban đầu của D và được gán nhãn bằng một phần tử S thuộc H , và kết thúc bởi một cung được gán nhãn là một phần tử của $\varepsilon \times P$. Nhãn này biểu thị một trọng số ước lượng (estimated cost) (lấy log của xác suất).

Tiếp theo, tác giả biểu diễn câu cần tách nhập là một acceptor trạng thái hữu hạn không có trọng số (**FSA - Finite State Acceptor**) I trên H .

Giả sử đã tồn tại một hàm Id mà có input là FSA A, và output là một 1 transducer mà các các phần tử trong đó chỉ bao gồm các phần tử thuộc A (gọi là D^*).

Cuối cùng, tác giả định nghĩa một trường hợp tách từ đúng nhất trong câu là một câu có trọng số nhỏ nhất trong $Id(I) \times D^*$.

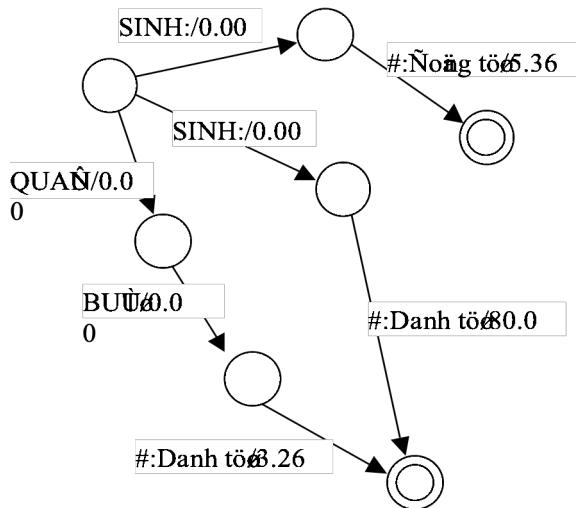
2.1.3.2 Biểu diễn từ điển

Như phân trình bày trên, từ điển gồm một loạt các nút và các cung. Mỗi từ được kết thúc bởi một cung biểu diễn một sự chuyển đổi giữa ε và từ loại của chúng.Thêm vào đó, chúng còn được gán một trọng số ước lượng cho từ đó. Tuy nhiên, nếu lấy xác suất của từ làm trọng số thì các phép tính sẽ khó mà thực hiện do các số này thường rất nhỏ. Do đó ta gán trọng số bằng cách lấy log của xác suất từ cụ thể là:

$$C = -\log\left(\frac{f}{N}\right) \quad [2.1]$$

Trong đó f : là tần số của từ

N : kích thước tập mẫu.



Hình 5: Đồ thị trọng số biểu diễn cung từ điển

2.1.3.3 Phân tích hình thái

Phần trên chỉ mới đề cập đến một phương pháp tách từ dựa trên từ điển. Tuy nhiên, có một số lớn một lớp các từ mà nhiều khi ta không tìm thấy trong các từ điển chuẩn. Trong đó, nổi bật là có một lớp các từ dẫn xuất, biến cách, (hình thái học) được tạo ra bằng cách thêm vào các hình thái của nó.

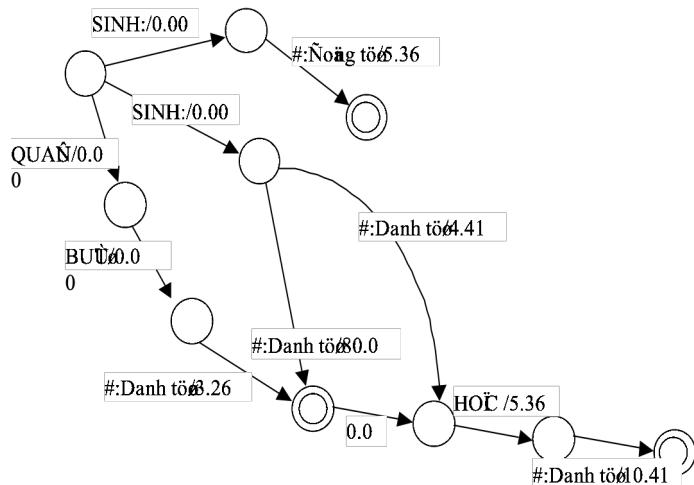
Ví dụ: xã hội → xã hội hóa

Tuy nhiên, vấn đề hình thái đã được các nhà tin học biểu diễn bởi một kỹ thuật nổi tiếng, đó là kỹ thuật dùng mô hình Trạng thái hữu hạn.

Trong trường hợp này, tác giả biểu diễn lại mô hình từ điển để có thể giải quyết được vấn đề này bằng cách thêm một loạt các cung thể hiện sự phân tích hình thái học. Cụ thể là nối thêm một cung có nhãn là "**học**" sẽ nối từ một entry kết thúc của một danh từ đến một cây WFST nhỏ biểu thị cho từ "**học**". Tuy nhiên sẽ không thể nào thống kê từng trường hợp của hình thái và đưa nó vào trong từ điển như cách trên.

Để giải quyết vấn đề trên, tác giả sử dụng một cách tính toán trọng số của một từ dẫn xuất thông qua xác suất của các thành phần trong từ dẫn xuất bằng cách tính xác suất Bayes. Khi đó xác suất tập hợp (aggregate probability) của một thể hiện chưa biết trước của một cấu trúc được ước lượng là $\frac{n_i}{N}$, khi đó N là tổng số

của những dấu hiệu quan sát được và n_1 là số của loại quan sát chỉ một lần. Ký hiệu tập hợp của những cái chưa biết trước, từ mới, là tập hợp X là unseen(X). Khi đó một tập hợp chưa biết mà có từ dẫn xuất từ X là định nghĩa là unseen("học"). Hình dưới chỉ ra cách mà mô hình được khai báo như là một phần của từ điển WFST. Ở đó có một sự chuyển tiếp giữa một nút gán nhãn "Danh từ" và nút có gán nhãn từ "học".



Hình 6: Đồ thị WFST biểu diễn từ điển

2.1.3.4 Mô hình mạng Neuron

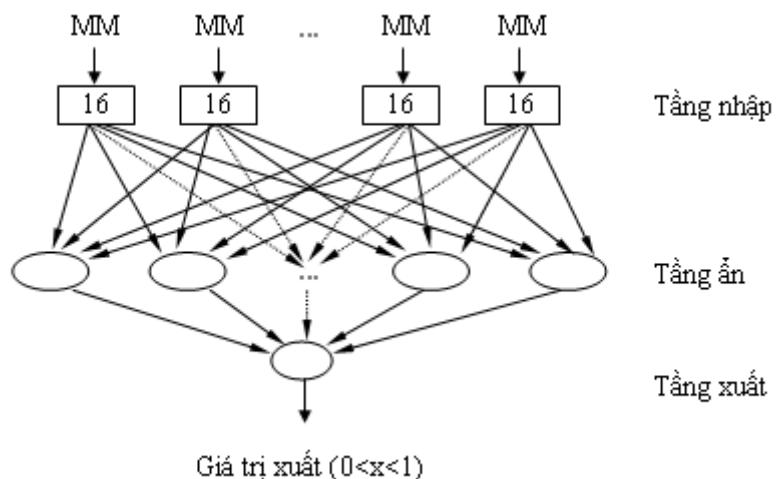
Sau khi cho câu được tách qua mô hình WFST. Để xác định kết quả tách từ trên có thực sự hợp lệ hay không, tác giả định nghĩa một ngưỡng giá trị t_0 với ý nghĩa như sau: Nếu sự chênh lệch về trọng số trong các câu được tách so với câu có trọng số bé nhất lớn hơn t_0 thì đó là kết quả tách từ trên thực sự được chấp nhận. Còn nếu có một vài câu mà sự chênh lệch về trọng số của chúng so với câu có trọng số bé nhất nhỏ hơn t_0 thì mô hình WFST chưa thể xác định được ranh giới từ trong câu, lúc này đưa những câu này qua mô hình neural xử lý.

Ví dụ:

Sau khi qua phần xử lý WFST chỉ được ba câu sau, do có một số trường hợp không chọn do sự chênh lệch trọng số của câu quá nhỏ:

1. *hoc sinh(N) hoc(V) sinh hoc(N)*
2. *hoc sinh(N) hoc sinh(N) hoc(V)*
3. *hoc(V) sinh hoc(N) sinh hoc(N)*

Thực tế trong tiếng Việt thì nhiều khi có một dãy các loại từ không thể tuân tự đứng cạnh nhau. Ví dụ như tính từ và giới từ không thể đứng cạnh nhau. Nếu áp dụng luật cho việc giải quyết nhập nhằng ở đây sẽ không thích hợp. Điều ấy là do tiếng Việt không có một cú pháp chặt chẽ. Hơn nữa, nếu có áp dụng mô hình luật đĩ chǎng nữa thì vẫn vấp phải một vấn đề rất khó khăn là số lượng luật bao nhiêu là vừa. Nếu ít quá thì sẽ thiếu sót một số trường hợp đúng. Còn nếu nhiều quá thì vô hình chung câu nào cũng đúng.



Hình 7: Mô hình mạng nơron dùng cho khử nhằng câu

Để giải quyết vấn đề này, mô hình đề nghị cho máy học các câu nhập nhằng dùng Mạng Neuron. Và mô hình máy học mà tác giả đề xuất ở đây là mạng neural. Tác giả sử dụng mô hình này để lượng giá sự hợp lệ của dãy từ loại trong câu.

Thực tế mô hình mạng chúng tôi gồm 6 nút nhập, 10 nút ẩn và một nút xuất. Mỗi nút nhập là một véc tơ 16 chiều ký hiệu cho từ loại mà từ đó mang. Nếu là dấu câu thì cũng coi như đó là một loại từ và tùy theo dạng dấu câu sẽ gán cho nó một giá trị nhất định.

Tầng nhập của mạng neural được kết nối hoàn toàn với một tầng ẩn gồm 10 nút với một hàm truyền. Những nút ẩn này lại được kết nối hoàn toàn với một tầng xuất chỉ gồm 1 nút.

Nút xuất là một giá trị thực nằm giữa 0..1. Biểu thị cho khả năng hợp lệ của một dãy các từ loại đứng liền nhau trong một cửa sổ trượt. Khi cửa sổ trượt từ

đầu câu đến cuối câu, cộng dồn các kết quả lại với nhau và gán giá trị này vào thành trọng số của câu.

Hàm truyền, được chọn hàm sigmoid

$$f(h_i) = \frac{1}{1 + e^{-\frac{h_i}{T}}} \quad [2.2]$$

Đây là một hàm thông dụng trong các mạng neural. Một câu được chọn tức là câu có trọng số lớn nhất.

2.1.4 Tách từ tiếng Việt dùng mô hình Maximum Matching

Maximum Matching (MM) được xem như là phương pháp tách từ dựa trên từ điển đơn giản nhất. **MM** cố gắng so khớp với từ dài nhất có thể có trong từ điển. Đó là một thuật toán ăn tham (Greedy Algorithms) nhưng bằng thực nghiệm đã chứng minh được rằng thuật toán này đạt được độ chính xác > 90% nếu từ điển đủ lớn [18]. Tuy nhiên, nó không thể giải quyết vấn đề nhập nhằng và không thể nhận diện được các từ chưa biết bởi vì chỉ những từ tồn tại trong từ điển mới được phân đoạn đúng.

Giải quyết **MM** gồm hai giải thuật con: **FMM** (Forward Maximum Matching: so khớp cực đại theo chiều tiến) và **BMM** (Backward Maximum Matching: so khớp cực đại theo chiều lùi). Nếu chúng ta nhìn vào kết quả của **FMM** và **BMM** thì sự khác biệt này cho chúng ta biết nơi nào nhập nhằng xảy ra. Ngoài ra, **MM** là phương pháp tách từ hoàn toàn phụ thuộc vào từ điển, từ điển phải đủ lớn, đủ chính xác và độ tin cậy phải cao thì mới cho kết quả tách từ chấp nhận được. Đây cũng là nhược điểm rất lớn của phương pháp này.

Ví dụ: Người nông dân ra sức cải tiến bộ công cụ lao động của mình.

Đầu ra FMM:

Người#nông dân#ra sức#cải tiến#bộ#công cụ#lao động#của#mình#.

Đầu ra BMM:

Người#nông dân#ra sức#cải#tiến bộ#công cụ#lao động#của#mình#.

Kết quả thử nghiệm phương pháp này trên tiếng Việt sẽ được trình bày cụ thể trong **chương 4**, chương sẽ trình bày cách tiếp cận mới cho bài toán tách từ tiếng Việt so sánh và rút ra kết luận.

2.1.5 Tách từ tiếng Việt dùng mô hình MMSeg⁵

Thực chất đây là mô hình tách từ của tiếng Trung Quốc [19] và mô hình này cho kết quả tương đối khả quan. Nói chính xác là đây là mô hình bổ sung cho mô hình **MM** (phần 2.1.4) sử dụng thêm 1 số luật heuristic trên ngôn ngữ để đánh giá dựa trên 2 mô hình **MM**.

2.1.5.1 Thuật toán MM và các biến thể của nó

Các nghiên cứu khác nhau cho thấy vấn đề tách từ khác nhau ở chỗ là giải quyết vấn đề nhập nhằng. **MM** có nhiều hình thức khác nhau:

- **MM đơn giản (Simple MM)**: hình thức cơ bản nhất là giải quyết nhập nhằng của từ đơn. Ví dụ: giả sử C_1, C_2, \dots, C_n biểu diễn cho dãy tiếng của 1 chuỗi. Chúng ta ở đầu của 1 chuỗi và muốn biết đâu là từ. Đầu tiên, chúng ta tìm trong tập từ vựng, nếu $_C_1$ là 1 từ đơn 1 tiếng thì tìm tiếp $_C_1C_2$ thử, và nếu nó là từ 2 tiếng và cứ tiếp tục như vậy cho đến khi sự kết hợp các tiếng tạo thành từ dài nhất trong tập từ vựng. Từ hợp lý nhất sẽ là từ được so khớp dài nhất. Chúng ta lấy từ này, sau đó tiếp tục tiến trình cho đến khi từ cuối cùng của chuỗi được nhận ra.

- **MM phức tạp (Complex MM)**: 1 biến thể khác của giải thuật **MM** được hoàn thành bởi [20] có hình thức phức tạp hơn hình thức cơ bản trên. Họ thêm 1 số luật cho rằng việc tách từ hợp lý nhất là 1 dãy ba từ với chiều dài dài nhất. Một lần nữa, chúng ta lại bắt đầu ở đầu chuỗi và muốn biết đâu là từ. Nếu chúng ta bắt gặp những đoạn nhập nhằng xảy ra ($_C_1$ là 1 từ, nhưng $_C_1C_2$ cũng là 1 từ, ...) thì chúng ta sẽ nhìn tiếp theo 2 từ để tìm ra dãy 3 từ có thể có bắt đầu với $_C_1$ và $_C_1C_2$. Ví dụ:

1. $_C_1_C_2_C_3C_4_$
2. $_C_1C_2_C_3C_4_C_5_$

⁵ <http://technology.chtsai.org/mmsseg/>

3. $_C_1C_2__C_3C_4__C_5C_6__$

Chuỗi với chiều dài dài nhất là cái thứ 3. Từ đầu tiên, $_C_1C_2__$ của chuỗi thứ 3 sẽ được xem như là 1 từ đúng. Chúng ta chọn từ này, tiếp tục tiến trình từ tiếng C_3 cho đến khi từ cuối cùng của chuỗi được nhận ra.

2.1.5.2 Các luật khử nhập nhằng (Ambiguity Resolution Rules)

Dựa vào đặc điểm riêng của tiếng Việt so với tiếng Trung Quốc, các luật sau đây sẽ được áp dụng:

Luật 1: sử dụng Simple Maximum Matching lấy từ với chiều dài dài nhất, Complex maximum matching lấy từ đầu tiên từ dãy với chiều dài dài nhất. Nếu có nhiều dãy với chiều dài dài nhất, áp dụng luật kế tiếp.

Luật 2: hai từ 2 tiếng không đi liền nhau. Điều này hoàn toàn đúng trong tiếng Việt, chúng ta xem ví dụ sau đây:

Học sinh học sinh học

Có 1 số cách tách từ sau đây:

Học sinh#học sinh#học

Học#sinh học#sinh học

Hai từ “Học sinh” và “học sinh” không bao giờ đi liền nha, cũng như “sinh học” không bao giờ đi liền với “sinh học”.

Luật 3: chiều dài biến động nhỏ nhất (smallest variance of word lengths). Có 1 số ít điều kiện nhập nhằng mà trong luật 1 và luật 2 không thể giải quyết được. Ví dụ, có 2 chuỗi (chunks) có cùng chiều dài:

1. $_C_1C_2__C_3C_4__C_5C_6__$

2. $_C_1C_2C_3__C_4__C_5C_6__$

Luật 3 sẽ lấy cái đầu tiên từ dãy với chiều dài biến động nhỏ nhất. Trong ví dụ trên, nó lấy $_C_1C_2__$ từ dãy đầu tiên. Luật này sẽ được áp dụng sau khi áp dụng luật 1. Giả sử của luật này là chiều dài từ phải được phân bố thường ngang nhau. Nếu có nhiều hơn 1 dãy (chunk) thỏa mãn yêu cầu, áp dụng luật kế tiếp.

Luật 4: tần số tiếng cao nhất hay log thấp nhất. Ví dụ sau đây thể hiện rõ 2 chuỗi với cùng chiều dài, biến động:

1. _C₁_ _C₂_ _C₃C₄_

2. _C₁_ _C₂C₃_ _C₄_

Cả hai dãy đều có những từ 1 tiếng và 1 từ 2 tiếng. Nhưng cái nào sẽ đúng hơn. Ở đây, chúng ta sẽ tập trung vào những từ 1 tiếng. Các tiếng sẽ khác nhau ở mức độ tự do hình vị (degree of morphemic freedom). Một vài tiếng hiếm khi được dùng như hình vị tự do. Tần số xuất hiện của các tiếng có thể xem như là chỉ mục của mức độ tự do hình vị. Tiếng có tần số cao rõ ràng là từ đơn 1 tiếng và ngược lại.

Công thức tính tổng độ tự do hình vị là tính tổng của log(frequency) của tất cả các từ 1 tiếng trong dãy. Lý do cho việc dùng biến đổi logarit là cùng lượng khác biệt tần số sẽ không có ảnh hưởng phù hợp thông quan tất cả vùng tần số.

Sau đó, luật 4 sẽ lấy từ đầu tiên của dãy với tổng log(frequency) lớn nhất. Khi 2 tiếng có cùng giá trị tần số, lúc đó sẽ không có nhập nhằng sau khi luật này được áp dụng.

Chúng ta lấy ví dụ: Học sinh học sinh học

FMM: Học sinh# học sinh# học

BMM: Học# sinh học# sinh học

MMSeg(Luật 2): Học sinh# học# sinh học.

2.1.6 Tách từ tiếng Việt dùng mô hình Maximum Entropy

Mô hình tách từ bằng phương pháp **Maximum Entropy** dựa trên ý tưởng của mô hình gán nhãn từ loại (**POS Tagger**) dùng phương pháp **Maximum Entropy** cho tiếng Anh của Adwait Ratnaparkhi [11]. Các tác giả của công trình [11] đã cài đặt thành công mô hình này cho tiếng Việt.

2.1.6.1 Ý tưởng của phương pháp

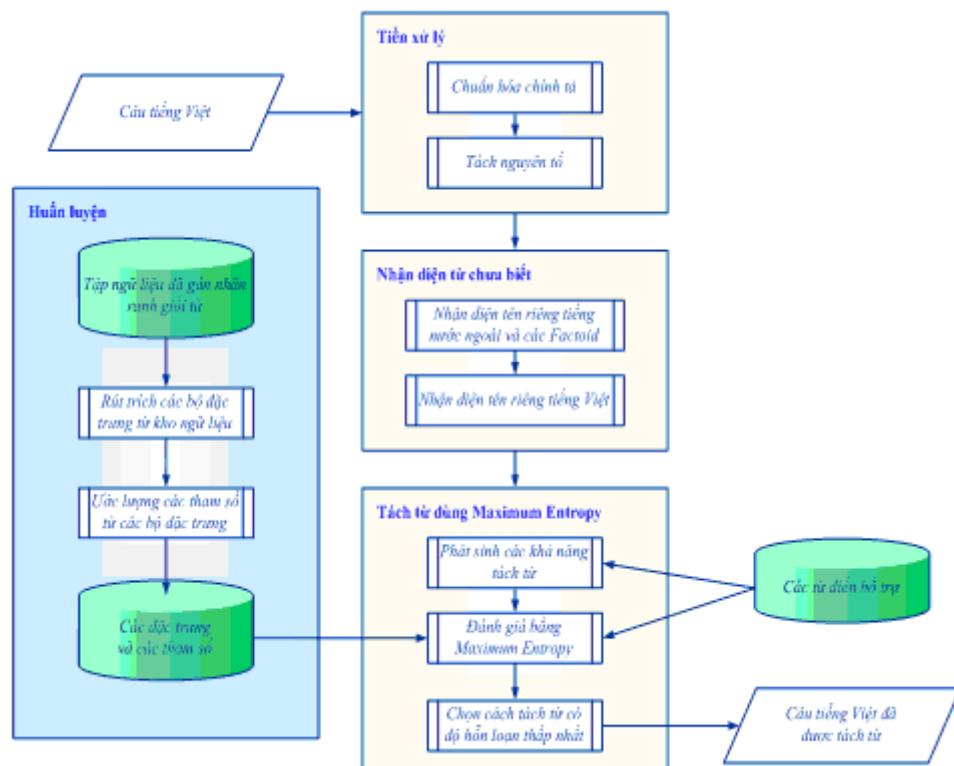
Cho một câu $S=c_1c_2\dots c_n$ có chiều dài n tiếng. Ta thực hiện tách từ cho câu S bằng cách gán nhãn vị trí t_i cho từng tiếng trong câu S các nhãn vị trí trong từ **PIV** (**Position In Word**). Có 4 nhãn vị trí như sau:

- **LL:** dùng để gán cho tiếng nằm bên trái của một từ, và tạo ra 1 từ khi kết hợp với các tiếng nằm bên phải của nó.

- **RR:** dùng để gán cho tiếng nằm bên phải của một từ, và tạo ra một từ khi kết hợp với những tiếng bên trái của nó.
- **MM:** dùng để gán cho tiếng nằm giữa 1 từ.
- **LR:** nếu bản thân nó đứng một mình và đóng vai trò là một từ.

Sau khi các tiếng c_i trong câu S đều đã được gán nhãn vị trí, ta chuyển kết quả này thành ranh giới từ dựa trên vị trí của từng tiếng trong từ.

2.1.6.2 Mô hình thuật toán [11]



Hình 8: Mô hình thuật toán Maximum Entropy

Mô hình xác suất được định nghĩa bằng $H \times T$ trong đó H là tập các ngữ cảnh và T là tập các nhãn. Mô hình của xác suất kết hợp của ngữ cảnh h và t được định nghĩa như sau:

$$p(h, t) = \pi \prod_{j=1}^k \alpha_j^{f_j(h, t)} \quad [2.3]$$

Trong đó,

π là hằng số

$\{\alpha_1, \dots, \alpha_k\}$ là các tham số của mô hình

$\{f_1, \dots, f_k\}$ là những đặc trưng và $f_j(h, t) \in \{0, 1\}$

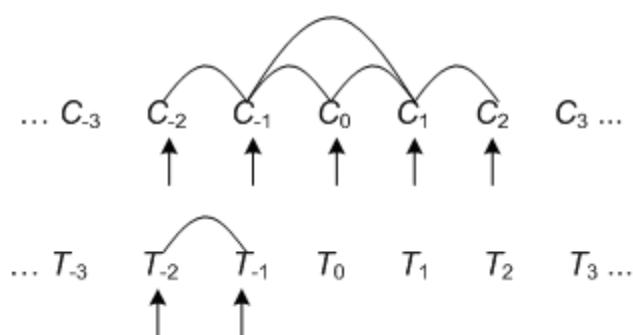
Mỗi đặc trưng f_j đều có tương ứng một tham số α_j

Trong giai đoạn huấn luyện, một chuỗi các tiếng $\{c_1, \dots, c_n\}$ và các nhãn tương ứng của chúng $\{t_1, \dots, t_n\}$ là thành phần dữ liệu huấn luyện. Và mục tiêu chính là xác định tập tham số $\{\alpha_1, \dots, \alpha_k\}$ mà cực đại hàm likelihood của dữ liệu huấn luyện sử dụng p :

$$L(p) = \prod_{i=1}^n p(h_i, t_i) = \prod_{i=1}^n \pi \prod_{j=1}^k \alpha_j^{f_j^{(h_i, t_i)}} \quad [2.4]$$

Sự thành công của mô hình trong việc gán nhãn phụ thuộc vào độ rộng của phạm vi trong việc lựa chọn các đặc trưng phù hợp. Cho một bộ (h, t) , một đặc trưng phải mã hóa được thông tin phục vụ cho việc dự đoán t . Các đặc trưng được sử dụng cho mô hình thử nghiệm này nằm trong danh sách sau: Đặc trưng mặc định: được kích hoạt khi không tồn tại đặc trưng nào bên dưới.

- ❖ Tiếng hiện tại (C_0)
- ❖ Hai cặp tiếng đi liền trước và sau (C_{-2}, C_{-1}, C_1, C_2)
- ❖ Hai tiếng đi liền trước và sau và tiếng hiện tại (C_{-1}, C_0, C_0, C_1), hai tiếng đi trước (C_{-2}, C_{-1}) và hai tiếng đi sau (C_1, C_2)
- ❖ Hai tiếng đi trước và đi sau (C_{-1}, C_1)
- ❖ Nhãn của tiếng liền trước (T_{-1}) và tiếng liền trước trước đó (T_{-2}).



Tổng quát, cho một bộ (h, t) , những đặc trưng này là những dạng quan hệ đồng hiện giữa t và một số loại ngữ cảnh h , hoặc giữa t và các thuộc tính của ký tự hiện tại.

Ví dụ:

$$f_i(h_i, t_i) = \begin{cases} 1 & t_{i-1} = LL \wedge t_i = RR \\ 0 & \text{khač} \end{cases} \quad [2.5]$$

Các đặc trưng kể trên có thể mã hóa 3 loại ngữ cảnh.

- ❖ Đặc trưng dựa trên tiếng hiện tại và các tiếng xung quanh (2, 3, 4, 5)
- ❖ Đặc trưng dựa trên các nhãn đi trước (6)
- ❖ Đặc trưng mặc định: được dùng khi không có đặc trưng nào tồn tại.

Ví dụ ta có câu sau: *tôi tư duy nghĩa là tôi tồn tại*

Câu này được tách từ như sau: *tôi#tư duy#nghĩa là#tôi#tồn tại#*

Tiếng	<i>tôi</i>	<i>tư</i>	<i>duy</i>	<i>nghĩa</i>	<i>là</i>	<i>tôi</i>	<i>tồn</i>	<i>tại</i>
Nhãn	LR	LL	RR	LL	RR	LR	LL	RR
Vị trí	1	2	3	4	5	6	7	8

Bảng 1: Ví dụ về nhãn vị trí tiếng trong từ

Các đặc trưng được phát sinh của tiếng *nghĩa* trong câu trên:

$$C_0 = \text{nghĩa} \quad \& \quad t_i = LL$$

$$C_{-2}C_{-1}C_1C_2 = \text{tư duy là tôi} \quad \& \quad t_i = LL$$

$$C_{-2}C_{-1}C_0C_1C_2 = \text{tư duy nghĩa là tôi} \quad \& \quad t_i = LL$$

$$C_{-1}C_1 = \text{duy là} \quad \& \quad t_i = LL$$

$$T_{-2}T_{-1} = LL RR \quad \& \quad t_i = LL$$

Khi kết thúc quá trình huấn luyện, các đặc trưng và các tham số tương ứng sẽ được dùng để tính xác suất của chuỗi gán nhãn của câu khi bộ gán nhãn gán nhãn trên câu truyền vào. Cho một chuỗi các tiếng $\{c_1, \dots, c_n\}$, và bộ gán nhãn sẽ đưa ra chuỗi nhãn $\{t_1, \dots, t_n\}$ có xác suất cao nhất.

$$P(t_1, \dots, t_n | c_1, \dots, c_n) = \prod_{i=1}^n P(t_i | h_i) \quad [2.6]$$

Và xác suất có điều kiện cho mỗi nhãn được gán t và ngữ cảnh h được tính như sau:

$$P(t|h) = \frac{p(h,t)}{\sum_{t' \in T} p(h,t')} \quad [2.7]$$

2.1.6.3 ***Ước lượng bộ tham số cho mô hình [11]***

Thuật toán **Generalized Iterative Scaling – GIS** – là thuật toán giúp xác định các trọng số α cho mô hình Maximum Entropy. Thuật toán này có 3 yêu cầu về tập đặc trưng:

- ❖ Giá trị của các đặc trưng là 1 hoặc 0
- ❖ Với mỗi bộ mẫu thì phải có ít nhất một đặc trưng có giá trị 1
- ❖ Tổng các giá trị đặc trưng cho mỗi bộ mẫu phải có cùng giá trị. Nghĩa là phải tồn tại một hằng số C sao cho

$$\sum_i f_i(h,t) = C$$

Trong đó, yêu cầu thứ hai có thể được thỏa bằng cách thêm một đặc trưng luôn trả ra giá trị 1. Trong mô hình trên, đặc trưng mặc định đóng vai trò thỏa mãn yêu cầu này.

Yêu cầu thứ ba có thể được thỏa bằng cách thêm một đặc trưng sửa lỗi có giá trị là $C - \sum_i f_i(h,t)$. Giá trị của hằng số C được chọn theo công thức sau:

$$C = \max_{x \in E} \sum_{i=1}^k f_i(h,t)$$

Đặc trưng mới này là một đặc trưng đặc biệt và có thể lớn hơn 1.

Về mặt lý thuyết, khi ta thêm vào các đặc trưng mới thì nó sẽ ảnh hưởng đến việc phân loại của hệ thống. Tuy nhiên, hai đặc trưng mới này về ý nghĩa đều phụ thuộc hoàn toàn vào các đặc trưng còn lại, và nó hầu như không thêm bất kỳ một thông tin mới nào. Và do đó, nó không ảnh hưởng đến việc phân loại của hệ thống

Thuật toán **GIS** được thực hiện qua các bước sau:

❖ **Bước 1:** Dựa trên tập mẫu học $S = \{(h_i, t_i) : i \in N\}$ với N là số lượng mẫu học có được từ tập huấn luyện, ta tính các giá trị ước lượng mong muốn (desired expectation) K_j theo công thức:

$$K_j = \tilde{E}f_j = \frac{1}{N} \sum_{i=1}^N f_j(h_i, t_i) \quad [2.8]$$

❖ **Bước 2:** Khởi tạo ngẫu nhiên các giá trị cho α_j (ta chọn giá trị khởi tạo mặc định là 1 để dễ hội tụ).

$$\alpha_j^{(0)} = 1$$

❖ **Bước 3:** Tính các giá trị mong muốn ứng với bộ tham số α_j hiện tại theo công thức:

$$\alpha_j^{(n+1)} = \alpha_j^{(n)} \left[\frac{\tilde{E}f_j}{E^{(n)}f_j} \right]^{\frac{1}{C}} \quad [2.9]$$

$$E^{(n)}f_j = \sum_{(h,t) \in (H \times T)} p^{(n)}(h,t) f_j(h,t) \quad [2.10]$$

$$p^{(n)}(h,t) = \pi \prod_{j=1}^k (\alpha_j^{(n)})^{f_j(h,t)} \quad [2.11]$$

❖ **Bước 4:** Nếu chưa hội tụ, quay lại Bước 3.

2.2 Bài toán phân loại văn bản

Phân loại văn bản là bài toán đã được nghiên cứu khá lâu trên nhiều ngôn ngữ. Tuy nhiên chúng tôi chỉ đi vào hai ngôn ngữ chính:

- **Tiếng Anh:** đây là ngôn ngữ đã được quan tâm khá sớm trong bài toán phân loại văn bản, là ngôn ngữ có tính phổ dụng trên toàn thế giới.
- **Tiếng Việt:** đây là ngôn ngữ đang được quan tâm trong dự án, đơn giản vì Việt Nam chúng ta nằm trong vùng đại dịch cúm có thể xảy ra nhiều.

2.2.1 Một số khái niệm cơ bản

❖ Hạng (Term)

Hạng trong một văn bản có thể là từ đơn (Single words) hoặc ngữ (phrases) ví dụ như ngữ danh từ, ngữ động từ...

❖ Lớp (Category)

Lớp của các tài liệu là sự gom nhóm các tài liệu có nội dung tương tự nhau.

❖ Trọng số (Weight)

Là một giá trị đặc trưng cho hạng, giá trị này thường là số thực. Công thức người ta thường dùng là **TFIDF** (Term Frequency and Inverse Document Frequency) và một số mở rộng của nó như **logTF_IDF**, **TF_IWF**,... (sẽ được trình bày kỹ hơn trong chương sau).

❖ Đặc trưng (Feature)

Đặc trưng của văn bản là những **hạng (term)** trong văn bản. Cơ bản thì có 2 loại thuật toán (algorithm) [24] để biểu diễn không gian đặc trưng (feature space) trong quá trình phân lớp. Một loại được gọi là “chọn lựa đặc trưng” (feature selection) và một loại được gọi là “rút trích đặc trưng” (feature extraction).

▪ Chọn lựa đặc trưng (Feature Selection) [24]

Chọn lựa đặc trưng là chọn lựa 1 tập con (subset) các đặc trưng biểu diễn từ không gian đặc trưng gốc.

▪ **Rút trích đặc trưng (Feature Extraction) [24]**

Rút trích đặc trưng là biến đổi (transform) không gian đặc trưng gốc (đầu vào) thành một không gian đặc trưng nhỏ hơn để giảm chiều đặc trưng. So sánh với chọn lựa đặc trưng, rút trích đặc trưng không chỉ có thể giảm chiều đặc trưng mà còn thành công trong việc giải quyết các vấn đề tính nhiều nghĩa (**polysemy**) và tính đồng nghĩa (**synonym**) (của từ) ở một mức độ có thể chấp nhận được.

2.2.2 Tổng quan về bài toán phân loại văn bản tự động trên tiếng Anh

Phân loại văn bản tự động là một bài toán rất được quan tâm trong những năm gần đây. Để phân loại, rất nhiều cách tiếp cận đã được áp dụng như dựa vào từ khóa, dựa vào thống kê tần số xuất hiện của các từ trong văn bản... Đối với tiếng Anh, bài toán phân loại đã sớm xuất hiện từ những năm đầu thập kỷ 60, tuy vậy, đến đầu thập kỷ 90, nó mới được đầu tư nghiên cứu sâu rộng khi cách tiếp cận máy học ngày càng trở nên phổ biến. Trong cách tiếp cận này, một quá trình quy nạp tổng quát (hay còn gọi là quá trình học) sẽ tự động xây dựng một “người phân lớp” cho phân lớp c_i bằng cách ghi nhận những đặc trưng có được của tài liệu thuộc lớp c_i và những tài liệu không thuộc phân lớp c_i . Từ những đặc trưng này, quá trình thu thập có tính chất quy nạp sẽ dự đoán các đặc trưng sẽ phải có đối với những tài liệu thuộc phân lớp c_i . Trong lĩnh vực máy học, quá trình học cách phân loại như trên được xem là quá trình học có giám sát.

Một số phương pháp máy học đã áp dụng thành công trên bài toán phân loại văn bản tiếng Anh có thể kể đến như: mô hình hồi quy [45], phương pháp phân loại k – người láng giềng gần nhất (k- Nearest Neighbors) [2], phương pháp xác suất Naïve Bayes [49], cây quyết định (decision trees) [2], phương pháp tăng cường (boosting method) [51], học luật quy nạp (inductive rule learning) [46], mạng nơron (Neural Network) [40], máy học sử dụng véc tơ hỗ trợ (Support vector machines) [28] ...

Mô hình tổng quát cho bài toán phân loại văn bản:



Hình 9: Quy trình của bài toán phân loại văn bản

Hầu hết các phương pháp máy học áp dụng cho bài toán phân loại văn bản đều sử dụng cách biểu diễn văn bản dưới dạng véc tơ đặc trưng. Điểm khác biệt duy nhất chính là không gian đặc trưng được chọn lựa. Tuy nhiên ở đây ta thấy nảy sinh một vấn đề cơ bản: Số lượng từ xuất hiện trong văn bản sẽ rất lớn. Như vậy, mỗi véc tơ có thể có hàng ngàn đặc trưng, hay nói cách khác mỗi véc tơ sẽ có số chiều rất lớn. Do vậy các véc tơ sẽ không đồng nhất về kích thước.

Để giải quyết vấn đề thông thường chúng ta sẽ chọn lựa những đặc trưng được đánh giá là hữu ích, bỏ đi những đặc trưng không quan trọng. Đối với phân loại văn bản, quá trình này rất quan trọng bởi vì véc tơ văn bản có số chiều rất lớn ($>>10000$), trong đó số thành phần dư thừa cũng rất nhiều. Vì vậy các phương pháp chọn lựa đặc trưng rất hiệu quả trong việc giảm chiều của véc tơ đặc trưng văn bản, chiều của véc tơ văn bản sau khi được giảm chỉ còn lại khoảng 1000 đến 5000 mà không mất đi độ chính xác phân loại.

2.2.3 Các phương pháp tiếp cận cho bài toán

Qua nghiên cứu nhiều công trình trên thế giới nói chung cũng như cho tiếng Việt nói riêng, chúng tôi nhận thấy, bài toán phân loại tài liệu thường có 3 cách tiếp cận chính:

- ❖ Tiếp cận theo hướng dãy các từ (**Bag of Words – BOW**) [25]
- ❖ Tiếp cận theo hướng mô hình ngôn ngữ thống kê **N-Gram** [26]
- ❖ Kết hợp 2 phương pháp trên [27]

Trong luận văn này, chúng tôi quan tâm đến 2 cách tiếp cận đầu tiên cho tiếng Việt. Trong mỗi cách tiếp cận chúng tôi sẽ phân tích, so sánh và đánh giá các ưu, khuyết điểm, những khó khăn và thuận lợi khi áp dụng cho ngôn ngữ tiếng Việt.

2.2.4 Phân loại văn bản tiếp cận theo hướng dãy các từ (**Bag of words –BOW based Approach**) [25]

2.2.4.1 Phương pháp xác suất Naïve Bayes

Naïve Bayes là phương pháp phân loại dựa trên xác suất được sử dụng rộng rãi trong lĩnh vực máy học, được sử dụng lần đầu tiên trong lĩnh vực phân loại bởi Maron năm 1961 và ngày càng trở nên phổ biến. Trong bài toán phân loại văn bản, phương pháp xác suất Naïve Bayes đã được áp dụng rộng rãi [28][2][3][4][5][33][34].

❖ Giới thiệu về bộ phân lớp Naïve Bayes

Naïve Bayes là phương pháp phân lớp dựa trên thống kê

$$\text{Naïve Bayes} = \text{Định lý Bayes} + \text{Các giả định độc lập}$$

❖ Định lý Bayes

Cho X, Y là hai tập hợp. Ta gọi tần suất hiện của X trong Y là

$$P(X | Y) = \frac{P(Y | X)P(X)}{P(Y)} \quad [2.12]$$

Trong đó:

- $P(X|Y)$: số phần tử của tập hợp X trong tập hợp Y
- $P(Y|X)$: số phần tử của tập hợp Y trong tập hợp X

- $P(X)$: số phần tử của tập hợp X
- $P(Y)$: số phần tử của tập hợp Y

❖ Ý tưởng của phương pháp Naïve Bayes áp dụng cho bài toán phân loại văn bản

Ý tưởng cơ bản của phương pháp xác suất Bayes là dựa vào xác suất có điều kiện của từ hay đặc trưng xuất hiện trong văn bản với chủ đề để dự đoán chủ đề của văn bản đang xét. Điểm quan trọng cơ bản của phương pháp này là các giả định độc lập:

- Các từ hay đặc trưng của văn bản xuất hiện là độc lập với nhau.
- Vị trí của các từ hay các đặc trưng là độc lập và có vai trò như nhau.

Giả sử ta có:

- n chủ đề (lớp) đã được định nghĩa $c_1, c_2, c_3, \dots, c_n$
- Tài liệu mới cần được phân loại d_j

Để tiến hành phân loại tài liệu d_j , chúng ta cần phải tính được tần suất xuất hiện của các lớp c_i ($i=1,2,\dots,n$) trong tài liệu d_j

Sau khi tính được xác suất của văn bản đối với các chủ đề, theo luật Bayes, văn bản sẽ được phân lớp vào chủ đề c_i nào có xác suất cao nhất.

❖ Công thức

$$P(c_i | \vec{d}_j) = \frac{P(c_i)P(\vec{d}_j | c_i)}{P(\vec{d}_j)} \quad [2.13]$$

$$\vec{d}_j = \langle w_{1j}, \dots, w_{|T|j} \rangle$$

$$P(\vec{d}_j | c_i) = \prod_{k=1}^{|T|} P(w_{kj} | c_i) \quad [2.14]$$

$$H_{Bayes}(\vec{d}_j) = \arg \max_{c_i \in c} \left(\frac{P(c_i) \cdot \prod_{k=1}^{|T|} P(w_{kj} | c_i)}{P(\vec{d}_j)} \right) \quad [2.15]$$

Trong đó :

$P(c_i | \vec{d}_j)$: xác suất mà lớp c_i là lớp mà tài liệu d_j thuộc về

$P(c_i)$: xác suất lớp c_i trong tập huấn luyện

$|T|$: số lượng đặc trưng của văn bản

❖ **Mô hình phân loại bằng phương pháp xác suất Bayes**



Hình 10: Mô hình phân loại văn bản Naive Bayes

2.2.4.2 Phương pháp phân loại k người láng giềng gần nhất

kNN là phương pháp truyền thống khá nổi tiếng về hướng tiếp cận dựa trên thống kê đã được nghiên cứu trên nhận dạng mẫu hơn suốt bốn thập kỷ qua. kNN [2][3] là phương pháp đơn giản và không cần huấn luyện để nhận dạng mẫu trong tập huấn luyện như các phương pháp khác.

❖ **Ý tưởng**

Cho một tài liệu đầu vào cần kiểm nghiệm, hệ thống sẽ tìm k tài liệu trong tập huấn luyện thỏa mãn điều kiện có độ tương đồng với tài liệu cần kiểm nghiệm là cao nhất. Sau đó, hệ thống sẽ tính toán tổng trọng số của mỗi phân lớp dựa vào k tài liệu tìm được và sử dụng sự phân loại của k tài liệu đó để dự đoán lớp mà tài liệu đưa vào thuộc về. Phân lớp của tài liệu kiểm nghiệm chính là phân lớp có trọng số cao nhất và thỏa điều kiện lớn hơn một giá trị ngưỡng cho phép.

Đôi khi trong phương pháp kNN, chúng ta thường xây dựng thêm giai đoạn học với mục tiêu chính là véc tơ hóa các văn bản trong tập huấn luyện nhằm giảm thiểu thời gian lặp đi lặp lại quá trình này trong giai đoạn phân loại mẫu.

❖ **Giai đoạn huấn luyện của thuật toán kNN**

Trong giai đoạn này tất cả các tài liệu đã được chuyển sang dạng không gian véc tơ n chiều. Ví dụ một tài liệu X_i được định nghĩa như sau:

$$\vec{X}_i = (f_{i1}, f_{i2}, f_{i3}, \dots, f_{in})$$

Trong đó f_{ij} được định nghĩa là giá trị đặc trưng thứ j của văn bản X_i .

Đặc trưng của tài liệu là những thành phần mà dùng để biểu diễn và mô hình tài liệu. Tập hợp các đặc trưng này ta gọi là tập đặc trưng (feature set) và một lược đồ trọng số được áp dụng để biểu diễn tầm quan trọng của mỗi đặc trưng trong tài liệu.

❖ **Phương pháp tính khoảng cách hay độ tương đồng giữa các văn bản**

Một tài liệu được biểu diễn dưới dạng véc tơ, trong đó các thành phần của vecto là trọng số tương ứng với những đặc trưng của tài liệu. Giữa hai tài liệu X_i , X_j khoảng cách hay độ tương đồng được định nghĩa như sau:

Khi sử dụng giá trị khoảng cách Euclide:

$$d(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{r=1}^n (x_{ir} - x_{jr})^2} \quad [2.16]$$

(r là kích thước hay số chiều của véc tơ đặc trưng)

Khi sử dụng giá trị đo Cosin:

$$d(\vec{x}_i, \vec{x}_j) = \frac{\vec{x}_i \cdot \vec{x}_j}{|\vec{x}_i| \cdot |\vec{x}_j|} \quad [2.17]$$

❖ **Hàm quyết định trong giải thuật phân loại kNN**

Hàm quyết định trong giải thuật phân loại văn bản **kNN** là hàm gán nhãn cho tài liệu đưa vào sử dụng k lân cận gần nhất. Có nhiều hàm quyết định ta có thể sử dụng như: **DVF** (discrete value function), **SWF** (similarity – Weighted Function), **ASWF** (Average similarity – Weighted Function).

Hàm **DVF** được định nghĩa như sau:

$$TC(\vec{x}_q) \leftarrow \arg \max_{c_j \in C} \sum_{\substack{d_i \in kNN}}^k y(d_i, c_j) \quad [2.18]$$

Trong đó: $y(d_i, c_j) \in \{0,1\}$ là kết quả của quá trình phân loại tài liệu d_i với lớp c_j ($y = 1$ nếu d_i thuộc về phân lớp c_j và $y = 0$ nếu d_i không thuộc về phân lớp c_j)

Hàm **DVF** sẽ quyết định phân lớp của tài liệu x_q là c_i khi c_i là nhãn của phân lớp xuất hiện nhiều nhất trong k tài liệu lân cận tìm được.

Với công thức trên, ta thấy **DVF** rất đơn giản, nó không tính khoảng cách hay sự tương đồng của mỗi tài liệu “lân cận” với tài liệu truy vấn mà chỉ dựa vào thông tin về phân lớp có sẵn của tài liệu.

SWF là sự cải tiến của **DVF**. Nó chọn những tài liệu “lân cận” dựa vào sự tương đương của tài liệu cần phân lớp với các tài liệu huấn luyện. Do vậy những tài liệu giống hơn hay gần hơn sẽ có trọng lớn hơn. Hàm quyết định **SWF** được định nghĩa như sau:

$$TC(\vec{x}_q) \leftarrow \arg \max_{c_j \in C} \sum_{\vec{d}_i \in kNN}^k sim(\vec{x}_q, \vec{d}_i) y(\vec{d}_i, c_j) \quad [2.19]$$

Trong đó: $sim(\vec{x}_q, \vec{d}_i)$ là mức độ tương đồng giữa tài liệu đưa vào x_q và tài liệu trong tập huấn luyện d_i . Mặc dù có nhiều cách đo khác nhau, chúng ta sử dụng giá trị đo Cosine để tính mức độ tương đồng giữa 2 tài liệu. **SWF** sẽ có hiệu quả cao nếu tập hợp tài liệu huấn luyện của ta có độ lớn phù hợp.

Ngoài ra, **SWF** còn được định nghĩa như sau

$$TC(\vec{x}_q) \leftarrow \arg \max_{c_j \in C} p(\vec{x}_q, c_j) \quad [2.20]$$

$$p(x_q, c_j) = \begin{cases} 1 & \text{nếu } \sum_{d_i \in kNN} sim(x_q, d_i) y(d_i, c_j) - b \geq 0 \\ 0 & \text{nếu ngược lại} \end{cases} \quad [2.21]$$

Với b là giá trị ngưỡng định trước.

ASWF là cải tiến của **SWF**. Nó tính giá trị trọng số trung bình của k tài liệu gồm tài liệu đưa vào nhất

$$TC(\vec{x}_q) \leftarrow \frac{\arg \max_{c_j \in C} \sum_{\vec{d}_i \in kNN}^k sim(\vec{x}_q, \vec{d}_i) y(\vec{d}_i, c_j)}{\sum_{\vec{d}_i \in kNN}^k y(\vec{d}_i, c_j)} \quad [2.22]$$

❖ Giá trị của k trong kNN

Giá trị của k trong **kNN** cho ta biết số tài liệu “lân cận” gần nhất mà ta chọn. Nếu toàn bộ các tài liệu huấn luyện trong tập huấn luyện được sử dụng khi

phân lớp một tài liệu mới thì ta gọi là một phương thức toàn cục (global method) còn nếu chỉ sử dụng một vài tài liệu huấn luyện thì ta gọi là phương thức cục bộ (local method).

Nếu ta sử dụng hàm **DVF** là hàm quyết định thì phương thức toàn cục (global method) không thể sử dụng vì khi sử dụng hàm này thì những tài liệu có khoảng cách rất xa tài liệu cần phân loại cũng có tác động nhiều như là những tài liệu có mức độ tương đồng rất lớn với tài liệu cần phân loại. Ngược lại, nếu chúng ta sử dụng hàm **SWF** hay **ASWF** làm hàm quyết định thì ta có thể dùng phương thức toàn cục này bởi vì những tài liệu có khoảng cách xa (mức độ tương đồng ít) với tài liệu cần phân loại sẽ có ảnh hưởng không đáng kể đến hiệu quả của hàm quyết định. Nhưng một bất lợi của phương thức toàn cục là thuật toán sẽ chạy chậm. Chính vì vậy, việc quyết định chọn k bao nhiêu cho thích hợp cũng sẽ cho ra kết quả khác nhau trong phương pháp **kNN**. Thông thường trong phương pháp phân loại văn bản bằng **kNN** trên tiếng Anh, k được chọn từ 30 đến 45 [35].

2.2.4.3 Phương pháp sử dụng mạng nơron

Mạng nơron nhân tạo (Neural network – NN) là phương pháp máy học cung cấp phương pháp hiệu quả để tạo ra các giá trị xấp xỉ của những hàm có giá trị thực, giá trị rời rạc, véc tơ. NN mô phỏng theo hệ thống sinh học thực tế, với các tế bào thần kinh gọi là nơron liên kết với nhau tạo thành một mạng gọi là mạng nơron. Mỗi nơron nhận một hoặc nhiều giá trị đầu vào và tạo ra một giá trị thực duy nhất ở đầu ra, giá trị ở đầu ra này có thể trở thành đầu vào cho một nơron khác.

❖ Khái niệm mạng nơron

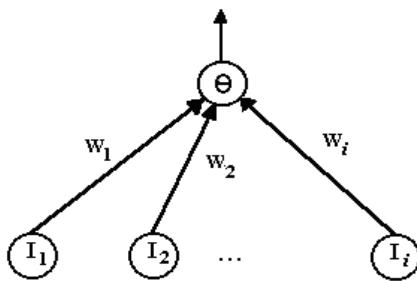
Mạng nơron nhân tạo là phương pháp máy học cung cấp phương pháp hiệu quả để tạo ra các giá trị xấp xỉ của những hàm có giá trị thực, giá trị rời rạc, véc tơ. NN mô phỏng theo hệ thống sinh học thực tế, với các tế bào thần kinh gọi là nơron liên kết với nhau tạo thành một mạng gọi là mạng nơron. Mỗi nơron nhận một hoặc nhiều giá trị đầu vào và tạo ra một giá trị thực duy nhất ở đầu ra, giá trị ở đầu ra này có thể trở thành đầu vào cho một nơron khác.

❖ Perceptron

Perceptron là một dạng đơn giản nhất của mạng nơron. Nó chỉ gồm có một nơron. Nó nhận một véc tơ giá trị thực ở đầu vào, tính toán, kết hợp những giá trị đầu vào này và cho ra một giá trị đầu ra duy nhất. Perceptron là thành phần quyết định giá trị đầu vào thuộc một trong 2 lớp (ứng với 2 giá trị đầu ra của perceptron là 0 và 1. Cho một véc tơ đầu vào ($x_1, x_2, x_3, \dots, x_n$), giá trị đầu ra của perceptron được tính như sau:

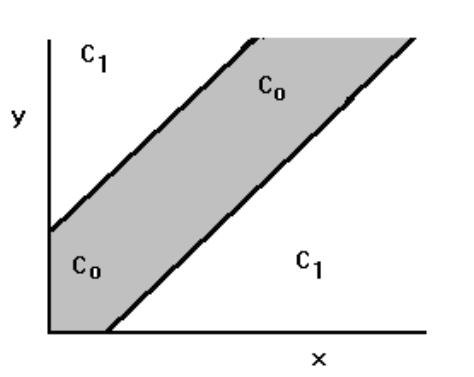
$$y(x_1, x_2, \dots, x_n) = \begin{cases} 1 \text{ nếu } \sum_{i=1}^n w_i x_i + \theta > 0 \\ 0 \text{ nếu ngược lại} \end{cases} \quad [2.23]$$

Trong đó w_i là trọng số xác định mức độ ảnh hưởng của đầu vào tương ứng x_i , θ là ngưỡng. Quá trình học trong một perceptron bao gồm chọn ra giá trị tốt nhất các giá trị w và θ dựa trên tập mẫu huấn luyện.



Hình 11: Mạng Perception

Tuy nhiên perceptron có một giới hạn là không giải quyết được các vấn đề rời rạc phi tuyến (minh họa ở hình 13).



Hình 12: Minh họa phân lớp

Để khắc phục tình trạng này ta sử dụng mạng nơron nhiều lớp dùng thuật giải lan truyền ngược để học.

❖ Mạng nơron nhiều lớp

Mạng nơron nhân tạo nhiều lớp bao gồm tập hợp các liên kết qua lại bên trong giữa các nơron trên nguyên tắc: đầu ra của mỗi nơron được liên kết thông qua các trọng số đến các nơron khác hoặc đến chính nó. Như vậy, việc bố trí các nơron và sơ đồ liên kết qua lại giữa chúng sẽ hình thành một kiểu mạng nơron nhân tạo.

Có nhiều thuật toán có thể dùng để huấn luyện mạng nơron: Học có giám sát và học không có giám sát. Trong phương pháp học có giám sát chúng ta sử dụng một tập hợp các mẫu chứa tập hợp các đặc trưng đầu vào và đầu ra mong muốn cho mỗi mẫu đó. Ta gọi là học có giám sát vì trong suốt quá trình học trọng số của mạng sẽ được điều chỉnh cho đến khi đầu ra đạt gần đến giá trị mong muốn nhất. Thuật giải lan truyền ngược là một phương pháp rất hữu hiệu trong học có giám sát. Trong học không giám sát chúng ta chỉ có tập giá trị của đặc trưng đầu vào mạng sẽ thi hành các thủ tục gom nhóm, kết hợp để học những lớp đã được đưa ra trong tập huấn luyện.

❖ Thuật giải lan truyền ngược

Cho không gian các mẫu học (x,t) , x là giá trị cần huấn luyện, t là giá trị kết quả đích (đầu ra mong muốn) của quá trình huấn luyện, hệ số học η . Qui định chỉ số của lớp là tăng dần từ lớp đầu vào đến lớp đầu ra. Thuật giải lan truyền ngược được tóm tắt như sau:

- Tạo mạng truyền thẳng có n_{in} Nơron đầu vào, n_{Hidden} Nơron trên mỗi lớp ẩn và h lớp ẩn trong mạng, với n_{out} Nơron đầu ra.
- Khởi tạo bộ trọng cho mạng với giá trị nhỏ.
- Trong khi <Điều kiện kết thúc chưa thỏa> làm

Với mỗi cặp (x,t) trong không gian mẫu huấn luyện thực hiện:

- **Xét lớp nhập:**

Truyền x qua mạng, tại mỗi lớp xác định đầu ra của mỗi Noron, quá trình này được thực hiện cho đến lớp xuất tuỳ theo cấu trúc mạng cụ thể.

- **Xét lớp xuất:**

Đối với đầu ra o_k của Noron k trong lớp xuất K, xác định sai số σ_k của nó:

$$\sigma_k = o_k(1 - o_k)(t_k - o_k) \quad [2.24]$$

Chuyển sang lớp ẩn L kế nó, đặt L = K-1

- **Xét các lớp ẩn:**

Với mỗi Noron l trên lớp ẩn L, xác định sai số σ_l của nó:

$$\sigma_l = o_l(1 - o_l) \sum_{i \in L+1} w_{il} \sigma_i \quad [2.25]$$

Cập nhật lại trọng số có trong mạng w_{il}

$$w_{ji} = w_{ji} + \Delta w_{ji} \quad \text{Với } \Delta w_{ji} = \eta \sigma_j o_{ji} \quad [2.26]$$

- Nếu $L > 1$ thì chuyển sang lớp ẩn kế tiếp: $L = L - 1$ và quay lại bước 3. Ngược lại thì chọn cặp (x, t) mới trong không gian học và quay lại bước 3.

❖ Mạng Noron dùng trong phân loại văn bản

Trong lĩnh vực phân loại văn bản, mô hình mạng Noron (NN) đã được áp dụng khá phổ biến. Hình thức đơn giản nhất của bộ phân loại dùng mạng Noron là Perceptron với phương pháp phân loại tuyến tính [36][37]). Một hình thức phân loại tuyến tính khác sử dụng mạng Noron là dạng hồi quy đã được đưa ra và kiểm nghiệm bởi [39][12].

Phương pháp mạng Noron phi tuyến trong phân loại văn bản (Lam và Lee, 1999 [41]; Ruiz và Srinivasan, 1999 [42]; Weigend et al, 1999 [40]) là mô hình áp dụng mạng Noron với nhiều tầng trong đó thể hiện ảnh hưởng tác động lẫn nhau cao hơn giữa các đặc trưng mà mạng Noron học được.

Trong phân loại văn bản hiện nay, mạng Noron thường dùng là mạng chỉ có một tầng nhập và một tầng xuất, không có tầng ẩn, và dùng thuật toán lan truyền ngược để huấn luyện.

2.2.4.4 Phương pháp phân loại văn bản bằng cây quyết định

Phương pháp dựa trên xác suất là một phương pháp có bản chất tự nhiên mà ảnh hưởng của nó khó thể giải thích rõ hết bởi con người. Tuy nhiên, cũng có một lớp các thuật toán không sử dụng xác suất hay còn gọi là không sử dụng số học mà thay vào đó là sử dụng các mô hình thể hiện. Trong những phương pháp này có thể kể đến hai phương pháp điển hình là phương pháp học luật quy nạp và cây quyết định.

Phương pháp phân loại văn bản bằng cây quyết định đã được sử dụng như một công cụ phân lớp chính trong các nghiên cứu của Fuhr và đồng nghiệp năm 1991 [45], Lewis và Catlett năm 1994 [49], Lewis và Ringuette năm 1994 [47]. Hay được xem như là bộ phân lớp cơ bản trong nghiên cứu của Cohen và Singer năm 1999 [46], Joachims năm 1998 [28]. Ngoài ra, phương pháp phân loại văn bản bằng cây quyết định cũng được sử dụng như một thành phần trong bộ phân lớp tổng hợp trong nghiên cứu của Li và Jain năm 1998 [33], Schapire và Singer năm 2000 [51], Weiss và đồng sự năm 1999 [25].

Bộ phân lớp cây quyết định Mitchell, 1996 [53] là một dạng cây mà mỗi nút được gán nhãn là một đặc trưng, mỗi nhánh là giá trị trọng số xuất hiện của đặc trưng trong văn bản cần phân lớp, và mỗi lá là nhãn của phân lớp tài liệu. Việc phân lớp của một tài liệu d_j sẽ được duyệt đệ quy theo trọng số của những đặc trưng có xuất hiện trong văn bản d_j . Thuật toán lặp đệ quy đến khi đạt đến nút lá và nhãn của d_j chính là nhãn của nút lá tìm được. Thông thường việc phân lớp văn bản nhị phân sẽ tương thích với việc dùng cây nhị phân.

Có rất nhiều bộ chuẩn cho phương pháp học của cây quyết định, và hầu hết các cách tiếp cận đều có thể dùng cho bài toán phân loại văn bản bằng cây quyết định. Các phương pháp có thể kể đến như ID3 (Fuhr et al, 1991 [45]), C4.5 (Cohen

và Hirsh, 1998 [48]), (Cohen và Singer, 1999 [46]), (Joachims, 1998 [28]), (Lewis và Catlett, 1994 [49]) hay C5 (Li và Jain, 1998 [33])

❖ **Gán nhãn phân lớp cho quá trình huấn luyện của cây quyết định**

Một vấn đề khả thi thường dùng cho việc học của cây quyết định trong phân lớp c_i chính là ở đặt điểm “chia để trị”, có thể mô tả như sau:

Kiểm tra có phải tất cả các tài liệu huấn luyện có được gán cùng nhãn (bao gồm cả nhãn c_i và không phải c_i)

Nếu không, chọn lựa đặc trưng t_k , phân chia tập huấn luyện thành các lớp tài liệu mà có cùng giá trị với t_k , sau đó đặt mỗi lớp vào một cây con. Quá trình sẽ lặp đi quy cho đến khi đạt đến các nút lá của cây. Như thế tất cả tài liệu mỗi nút đều có cùng nhãn với nhãn của nút lá.

Trong quá trình phân chia, bước quan trọng là chọn được nhãn t_k mà nó có tác dụng cho quá trình phân chia. Tuy nhiên đối với những cây quá đầy đủ, các nhánh của cây có thể trở nên quá đặc biệt cho việc huấn luyện. Chính vì thế, hầu hết các phương pháp huấn luyện của cây quyết định đều bao gồm quá trình tia cành, tức là cắt bỏ đi những cành nào quá đặc biệt có thể ảnh hưởng gây tình trạng “quá khớp”.

2.2.4.5 Phân loại văn bản bằng phương pháp hồi quy

Hồi quy được định nghĩa là hàm xấp xỉ giá trị thực ϕ thay cho giá trị nhị phân trong bài toán phân lớp. Hàm ϕ sẽ có nhiệm vụ học từ dữ liệu huấn luyện. Nhiều nghiên cứu về phương pháp phân loại văn bản ứng dụng mô hình hồi quy có thể kể đến như Fuhr và Pfeifer năm 1994 [54], Ittner và đồng sự năm 1995 [55], Lewis và Gale năm 1994 [5], hay Schutze và đồng sự năm 1995[39]. Ở đây, chúng tôi chỉ đề cập đến một mô hình, **LLSF** (Linear Least-Squares Fit) ứng dụng trong bài toán phân loại văn bản.

LLSF là cách tiếp cận ánh xạ được phát triển bởi Yang và Chute năm 1992. Đầu tiên tác giả thử nghiệm phương pháp này trong bài toán xác định từ đồng nghĩa và sau đó tiếp tục áp dụng vào bài toán phân loại văn bản năm 1994 [29].

Trong phương pháp LLSF, mỗi văn bản trong tập huấn luyện sẽ được biểu diễn dưới dạng một cặp vec tơ vào và ra. Vec tơ đầu vào bao gồm các đặc trưng và trọng số của nó. Vec tơ đầu ra bao gồm các chủ đề với các trọng số nhị phân của văn bản ứng với vec tơ đầu vào.

Quá trình phân lớp chính là việc giải phương trình các cặp vec tơ đầu vào và ra, cũng đồng nghĩa là ta sẽ tính toán được một ma trận đồng hiện của hệ số hồi quy giữa từ và chủ đề. Mục đích của phương pháp này là tìm giá trị lỗi nhỏ nhất từ ma trận đồng hiện cho bởi công thức.

$$F_{ls} = \arg \min_F \|FA - B\|^2 \quad [2.27]$$

Trong đó:

- A, B là đại diện cho tập ngữ liệu huấn luyện (các cột trong ma trận tương ứng các giá trị của vec tơ đầu vào và ra).

- F_{ls} là ma trận kết quả chỉ ra một ánh xạ từ một văn bản đầu vào bất kỳ vào vec tơ chủ đề đã được gán trọng số, hay nói cách khác chính là thể hiện mức độ quan hệ giữa các đặc trưng đầu vào và các chủ đề phân lớp.

Nhờ vào việc sắp xếp trọng số các chủ đề, ta được danh sách các chủ đề có thể gán cho văn bản cần phân loại. Từ danh sách đó ta sẽ tìm được chủ đề cho văn bản cần phân loại.

2.2.4.6 Phân loại văn bản sử dụng Support Vector Machines – SVM

❖ Giới thiệu

SVM là phương pháp nhận dạng do Vladimir N. Vapnik đề xuất năm 1995. **SVM** là phương pháp nhận dạng dựa trên lý thuyết học thống kê ngày càng được sử dụng phổ biến trong nhiều lĩnh vực, đặc biệt là lĩnh vực phân loại mẫu và nhận dạng mẫu. Đồng thời có nhiều tính năng ưu việt so với các phương pháp cổ điển khác: dễ dàng xử lý, xử lý với tính ổn định cao trên dữ liệu phức tạp, có thể có số chiều lớn và quan trọng hơn cả là khả năng xử lý tổng quát.

❖ Cơ sở lý thuyết SVM:

✓ Hàm phân cách tuyến tính(Linear Discriminant Function)

Nếu chúng ta có 1 hàm tuyến tính gọi là hàm phân cách được định nghĩa như sau:

$$g(x) = w^t x + w_o, \text{ thì :}$$

w: véctơ trọng số (weight véctơ)

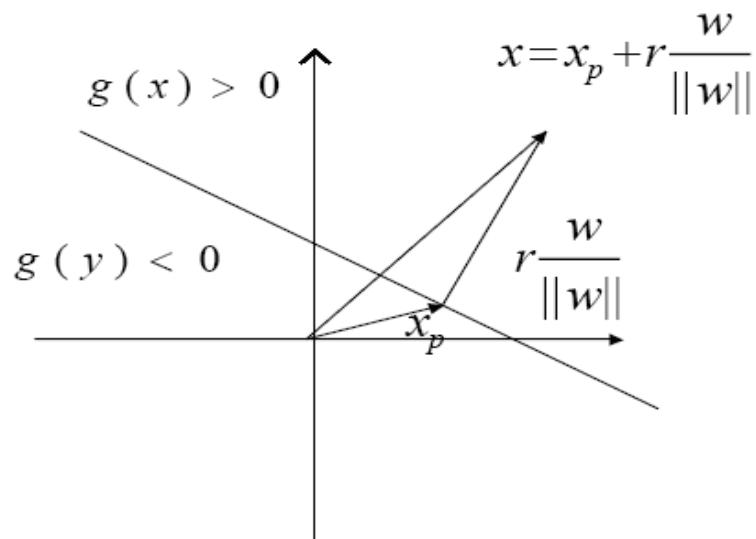
w_o : trọng số ngưỡng (threshold weight)

Nếu chúng ta đặt:

$$x = x_p + r \frac{w}{\|w\|} \text{ trong đó } r \text{ là khoảng cách từ } x \text{ đến mặt phẳng thì}$$

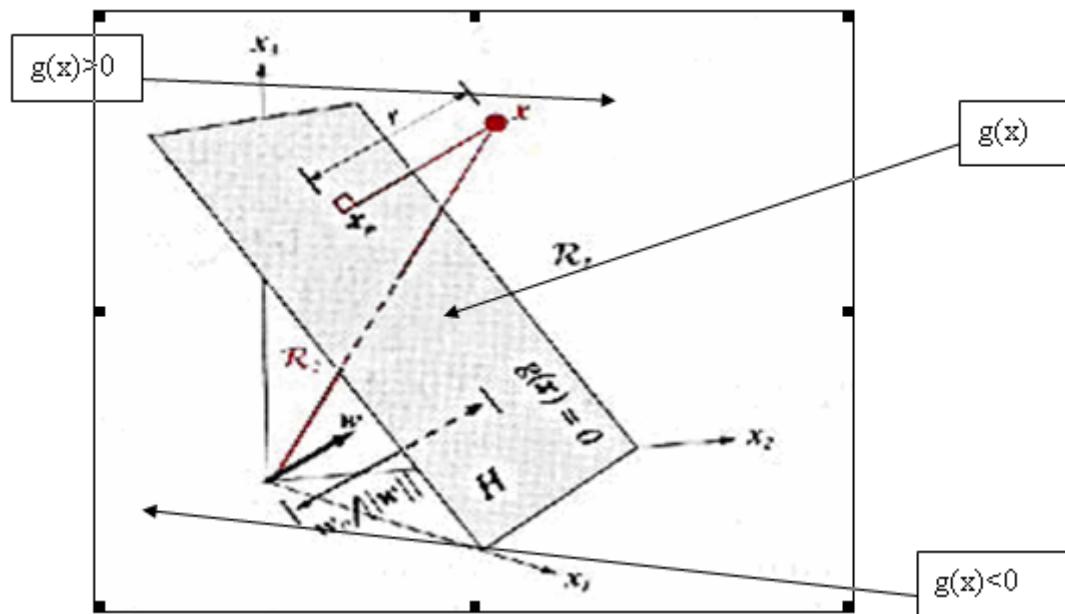
$$\begin{aligned} g(x) &= w^t (x_p + r \frac{w}{\|w\|}) + w_o = w^t x_p + w_o + r \frac{w^t w}{\|w\|} \\ &= g(x_p) + r \frac{\|w\|^2}{\|w\|} = g(x_p) + r \|w\| = r \|w\|, g(x_p) = 0 \end{aligned} \quad [2.28]$$

$$r = \frac{g(x)}{\|w\|} \quad [2.29]$$



Hình 13: Không gian mặt phẳng với x_p

Nhìn hình vẽ trên ta thấy không gian mặt phẳng được chia thành 2 nửa không gian $g(x)>0$ và $g(x)<0$:



Hình 14: Ý tưởng mặt phân cách

Xuất phát từ ý tưởng phân cách 2 mặt phẳng con trên, hàm phân cách tuyến tính được định nghĩa như sau

$$g(x) = w^t x + w_0 \quad [2.30]$$

w: véctơ trọng số (weight vector)

w_o : trọng số ngưỡng (threshold weight)

Trường hợp phân loại 2 lớp: ω_1, ω_2 , lớp ω_i được chọn nếu như:

$$g(x) = w^t x + w_o > 0 \text{ và } \omega_2 \text{ nếu như } g(x) = w^t x + w_o < 0$$

- ✓ **Định nghĩa siêu mặt:** mặt phẳng $g(x)=0$ σ trên sẽ tạo ra mặt quyết định(decision surface) mà nó chia cắt những điểm được gán cho ω_1, ω_2 . Nếu $g(x)$ là tuyến tính thì mặt quyết định trên gọi là siêu mặt(hyperplane)

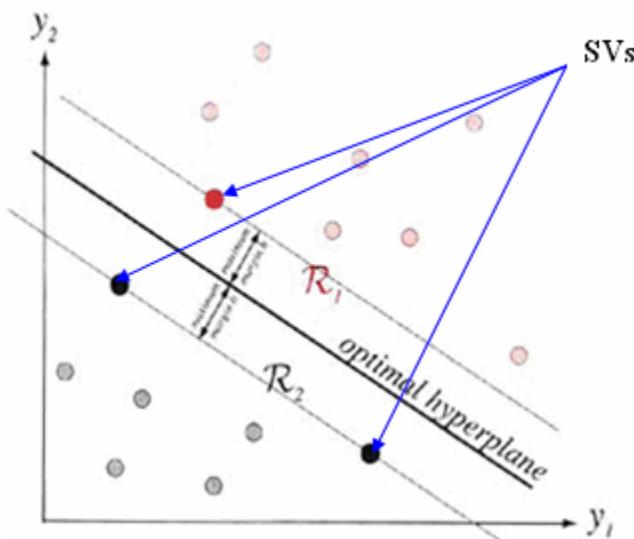
Trường hợp đa lớp: phức tạp hơn nhiều so với trường hợp 2 lớp, chúng ta có 2 phương pháp:

Cách 1: chúng ta chia cắt những điểm được gán cho w_i từ những điểm khác không được gán cho w_i

Cách 2: sử dụng 1 siêu mặt cho mỗi cặp lớp, lúc đó chúng ta sẽ cần $c(c-1)/2$ hàm phân cách tuyến tính.

❖ Support Vectors(SVs)

SVs là những mẫu gần nhất có khoảng cách b từ siêu mặt.



Hình 15: Minh họa SVs

✓ **SVM tuyế́n tính**

Nhiệm vụ phương pháp **SVM** sẽ tìm ra siêu mặt với khoảng cách cực đại từ những mẫu được huấn luyện gần nhất, gọi là các siêu mặt tối ưu(optimal hyperplane).

Giả sử cho tập học $S = \{x_i, y_i\}_{i=1}^n$, trong đó $x_i \in R^n$ và

$y_i \in \{+1, -1\}$. Mục đích của **SVM** là tìm ra siêu mặt phân cách thỏa:

$$\begin{aligned} w^T x_i + b &\geq 1 \text{ khi } y_i = +1 \\ w^T x_i + b &\leq -1 \text{ khi } y_i = -1 \end{aligned} \quad [2.31]$$

Trong đó $w \in R^n$ và cũng là véctơ pháp tuyế́n của siêu mặt. Nếu như các thành phần của tập học S thỏa bất phương trình (1) thì đây là trường hợp phân cách tuyế́n tính. Trong việc huấn luyện siêu mặt , **SVM** tìm cách cực đại khoảng cách bờ p giữa 2 lớp với $\rho = \frac{2}{|w|}$. Do đó, trường hợp trường hợp truyền tính việc

tìm khoảng cách siêu mặt cũng chính là giải hệ phương trình sau:

Cực tiêu $\Phi_{(w)} = \frac{1}{2} w^T w$ với ràng buộc

$y_i(w^T x_i + b) \geq 1, i = 1, \dots, n$. Để giải quyết bài toán này, người ta đưa ra hàm Lagrangian:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i (y_i \cdot (x_i \cdot w) + b) - 1 \quad [2.32]$$

Trong đó:

$$\frac{\partial}{\partial b} L(w, b, \alpha) = - \sum_{i=1}^l \alpha_i y_i = 0$$

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^l \alpha_i y_i x_i = 0$$

Điều kiện KKT cho bài toán:

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad w = \sum_{i=1}^l \alpha_i y_i x_i, \quad \alpha_i \text{ của Support Vectors là khác } 0:$$

$$\alpha_i \cdot [y_i((x_i \cdot w) + b) - 1] = 0, \quad i = 1, \dots, l \quad [2.33]$$

Vấn đề tối ưu trở thành:

$$\text{Làm cực đại: } W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad [2.34]$$

Với ràng buộc:

$$\alpha_i \geq 0, \quad i = 1, \dots, l, \quad \text{and} \quad \sum_{i=1}^l \alpha_i y_i = 0$$

Suy ra hàm quyết định:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i \alpha_i \cdot (x \cdot x_i) + b \right) \quad [2.35]$$

Trong trường hợp phi tuyến, ta cần ánh xạ tập véctơ dữ liệu vào không gian đặc trưng cao hơn nhiều, bằng cách sử dụng hàm:

$$\phi: R^N \rightarrow F$$

Với hàm xử lý chính tích vô hướng $k(x,y)$ ánh xạ không gian phi tuyến vào không gian đặc trưng:

$$k(x, y) = (\phi(x) \cdot \phi(y)) \quad [2.36]$$

Hàm này gọi là hàm Kernel. Một số hàm kernel cơ bản:

- Linear Kernel: $k(x, y) = (xy)^d$
- Polynominal Kernel: $k(x, y) = (xy + 1)^d$
- RBF(Radial Basis Function) Kernel:

$$k(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\delta^2} \right)$$

- Sigmoid Kernel: $\tanh(\eta(xy) + c)$

✓ SVM phi tuyến sử dụng Kernel

$$\begin{aligned} f(x) &= \operatorname{sgn}\left(\sum_{i=1}^l v_i (\phi(x) \cdot \phi(x_i)) + b\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^l v_i k(x, x_i) + b\right) \end{aligned} \quad [2.37]$$

$$v_i = \alpha_i y_i$$

❖ Ứng dụng SVM

Đây là phương pháp mới, tỏ ra rất hiệu quả trong nhiều lĩnh vực, đặc biệt trong nhận dạng mẫu:

- Face Detection
- Face Recognition
- Hand Written Digit Recognition
- Text Categorization
- 3D object recognition
- Face Pose Recognition
- Color based Classification
- Bio-informatics(protein Homology Detection)

❖ Ứng dụng SVM trong phân loại văn bản

Thorsten Joachims trong công trình [28] của mình đã chứng minh được SVM là phương pháp máy học thích hợp nhất cho bài toán phân loại văn bản tiếng Anh (thử nghiệm trên bộ ngữ liệu của **Reuters**, **20Newsgroups**, **Ohsumed**⁶). Trong luận văn này, chúng tôi chọn SVMs là phương pháp máy học chủ đạo cho bài toán phân loại văn bản tiếng Việt trong cách tiếp cận dãy các từ.

⁶ <http://ai-nlp.info.uniroma2.it/moschitti/corpora.htm>

2.2.4.7 **Những phương pháp khác**

Trong những phần trên, chúng tôi đã cố gắng đưa ra một cái nhìn tổng quan nhất về việc ứng dụng các phương pháp máy học cho bài toán phân loại văn bản. Tuy nhiên, thật khó khăn để có thể liệt kê hết tất cả các phương pháp đã áp dụng đối với bài toán này trên thế giới [2]. Ngoài ra, hiện nay còn có xu hướng sử dụng kết hợp nhiều phương pháp máy học với nhau nhằm tận dụng ưu thế của nhiều phương pháp. Có thể kể đến một số phương pháp kết hợp đáng chú ý: mạng Nơron và Bayes, thuật giải di truyền (Genetic Algorithms) kết hợp với mạng nơron, mô hình độ hỗn loạn cực đại (Maximum Entropy)...

2.2.5 **Phân loại văn bản tiếp cận theo hướng mô hình ngôn ngữ thống kê N-Gram (Statistical N-Gram Language modeling based Approach) [26]**

Như đã trình bày trong phần trước, nhiều thuật toán máy học đã được áp dụng cho bài toán phân loại văn bản tự động như là: Naïve Bayes, **SVMs**, **LLSF**, **Neural Networks**, **k-Nearest neighbor classifiers** ... ([2]). Các phương pháp này đều làm việc dựa trên các hạng hay còn gọi là các thuộc tính (attribute) và các thuộc tính này đều được giả sử là độc lập với nhau. Trong các phương đó, Naïve Bayes là một phương pháp đã được chứng minh là phương pháp ứng dụng thành công cho bài toán phân loại văn bản [2] mặc dù tính đơn giản và giới hạn trong việc giả sử độc lập. Domingos và Pazzani trong công trình [57] thấy rằng bộ phân lớp Naïve Bayes có thể đạt được tỷ lệ phân lớp lỗi tối ưu nếu như có thể can thiệp đến giả sử độc lập của phương pháp này. Trong thực tế, các thuộc tính phụ thuộc lẫn nhau có thể tăng độ chính xác phân lớp trong một vài trường hợp [58][59].

Có khá nhiều nghiên cứu đã hướng vào khuyết điểm này để mong rằng có thể nâng độ chính xác phân lớp lên cao hơn. Một công trình khá nổi tiếng là Tree Augmented Naïve Bayes (**TAN**) classifier của Friedman et al 1997 [58]. Bộ phân lớp này cho phép sự phụ thuộc có cấu trúc cây giữa các biến quan sát ngoài phụ thuộc truyền thống trên biến gốc (root) của cây. Tuy nhiên, quá trình học mạng Bayes theo cấu trúc cây có chi phí khá lớn, và vì thế mô hình này hiếm khi được sử

dụng bài toán phân loại văn bản. Fuchun Peng et al trong công trình [26] đã đề nghị một phương pháp rất hay để giải quyết vấn đề giả sử độc lập, công trình này có tên là mô hình **Chain Augmented Naïve Bayes (CAN)**. Mô hình này đơn giản hơn mô hình trước bằng cách là nó giới hạn tính phụ thuộc giữa các biến đến 1 dây Markov (**Markov chain**) thay vì dùng cây (**tree**). Nhờ đó, mô hình này lại gián tiếp liên quan đến mô hình ngôn ngữ n-gram. Cũng trong công trình này, các tác giả đã đề xuất mô hình ngôn ngữ thống kê ngram kết hợp phương pháp Naïve Bayes ứng dụng cho bài toán phân loại văn bản. Trong luận văn này, chúng tôi sẽ tìm hiểu và thí nghiệm lại phương pháp này cho bài toán phân loại văn bản tiếng Việt, từ đó so sánh các ưu khuyết điểm so với phương pháp truyền thống là cách tiếp cận dây các từ.

Trong chương kế tiếp, chúng tôi sẽ trình bày kỹ hơn về phương pháp này.

2.2.6 Tiếp cận theo hướng kết hợp 2 loại trên (**Combining approach**) [27]

❖ Giới thiệu về cụm từ hay ngữ trong văn bản

Trong quá khứ, rất nhiều nhà nghiên cứu về lĩnh vực rút trích thông tin văn bản đã chứng tỏ được sự hạn chế trong việc tiếp cận bằng cách sử dụng các từ rời rạc. Thay vào đó họ hướng tới việc khai thác bằng các đặc trưng giàu tính ngữ nghĩa hơn, đó chính là việc sử dụng các cụm từ có tính liên kết cao hơn. Về ý nghĩa ngôn ngữ, các cụm từ này thường mang ý nghĩa cao hơn từ nhưng lại không đầy đủ như câu. Những cụm từ như vậy được gọi là “cụm từ ngữ pháp”, tức là được cấu thành theo đúng cấu trúc ngữ pháp của ngôn ngữ. Chính vì thế, trong việc khai thác ý nghĩa, nội dung thông tin văn bản, sử dụng các “cụm từ ngữ pháp” có các thuận lợi sau:

- Sử dụng cụm từ có tính liên kết sẽ có ý nghĩa giàn hơn so với việc sử dụng từ đơn hay các từ gốc trong việc thể hiện các khái niệm chung.
- Sử dụng cụm từ có tính liên kết sẽ cho mức độ nhập nhằng thấp và dễ diễn đạt ý nghĩa hơn so với việc phải dùng các từ riêng lẻ hợp thành.

- Bằng cách sử dụng các cụm từ như những đặc trưng, tài liệu mà trong nó bao hàm các cụm từ sẽ được đánh giá cao hơn so với việc phải dùng các từ riêng biệt không có quan hệ về nội dung với nhau.

Tuy nhiên, không phải hầu hết các cụm ngữ pháp đều thể hiện chính xác nội dung. Do vậy, một số nhà nghiên cứu ngôn ngữ học đã cố gắng giải quyết những khó khăn này bằng cách tìm hiểu khái niệm của một cụm từ trong ý nghĩa thống kê còn gọi là ngữ thống kê thay cho việc tìm hiểu khái niệm trong ý nghĩa ngữ pháp. Ngữ thống kê được định nghĩa là một dãy các từ liên tiếp nhau trong văn bản. Trong xử lý ngôn ngữ, việc sử dụng ngữ thống kê có một số điểm thuận lợi hơn so với “cụm từ ngữ pháp”:

- Ngữ thống kê sẽ được nhận diện dễ dàng hơn mà không phải phụ thuộc quá nhiều vào việc tính toán của các thuật toán.
- Ảnh hưởng của những giá trị ngữ pháp không liên quan có thể được bỏ qua.

Tuy nhiên, việc sử dụng ngữ thống kê cũng sẽ có những bất lợi như việc một số ngữ sẽ không được nhận dạng tốt, hay việc nhận dạng lầm lẫn.

Từ các thuận lợi trên, ngữ thống kê đã cho thấy được giá trị khi sử dụng cho việc chỉ mục văn bản theo nội dung hay nói cách khác là gán nhãn cho tài liệu trong bài toán phân loại văn bản theo những chủ đề đã định nghĩa.

❖ Véc tơ hóa văn bản bằng phương pháp ngữ thống kê

Trong cách tiếp cận này, tác giả đã dùng ngữ thống kê, hay còn được gọi là n-gram để xây dựng véc tơ đặc trưng cho văn bản cần phân loại. n-gram ở đây được hiểu theo hai nghĩa:

- n-gram chính là một dãy liên tiếp các từ trong văn bản
- n-gram chính là một dãy các ký tự liên tiếp nhau trong văn bản, có thể là một phần của từ hay nhiều từ liên tục. Cách hiểu này sẽ thuận lợi cho việc áp dụng đối với những ngôn ngữ thuộc khu vực Đông Nam Á

Từ ngữ liệu huấn luyện, mô hình sẽ thống kê được một tập các k-gram khác nhau xuất hiện, trong đó k sẽ biến thiên từ 1 đến n. Khi k = 1 ta sẽ định nghĩa nó là unigram hay chính là một từ gốc. n-gram chính là một dãy theo thứ

tự g_k của n unigram. Sau đó, bằng cách dùng hàm ước lượng đặc trưng, mô hình sẽ tính điểm cho từng đặc trưng tương ứng và sắp xếp các đặc trưng theo giá trị điểm đạt được. Tính hiệu quả của n-gram ứng dụng trong bài toán phân loại văn bản là xác định được tần số nào xuất hiện trong nhóm đầu của danh sách đã sắp xếp. Trong thực tế, số lượng k-gram sẽ tăng nhanh đáng kể vì cứ một k-gram xuất hiện sẽ có $2(k+1)$ -gram xuất hiện kèm theo. Bằng cách sử dụng bộ lọc đặc trưng, mô hình sẽ loại bỏ bớt những gram nào có tần số nhỏ hơn giá trị k-gram trung bình. Sau khi có được danh sách gram, mô hình sẽ tiến hành véc tơ hóa cho các văn bản trong tập huấn luyện. Tiếp theo, mô hình sẽ xây dựng bộ học mẫu từ những văn bản đã véc tơ hóa. Quá trình học mẫu có thể áp dụng các phương pháp máy học khác nhau để giải quyết bài toán.

2.2.7 **Tổng quan về bài toán phân loại văn bản trên tiếng Việt**

So với bài toán phân loại văn bản áp dụng trên tiếng Anh, phân loại văn bản tiếng Việt mới có trong thời gian gần đây. Nhiều áp dụng thử nghiệm các phương pháp phân loại đã kiểm chứng cho kết quả tốt trên tiếng Anh được áp dụng cho văn bản tiếng Việt. Tuy nhiên giữa tiếng Anh và tiếng Việt có một điểm khác nhau cơ bản: tiếng Anh là ngôn ngữ thuộc loại hình hòa kết, trong khi tiếng Việt là ngôn ngữ thuộc loại hình đơn lập. Với các ngôn ngữ thuộc loại hình đơn lập, ranh giới từ không phải chỉ là những khoảng trắng, do vậy việc xác định ranh giới từ không hề đơn giản. Chính vì lẽ đó, giải quyết tốt bài toán phân loại văn bản đòi hỏi chúng ta phải giải quyết tốt bài toán tách từ tiếng Việt.

So với bài toán phân loại văn bản tiếng Anh, bài toán phân loại văn bản tiếng Việt hoàn toàn chưa có một kết quả nào được công bố chính thức trên thế giới. Điều này cũng có thể lý giải vì tiếng Việt chưa hề xây dựng được một kho ngữ liệu chung đạt tiêu chuẩn thống nhất cho việc kiểm chứng kết quả như các ngôn ngữ khác (Reuter, NewGroups (tiếng Anh),...).

Tuy nhiên qua quá trình tìm hiểu, chúng tôi biết được một số công trình đã làm cho bài toán phân loại văn bản tiếng Việt như sau:

2.2.7.1 Phân loại văn bản tiếng Việt bằng phương pháp Naïve Bayes [21]

Công trình này đã áp dụng phương pháp **Naïve Bayes** cho bài toán phân loại văn bản cùng với phương pháp Internet and Genetics Algorithm-based Text Categorization (**IGATEC**) của H.Nguyen et al (2005) [69] để tiếp cận bài toán tách từ.

Ngữ liệu thử nghiệm sử dụng trong công trình này được lấy từ báo điện tử VnExpress (www.vnexpress.net) vào thời điểm 6/2005. Ứng với mỗi chủ đề chỉ lấy 100 tin. Tổng cộng có tất cả 1500 tin cho 15 chủ đề.

Công trình này đã kiểm nghiệm trên cây chủ đề có 2 phân cấp:

- **Cấp 1:** gồm 4 chủ đề với kết quả cao nhất đạt được:

Xã hội (R = 77,2%, P = 78,4%, F1 = 77,8%)

Khoa học (R = 85,7%, P = 87,3%, F1 = 86,5%)

Thể thao (R = 92,4%, P = 91,8%, F1 = 92,1%)

Kinh doanh (R = 63%, P = 74%, F1 = 68,1%)

- **Cấp 2:** gồm 15 chủ đề: Giáo dục, Du học, Lối sống, Du lịch, Khoa học, Bóng đá, Quần vợt, Bất động sản, Chứng khoán, Quốc tế, Âm nhạc, Thời trang, Điện ảnh, Làm đẹp, Giới tính.

Kết quả đạt được của các chủ đề khá thấp dao động từ **15% đến 75%**.

❖ Đánh giá ưu khuyết điểm:

- **Ưu điểm:**

♦ Trong công trình này, hướng tiếp cận để giải quyết bài toán tách từ là một hướng khá mới mẻ. Bằng cách sử dụng Google kết hợp thuật toán di truyền, hướng tiếp cận này đã khai thác được thế mạnh về tài nguyên trên Internet, một thư viện khổng lồ cho toàn nhân loại. Việc sử dụng hướng tiếp cận Naïve Bayes cho phân loại văn bản dựa trên Google có thể nói là bước cải tiến đáng khích lệ.

♦ Phương pháp phân loại văn bản dựa trên công thức của **IGATEC** và phương pháp Naïve Bayes đều tương đối đơn giản, không bị hạn chế về tập huấn luyện như khi sử dụng các phương pháp khác. Ngoài

ra, các phương pháp trên cũng không gặp trường hợp sai lạc do có sự thay đổi trong tập huấn luyện bởi tính linh hoạt đối với sự thay đổi nhờ dùng thông tin thống kê từ **Google**.

- **Khuyết điểm:**

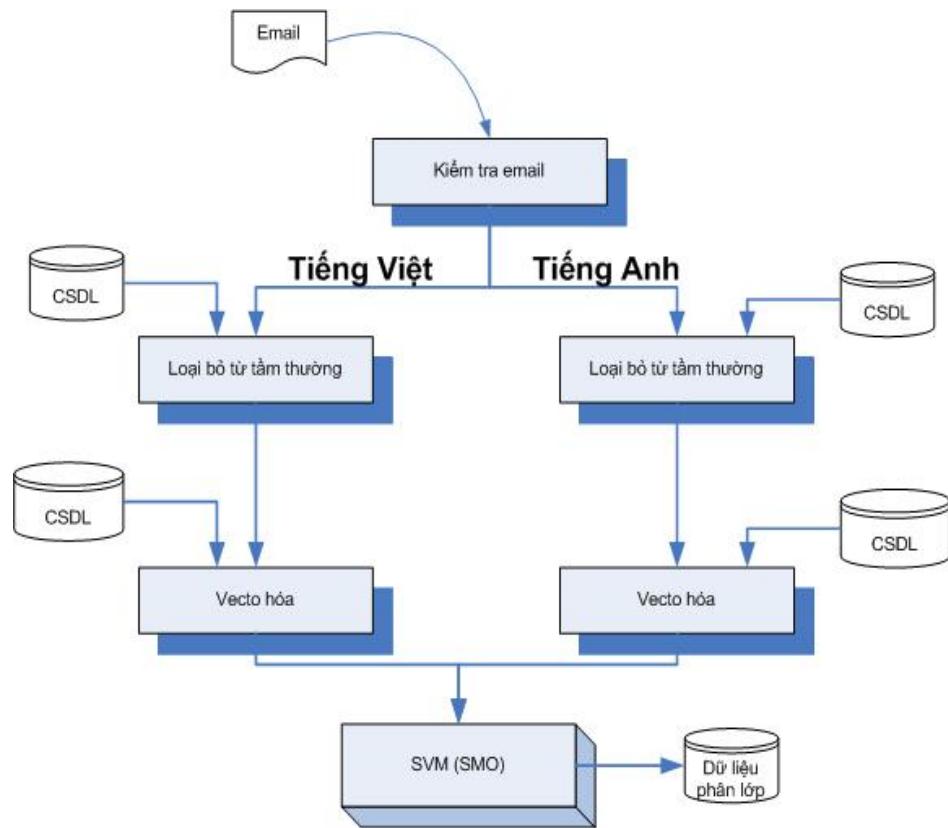
- ◆ Bước đầu thực hiện việc tách từ sẽ khá lâu vì phải mất thời gian lấy thông tin từ công cụ tìm kiếm trên mạng.
- ◆ Chưa nghiên cứu việc chọn lựa đặc trưng văn bản trước khi tiến hành quá trình phân loại, do vậy về tốc độ lẫn chất lượng phân loại đều cho hiệu quả chưa thuyết phục.
- ◆ Việc dùng Google để lấy thông tin về tàn số từ cho giai đoạn tách từ là chưa hợp lý. Lý do cơ bản là việc dùng các câu truy vấn trên Google sẽ không bảo đảm bản chất từ tiếng Việt. Google hoàn toàn không thể phân biệt thế nào là từ mà chỉ có thể tìm kiếm như một cụm từ truy vấn.
- ◆ Việc tìm kiếm bằng cách sử dụng thư viện Google API mỗi ngày bị giới hạn. Như thế việc phát triển thành hệ thống lớn chạy liên tục sẽ không thể thực hiện được.

2.2.7.2 SVM - Ứng dụng lọc email [22]

Công trình này ứng dụng **SVM** trong phân lớp dữ liệu để lọc email: dựa vào nội dung email để sắp xếp chúng vào các thư mục được chỉ định trước của người dùng một cách tự động. Dựa vào các đánh giá của các phương pháp phân loại văn bản áp dụng thử nghiệm trên tiếng Anh: **SVM** là phương pháp thành công hơn các phương pháp khác trong phân loại văn bản về tốc độ thực thi, tốc độ huấn luyện, tốc độ xử lý dữ liệu và có thể áp dụng **SVM** cho tập dữ liệu rất lớn, tác giả đã chọn **SVM** để cài đặt cho ứng dụng của mình.

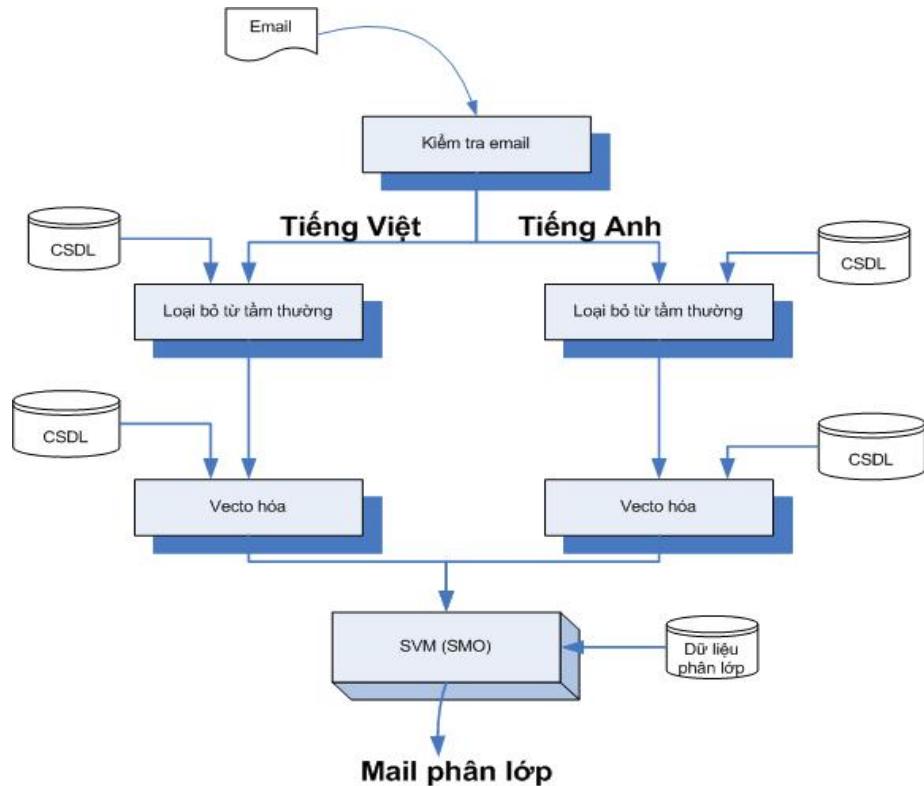
Trong công trình này, tác giả đã chọn thuật toán **SMO** có cải tiến để cài đặt ứng dụng cho mô hình **SVM**.

- ❖ **Mô hình hóa quá trình học**



Hình 16: Quá trình học

❖ Mô hình hóa quá trình phân lớp



Hình 17: Quá trình phân lớp

❖ Kết quả thực nghiệm [22]

- Số chiều véc to : 100
- Kết quả trên tập huấn luyện

	Số mail	Tỷ lệ (%)
Main class	46	92.95
Junk class	52	92.43

Bảng 2: Kết quả thử nghiệm trên tập huấn luyện

- Kết quả trên tập kiểm nghiệm [22]

	Số mail thử	Số mail nhận	Chính xác	Không chính xác	Tỷ lệ (%)
Main class	21	19	19	2 (Junk class)	90.5
Junk class	25	21	19	6 (Other class)	76

Bảng 3: Kết quả thử nghiệm trên tập kiểm nghiệm

❖ Đánh giá ưu khuyết điểm

• **Ưu điểm**

- ◆ Xây dựng một ứng dụng cụ thể cho bài toán phân loại văn bản, có kết quả chấp nhận được.

• **Khuyết điểm**

- ◆ Ứng dụng vẫn còn mang tính thử nghiệm, chưa mang tính thực tiễn.
- ◆ Lọc email tiếng Việt còn rất hạn chế vì chưa có sử dụng hỗ trợ của kỹ thuật phân cụm từ tiếng Việt (tách từ tiếng Việt).
- ◆ Số phân lớp thử nghiệm còn rất ít (khoảng 3 lớp: mail, junk, others)

2.2.7.3 Ứng dụng lý thuyết tập thô trong bài toán phân loại văn bản [23]

Công trình này áp dụng lý thuyết tập thô vào bài toán phân loại văn bản mà cụ thể là cài đặt thử nghiệm phân loại email trong **Microsoft Outlook**.

Tác giả đã dùng lý thuyết tập thô để rút gọn không gian từ phô biến, cũng như áp dụng trong việc tạo bộ luật phân loại.

Công trình này sử dụng ngũ liệu lấy từ trang báo điện tử Tuoi Tre Online (www.tuoitre.com.vn) và phân chia thành 10 chủ đề

Tên chủ đề	Số lượng văn bản
Công nghệ thông tin	168
Du lịch	58
Giáo dục	223
Khoa học	125
Kinh tế	490
Pháp luật	205
Sức khỏe	194
Thế giới	390
Thể thao	258
Xã hội	457

Bảng 4: Mô tả ngũ liệu

Do điều kiện hạn chế về tài nguyên máy tính nên mỗi chủ đề, tác giả chỉ chọn 50 văn bản. Vì vậy tập huấn luyện bao gồm 500 văn bản. Qua quá trình huấn luyện, tác giả xây dựng được bộ luật phân loại gồm 6632 luật.

❖ **Nguyên lý áp dụng**

Úng với mỗi văn bản, từ tập luật ta sẽ xác định các luật mà nội dung văn bản cần phân loại thỏa mãn. Khi đó loại văn bản có số lượng luật được phân loại thỏa mãn nhiều nhất sẽ là loại của nội dung văn bản cần phân loại.

❖ **Kết quả thử nghiệm [23]**

Kết quả được đánh giá trên tập huấn luyện (50 văn bản / 1 chủ đề)

Tên chủ đề	Tỷ lệ (%)
Công nghệ thông tin	95.89
Du lịch	87.5
Giáo dục	95.83
Khoa học	60
Kinh tế	66.67
Pháp luật	85.71
Sức khỏe	41.18
Thế giới	92.11
Thể thao	94.59
Xã hội	40

Bảng 5: Kết quả đánh giá

❖ **Đánh giá ưu khuyết điểm**

• **Ưu điểm**

- ◆ Úng dụng được lý thuyết tập thô vào bài toán phân loại văn bản.
- ◆ Công trình này đã phát triển ứng dụng với việc xử lý văn bản trên nền font chữ Unicode nên có khả năng áp dụng vào thực tế

• **Khuyết điểm**

- ◆ Số lượng tập thử nghiệm còn rất thấp.
- ◆ Các chủ đề còn mang tính tổng quát, chưa phân chia thành nhiều chủ đề nhỏ chi tiết hơn để thử nghiệm.

2.2.7.4 Phân tích các ưu khuyết điểm trong bài toán phân loại văn bản tiếng Việt

Phân loại văn bản là một bài toán khó đòi hỏi có sự hiểu biết về xử lý ngôn ngữ. Trong khi đó chúng ta đều biết ngôn ngữ ngoài sự phong phú về từ vựng cú pháp, còn có sự biến đổi liên tục. Chính vì thế đây là một thách thức rất lớn đối với những người muốn nghiên cứu về xử lý ngôn ngữ tự nhiên.

Tuy nhiên phải thấy rằng, thực tế trên thế giới, rất nhiều mô hình máy học đã được đưa ra thử nghiệm kiểm chứng cho bài toán phân loại văn bản và đạt được kết quả rất khả quan. Đặc biệt trong số đó, bài toán phân loại văn bản áp dụng trên tiếng Trung Quốc, ngôn ngữ gần giống với tiếng Việt đã ghi nhận những thành công đáng kể. Đây là thuận lợi rất lớn giúp chúng ta có cơ sở lý thuyết để nghiên cứu, cài đặt thử nghiệm và kế thừa và phát triển từ những thành quả đạt được.

Tuy vậy, mỗi ngôn ngữ lại có những nét riêng mang tính bản sắc của ngôn ngữ. Chính những nét riêng này làm cho lĩnh vực xử lý ngôn ngữ tự nhiên trở nên thú vị. Đối với bài toán phân loại văn bản tiếng Việt, cái khó khăn lớn nhất mà chúng ta cần giải quyết là bài toán tách từ, bài toán cơ bản nhất của tất cả các bài toán liên quan đến xử lý ngôn ngữ tự nhiên áp dụng cho tiếng Việt. Thực tế hiện nay chưa có một công trình nào công bố một công cụ có khả năng tách từ cho kết quả mỹ mãn. Chính vì thế, kết quả đạt được của bài toán phân loại văn bản tiếng Việt mới chỉ đạt ở mức chấp nhận được.

Với mong muốn tham gia giải quyết những khó khăn cơ bản nêu trên, chúng tôi đã tiến hành nghiên cứu và xây dựng đề tài này nhằm đưa ra một số hướng giải quyết khác. Hướng tiếp cận mới cho bài toán tách từ tiếng Việt. Áp dụng mô hình máy học sử dụng véc tơ hỗ trợ (**Support vector machine – SVM**) cho bài toán phân loại văn bản tiếng Việt. Và hơn hết chúng tôi cũng đề ra một hướng giải quyết cho bài toán phân loại văn bản không theo mô hình truyền thống, có nghĩa là không sử dụng bài toán tách từ trực tiếp, sử dụng mô hình ngôn ngữ thông kê **N-Gram**. Đây là hướng giải quyết khá độc đáo cho bài toán phân loại văn bản tiếng Việt khi chúng ta chưa có một công cụ tách từ tiếng Việt hoàn hảo.

Chương 3: CƠ SỞ LÝ THUYẾT

Nội dung

Chương này sẽ trình bày một số cơ sở lý thuyết nền tảng về văn bản, phân loại văn bản, lý thuyết về từ tiếng Việt ứng dụng cho bài toán tách từ trong phân loại văn bản. Ngoài ra, trong chương cũng trình bày các cơ sở lý thuyết toán học về rút trích đặc trưng, chọn lựa đặc trưng, và cuối cùng là các mô hình phân loại văn bản phổ biến trên thế giới và các mô hình thích hợp vào bài toán tìm kiếm văn bản tiếng Việt theo chủ đề.

3.1 Lý thuyết ngôn ngữ cho bài toán tách từ tiếng Việt [11]

3.1.1 Khái niệm về từ

Trong quá trình học tập và sử dụng ngôn ngữ trong đời sống hằng ngày, mỗi chúng ta đều quen thuộc với khái niệm về “từ”. Nhưng để định nghĩa được chính xác *từ là gì* hoàn toàn không phải là một vấn đề đơn giản. Trong ngành ngôn ngữ học, đã có hàng trăm định nghĩa về từ được đưa ra, nhưng hầu như chưa có một định nghĩa nào có thể bao quát hết được mọi vấn đề liên quan đến khái niệm “từ”. Theo công trình [12] của Đinh Diên, có một số khái niệm tiêu biểu sau đây về từ:

- Theo L.Bloomfield thì: “*từ là một hình thái tự do nhỏ nhất*”.
- B.Golovin quan niệm: “*từ là đơn vị nhỏ nhất có nghĩa của ngôn ngữ, được vận dụng độc lập, tái hiện tự do trong lời nói để xây dựng nên câu*”.
- Còn Solncev thì lại quan niệm: “*Từ là đơn vị ngôn ngữ có tính hai mặt : âm và nghĩa. Từ có khả năng độc lập về cú pháp khi sử dụng trong lời*”

Trong tiếng Việt, cũng có nhiều định nghĩa về từ như:

- Theo Trương Văn Trình và Nguyễn Hiến Lê thì: “*Từ là âm có nghĩa, dùng trong ngôn ngữ để diễn tả một ý đơn giản nhất, nghĩa là ý không thể phân tích ra được*”.

- Nguyễn Kim Thản thì định nghĩa: “*Từ là đơn vị cơ bản của ngôn ngữ, có thể tách khỏi các đơn vị khác của lời nói để vận dụng một cách độc lập và là một khối hoàn chỉnh về ý nghĩa (từ vựng hay ngữ pháp) và cấu tạo*”.

Theo Hồ Lê, “*Từ là đơn vị ngữ ngôn có chức năng định danh phi liên kết hiện thực, hoặc chức năng mô phỏng tiếng động, có khả năng kết hợp tự do, có tính vững chắc về cấu tạo và tính nhất thể về ý nghĩa*”.

3.1.2 **Hình thái từ tiếng Việt**

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

3.1.2.1 **Hình vị tiếng Việt**

Đầu tiên, chúng tôi sử dụng quan niệm của công trình [12] như sau: tiếng là đơn vị cơ bản trong tiếng Việt vì nó có thể nhận diện tương đối dễ dàng bởi người bản ngữ cũng như nhận diện một cách tự động bởi máy tính. Xét về mặt kỹ thuật trên máy tính, ta cũng có thể thực hiện được các thao tác lưu trữ, xử lý, tìm kiếm và sắp xếp các tiếng một cách dễ dàng do số lượng cũng như chiều dài của các tiếng này là nhỏ⁷.

Ngoài ra, tiếng còn được xem là “từ chính tả”. Tuy nhiên, nếu xét trên các tiêu chí của ngôn ngữ học, thì tiếng không thể được xem là một từ thực sự. Thậm chí, tiếng cũng chưa hoàn toàn đủ tư cách để được xem là “hình vị thực sự” vì chưa thỏa tiêu chí về nội dung (phải có ý nghĩa hoàn chỉnh). Vì vậy, trong luận văn này, chúng tôi dựa theo quan điểm của Đinh Điền trong công trình [13] là xem tiếng chỉ là “hình vị tiếng Việt”:

⁷ Trong tiếng Việt, có khoảng 9270 tiếng các loại, và chiều dài của mỗi tiếng cũng được giới hạn là 7 ký tự (*nghiêng* là tiếng dài nhất với 7 ký tự).

Hình vị tiếng Việt ở đây phải được hiểu là: bên cạnh khái niệm *hình vị* như trong ngôn ngữ học đại cương, còn phải xét đến yếu tố *hình tố*, là yếu tố thuần túy hình thức biểu hiện những kiểu quan hệ bên trong giữa các thành tố trong từ. Ta có thể gọi đây là những “tha hình vị” hay “á hình vị”. Như vậy, trong tiếng Việt sẽ có 3 loại hình vị như sau:

- **Hình vị gốc:** là những nguyên tố, đơn vị nhỏ nhất, có nghĩa, chúng có thể là *hình vị thực* (là những từ vựng) hay *hình vị hư* (ngữ pháp), chúng có thể đứng độc lập hay bị ràng buộc.
- **Tha hình vị:** vốn cũng là hình vị gốc, nhưng vì mối tương quan với các thành tố khác trong từ mà chúng biến đổi đi về âm, nghĩa,... Tha hình vị bao gồm:
 - **Tha hình vị lấy nghĩa:** trong các từ ghép bội nghĩa, như: *giá cả, hỏi han, tuổi tác,...*; *nha cửa, yêu thương, ngược xuôi,...*
 - **Tha hình vị lấy âm:** *chúm chím, đo đở, chúm chím,...*; *lé dé, đứng đĩnh,...*
 - **Tha hình vị định tính:** là các yếu tố phụ để miêu tả thuộc tính, như: *xanh lè, tối om, cười khẩy,...*
 - **Tha hình vị tựa phụ tố:** là đơn vị hoạt động giống như những phụ tố (affix) trong các ngôn ngữ biến hình, như: *giáo viên, hiện đại hoá, tân tổng thống,...*
- **Á hình vị:** là những chiết đoạn ngữ âm được phân xuất một cách tiêu cực, thuần túy dựa vào hình thức, không rõ nghĩa, song có giá trị khu biệt, làm chức năng cấu tạo từ. Ví dụ: *dưa hấu, dưa gang, bí ủ, đậu nành, cà niêng,...*

3.1.2.2 Từ tiếng Việt

Trong luận văn này, chúng tôi sử dụng định nghĩa từ theo công trình [13], “*từ được cấu tạo bởi những hình vị*”. Theo công trình này, thì “*từ tiếng Việt được cấu tạo bởi những hình vị tiếng Việt*”.

Từ tiếng Việt ở đây bao gồm: *từ đơn, từ ghép, từ láy và từ ngẫu hợp*.

Xuất phát từ nhu cầu xử lý tự động ngữ liệu tiếng Việt bằng máy tính, Đinh Điện đã đề nghị cách thức hình thức hoá các quan niệm về hình vị tiếng Việt và từ tiếng Việt nói trên trong công trình [13] như sau:

- Do “*hình vị tiếng Việt*” cũng chính là *từ chính tả* (từng chữ độc lập), nên việc hình thức hoá rất đơn giản, không cần đặt ra. Trong ngữ liệu tiếng Việt cũng như tiếng Anh, đơn vị cơ bản được lưu cũng chính là từ chính tả này. Tuy nhiên, nếu chỉ lưu trữ ở cấp độ hình vị như vậy, thì lượng thông tin trong kho ngữ liệu sẽ rất hạn chế và chúng ta sẽ không thể khai thác hiệu quả vốn có của nó được.
- Để lưu trữ thông tin về ranh giới từ tiếng Việt, chúng tôi sử dụng khái niệm *từ từ điển học* được trình bày trong công trình [13]. *Từ từ điển học* ở đây được định nghĩa là “những đơn vị mà căn cứ vào đặc điểm ý nghĩa của nó phải xếp riêng trong từ điển và có đánh dấu đây là đơn vị từ của ngôn ngữ”. Việc chọn lựa những từ nào sẽ đưa vào từ điển là hoàn toàn do các nhà ngôn ngữ hay người xây dựng kho ngữ liệu quyết định, dựa theo quan điểm về từ đã nêu trên. Trong luận văn này chúng tôi sử dụng từ điển tiếng Việt của công trình [14] của GS Hoàng Phê.

Do có nhiều thuật ngữ về “từ” khác nhau (*từ chính tả*, *từ từ điển học* ...), vì vậy, từ đây trở về sau, thuật ngữ “từ” được sử dụng trong luận văn được quy ước là để chỉ “*từ từ điển*”.

3.2 Cơ sở lý thuyết về văn bản, phân loại văn bản

3.2.1 Khái niệm văn bản

Theo Wikipedia (<http://en.wikipedia.org/wiki/Text>) thì văn bản (text, document) có 1 số khái niệm sau:

Trong ngôn ngữ (language), văn bản là 1 thuật ngữ rộng nói về 1 thứ gì đó mà chứa các từ ngữ diễn đạt 1 sự việc.

Trong ngôn ngữ học (linguistics), văn bản là 1 hoạt động giao tiếp, thi hành 7 nguyên tắc câu thành cơ bản và 3 nguyên tắc điều khiển của văn bản học. Cả tiếng

nói, ngôn ngữ viết hay ngôn ngữ thông thường đều có thể xem như văn bản trong ngôn ngữ học.

Trong lý thuyết văn học, văn bản là 1 đối tượng (object) được nghiên cứu, dù nó là 1 cuốn tiểu thuyết, 1 bài thơ, 1 vở phim, 1 mẫu quảng cáo hay bất cứ thứ gì có thành phần thuộc về ký hiệu. Cách dùng rộng rãi thuật ngữ này được bắt nguồn từ sự xuất hiện của ký hiệu những năm 1960 và được củng cố vững chắc bằng những nghiên cứu văn hóa sau đó trong những năm 1980.

Trong truyền thông các thiết bị di động, văn bản (hay tin nhắn văn bản) là 1 đoạn tin nhắn số hóa ngắn giữa những thiết bị.

Trong tin học, văn bản liên hệ đến dữ liệu ký tự (character data), hay đến 1 trong những thành phần của chương trình trong bộ nhớ.

Trong học thuật, văn bản thường được dùng như là 1 hình thức viết tắt của sách giáo khoa.

3.2.2 Khái niệm phân lớp

Theo Wikipedia (<http://en.wikipedia.org/wiki/Categorization>)

Phân lớp (classification, categorization) là 1 tiến trình trong đó các đối tượng và sự việc được nhận ra, được phân biệt và hiểu được. Sự phân lớp hàm ý rằng các đối tượng được nhóm thành các bộ phân loại, thường thì phục vụ cho 1 vài mục đích đặc biệt. Nói 1 cách cơ bản, 1 bộ phân loại mô tả mối quan hệ giữa các chủ đề và đối tượng tri thức. Có rất nhiều cách tiếp cận phân lớp, nhưng nói chung có 2 cách cơ bản nhất:

- Phân lớp học có giám sát (supervised learning)
- Phân lớp học không có giám sát (unsupervised learning).

3.2.3 Khái niệm phân loại văn bản

Phân loại văn bản (text/document classification/categorization - TC) là 1 quá trình gán nhãn cho những tài liệu được diễn đạt trong ngôn ngữ tự nhiên vào 1 trong những bộ phân lớp (category, class), các bộ phân lớp này đã được định nghĩa trước [3].

Nói 1 cách toán học, phân loại văn bản là 1 quá trình xấp xỉ hàm mục tiêu chưa biết $\Psi : D \times C \rightarrow \{T, F\}$ bằng trung gian của hàm $\Phi : D \times C \rightarrow \{T, F\}$, hàm này được gọi là hàm phân lớp. Trong đó:

- $C = \{c_1, \dots, c_m\}$ là tập các nhãn phân lớp có kích thước cố định đã được định nghĩa trước.

- D là phạm vi các tài liệu.

- Giá trị của **T (True)** được gán cho (d_j, c_i) chỉ định rằng 1 quyết định tài liệu d_j thuộc về lớp c_i .

- Giá trị của **F (False)** cho biết quyết định d_j không thuộc về lớp c_i .

Một số lưu ý:

- Chúng ta thường có giả sử rằng các bộ phân loại chỉ là những nhãn ký hiệu. Không có 1 tri thức bổ sung nào từ ý nghĩa (meaning) của các bộ phân loại có thể giúp xây dựng bộ phân lớp.

- Các thuộc tính của các tài liệu liên quan đến bộ phân lớp nên được nhận ra dựa trên cơ bản là nội dung của tài liệu.

- Đưa ra nội dung của 1 tài liệu mang tính chủ quan, điều này có nghĩa tài liệu trong bộ phân loại này không được quyết định 1 cách chắc chắn.

Tùy vào từng ứng dụng cụ thể mà phân loại văn bản có thể chia thành [2]:

3.2.3.1 Phân loại văn bản đơn nhãn và đa nhãn

Ràng buộc khác biệt ở đây có lẽ bị phụ thuộc vào nhiệm vụ phân loại (TC task), vào ứng dụng cụ thể. Chúng ta có thể lấy ví dụ như sau: cho trước 1 số nguyên k (hoặc lớn hơn k hoặc nhỏ hơn k), k thành phần của tập C (tập các loại) được gán cho mỗi tài liệu $d_j \in D$.

- Trường hợp chỉ có chính xác 1 phân lớp (category) được gán cho tài liệu $d_j \in D$ được gọi là phân loại nhãn đơn (single-label, nonoverlapping category).

- Trường hợp có 1 số lượng nhãn (từ 0 cho đến $|C|$) được gán cho tài liệu $d_j \in D$ được gọi là phân loại đa nhãn (multi-label, overlapping category).

- Trường hợp đặc biệt của phân loại nhãn đơn là phân loại nhị phân trong đó mỗi tài liệu $d_j \in D$ có thể được gán cho bộ phân loại c_i hay không thuộc bộ phân loại c_i .

Trên quan điểm lý thuyết, trường hợp phân loại đơn nhãn (nhị phân) tổng quát hơn trường hợp đa nhãn. 1 thuật toán (algorithm) cho phân lớp đơn nhãn cũng có thể áp dụng cho phân lớp đa nhãn, chỉ đơn giản là chúng ta biến đổi vấn đề phân lớp đa nhãn trên tập $\{c_1, \dots, |C|\}$ thành $|C|$ vấn đề phân lớp đơn nhãn độc lập với nhau. Tuy nhiên, điều ngược lại là không đúng, 1 thuật toán cho phân lớp đa nhãn không thể áp dụng cho phân lớp đơn nhãn (cũng như phân lớp nhị phân).

3.2.3.2 Phân loại văn bản phụ thuộc lớp/loại văn bản so với phụ thuộc tài liệu

Có rất nhiều cách khác nhau để sử dụng bộ phân loại văn bản (text classifier). Cho trước tài liệu $d_j \in D$, chúng ta muốn tìm tất cả các lớp $c_i \in C$ mà tài liệu thuộc vào, cách này được gọi là phân loại dựa vào tài liệu (document-pivoted categorization-DPC). Ngược lại, nếu cho trước $c_i \in C$, chúng ta muốn tìm tất cả các tài liệu $d_j \in D$ mà thuộc vào nó, cách này được gọi là phân loại dựa vào lớp (loại) văn bản (category-pivoted categorization-CPC). Sự khác biệt này thể hiện rõ ở thực tế hơn là ở khái niệm trừu tượng.

DPC thích hợp hơn khi mà các tài liệu sẵn có ở những thời điểm khác nhau ví dụ như bài toán lọc e-mail. Còn CPC thích hợp hơn khi 1 tài liệu c $|C|+1$ sẵn sàng thêm vào tập có sẵn $C=\{c_1, \dots, c_{|C|}\}$ sau khi các tài liệu đã được phân lớp dưới C lớp và các tài liệu này cần được xem xét phân lớp lại dưới $c|C|+1$ lớp. Trên thực tế DPC được dùng nhiều hơn CPC.

3.2.3.3 Phân loại văn bản “cứng” so với “mềm”

Trong khi việc tự động hoàn toàn của quá trình phân lớp cần 1 quyết định T hay F cho mỗi cặp (d_i, c_j) thì việc tự động từng phần của tiến trình có lẽ lại cần những nhu cầu khác nhau.

- Phân loại văn bản “cứng” (hard TC) tức là cung cấp 1 giá trị trong {T,F} cho biết d_j có hay không có nằm trong c_i . Điều này rất hữu ích cho các ứng dụng phân lớp tự động (autonomous) [3].

- Phân loại văn bản mềm (soft TC) tức là cung cấp 1 giá trị trong [0,1] cho biết mức độ tin cậy của hệ thống khi quyết định sự phụ thuộc của d_j vào trong c_i . Điều này lại phù hợp hơn với các ứng dụng phân lớp tương tác (interactive) [3].

3.2.3.4 Các ứng dụng của phân loại văn bản [2]

Phân loại văn bản là bài toán nền tảng trong lĩnh vực truy hồi thông tin (information retrieval) có liên quan 1 phần đến Xử lý ngôn ngữ tự nhiên (Natural Language Processing-NLP). Phân loại văn bản là bài toán ứng dụng rất nhiều trong lĩnh vực xử lý ngôn ngữ hiện nay, ví dụ như: search engines, hệ thống lọc Spam mail, hệ thống phân loại để phục vụ cho việc lưu trữ và tìm kiếm... Ngoài ra, phân loại văn bản kết hợp với một số bài toán khác là cơ sở cho một số ứng dụng như: phân loại giọng nói bằng cách kết hợp giữa nhận dạng giọng nói và phân loại văn bản [4][5], phân loại tài liệu số (multimedia) thông qua phân tích chủ thích văn bản [6], định danh tác giả (author identification) cho dạng văn bản tiểu thuyết của những tác giả chưa biết [7], nhận dạng ngôn ngữ (language identification) của những văn bản chưa biết loại ngôn ngữ [8], định danh tự động thể loại văn bản (text genre) [9], và chấm điểm bài luận tự động (automated essay grading) [10], ...

Đối với tiếng Anh và 1 số ngôn ngữ khác, việc nghiên cứu TC từ khá sớm và đã đạt được nhiều kết quả rất khả quan. Đối với tiếng Việt, các kết quả nghiên cứu đối với bài toán này còn hạn chế và thật sự vẫn chưa có một kết quả khả quan nào. Đây cũng chính là thách thức cho luận văn này.

Chương 4: MÔ HÌNH – THIẾT KẾ – CÀI ĐẶT

Nội dung

Trong chương này, chúng tôi sẽ trình bày về các mô hình thuật toán được sử dụng cho bài toán phân loại tài liệu tiếng Việt được dùng trong hệ thống tìm kiếm tài liệu tiếng Việt theo chủ đề. Hơn nữa, chúng tôi còn trình bày về bài toán tách từ tiếng Việt, bài toán rất quan trọng trong cách tiếp cận dãy cách từ phục vụ cho bài toán phân loại tài liệu tiếng Việt.

4.1 Chuẩn bị ngữ liệu

Chúng tôi tiến hành xây dựng ngữ liệu chuẩn cho tiếng Việt dựa trên nguồn tài nguyên chính là 4 trang báo điện tử lớn nhất Việt Nam hiện nay: VnExpress (www.vnexpress.net), Tuổi trẻ Online (www.tuoitre.com.vn), Thanh niên Online (www.thanhnien.com.vn), Người lao động Online (www.nld.com.vn). Ngữ liệu thu thập được từ web được qua 1 quá trình tiền xử lý tự động trên máy tính (gỡ bỏ các tag HTML, chuẩn hóa chính tả, ...), sau đó toàn bộ được chỉnh sửa và chọn lựa bằng tay bởi các nhà ngôn ngữ học. Minh họa cách thức lấy tin từ web:



Hình 18: Hình ảnh minh họa cách lấy thông tin từ web

Kết quả là chúng tôi có được bộ ngữ liệu tương đối lớn và đầy đủ, số lượng khoảng ~100000 tài liệu được chia thành 2 dạng:

❖ **Dạng thô (raw form):**

Gồm **10** chủ đề lớn chia thành: chính trị-xã hội, đời sống, khoa học, kinh doanh, pháp luật, sức khỏe, thế giới, thể thao, văn hóa, vi tính.

Chứa khoảng **33759** tài liệu dùng cho tập huấn luyện và **50373** tài liệu dùng cho tập kiểm nghiệm:

Tên chủ đề	Số file huấn luyện	Số file kiểm chứng
Chính trị xã hội	5219	7567
Đời sống	3159	2036
Khoa học	1820	2096
Kinh doanh	2552	5276
Pháp luật	3868	3788
Sức khỏe	3384	5417
Thế giới	2898	6716
Thể thao	5298	6667
Văn hóa	3080	6250
Vi tính	2481	4560
Tổng cộng	33759	50373

Bảng 6: Mô tả ngũ liệu dạng thô

❖ **Dạng mịn (smooth form):**

Gồm **27** chủ đề nhỏ chia thành: âm nhạc, âm thực, bất động sản, bóng đá, chứng khoán, cúm gà, cuộc sống đó đây, du học, du lịch, đường vào WTO, gia đình, giải trí tin học, giáo dục, giới tính, Hackers và Viruses, hình sự, không gian sống, kinh doanh quốc tế, làm đẹp, lối sống, mua sắm, mỹ thuật, sân khấu điện ảnh, sản phẩm tin học mới, Tennis, thế giới trẻ, thời trang.

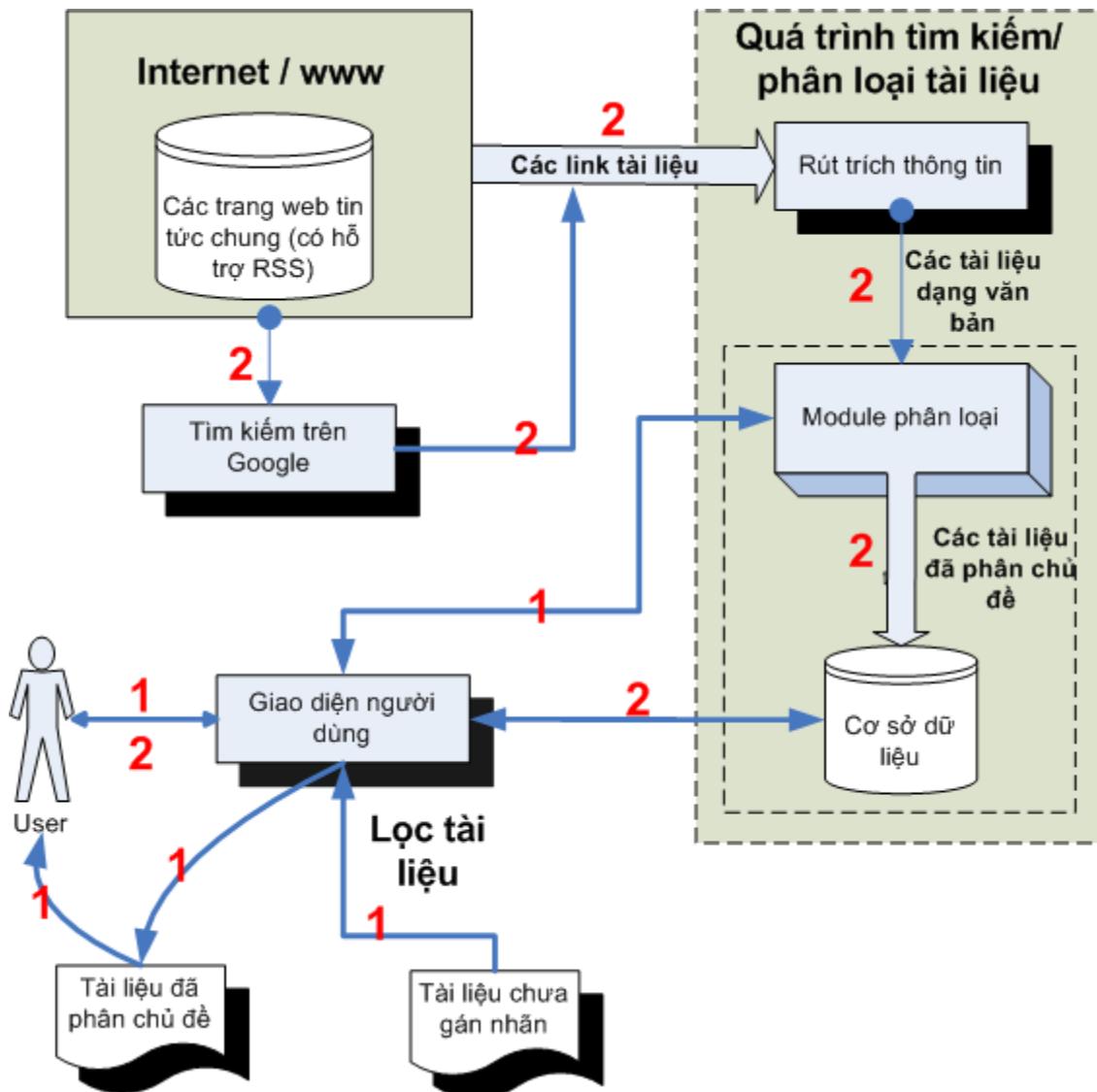
Chứa khoảng **14375** tài liệu dùng cho huấn luyện và **12076** dùng cho kiểm nghiệm.

Tên chủ đề	Số file huấn luyện	Số file kiểm chứng
Âm nhạc	900	813
Âm thực	265	400
Bất động sản	246	282
Bóng đá	1857	1464
Chứng khoán	382	320
Cúm gà	510	381
Cuộc sống đó đây	729	405
Du học	682	394
Du lịch	582	565
Đường vào WTO	208	191
Gia đình	213	280
Giải trí tin học	825	707
Giáo dục	821	707
Giới tính	343	268
Hackers và Virus	355	319
Hình sự	155	196
Không gian sống	134	58
Kinh doanh quốc tế	571	559
Làm đẹp	776	735
Lối sống	223	214
Mua sắm	187	84
Mỹ thuật	193	144
Sân khấu điện ảnh	1117	1030
Sản phẩm tin học mới	770	595
Tennis	588	283
Thế giới trẻ	331	380
Thời trang	412	302
Tổng cộng	14375	12076

Bảng 7: Mô tả ngũ liệu dạng mìn

4.2 Kiến trúc tổng quát của hệ thống (The General Architecture)

Kiến trúc tổng quát của hệ thống tìm kiếm tài liệu được mô tả trong hình sau đây:



Hình 19: Kiến trúc của hệ thống tìm kiếm trong luận văn

4.2.1 Các module chính của hệ thống

- ❖ **Module rút trích thông tin:** module này có khả năng rút trích động các tin tức tài liệu từ các trang web hoặc từ kết quả trả về khi tìm kiếm từ 1 search engines (ví dụ: Google). Lưu ý các trang web có hỗ trợ RSS (Really Simple Syndication).
- ❖ **Module Phân Loại,** module này sẽ chọn lựa các tài liệu có liên quan và tập hợp trong các chủ đề đã định nghĩa trước cho người dùng.
- ❖ **Giao diện người dùng (user interface):** quản lý giao diện người dùng.

4.2.2 Các chức năng chính của hệ thống

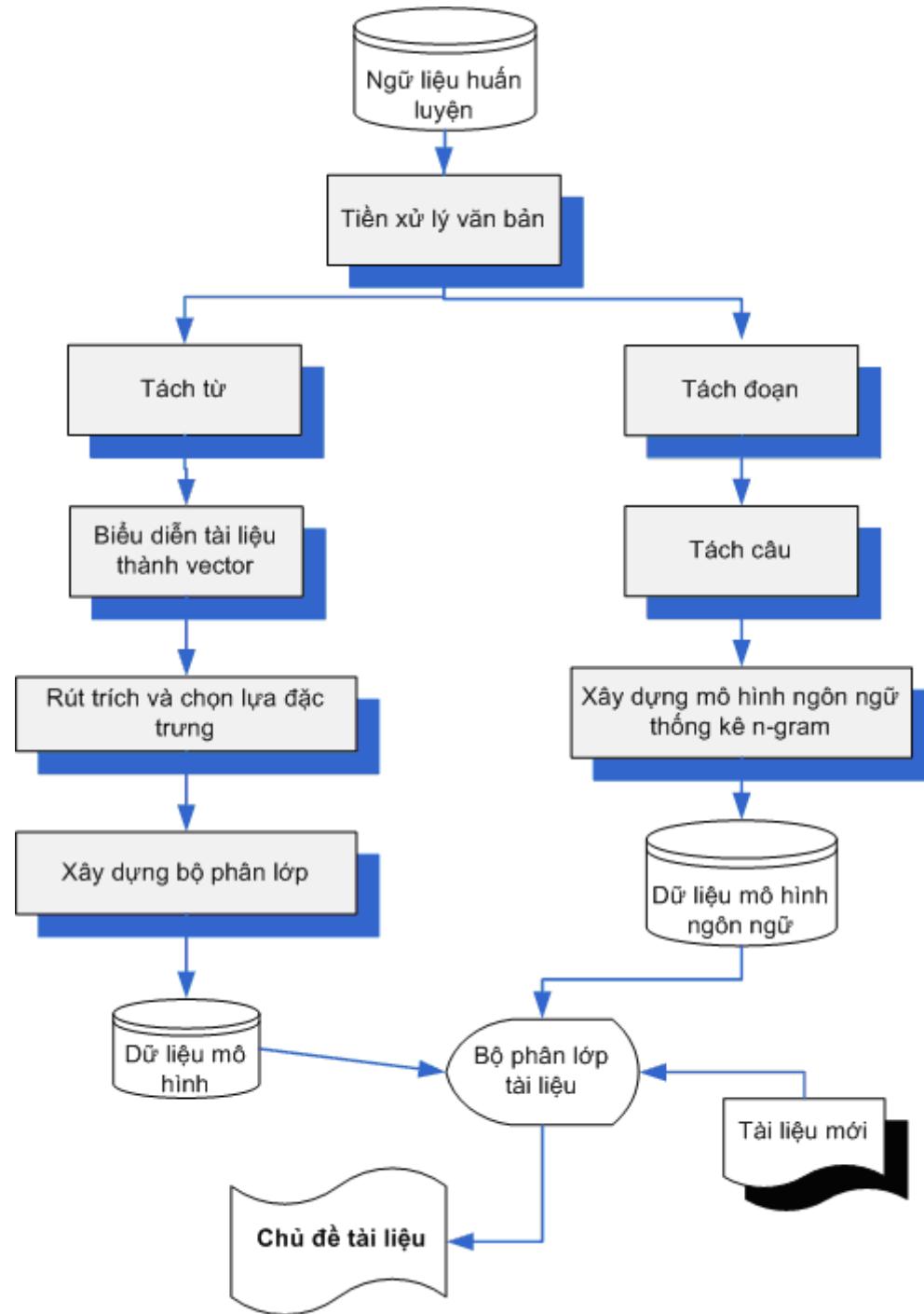
- ❖ **Chức năng Offline (1):** hệ thống cho phép người dùng đưa vào các file cần phân loại, hoặc tìm kiếm các file có chủ đề theo yêu cầu
- ❖ **Chức năng Online (2):** hệ thống cho phép lấy các tin từ các báo có sử dụng RSS hoặc trang tin VnExpress. Ngoài ra hệ thống dự kiến còn có khả năng lấy tin mở rộng dựa trên kết quả tìm kiếm bằng Google. Qua module rút trích thông tin, các trang thông tin sẽ được lấy về dạng file text. Tiếp theo đó sẽ được phân loại theo các chủ đề và lưu trữ dưới cơ sở dữ liệu. Người dùng có thể trực tiếp xem tin lấy về hoặc có thể truy vấn các file này thông qua hệ thống cơ sở dữ liệu

Module phân loại tài liệu/văn bản được chia thành các phần sau đây:

4.3 Module phân loại tài liệu (*Vietnamese Document Classification Module*)

4.3.1 Mô hình tổng quát (General Model)

Mô hình tổng quát của module **phân loại tài liệu** được trình bày như trong hình sau:



Hình 20: Mô hình thuật toán

Module phân loại tài liệu được chúng tôi phát triển dựa trên 2 cách tiếp cận chính:

4.3.2 Cách tiếp cận dựa trên dãy các từ (*The BOW-based Approach*)

Trong cách tiếp cận này, tài liệu văn bản được biến đổi thành véc tơ đặc trưng trong đó đặc trưng là 1 token đơn hay là 1 từ. Trong khi tiếng Anh là ngôn ngữ thuộc loại hình hòa kết, thì tiếng Việt lại là ngôn ngữ thuộc loại hình đơn lập. Với các ngôn ngữ thuộc loại hình đơn lập, ranh giới từ không phải chỉ là những khoảng trắng, do vậy việc xác định ranh giới từ không hề đơn giản. Chính vì lẽ đó, trong cách tiếp cận này giải quyết tốt bài toán phân loại văn bản đòi hỏi chúng ta phải giải quyết tốt bài toán tách từ tiếng Việt.

4.3.2.1 Bài toán tách từ tiếng Việt

1) Khởi đầu

Phương pháp của chúng tôi dựa trên ý tưởng bởi tác giả Goh [18] trong đó giải thuật **MM** (Maximum Matching: so khớp cực đại) kết hợp với mô hình máy học **SVM** (Support Vector Machines : cơ chế véc tơ hỗ trợ) để giải quyết cả 2 vấn đề nhập nhằng và nhận diện từ chưa biết. Trong bài báo này, chúng tôi tập trung vào phương pháp kết hợp 2 phương pháp tách từ dựa trên từ điển **MM** và phương pháp dựa trên học **SVM**. Việc kết hợp như vậy tỏ ra khá hiệu quả vì ưu điểm của **MM** sẽ được bổ sung cho **SVM** và ngược lại nhược điểm của **MM** sẽ được sửa sai bởi **SVM**. Chúng tôi nghĩ rằng mô hình này rất thích hợp cho bài toán tách từ tiếng Việt hiện nay bởi vấn đề thiếu từ điển chuẩn và một ngữ liệu huấn luyện đầy đủ.

Trong phương pháp của chúng tôi, chúng tôi xem vấn đề tách từ trở thành vấn đề gán nhãn cho tiếng. Chúng tôi tin rằng mỗi tiếng trong tiếng Việt giữ một vai trò nhất định khi xuất hiện trong vị trí của 1 từ. Nói cách khác, nó có thể được đứng ở đầu 1 từ, hoặc đứng giữa 1 từ hoặc đứng cuối 1 từ hoặc cũng có thể đứng 1 mình như từ đơn. Bằng cách dựa vào cách dùng tiếng như vậy, chúng tôi sẽ quyết định vị trí của tiếng bằng cách sử dụng mô hình máy học **SVM** hay nói cách khác dùng **SVM** để gán nhãn vị trí cho từng tiếng trong câu. Như vậy **SVM** được áp dụng để giải quyết vấn đề nhập nhằng mà mô hình **MM** không thể giải quyết được.

Chúng tôi định nghĩa 4 nhãn (tag) đánh cho tiếng, thể hiện vị trí của tiếng trong 1 từ [2]:

Nhãn	Mô tả
O	Từ đơn 1 tiếng
B	Tiếng đầu tiên trong 1 từ đa tiếng
I	Tiếng trung gian trong 1 từ đa tiếng (đối với từ > 2 tiếng)
E	Tiếng cuối cùng trong 1 từ đa tiếng

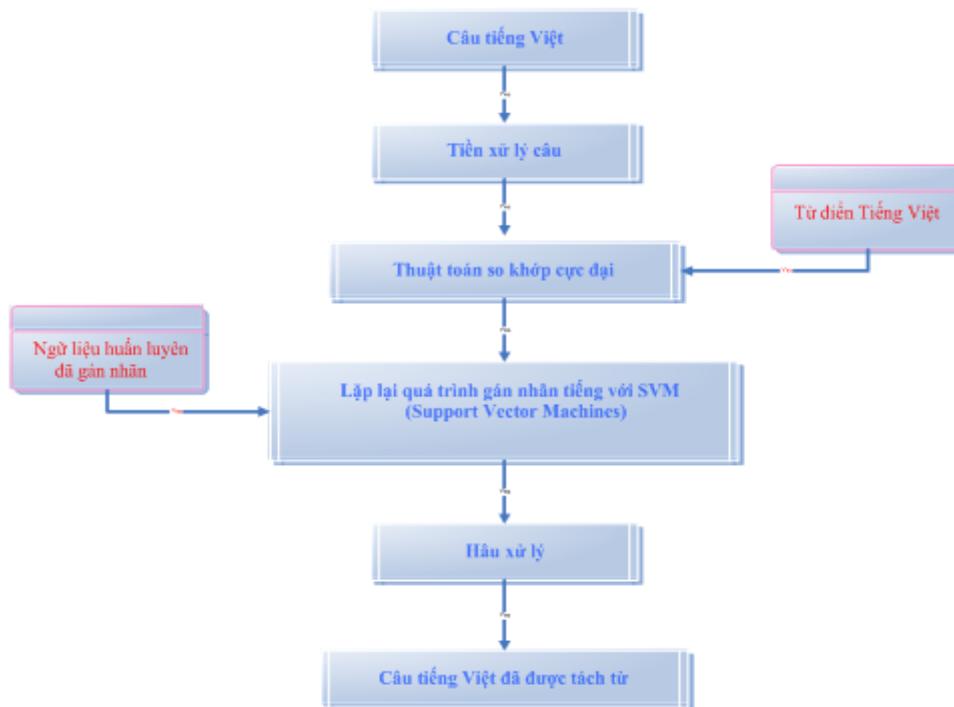
Bảng 8: Thông tin mô tả vị trí của tiếng trong 1 từ

Ví dụ:

Tốc độ truyền#thông tin#được#nâng#cao

B I E B E O O O

2) Mô hình của chúng tôi



Hình 21: Kiến trúc của mô hình kết hợp

3) Chuẩn bị ngữ liệu đã gán nhãn

Kho ngữ liệu được dùng trong mô hình được trích từ kho ngữ liệu song ngữ CADASA, đó là kết quả của dự án xây dựng kho ngữ liệu dùng cho hệ thống dịch tự động Anh Việt [13]. Đây là một vài thống kê trên ngữ liệu này:

Tham số	Giá trị
---------	---------

Số câu	8881 câu
Số tiếng tiếng Việt	239214 tiếng (3654 tiếng không trùng lặp)
Số từ tiếng Việt	182257 từ (5684 từ không trùng lặp)
Chiều dài trung bình câu	20.52 từ /câu (26.93 tiếng/câu)

Bảng 9: Thống kê ngữ liệu CADASA

Đối với tiếng Việt, tiền xử lý đóng 1 vai trò rất quan trọng trong hệ thống tách từ. Vì tính đa dạng của tiếng Việt, chúng tôi không thể bao quát hết tất cả các trường hợp không chuẩn văn bản tiếng Việt mà chúng tôi chỉ chuẩn hóa trong 1 số trường hợp nhất định. Quá trình tiền xử lý dựa trên báo cáo của tác giả Đinh Diền et al [16] bao gồm:

a) Chuẩn hóa chính tả

Tiếng Việt có 2 loại chính tả khác nhau:

- ❖ Luật trên tiếng:

Ví dụ: hòa và hoà

- ❖ Sai khác mẫu tự

Ví dụ: thời kì → thời kỳ

b) Chuẩn hóa dấu chấm câu

Chúng tôi chia ra 3 loại xử lý:

- ❖ Xử lý dấu chấm cuối câu

Ví dụ:

Tôi đi học. và Tôi đi học .

- ❖ Xử lý dấu chấm giữa câu

Ví dụ:

www.yahoo.com là 1 trang web hay.

Cô ấy cho tôi 9.500\$.

- ❖ Xử lý trường hợp viết tắt

Ví dụ:

GS, GS., GS.TS , GS. TS. ,

4) Thuật toán so khớp cực đại (Maximum Matching Algorithms)

Chúng tôi tập trung vào việc giải quyết vấn đề nhập nhằng bằng cách kết hợp một cách tiếp cận dựa trên từ điển với mô hình thông kê máy học **SVM**. **MM** được xem như là phương pháp tách từ dựa trên từ điển đơn giản nhất. **MM** có găng so khớp với từ dài nhất có thể có trong từ điển. Đó là một thuật toán ăn tham (Greedy Algorithms) nhưng bằng thực nghiệm đã chứng minh được rằng thuật toán này đạt được độ chính xác > **90%** nếu từ điển đủ lớn [18]. Tuy nhiên, nó không thể giải quyết vấn đề nhập nhằng và không thể nhận diện được các từ chưa biết bởi vì chỉ những từ tồn tại trong từ điển mới được phân đoạn đúng.

Giải quyết **MM** gồm hai giải thuật con: **FMM** (Forward Maximum Matching: so khớp cực đại theo chiều tiến) và **BMM** (Backward Maximum Matching: so khớp cực đại theo chiều lùi). Nếu chúng ta nhìn vào kết quả của **FMM** và **BMM** thì sự khác biệt này cho chúng ta biết nơi nào nhập nhằng xảy ra.

Ví dụ: Người nông dân ra sức cải tiến bộ công cụ lao động của mình.

Đầu ra FMM: Người#nông dân#ra sức#cải tiến#bộ#công cụ#lao động#của#mình#.

Đầu ra BMM: Người#nông dân#ra sức#cải tiến#bộ#công cụ#lao động#của#mình#.

Như vậy, kết xuất của **FMM** và **BMM** sẽ được gán nhãn vị trí như sau:
FMM: Người#nông dân#ra sức#cải tiến#bộ#công cụ#lao động#của#mình#.

O B E B E B E O B E B E O O

BMM: Người#nông dân#ra sức#cải#tiến bô#công cụ#lao động#của#mình#.

O B E B E O B E B E B E O O

Sự khác biệt giữa thông tin **OBEBEBEBOBEBOOO** và **OBEBOBEBEBOOO** sẽ là các đặc trưng đầu vào cho mô hình **SVM**.

5) Vấn đề phân lớp các tiếng

Chúng tôi phân lớp các tiếng dựa trên bốn nhãn (xem phần 1): **B, I, E, O**. Thay vì tách một câu thành dãy các từ trực tiếp, các tiếng đầu tiên được gán nhãn vị trí. Dựa trên các nhãn vị trí này, các tiếng sẽ được chuyển đổi ngược trở lại thành dãy các từ. Đặc trưng cơ bản được sử dụng ở đây là tiếng. Tuy nhiên, ngữ liệu dùng để huấn luyện tương đối nhỏ, thông tin về tiếng như đặc trưng là chưa đủ. Vì thế,

chúng tôi cung cấp kết xuất của **FMM** và **BMM** như là một đặc trưng. Như vậy, việc học bởi **SVM** được hướng dẫn bởi một từ điển để phân đoạn các từ đã biết. Sự giống nhau và khác nhau giữa **FMM** và **BMM** được dùng như một đặc trưng trong huấn luyện **SVM** để giải quyết vấn đề nhập nhằng trong tách từ.

Như vậy, tập đặc trưng chúng tôi sử dụng bao gồm: các tiếng, kết xuất của **FMM** và **BMM**, và các nhãn kết xuất phía trước nhãn hiện tại. Ngữ cảnh sử dụng là cửa sổ 2 tiếng trước nó và 2 tiếng sau nó. Như vậy, tập đặc trưng sẽ gồm:

- ❖ Tiếng hiện tại (C_0)
- ❖ Các tiếng ngữ cảnh: cấu trúc đơn (C_{i-2} , C_{i-1} , C_i , C_{i+1} , C_{i+2}), cấu trúc phức ($C_{i-2}C_{i-1}$, $C_{i-1}C_i$, $C_{i-1}C_{i+1}$, C_iC_{i+1} , $C_{i+1}C_{i+2}$)
- ❖ Đầu ra của **FMM** và **BMM**: ($f_{i-2}b_{i-2}$, $f_{i-1}b_{i-1}$, f_ib_i , $f_{i+1}b_{i+1}$, $f_{i+2}b_{i+2}$)
- ❖ Các nhãn trước nhãn sau: (t_{i-2} , t_{i-1})

Sau đó, dựa trên những đặc trưng này, chúng tôi sẽ phân lớp các nhãn bằng cách sử dụng 2 bộ công cụ **YAMCHA** (Yet Another Multipurpose CHunk Annotator) [60]. Tiếp theo, các nhãn của các tiếng phân lớp có được sẽ được qua giai đoạn sửa sai trước khi chuyển đổi thành dãy các từ.

Vị trí	Tiếng	FMM	BMM	Output
i-2	Tối	O	O	O
i-1	Cung	B	O	B
i	Cấp	E	B	E
i+1	Số	O	E	O
i+2	Vaccine	O	O	O

Bảng 10: Khung đặc trưng sử dụng cho mô hình SVM

Sau khi các tiếng được gán nhãn, các nhãn này được qua một quá trình sửa sai, các luật sửa sai như sau [61]:

Điều kiện	Sửa
nhãn trước="I" và nhãn hiện tại="O"	nhãn hiện tại="E"
nhãn trước="B" và nhãn hiện tại="O"	nhãn trước="O"
nhãn trước="O" và nhãn hiện tại="E"	nhãn trước="B"
nhãn trước="O" và nhãn hiện tại="I"	nhãn hiện tại="B"
nhãn trước="I" và nhãn hiện tại="B" và nhãn tiếp theo="B"	nhãn hiện tại="E"
nhãn trước="B" và nhãn hiện tại="B" và nhãn tiếp theo="E"	nhãn trước="O"
nhãn trước="I" và nhãn hiện tại="B" và nhãn tiếp theo="O"	nhãn hiện tại="E"
nhãn trước="B" và nhãn hiện tại="B" và nhãn tiếp theo="B"	nhãn hiện tại="E"
nhãn trước="B" và nhãn hiện tại="E" và nhãn tiếp theo="E"	nhãn hiện tại="I"

Bảng 11: Quá trình sửa sai

6) Nhận diện từ chưa biết

Chúng tôi chia từ chưa biết thành 3 loại:

a) Tên riêng tiếng Việt

- ❖ **Tên người:** Hoàng Công Duy Vũ, Nguyễn Lê Nguyên,
- ❖ **Tên địa danh:** Hà Nội, Sài Gòn,

b) Tên riêng tiếng nước ngoài: Luis Figo, David Fillipe,...

c) Factoids

Factoid là một chuỗi diễn đạt các thông tin đặc biệt. **Factoids** trong bài báo này là ngày tháng, thời gian, phần trăm, tiền, con số, độ đo, e-mail, số điện thoại, và web-site...

7) Khử nhập nhằng

Mẫu chốt của vấn đề khử nhập nhằng là ngữ liệu huấn luyện có chứa những trường hợp nhập nhằng này hay nói cách khác **ngữ cảnh** gây nên sự nhập nhằng này đã có trong ngữ liệu huấn luyện. Hệ thống kết hợp **MM + SVM** tỏ ra giải quyết rất tốt các vấn đề nhập nhằng nêu trên.

Một số ví dụ:

Chính phủ#ra súc#cải tiến#bộ máy#quản lý#từ#Trung ương#đến#địa phương

Đời sống#nhân dân#cực kỳ#khó khăn#, #siêu#lạm phát#đạt#đến#đỉnh cao#năm#1998

4.3.2.2 Tiền xử lý văn bản tiếng Việt

❖ Tách từ

Chúng tôi sử dụng kết quả tách từ của mình trong phần trước cho tiền xử lý văn bản đầu tiên là tách token hay từ trong cách tiếp cận dãy các từ. Tất cả các tài liệu qua bước này đều được xử lý thành các từ là đầu vào cho bước xử lý tiếp theo.

❖ Gỡ bỏ từ vô nghĩa (stop words)

Trong giai đoạn này, các đặc trưng liên quan sẽ được trút trích từ các tài liệu. Tất cả các từ lấy từ tài liệu đều được xem như là đặc trưng khả thi. Chúng thường được gọi là các token (Ciya Liao et al, 2003 [25]). Sau đó, tập các token này sẽ được qua bước lọc bỏ các đặc trưng mà không mang thông tin hữu ích. Các từ chức năng hay các phụ từ, hư từ (ví dụ: “là”, “của”, “nhất là”, ...) sẽ được lược bỏ để tăng hiệu năng cũng như giảm bớt số lượng đặc trưng vốn đã rất lớn trong các mô hình phân loại văn bản. Để làm việc này, một danh sách các từ chức năng được chuẩn bị trước, danh sách này được cung cấp bởi nhóm chúng tôi VCL.

❖ Trọng số hóa đặc trưng

Cho tập đặc trưng $F=\{f_1, f_2, \dots, f_D\}$ và tập các lớp $C=\{C_1, C_2, \dots, C_{|C|}\}$, một đặc trưng $f \in F$ và một tài liệu d sẽ được trọng số hóa theo công thức như sau:

+ **Định nghĩa TF (Term Frequency):** tf_{fd} là số lần xuất hiện của đặc trưng f trong tài liệu d .

+ **TF_IDF (Terms Frequency Inverse Document Frequency):** được tính theo công thức sau:

$$\omega_{fd} = \frac{tf_{fd} \times \log \frac{D}{df_f}}{\sqrt{\sum_{r \in F} \left(tf_{rd} \times \log \frac{D}{df_r} \right)^2}} \quad [4.1]$$

Trong đó:

ω_{fd} là trọng số của đặc trưng f trong tài liệu d.

tf_{fd} là tần số xuất hiện của đặc trưng f trong tài liệu d.

D là tổng số tài liệu trong tập huấn luyện.

df_f là số tài liệu chứa đặc trưng f

+ **logTF_IDF** : được tính theo công thức sau:

$$\omega_{fd} = \frac{l_f^d \times \log \frac{D}{df_f}}{\sqrt{\sum_{r \in F} \left(l_r^d \times \log \frac{D}{df_r} \right)^2}} \quad [4.2]$$

Với:

$$l_f^d = \begin{cases} 0 & Neu \ tf_{fd} = 0 \\ \log(tf_{fd}) + 1 & Neu \ nguoc lai \end{cases} \quad [4.3]$$

+ **TF_IWF** được tính theo công thức sau:

$$\omega_f^d = \frac{tf_{df} \times (IWF(f)^2)}{\sqrt{\sum_{r \in F} \left(o_r^d \times (IWF(r))^2 \right)^2}} \quad [4.4]$$

Với:

$$IWF = \log\left(\frac{o}{o_f}\right) \quad [4.5]$$

Trong đó:

O : Tần suất xuất hiện của tất cả các đặc trưng

O_f là tần suất xuất hiện của đặc trưng f

+ Khác:

TF_CHI, TF_CRF, TF_OddRatio tham khảo trong tài liệu [63].

- Ở đây chúng tôi chọn 2 công thức **TF_IDF** và **logTF_IDF** vì tính dễ cài đặt và hiệu quả của nó [63].

4.3.2.3 Chọn lựa đặc trưng

Thực chất của quá trình chọn lựa đặc trưng là làm giảm số chiều của véc tơ đặc trưng bằng cách bỏ đi những thành phần đặc trưng không quan trọng. Đối với phân loại văn bản thì quá trình này rất quan trọng bởi vì véc tơ văn bản có số chiều rất lớn ($>>10000$ [28]), trong đó số thành phần dư thừa cũng rất nhiều. Vì vậy các phương pháp chọn lựa đặc trưng rất hiệu quả trong việc giảm chiều của véc tơ đặc trưng văn bản, chiều của véc tơ văn bản sau khi được giảm chỉ còn lại khoảng 1000 đến 5000 mà không mất đi độ chính xác. Ở đây chúng tôi tiếp cận 6 phương pháp khác nhau (Fabrizio, 2002 [2]) như là: **MI** (Mutual Information), **IG** (Information Gain), **GSS** (GSS coefficient), **CHI** (Chi-square), **OR** (Odds Ratio), **DIA** association factor, **RS** (Relevancy score) và **OCFS** [62] (Optimal Orthogonal Centroid Feature Selection).

Trong luận văn này, chúng tôi sẽ triển khai 6 phương pháp **MI, IG, GSS, CHI, OR**, và đặc biệt là **OCFS** để so sánh và đánh giá. Lý do mà chúng tôi chọn 6 phương pháp trên là vì chúng đã được chứng minh bằng thực nghiệm trên các bộ ngữ liệu chuẩn của thế giới cho tiếng Anh là tốt nhất.

Cho kho ngữ liệu, một cách toán học nó được biểu diễn bởi một ma trận $d \times n$ X, trong đó d là số lượng các hàng hay các đặc trưng, n là số lượng các tài liệu trong

kho dữ liệu. Mỗi tài liệu là một véc tơ cột x_i , $i=1,2,\dots,n$. X^T là ma trận chuyển vị của ma trận $X \in R^{d \times n}$. Vấn đề giảm chiều thực chất là tìm ánh xạ $f: R^d \rightarrow R^p$ với p là chiều của dữ liệu sau khi được giảm, p rất nhỏ so với d . Như vậy tài liệu $x_i \in R^d$ sẽ được biến đổi thành $y_i = f(x_i) \in R^p$. Theo phương pháp rút trích đặc trưng thì vấn đề giảm chiều là tìm ma trận chuyển đổi tối ưu $W \in R^{d \times p}$ để mà $y_i = f(x_i) = W^T x_i \in R^p$, $i=1,2,\dots,n$; p là chiều sau khi được giảm. Một cách khác theo phương pháp chọn lựa đặc trưng, mục đích của việc giảm chiều là tìm tập đặc trưng k_l , $l=1,2,\dots,p$ sao cho $y_i = f(x_i) = (x_i^{k_1}, x_i^{k_2}, \dots, x_i^{k_p})^T$.

Phương pháp chọn lựa đặc trưng tìm ma trận biến đổi đặc biệt $\tilde{W} \in R^{d \times p}$ trong đó \tilde{W} là một ma trận nhị phân mà tất cả các thành phần là 0 hoặc 1 và mỗi cột chỉ có duy nhất một thành phần là 1. Như vậy quá trình biến đổi sẽ là $y_i = f(x_i) = \tilde{W}^T \times x_i = (x_i^{k_1}, x_i^{k_2}, \dots, x_i^{k_p})^T$, giải pháp của vấn đề chọn lựa đặc trưng là tìm tất cả các ma trận \tilde{W} với tiêu chuẩn $J(\tilde{W})$ trên, chúng ta định nghĩa không gian này là $H^{d \times p}$.

❖ 5 phương pháp chọn lựa đặc trưng truyền thống [2] được tóm tắt như sau:

Phương pháp	Ký hiệu	Công thức
Information gain	IG(t_k , c_i)	$\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \cdot \log \frac{P(t, c)}{P(t) \cdot P(c)}$
Mutual information	MI(t_k , c_i)	$\log \frac{P(t_k, c_i)}{P(t_k) \cdot P(c_i)}$
Chi-square	$\chi^2(t_k, c_i)$	$\frac{ Tr \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]^2}{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}$
Odds ratio	OR(t_k , c_i)	$\frac{P(t_k c_i) \cdot (1 - P(t_k \bar{c}_i))}{(1 - P(t_k c_i)) \cdot P(t_k \bar{c}_i)}$
GSS coefficient	GSS(t_k , c_i)	$P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)$

Bảng 12: Tóm tắt các phương pháp chọn lựa đặc trưng

Trong đó các đại lượng được định nghĩa như sau :

$P(t_k)$: xác suất xảy ra đặc trưng t_k

$P(c_i)$: xác suất xảy ra phân lớp c_i

$P(t_k, c_i)$: xác suất khi chọn ngẫu nhiên tài liệu x, đặc trưng t_k sẽ xuất hiện trong x, và x thuộc về phân lớp c_i

$P(t_k, \bar{c}_i)$: xác suất khi chọn ngẫu nhiên tài liệu x, đặc trưng t_k sẽ xuất hiện trong x, và x không thuộc về phân lớp c_i

$P(\bar{t}_k, c_i)$: xác suất khi chọn ngẫu nhiên tài liệu x, đặc trưng t_k sẽ không xuất hiện trong x, và x thuộc về phân lớp c_i

$P(\bar{t}_k, \bar{c}_i)$: xác suất khi chọn ngẫu nhiên tài liệu x, đặc trưng t_k sẽ không xuất hiện trong x, và x không thuộc về phân lớp c_i

$P(t_k | c_i)$: xác suất xuất hiện t_k với điều kiện thuộc phân lớp c_i

$P(t_k | \bar{c}_i)$: xác suất xuất hiện t_k với điều kiện không thuộc phân lớp c_i

$|Tr|$: tổng số tài liệu đang xét

Như vậy, với việc chọn $J(\tilde{W}) = IG(t_k, c_i)$, $J(\tilde{W}) = MI(t_k, c_i)$, $J(\tilde{W}) = \chi^2(t_k, c_i)$, $J(\tilde{W}) = OR(t_k, c_i)$, $J(\tilde{W}) = GSS(t_k, c_i)$ ta sẽ được hàm chọn lựa đặc trưng trong không gian huấn luyện. Vẫn đề là ở chỗ: với hàm tiêu chuẩn $J(\tilde{W})$ ở trên làm cách nào ta chọn được các đặc trưng cho kết quả phân lớp cao nhất. Nói chung tất cả các hàm liệt kê trong bảng trên đều liên hệ cục bộ đến lớp c_i để mà nó đánh giá giá trị của hạng (term) t_k trong toàn cục (có nghĩa là nó độc lập với lớp). Trong thực nghiệm, người ta có thể dùng cả hàm tổng $f_{sum}(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i)$ hay hàm tổng có trọng số (weighted sum) $f_{wsum}(t_k) = \sum_{i=1}^{|C|} P(c_i) f(t_k, c_i)$ hay hàm cục đại $f_{max}(t_k) = \max_{i=1}^{|C|} f(t_k, c_i)$ trong đó $f(t_k, c_i)$ là hàm $J(\tilde{W})$ ở trên. Tất cả các hàm này đều cố gắng lấy ra được các đặc trưng tốt nhất cho cho lớp c_i mặc dù chúng được phân phối rất khác nhau trong tập huấn luyện. Kết quả thực nghiệm chi tiết sẽ được chúng tôi trình bày chi tiết trong chương 6.

Trong phần tiếp theo, chúng tôi muốn giới thiệu một phương pháp mới có cùng chức năng như các phương pháp kể trên nhưng hiệu quả thì lại có hiệu quả cao hơn. Phương pháp này được các tác giả tại trung tâm nghiên cứu Microsoft Research Asia đề xuất **2004 [62]**, có tên **Optimal Orthogonal Centroid Feature Selection - OCFS**. Các kết quả thực nghiệm trên bộ ngữ liệu chuẩn của tiếng Anh là rất ấn tượng, phương pháp này đã vượt qua 2 phương pháp khác là **MI** và **X² Test** vốn cho kết quả rất cao trước đây. Trong luận văn này, chúng tôi muốn thí nghiệm lại phương pháp này trên tiếng Việt, từ đó so sánh với các phương pháp khác và chọn ra phương pháp tốt nhất dùng cho module tìm kiếm như đã nói lúc đầu.

❖ Phương pháp OCFS [62]

Phương pháp **OCFS** dựa trên nền tảng là giải thuật Orthogonal Centroid - **OC**. Để hiểu rõ hơn về phương pháp **OCFS**, chúng ta hãy tìm hiểu sơ qua về giải thuật **OC**.

➤ Thuật toán OC

Thuật toán **OC** được đề nghị gần đây được xem như là một thuật toán rút trích đặc trưng có giám sát. Nó tận dụng phép biến đổi trực giao trên trọng tâm [64][65]. Thuật toán này đã được chứng minh là rất hiệu quả với các vấn đề phân lớp véc tơ liệu dưới dạng văn bản (text data) [64][65] và nó dựa trên phép tính toán không gian véctơ trong đại số tuyến tính [66] (cụ thể là QR matrix decomposition).

Thuật toán này cũng có mục tiêu là tìm ra được ma trận biến đổi $W \in R^{d \times p}$, tiêu chuẩn $J(W)$ được tính như sau:

$$\arg \max J(W) = \arg \max \text{trace}(W^T S_b W), \text{subject to } W^T W = I \quad [4.6]$$

Với:

$$S_b = \sum_{j=1}^c \frac{n_j}{n} (m_j - m)(m_j - m)^T \quad [4.7]$$

Trong đó:

+ m_j là véc tơ trung bình của lớp j được tính theo công thức

sau:

$$m_j = (1/n_j) \sum_{x_i \in c_j} x_i \quad [4.8]$$

+ n_j là kích thước của lớp j

+ m là véc tơ trung bình của tất cả các tài liệu, được tính theo công thức sau:

$$m = (1/n) \sum_{i=1}^n x_i = (1/n) \sum_{j=1}^c n_j m_j \quad [4.9]$$

Chi tiết chứng minh của nó có thể xem chi tiết trong [64][65]. Các tác giả của **OCFS** dựa trên tiêu chuẩn $J(W)$ trong thuật toán **OC** để đề xuất ra giải thuật chọn lọc đặc trưng tối ưu bằng cách tối ưu $J(W)$ trong không gian $H^{d \times p}$.

➤ **Giải thuật OCFS**

Thay vì tìm ma trận W ta tìm ma trận \tilde{W} với tiêu chuẩn tối ưu $J(\tilde{W})$ (xem thêm phần trước):

$$\arg \max J(\tilde{W}) = \arg \max \text{trace}(\tilde{W}^T S_b \tilde{W}) \text{ subject to } \tilde{W} \in H^{d \times p} \quad [4.10]$$

Trong đó các thành phần trong công thức được định nghĩa như trong phần OC, chỉ khác là ma trận \tilde{W} là ma trận nhị phân mà mỗi cột chỉ duy nhất có một phần tử khác 0, ta định nghĩa tập $K = \{k_i, 1 \leq k_i \leq d, i = 1, 2, \dots, p\}$ là tập các chỉ mục của đặc trưng, mặt khác ta lại có:

$$\text{trace}(\tilde{W}^T S_b \tilde{W}) = \sum_{i=1}^p \tilde{W}_i^T S_b \tilde{W}_i = \sum_{i=1}^p \sum_{j=1}^c \frac{n_j}{n} (m_j^{k_i} - m^{k_i})^2 \quad [4.11]$$

Như vậy thực chất của vấn đề OCFS là tìm tập K ở trên để làm cực đại:

$$\sum_{i=1}^p \sum_{j=1}^c \frac{n_j}{n} (m_j^{k_i} - m^{k_i})^2 \quad [4.12]$$

Từ đó giải thuật **OCFS** được đề nghị như sau:

- ❖ **Bước 1:** Tính centroid m_i $i=1,2,\dots,c$ của mỗi lớp cho dữ liệu huấn luyện
- ❖ **Bước 2:** Tính centroid m của tất cả các mẫu huấn luyện
- ❖ **Bước 3:** Tính điểm cho từng đặc trưng i theo công thức

$$s(i) = \sum \frac{n_j}{n} (m_j^i - m^i)^2$$
- ❖ **Bước 4:** Lấy k đặc trưng có điểm cao nhất

➤ **Phân tích:**

Độ phức tạp (complexity) của thuật toán **OCFS** là $O(cd)$, thuật toán **OCFS** dễ cài đặt và có thời gian tính toán nhanh hơn các phương pháp khác [62].

Phương pháp chọn lựa số đặc trưng k: giả sử tất cả các đặc trưng đã được tính điểm và sắp xếp theo thứ tự giảm dần

$$E(p) = \frac{\sum_{j=1}^p s(k_j)}{\sum_{i=1}^d s(i)}$$

s(k_1)>=s(k_2)>=...>=s(k_d) ta tính hàm như vậy p

được chọn phải thỏa mãn $p = \arg \min E(p)$ sao cho $E(p) \geq T$, trong đó $T \geq 80\%$.

4.3.2.4 Xây dựng bộ phân lớp

Trong luận văn này, chúng tôi chọn mô hình máy học cơ chế véc tơ hỗ trợ (Support Vector Machines – **SVM**) làm mô hình chính trong thí nghiệm của mình. **SVM** đã được chứng minh là một trong những giải thuật phân lớp tốt nhất hiện nay cho vấn đề phân loại văn bản [28]. **SVM** phù hợp với bài toán phân loại văn bản vì theo [28] **SVM** có khả năng đáp ứng các yêu cầu sau:

- Không gian đầu vào có số chiều rất lớn
- Các đặc trưng rời rạc, ít liên hệ lẫn nhau
- Các véc tơ tài liệu là thưa
- Các vấn đề phân lớp trong văn bản là có thể chia cắt được.

Bên cạnh đó, trong luận văn này chúng tôi cũng cài đặt phương pháp **kNN** (**k-Nearest Neighbours**) làm phương pháp cơ sở để so sánh (baseline algorithm) và đánh giá.

4.3.3 Tiếp cận theo hướng mô hình ngôn ngữ thống kê

4.3.3.1 Tiền xử lý văn bản tiếng Việt

Trong giai đoạn này, các tài liệu đầu tiên được chuẩn hóa chính tả. Sau đó, chúng được qua các bộ phận tách đoạn và tách câu để chuẩn bị cho các xử lý tiếp theo.

4.3.3.2 Xây dựng mô hình ngôn ngữ

Như đã giới thiệu trong phần 3.3.8, trong phần này chúng tôi sẽ giới thiệu cụ thể mô hình ngôn ngữ thông kê n-gram (Statistical N-Gram Language Modeling - SLM). Đây là cách tiếp cận mới cho bài toán phân loại văn bản được Fuchen Peng et al [26] đề xuất. Trong công trình của mình, các tác giả đã chứng minh bằng thực nghiệm phân loại văn bản dựa trên mô hình ngôn ngữ rất thành công trên các ngôn ngữ Trung Quốc, Nhật Bản và một phần trên tiếng Anh. Tuy nhiên, cho đến bây giờ vẫn chưa có một công trình nào áp dụng phương pháp này trên tiếng Việt. Dựa trên sự tương đồng của tiếng Việt và tiếng Trung Quốc và tiếng Nhật, trong luận văn này, chúng tôi bước đầu thí nghiệm phương pháp này cho bài toán phân loại và từ đó so sánh với phương pháp trong cách tiếp cận trước.

Mục đích của mô hình ngôn ngữ là đoán (predict) được xác suất của dãy từ (word sequences) hay nói đơn giản hơn là đặt xác suất cao trên dãy từ mà thật sự xảy ra nhiều. Dựa ra dãy từ $w_1 w_2 \dots w_T$ được dùng như là ngữ liệu đánh giá, chất lượng mô hình ngôn ngữ có thể được tính bằng độ phức tạp (perplexity) hay độ hỗn loạn (entropy) trên ngữ liệu này

$$Perplexity = \sqrt[T]{\frac{1}{P(w_1 \dots w_T)}}$$

$$Entropy = \log_2 Perplexity$$

[4.13]

Mục đích của mô hình ngôn ngữ là đạt được độ phức tạp nhỏ. Mô hình đơn giản và thành công nhất của mô hình ngôn ngữ là mô hình n-gram (xem phần định nghĩa n-gram trong 3.3.9). Chú ý rằng bằng dãy luật xác xuất chúng ta có thể viết ra xác suất của bất cứ dãy nào như sau:

$$P(w_1 w_2 \dots w_T) = \prod_{i=1}^T P(w_i | w_1 \dots w_{i-1})$$

[4.14]

Bất cứ mô hình n-gram nào cũng xác suất này bằng cách giả sử rằng những từ liên hệ để đoán xác suất $P(w_i | w_1 \dots w_{i-1})$ là n-1 từ trước đó. Ngoài ra, nó còn giả sử tính độc lập n-gram dãy Markov như sau

$$P(w_i | w_1 \dots w_{i-1}) = P(w_i | w_{i-n+1} \dots w_{i-1}) \quad [4.15]$$

Một ước lượng độ tương đồng (likelihood) cực đại dễ hiểu từ ngữ liệu được cho bởi tần số xuất hiện

$$P(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\#(w_{i-n+1} \dots w_i)}{\#(w_{i-n+1} \dots w_{i-1})} \quad [4.16]$$

Trong đó $\#(\cdot)$ là số lượng tần suất xuất hiện của gram chỉ định trong ngữ liệu huấn luyện. Bởi vì tính tự nhiên của ngôn ngữ (theo luật Zip), sẽ có trường hợp có n-gram chưa từng xuất hiện trong quá trình huấn luyện. Vì thế, các kỹ thuật gán xác suất bằng không cho n-grams mới là một vấn đề không thể tránh khỏi. Một cách tiếp cận chuẩn làm tròn (smoothing) ước lượng xác suất để đối phó với các vấn đề dữ liệu thưa (và để đối phó với các n-gram không tồn tại) là sử dụng một vài loại phép nội suy tuyến tính (linear interpolation) hay ước lượng “quay lui” (back-off estimator). Mô hình phép nội suy tuyến tính bao gồm thủ tục cực đại kỳ vọng EM (Expectation Maximum) để tối ưu trọng số mỗi thành phần. Trong khi đó, mô hình “quay lui” đơn giản hơn và rất thích hợp với mô hình kết hợp với Naïve Bayes.

$$P(w_i | w_{i-n+1} \dots w_{i-1}) = \begin{cases} \hat{P}(w_i | w_{i-n+1} \dots w_{i-1}) & \text{if } \#(w_{i-n+1} \dots w_i) > 0 \\ \beta(w_{i-n+1} \dots w_{i-1}) * P(w_i | w_{i-n+2} \dots w_{i-1}) & \text{otherwise} \end{cases} \quad [4.17]$$

Trong đó:

$$\hat{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\text{discount } \#(w_{i-n+1} \dots w_i)}{\#(w_{i-n+1} \dots w_{i-1})} \quad \text{là xác xuất không}$$

đếm được (discounted probability) và $\beta(w_{i-n+1} \dots w_{i-1})$ là hằng chuẩn hóa được tính như sau:

$$\beta(w_{i-n+1} \dots w_{i-1}) = \frac{1 - \sum_{x: \#(w_{i-n+1} \dots w_{i-1}x) > 0} \bar{P}(x | w_{i-n+1} \dots w_{i-1})}{1 - \sum_{x: \#(w_{i-n+1} \dots w_{i-1}x) > 0} \bar{P}(x | w_{i-n+2} \dots w_{i-1})} \quad [4.18]$$

Bất kỳ một n-gram nào đầu tiên được so khớp với mô hình ngôn ngữ sẽ được xem trong ngữ liệu huấn luyện. Nếu n-gram này không tồn tại, nó sẽ giảm xuống còn n-1-gram bằng cách là bỏ đi ngữ cảnh 1 từ. Xác suất không đếm được (discounted probability) có thể tính được bằng cách sử dụng các kỹ thuật làm tròn khác nhau (smoothing approaches). Các cách tiếp cận “làm tròn” được quan tâm trong luận văn này [26] gồm: **linear, absolute, Good-Turing, Witten-Bell**.

Để mô tả các kỹ thuật “làm tròn” này, chúng ta ký hiệu n_i là số lượng sự kiện (event) mà xảy ra i lần trong ngữ liệu huấn luyện

❖ **Absolute smoothing:**

$$\bar{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\#(w_{i-n+1} \dots w_i) - b}{\#(w_{i-n+1} \dots w_{i-1})} \quad [4.19]$$

Trong đó b thường được định nghĩa là $b = \frac{n_1}{n_1 + 2n_2}$

❖ **Linear smoothing:**

$$\bar{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \left(1 - \frac{n_1}{T}\right) * \frac{\#(w_{i-n+1} \dots w_i)}{\#(w_{i-n+1} \dots w_{i-1})} \quad [4.20]$$

Trong đó T là số lượng uni-gram

❖ **Good-Turing smoothing:**

$$\bar{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{GT_{\#(w_{i-n+1} \dots w_i)}}{\#(w_{i-n+1} \dots w_{i-1})} \quad [4.21]$$

Trong đó $GT_r = (r+1) \frac{n_r + 1}{n_r}$

❖ **Witten-Bell smoothing:**

$$\bar{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\#(w_{i-n+1} \dots w_i)}{\#(w_{i-n+1} \dots w_{i-1}) + D(w_{i-n+1} \dots w_{i-1})} \quad [4.22]$$

Trong đó $D(w_{i-n+1} \dots w_{i-1})$ là số lượng những từ riêng biệt có thể theo sau

$w_{i-n+1} \dots w_{i-1}$ trong ngữ liệu huấn luyện.

4.3.3.3 Sử dụng mô hình Naïve Bayes kết hợp với mô hình ngôn ngữ thống kê n-gram

Thông thường, bộ phân lớp văn bản cố gắng xác định các thuộc tính mà phân biệt các tài liệu giữa các chủ đề với nhau. Các thuộc tính có thể gồm các hạng từ vựng, chiều dài trung bình từ, các n-gram cục bộ, hay cấu trúc toàn cục và tính chất ngữ nghĩa. Các tính chất này nói một cách cơ bản có thể được xây dựng từ mô hình ngôn ngữ. Một mô hình ngôn ngữ n-gram có thể được áp dụng cho phân loại văn bản giống như phương pháp phân loại **Naïve Bayes**.

Dưa ra C là tập các chủ đề, d là 1 tài liệu đánh giá mới:

$$\begin{aligned} c^* &= \arg \max \{P(c)P(d | c)\} \\ &= \arg \max_{c \in C} \left\{ P(c) \prod_{i=1}^T P(w_i | w_{i-n+1} \dots w_{i-1}, c) \right\} \\ &= \arg \max_{c \in C} \left\{ P(c) \prod_{i=1}^T P_c(w_i | w_{i-n+1} \dots w_{i-1}) \right\} \end{aligned} \quad [4.23]$$

c^* là phân lớp tốt nhất cho tài liệu mới d . $P(c)$ có thể được tính từ tập huấn luyện. $P_c(w_i | w_{i-n+1} \dots w_{i-1})$ được tính bằng cách sử dụng mô hình “quay lui” (back-off model) như đã nói trong phần trước.

Như vậy, để biết được chủ đề của tài liệu mới d ta đơn giản chỉ tính độ phức tạp (perplexity) của tài liệu mới này so với mô hình ngôn ngữ (đã huấn luyện từ tập huấn luyện).

Một lưu ý nhỏ là đơn vị cơ bản nhất mô tả trong phần trên là từ. Trong cách tiếp cận này, chúng tôi sẽ áp dụng mô hình ngôn ngữ dựa trên tiếng đơn giản chỉ cần xem văn bản như là dãy các tiếng liền nhau. Nguyên nhân là do lợi ích của mô hình ngôn ngữ dựa trên tiếng sẽ trình bày trong phần tiếp theo.

4.3.3.4 Các lợi ích của mô hình ngôn ngữ

Các lợi ích của mô hình ngôn ngữ dựa trên tiếng:

- ❖ Tránh được bài toán tách từ vốn đã rất khó khăn đối với tiếng Việt
- ❖ Tránh được vấn đề chọn lựa đặc trưng (có thể gây những đặc trưng dư thừa)

- ❖ Mô hình ngôn ngữ dựa trên tiếng bé hơn mô hình dựa trên từ về kích thước và nó cũng giảm được vấn đề dữ liệu thưa (sparse data).

4.3.4 Lọc và tìm kiếm tài liệu

Module phân lớp tài liệu như đã trình bày trong **phần trước** sẽ được đóng gói thành thư viện **COM (Component Object Model)**. Tính tiện lợi khi sử dụng môi trường **COM** là có thể sử dụng ở bất cứ ngôn ngữ lập trình nào (dùng cho tính kế thừa và sử dụng) và nhiều hệ điều hành (**Windows, Linux, Mac, ...**).

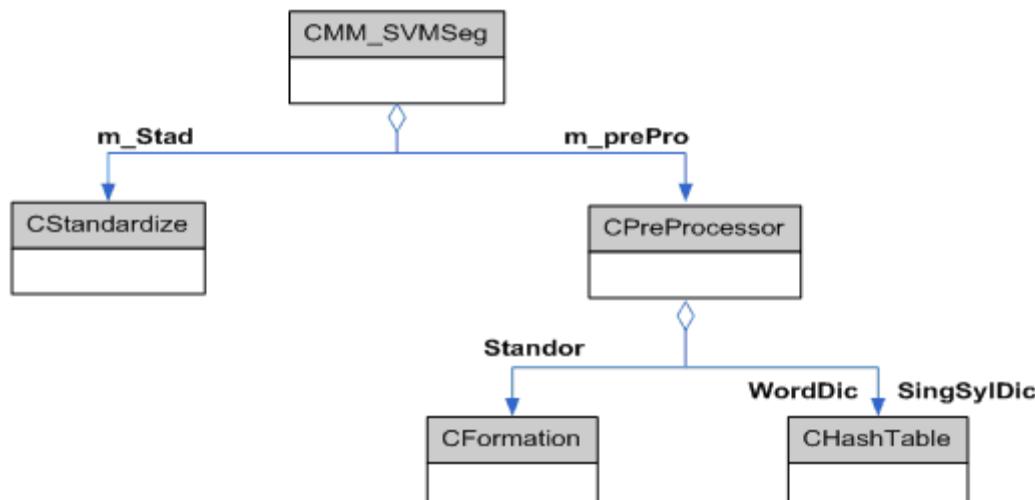
Module lọc tài liệu đầu tiên sẽ thu thập các văn bản hay những tài liệu từ web thông qua 1 web crawler. Sau đó, qua Module phân loại tài liệu các tài liệu sẽ được phân loại chủ đề (các chủ đề đã chọn lựa trước), chúng sẽ được lưu xuống database và sẽ được dùng cho các xử lý sau này.

4.4 Thiết kế cài đặt

Trong phần này, chúng tôi sẽ trình bày mô hình thiết kế cho các module chính: tách từ và phân loại văn bản. Chúng tôi cũng sẽ nêu ra một số phương thức chính được cài đặt trong chương trình. Cuối cùng, chúng tôi sẽ trình bày phần ứng dụng cụ thể của chương trình tìm kiếm văn bản tiếng Việt theo chủ đề có tên là **VCL_TVDoS** (*Topic-based Vietnamese Document Search*).

4.4.1 Thiết kế cài đặt thư viện tách từ

4.4.1.1 Sơ đồ lớp



Hình 22: Sơ đồ lớp ứng dụng tách từ

4.4.1.2 Cài đặt

❖ Lớp CMM_SVMSeg

◆ Chức năng

Chức năng chính của lớp này là thực hiện nhiệm vụ tách từ cho văn bản đầu vào thành hai dạng chính: tách từ và lưu thành file, tách từ và lưu thành 1 danh sách các từ. Đồng thời lớp CMM_SVMSeg cũng thực hiện chức năng chuẩn hóa các dấu câu đặc biệt.

◆ Cài đặt

//hàm chuẩn hóa dấu _

```
void Process_Line(CString &);  
//hàm chuẩn hóa chấm  
void Process_Dot(CString &);  
//tách từ cho 1 file và lưu thành file  
void SegmentText(CString ,CString );  
//tách từ cho 1 file thành 1 danh sách các từ  
void Segment2List(CString ,CStringList& );
```

❖ Lớp CpreProcessor

◆ Chức năng

Thừa kế từ lớp CMM_SVMSeg, thực hiện các công đoạn tiền xử lý cho quá trình tách từ bằng các phương pháp so khớp từ trái qua phải (LRMM_Segment) và từ phải qua trái (RLMM_Segment)

◆ Cài đặt

```
//tách từ theo thuật toán so khớp từ phải sang trái  
void RLMM_Segment(CString &Sentence);  
//tách từ theo thuật toán so khớp từ trái sang phải  
void LRMM_Segment(CString &Sentence);
```

❖ Lớp CStandardize

◆ Chức năng

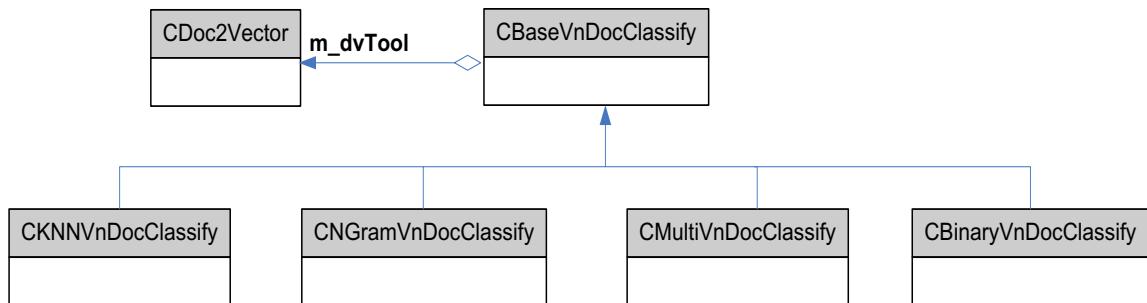
Thừa kế từ lớp CMM_SVMSeg, thực hiện chức năng chuẩn hóa văn bản đầu vào thành các danh sách phục vụ giai đoạn tách từ

◆ Cài đặt

```
//định dạng văn bản đầu vào  
void FormatText(CString& );  
//tách chuỗi thành danh sách các symbol  
void ParseSymbol(CString,CStringList&,char);  
//tách chuỗi thành danh sách các tiếng  
void ParseSyllable(CString,CStringList&,bool);
```

4.4.2 Thiết kế cài đặt module phân loại văn bản

4.4.2.1 Sơ đồ lớp



4.4.2.2 Cài đặt

❖ Lớp CDoc2Vector

◆ Chức năng

Thực hiện nhiệm vụ véc tơ hóa văn bản đầu vào thành các véc tơ trọng số. Trong đó bao gồm các chức năng chính như trọng số hóa véc tơ, giảm số chiều đặc trưng, chuyển văn bản thành các đặc trưng và gỡ bỏ những từ vô nghĩa (stop word)

◆ Cài đặt

Cấu trúc dữ liệu mô tả trọng số hóa tài liệu

```

struct docWeight
{
    //số thứ tự của lớp chứa tài liệu
    int docClass;
    //tổng trọng số các term mà tài liệu có chứa
    float totalWeight;
    //chỉ số của tài liệu trong tập ngữ liệu huấn luyện
    int index;
};
  
```

Thông tin chỉ mục của một tài liệu

```

struct docIndex
{
    //số lượng term mà tài liệu có tham gia
    int count;
    //chỉ số của tài liệu trong tập ngữ liệu huấn luyện
    int index;
  
```

```
//trọng số của tài liệu với term hiện hành  
float weight;  
//lớp hiện hành của tài liệu  
int docClass;  
//con trỏ tài liệu kế tiếp  
struct docIndex *nextIndex;  
};
```

Thông tin của một term

```
struct item  
{  
    //từ hiện tại  
    CString text;  
    //số lượng term xuất hiện  
    int wrdCount;  
    //tần số xuất hiện của tổng các term  
    long numTermTotal;  
    //số lượng tài liệu có chứa term hiện tại  
    int docCount;  
    //tần số IDF  
    float IDF;  
    //điểm của 1 term  
    float s;  
    //danh sách các tài liệu liên hệ có chứa term  
    struct docIndex *docList;  
    //con trỏ term kế tiếp  
    struct item *nextItem;  
};  
  
//tạo 1 term mới  
void CreateNewItem(struct item *prevIt, struct item **newIt, struct item *nextIt, CString wrd, int listSize, int docClass, int numDocs);  
//tạo một chỉ mục tài liệu mới  
void CreateNewDocIndex(struct docIndex *docInd, struct docIndex **newIndex, int docClass, int numDocs);  
//tính trọng số term  
void Calc_Term_weights();  
//tính trọng số tài liệu  
void Calc_Doc_weights();  
//lấy thông tin tài liệu  
void GetDocInfo(CString , CString& );  
//chọn lựa đặc trưng bằng phương pháp OCFS  
void ReduceDim_OCFS(CString , CString );  
//chọn lựa đặc trưng bằng phương pháp IG  
void ReduceDim_GSS(CString , CString );
```

```
//chọn lựa đặc trưng bằng phương pháp MI
void ReduceDim_MI(CString ,CString );
//chọn lựa đặc trưng bằng phương pháp OR
void ReduceDim_OR(CString ,CString );
//chọn lựa đặc trưng bằng phương pháp CHI
void ReduceDim_CHI(CString ,CString );
//chọn lựa đặc trưng bằng phương pháp IG
void ReduceDim_IG(CString ,CString );
//gỡ bỏ stopwords
void RemoveStopWords(CStringList& lstTokens);
//tiền xử lý các token
void PreProcessing(CStringList& );
//lấy đặc trưng term của tài liệu
struct svm_node* GetFeatures(Cstring, int&);
//thực hiện quá trình chọn lựa đặc trưng
void Process_ReduceDim(CString, CString, int);
```

❖ Lớp CBaseVnDocClassify

◆ Chức năng

Thực hiện các quá trình chính cho mô hình phân loại văn bản, bao gồm học và kiểm nghiệm. Trong lớp này sẽ thực hiện thống kê đặc trưng từ ngữ liệu huấn luyện. Lớp CBaseVnDocClassify sẽ là lớp chứa hàm ảo cho các lớp con thừa kế. Trong các lớp thừa kế sẽ định nghĩa lại các hàm ảo dành cho quá trình huấn luyện và kiểm nghiệm phù hợp theo từng mô hình cụ thể

◆ Cài đặt

```
//thống kê từ ngữ liệu huấn luyện
void Statistics(CString, CString, CString);
//lấy danh sách chủ đề
void LoadCategs(CString );
//khởi tạo danh sách các đặc trưng đã huấn luyện
void Init(CString );
//huấn luyện
virtual void Train(CString ,CString ,CString ,
CString, int);
//kiểm nghiệm 1 file
virtual CString Test(CString );
//kiểm nghiệm theo đường dẫn thư mục
virtual float TestPath(CString ,CString ,CString );
```

❖ Lớp CMultiVnDocClassify

◆ Chức năng

Thừa kế từ lớp CBaseVnDocClassify, thực hiện huấn luyện và kiểm nghiệm với mô hình SVM-Multi

◆ **Cài đặt**

```
//khởi tạo mô hình SVM đã huấn luyện  
void Init_SVM(char* );
```

❖ **Lớp CBinaryVnDocClassify**

◆ **Chức năng**

Thừa kế từ lớp CBaseVnDocClassify, thực hiện huấn luyện và kiểm nghiệm với mô hình SVM-Binary

◆ **Cài đặt**

```
//khởi tạo các mô hình đã huấn luyện  
void LoadListModelFile(CString);
```

❖ **Lớp CKNNVnDocClassify**

◆ **Chức năng**

Thừa kế từ lớp CBaseVnDocClassify, thực hiện huấn luyện và kiểm nghiệm với mô hình kNN

◆ **Cài đặt**

```
//khởi tạo các mô hình đã huấn luyện  
void LoadTrainningFile(CString);  
//ghi nhận số term chọn lựa cho mô hình  
void SetNumTerms(int);
```

❖ **Lớp CNGramVnDocClassify**

◆ **Chức năng**

Thừa kế từ lớp CBaseVnDocClassify, thực hiện huấn luyện và kiểm nghiệm với mô hình N-Gram

◆ **Cài đặt**

Cấu trúc tham số cấu hình

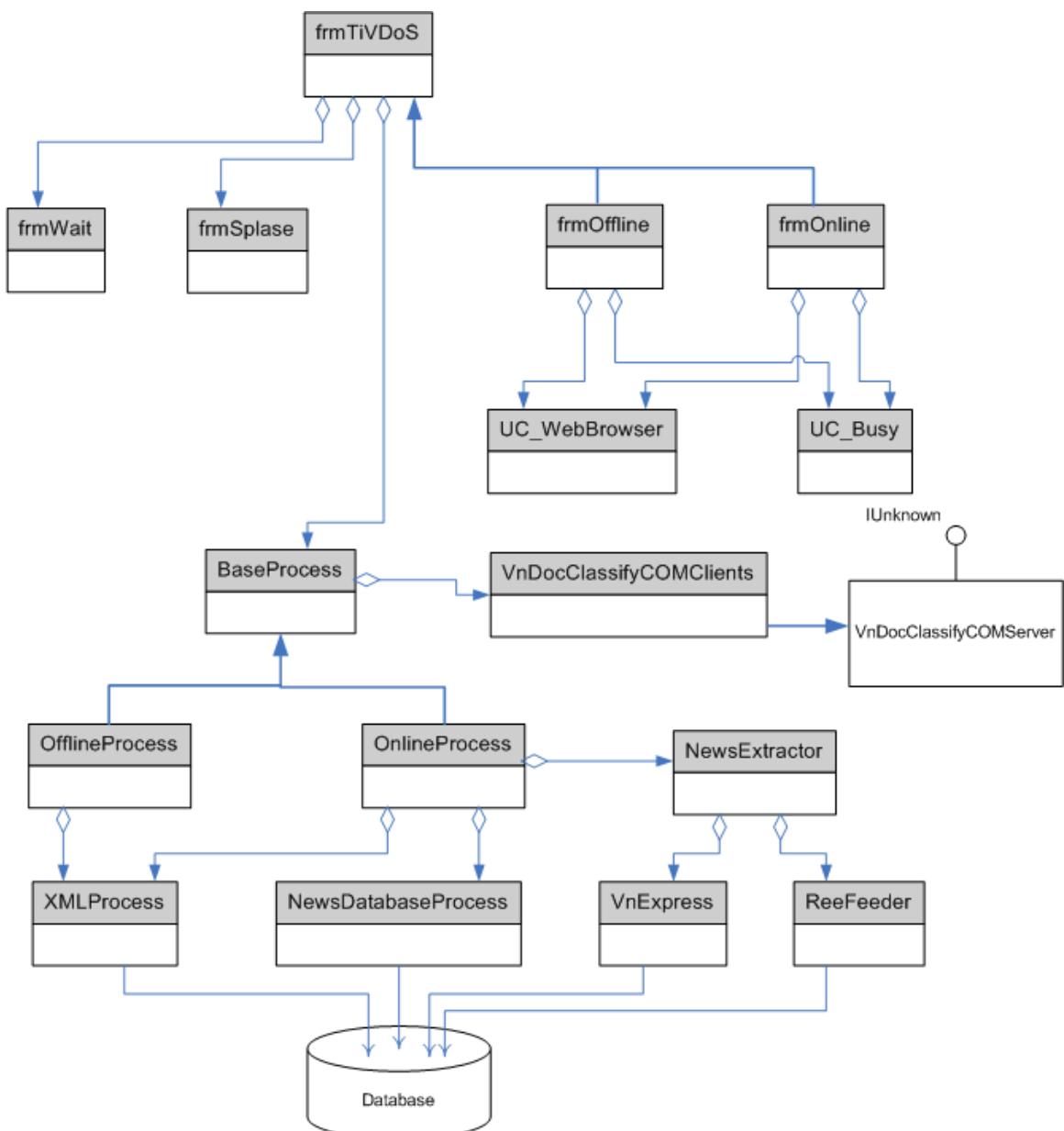
```
typedef struct cmd_param  
{  
    int n;  
    //text2idngram  
    int nBuffer;  
    BOOL write_ascii;  
    //idngram2lm  
    int vocab_type;
```

```
int discounting_method;
CString cut_offs;
BOOL bin_input;
}CMD_PARAM;

//chuyển file tài liệu thành nội dung chuỗi
void ReadU2V(CString docName, BYTE *&vOut, int& nLen);
void ReadU2V(CString docName, CString vOutFile);
//huấn luyện N-Gram
void TrainNGram(CString path, CString pathOut);
//xử lý chính tả
void FormatString(CString& content);
//tiền xử lý văn bản đầu vào
void PreProcessing(CString &content);
```

4.4.3 Thiết kế cài đặt ứng dụng tìm kiếm văn bản tiếng Việt theo chủ đề

4.4.3.1 Sơ đồ lớp

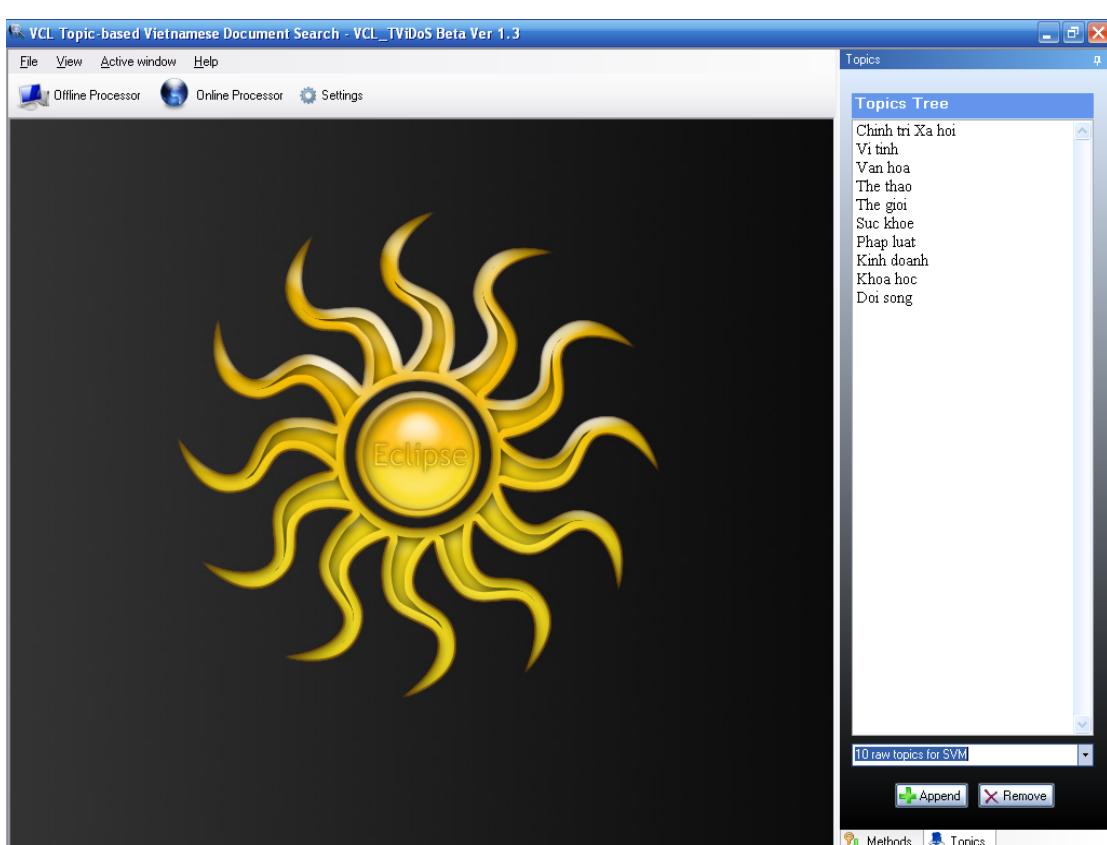


Hình 24: Sơ đồ lớp ứng dụng tìm kiếm văn bản tiếng Việt theo chủ đề

4.4.3.2 Giao diện ứng dụng

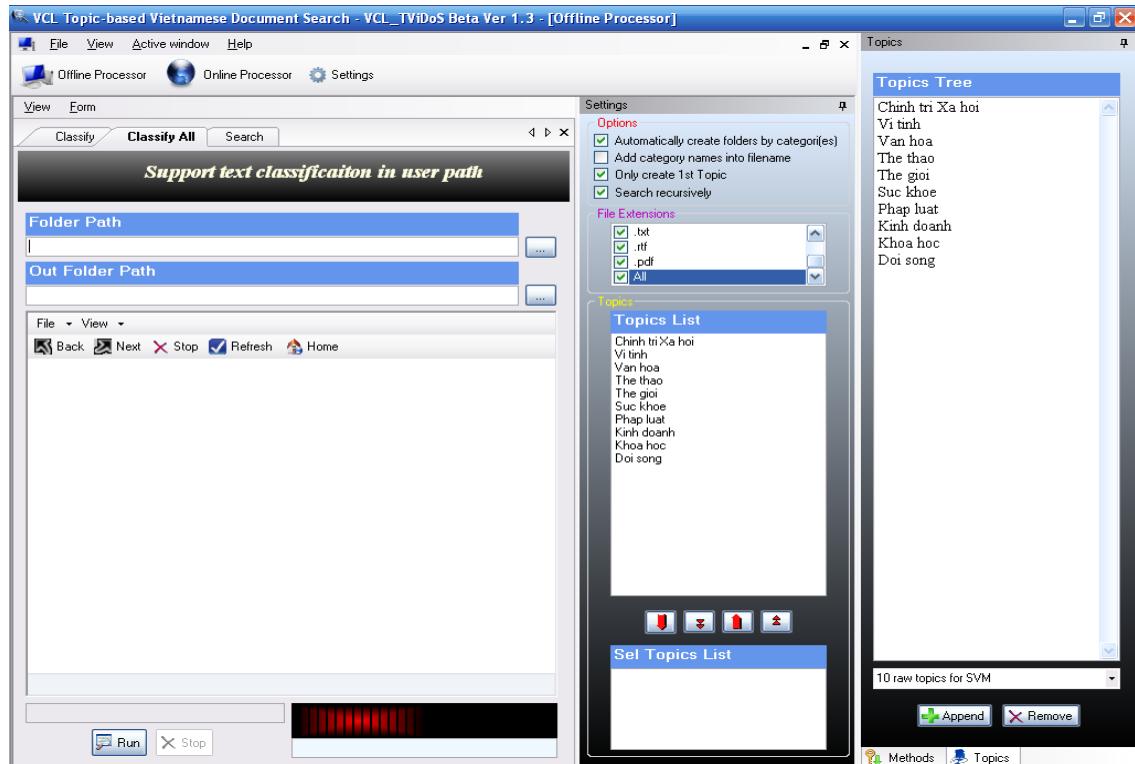


Hình 25: Màn hình Splash

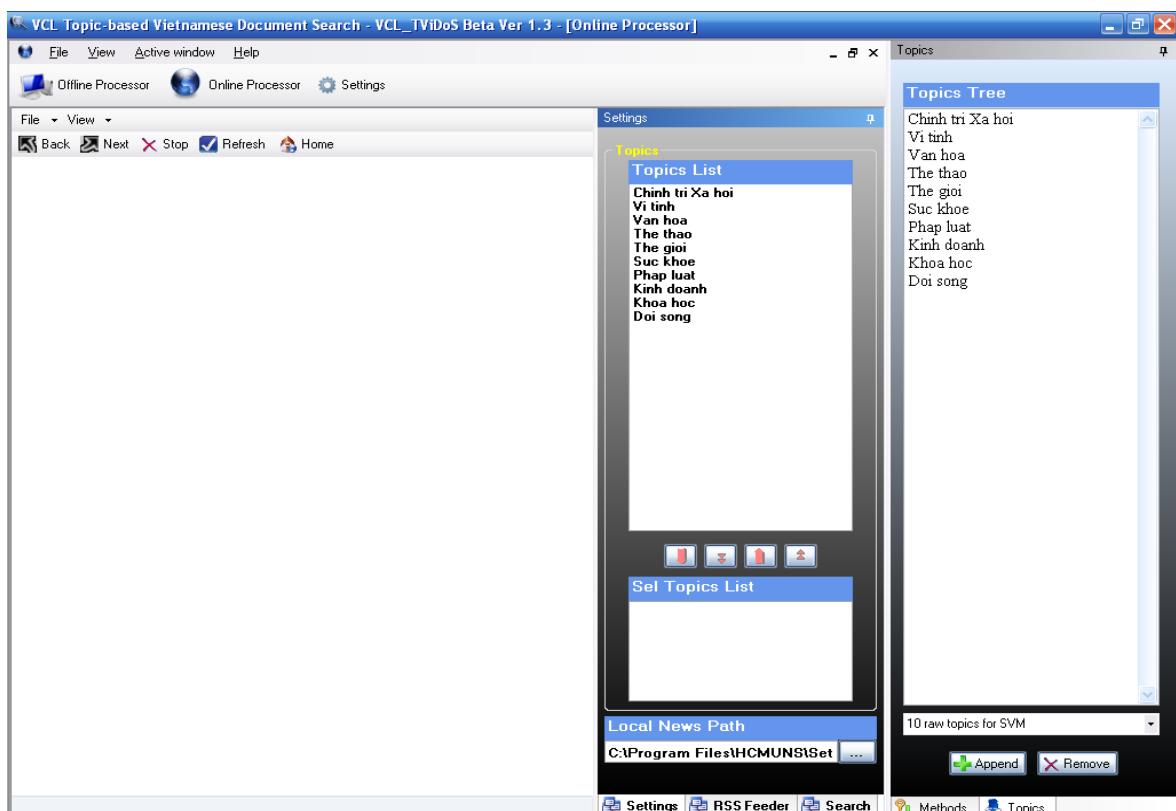


Hình 26: Giao diện màn hình chính ứng dụng

Chương 4: MÔ HÌNH – THIẾT KẾ – CÀI ĐẶT



Hình 27: Giao diện màn hình chức năng Offline



Hình 28: Giao diện màn hình chức năng online

4.4.3.3 Cài đặt

- ❖ Lớp **frmTViDos**, **frmOffline**, **frmOnline**: các lớp giao diện màn hình chính, màn hình xử lý offline, màn hình xử lý online
- ❖ Lớp **UC_WebBrowser**, **UC_Busy**: các lớp user control
- ❖ Lớp **BaseProcess**, **OfflineProcess**, **OnlineProcess**: các lớp xử lý thực hiện các nhiệm vụ liên quan đến phân loại văn bản, tìm kiếm văn bản
- ❖ Lớp **VnDocClassifyCOMClient**: thực hiện liên kết COM phân loại văn bản
- ❖ Lớp **NewsExtractor**: thực hiện nhiệm vụ lấy tin tức từ báo điện tử
- ❖ Lớp **XMLProcess**, **NewsDatabaseProcess**, **VnExpress**, **RssFeeder**: thực hiện nhiệm vụ thao tác cơ sở dữ liệu, ghi và đọc file Xml và lưu trữ file text lấy được xuống đĩa cứng

Chương 5: KẾT QUẢ THỰC NGHIỆM

Nội dung

Chương này sẽ trình bày các kết quả thực nghiệm đạt được trên các tập dữ liệu cho các bài toán tách từ và phân loại văn bản tiếng Việt mà chúng tôi đã cài đặt thực nghiệm.

5.1 Bài toán tách từ tiếng Việt

5.1.1 Thí nghiệm

Chúng tôi sử dụng kho ngữ liệu vnQTAG [67] và CADASA (VCL Group) để thí nghiệm. Ngữ liệu CADASA được mô tả trong phần 4.3.2.1. Ngữ liệu QTAG gồm 7 file văn bản : 75594 từ (6058 từ khác nhau), 87993 tiếng (3724 tiếng khác nhau). Chúng tôi tạo ra từ điển dùng cho thuật toán MM với tất cả các từ trong ngữ liệu.

Văn bản / Văn phong	Số từ
Chuyện tình 1 / Tiếu thuyết	16787
Chuyện tình 2 / Tiếu thuyết	14698
Hoàng tử bé / Truyện nước ngoài	18663
Lược sử thời gian / Sách khoa học	11626
Muối của rừng / Truyện ngắn	3573
Những bài học / Truyện ngắn	8244
Công nghệ / Báo chí	1162

Bảng 13: Thống kê trên ngữ liệu QTAG

❖ **Huấn luyện:** chúng tôi sử dụng ngữ liệu CADASA. Từ điển từ dùng trong thí nghiệm được tạo ra từ ngữ liệu này.

❖ **Kiểm nghiệm:** chúng tôi sử dụng ngữ liệu QTAG để đánh giá.

Ngoài ra, chúng tôi còn thử nghiệm 5 mô hình khác nhau để đánh giá và tìm ra mô hình tốt nhất cho bài toán đặt ra.

❖ **Mô hình SVM:** chỉ sử dụng thông tin các tiếng (không dùng thông tin đầu ra của FMM và BMM)

❖ **Mô hình FMM + SVM:** sử dụng đầu ra FMM và các tiếng

- ❖ **Mô hình BMM + SVM:** sử dụng đầu ra **BMM** và các tiếng
- ❖ **Mô hình FMM + BMM + SVM:** sử dụng đầu ra **FMM, BMM** và các tiếng.
- ❖ **Mô hình FMM + BMM + SVM + PN(Proper Nouns):** sử dụng đầu ra **FMM, BMM**, các tiếng và danh sách tên riêng (không có ngữ cảnh).

5.1.2 Đánh giá

Để đánh giá kết quả của hệ thống, chúng tôi sử dụng cách đánh giá trong [68]:

- Số lượng từ của kho ngữ liệu đánh giá (word count – N)
- Số lượng từ nhận diện đúng trên tổng số từ có trong ngữ liệu đánh giá (**recall** – R)

$$R = \frac{N_3}{N_1}$$

- Số lượng từ nhận diện đúng trên tổng số từ nhận diện được (**precision** – P)

$$P = \frac{N_3}{N_2}$$

- Hệ số cân bằng F:

$$F = \frac{(1+\beta)PR}{\beta P+R}$$

Trong thí nghiệm này chúng tôi chọn $\beta=1$ đảm bảo tính khách quan khi so sánh với các thuật toán khác. Trong đó:

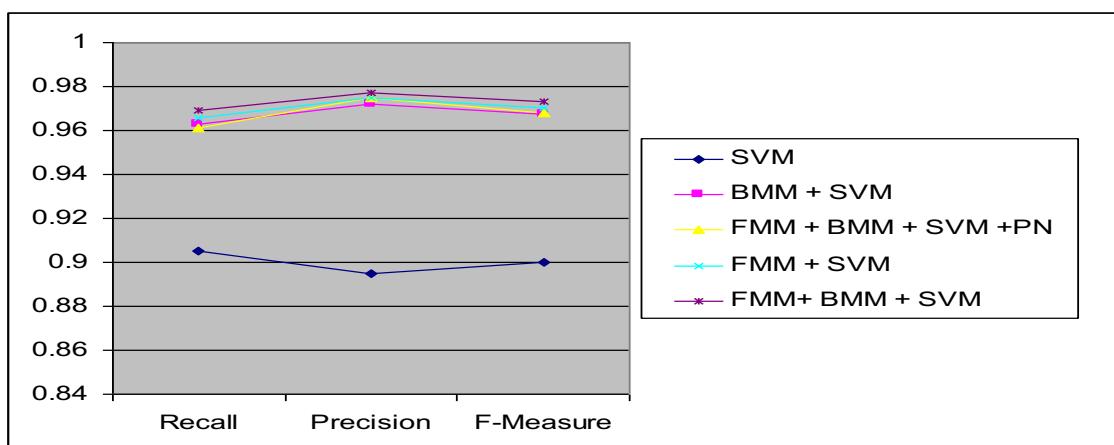
- N_3 : số từ được nhận diện đúng
- N_2 : số từ được nhận diện bằng mô hình thuật toán
- N_1 : số từ trong ngữ liệu huấn luyện

5.1.3 Kết quả

❖ *Thí nghiệm so sánh các mô hình tách từ:*

Phương pháp	Recall	Precision	F-Measure
SVM	0.905	0.895	0.9
BMM + SVM	0.963	0.972	0.9675
FMM + BMM + SVM +PN	0.961	0.975	0.9679
FMM + SVM	0.966	0.975	0.9705
FMM+ BMM + SVM	0.969	0.977	0.973

Bảng 14: Kết quả thực nghiệm so sánh các mô hình tách từ

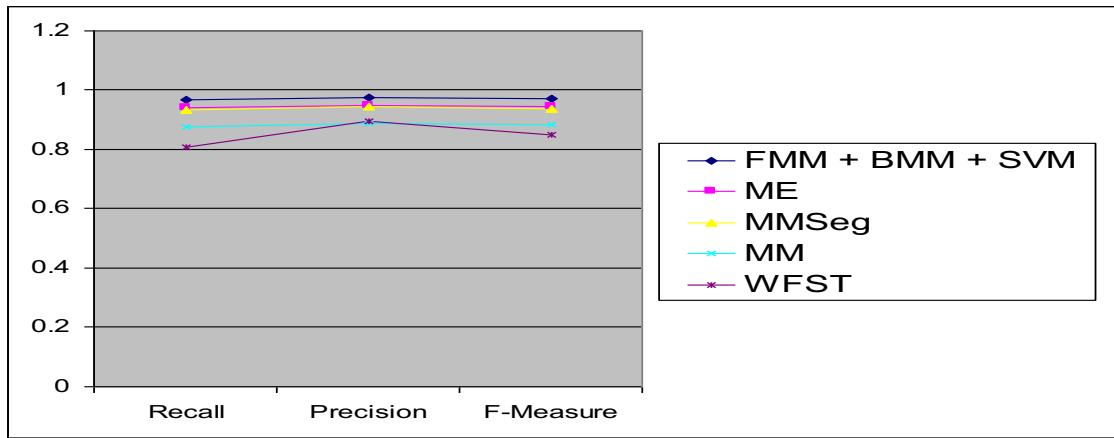


Biểu đồ 1: So sánh hệ thống với các mô hình tách từ

❖ *Thí nghiệm so sánh với các phương pháp trước:*

Phương pháp	Recall	Precision	F-Measure
FMM + BMM + SVM	0.969	0.974	0.9715
ME	0.941	0.949	0.945
MMSeg	0.933	0.945	0.939
MM	0.878	0.887	0.8825
WFST	0.808	0.894	0.8488

Bảng 15: Kết quả thực nghiệm so sánh với các phương pháp trước



Biểu đồ 2: So sánh hệ thống với các phương pháp trước

Trong đó:

- **WFST** (Weighted Finite State Transducer : chuyển đổi trạng thái trọng số hữu hạn) [16]
- **ME** (Maximum Entropy : độ hỗn loạn cực đại) [11]
- **MMSeg** (Maximum Matching Segmentation: so khớp cực đại mở rộng) [11]

5.1.4 Nhận xét

Kết quả đạt được của hệ thống chúng tôi cho thấy tính khả thi của mô hình kết hợp nói trên. Hiệu năng của hệ thống tốt hơn so với các hệ thống khác hiện có, chúng tôi đạt được **97.2%** so với 94.5% **ME**, 93.93% **MMSeg**, 84.88% **WFST**, 88.28% **MM**. Với kết quả đạt được như vậy, chúng tôi hi vọng sẽ làm tăng kết quả cho bài toán phân loại văn bản.

5.2 Bài toán phân loại văn bản tiếng Việt

5.2.1 Thí nghiệm

Chúng tôi sử dụng ngữ liệu là các file văn bản đã được nêu trong phần 5.1.1 để tiến hành thí nghiệm. Dữ liệu này sau khi thu thập đã được chỉnh sửa bởi các nhà ngôn ngữ học dưới sự giới thiệu và giúp đỡ của Đinh Điền.

Trong thí nghiệm này, chúng tôi thử nghiệm trên 2 bộ dữ liệu: dữ liệu thô (10 chủ đề) và dữ liệu mịn (27 chủ đề). Trên mỗi bộ dữ liệu sẽ phân chia thành 3 phần chính.

- Thí nghiệm so sánh kết quả phân loại văn bản bằng SVM với các phương pháp chọn lựa đặc trưng khác nhau
- Thí nghiệm so sánh kết quả phân loại văn bản bằng N-gram với các phương pháp “làm tròn” (discounting smoothing methods) khác nhau
- Thí nghiệm so sánh kết quả phân loại văn bản bằng các mô hình khác nhau SVM-Multi, SVM-Binary, kNN, N-gram

Ngoài ra chúng tôi cũng sẽ so sánh kết quả thử nghiệm với dữ liệu được cung cấp bởi những luận văn trước

5.2.1.1 Định nghĩa các giá trị độ đo

- Recall

$$\text{Recall} = \frac{N_1}{T}$$

- Precision

$$\text{Precision} = \frac{N_2}{N_3}$$

- F1

$$F1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

Trong đó:

- N_1 : tổng số tài liệu trong tập huấn luyện cho kết quả kiểm nghiệm đúng bởi mô hình
- T : tổng số tài liệu huấn luyện
- N_2 : tổng số tài liệu trong tập kiểm nghiệm cho kết quả kiểm nghiệm đúng bởi mô hình
- N_3 : tổng số tài liệu trong tập kiểm nghiệm

5.2.1.2 Các mô hình dùng trong bài toán phân loại văn bản:

- **SVM-Multi:** sử dụng phương pháp máy học SVM, trong đó mỗi phân lớp tài liệu được gán một nhãn c_i , $i = 1..n$. Như vậy, mỗi tài liệu chỉ có thể thuộc về một phân lớp duy nhất.
- **SVM-Binary:** sử dụng phương pháp máy học SVM, trong đó một phân lớp c_i được gán nhãn 1 và các phân lớp c_j còn lại đều gán nhãn -1, $j = 1..n$, $j \neq i$. Như vậy, một tài liệu có thể thuộc về nhiều phân lớp khác nhau.
- **kNN:** sử dụng phương pháp kNN
- **N-gram:** sử dụng mô hình ngôn ngữ thống kê

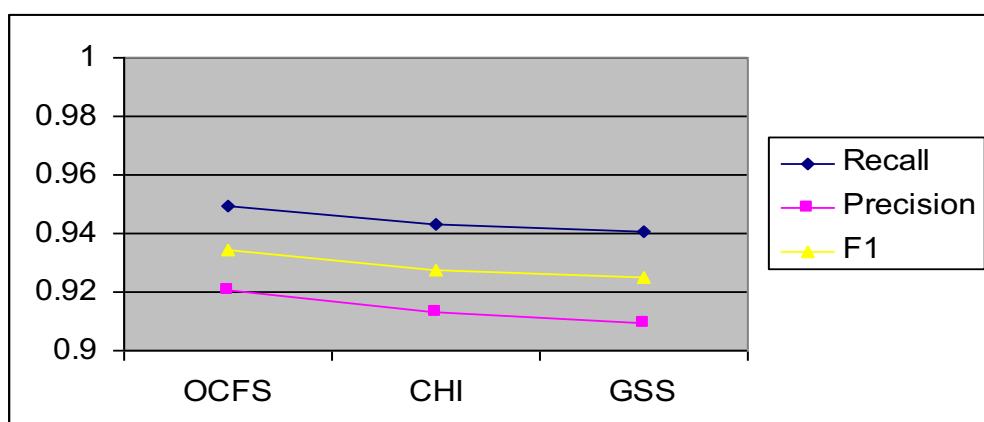
5.2.2 Dữ liệu thô (10 chủ đề)

5.2.2.1 So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 3 phương pháp chọn đặc trưng OCFS, CHI, GSS

❖ Với số đặc trưng chọn lựa là 2500

Mô hình SVM-Multi			
	OCFS	CHI	GSS
Recall	0.94908	0.943126	0.940875
Precision	0.9207244	0.9128193	0.909577
F1	0.934687194	0.927725203	0.924961317

Bảng 16: Kết quả trung bình 10 chủ đề với 2500 terms

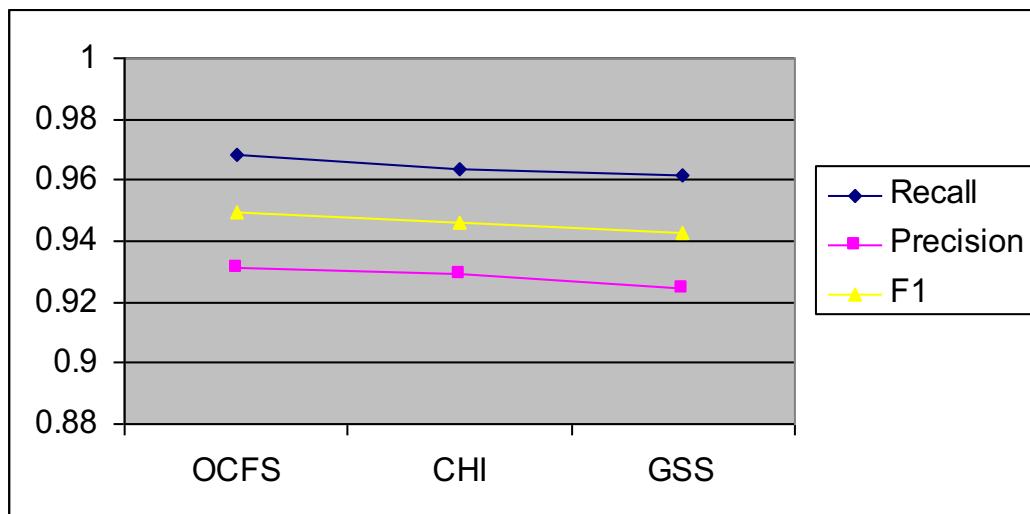


Biểu đồ 3: So sánh kết quả trung bình 10 chủ đề với 2500 terms

❖ Với số đặc trưng chọn lựa là 5000

<i>Mô hình SVM-Multi</i>			
	OCFS	CHI	GSS
Recall	0.968542	0.963299	0.961403
Precision	0.93113	0.9290102	0.92435
F1	0.949467605	0.945843942	0.942512474

Bảng 17: Kết quả trung bình 10 chủ đề với 5000 terms



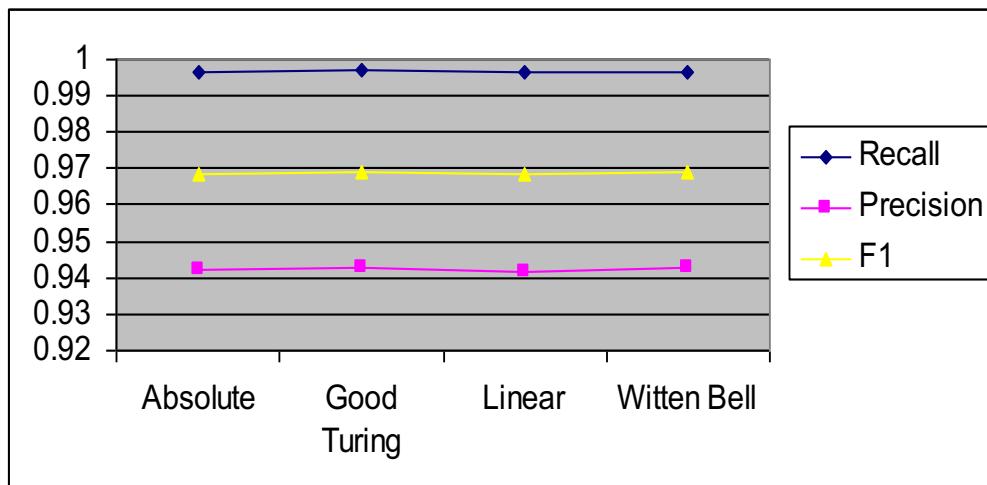
Biểu đồ 4: So sánh kết quả trung bình 10 chủ đề với 5000 terms

5.2.2.2 So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm tròn” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell

❖ Với N = 2

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Recall	0.99654	0.99702	0.996478	0.996427
Precision	0.942289	0.942968	0.941863	0.942629
F1	0.968655493	0.969241001	0.96840108	0.968781703

Bảng 18: Kết quả trung bình 10 chủ đề với N = 2

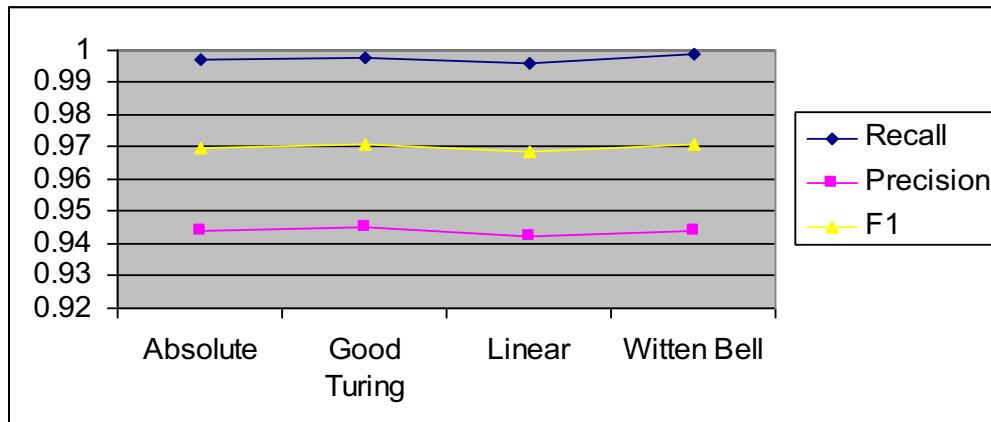


Biểu đồ 5: So sánh kết quả trung bình 10 chủ đề với $N = 2$

❖ Với $N = 3$

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Recall	0.997056	0.997836	0.996012	0.998634
Precision	0.94373	0.945167	0.942424	0.944237
F1	0.969660394	0.97078765	0.968477281	0.970673989

Bảng 19: Kết quả trung bình 10 chủ đề với $N = 3$

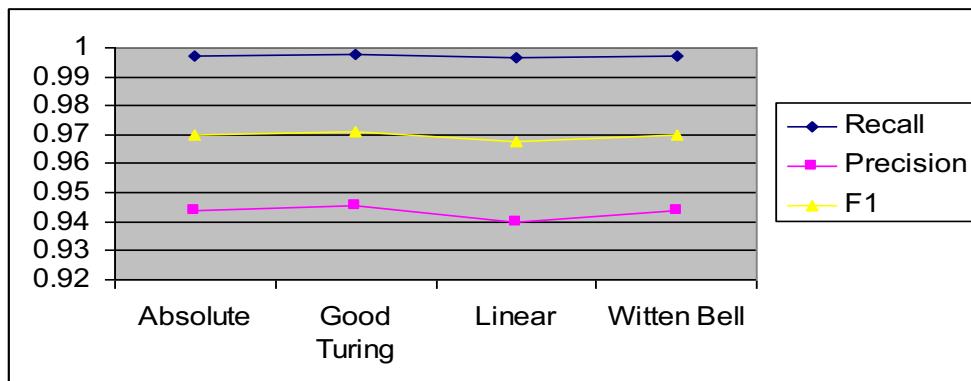


Biểu đồ 6: So sánh kết quả trung bình 10 chủ đề với $N = 3$

❖ Với $N = 4$

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Recall	0.997215	0.99784	0.996638	0.997058
Precision	0.944085	0.945533	0.940044	0.943894
F1	0.969922962	0.970982564	0.967514101	0.969747901

Bảng 20: Kết quả trung bình 10 chủ đề với $N = 4$



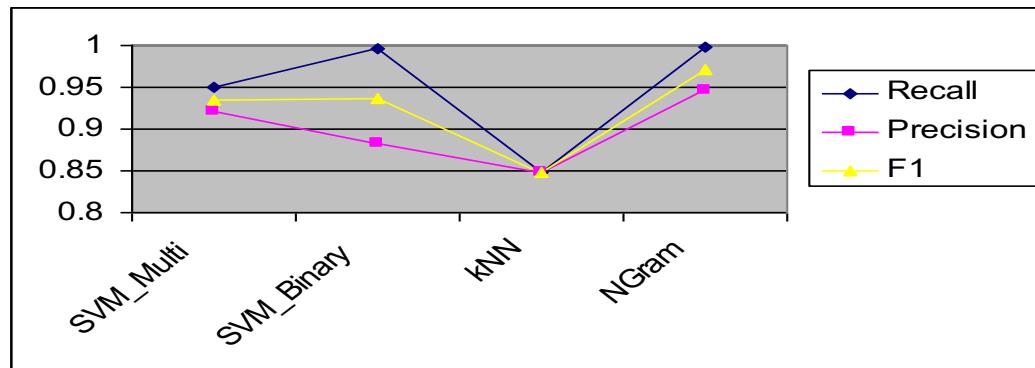
Biểu đồ 7: So sánh kết quả trung bình 10 chủ đề với $N = 4$

5.2.2.3 So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram

❖ Với số đặc trưng chọn lựa là 2500 và $N = 2$

So sánh các mô hình	SVM_Multi	SVM_Binary	kNN	NGram
Recall	0.94908	0.9967949	0.847766	0.99784
Precision	0.9207244	0.8817308	0.847766	0.9455334
F1	0.934687194	0.935738877	0.847766	0.970982774

Bảng 21: Kết quả trung bình 10 chủ đề với 4 mô hình (2500 terms, $N = 2$)

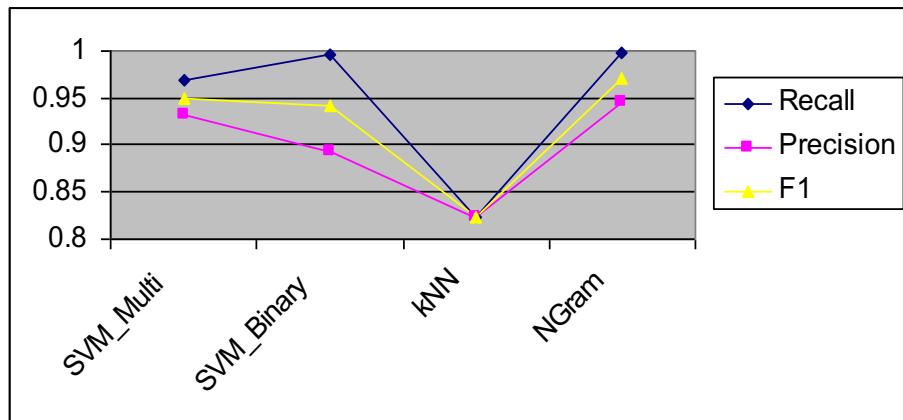


Biểu đồ 8: So sánh kết quả trung bình 10 chủ đề 4 mô hình (2500 terms, $N=2$)

❖ Với số đặc trưng chọn lựa là 5000

So sánh các mô hình	SVM_Multi	SVM_Binary	kNN	NGram
Recall	0.968542	0.996795	0.8235473	0.99784
Precision	0.93113	0.8930464	0.8235473	0.9455334
F1	0.949467605	0.942072902	0.8235473	0.970982774

Bảng 22: Kết quả trung bình 10 chủ đề với 4 mô hình (5000 terms, $N = 2$)

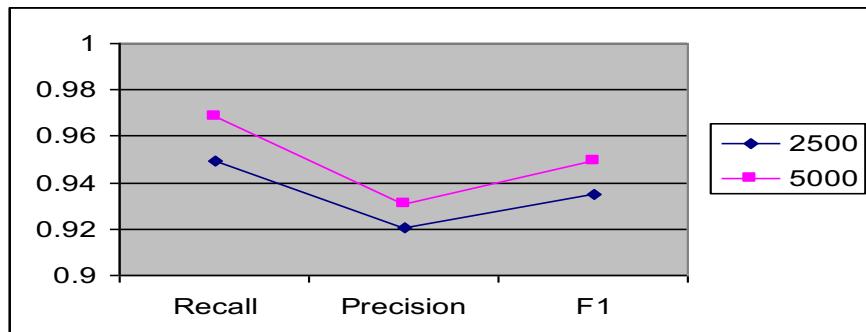


Biểu đồ 9: So sánh kết quả trung bình 10 chủ đề 4 mô hình (5000 terms, N= 2)

5.2.2.4 So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng chọn lựa với mô hình SVM-Multi

Mô hình SVM-Multi		
	2500	5000
Recall	0.94908	0.968542
Precision	0.9207244	0.93113
F1	0.934687194	0.949467605

Bảng 23: Kết quả trung bình 10 chủ đề với số lượng đặc trưng khác nhau

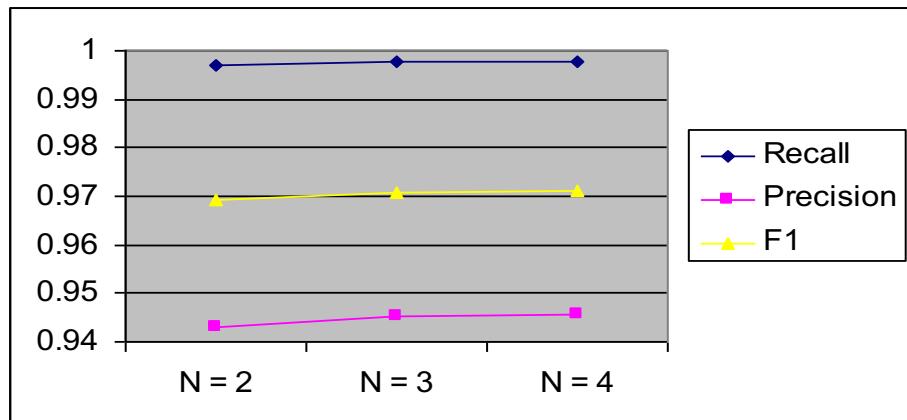


Biểu đồ 10: So sánh kết quả trung bình 10 chủ đề với số lượng đặc trưng khác nhau

5.2.2.5 Mô hình N-gram với phương pháp “làm trơn” (discounting smoothing method) Good Turing

Mô hình N-gram (Good Turing)			
	N = 2	N = 3	N = 4
Recall	0.99702	0.997836	0.99784
Precision	0.942968	0.945167	0.945533
F1	0.969241001	0.97078765	0.970982564

Bảng 24: Kết quả trung bình 10 chủ đề với N khác nhau



Biểu đồ 11: So sánh kết quả trung bình 10 chủ đề với N khác nhau

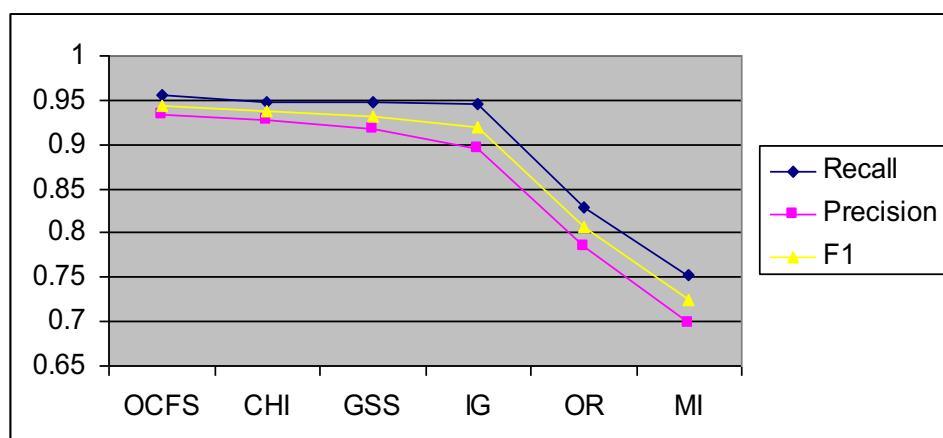
5.2.3 Dữ liệu mìn (27 chủ đề)

5.2.3.1 So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 6 phương pháp chọn đặc trưng OCFS, CHI, GSS, IG, OR, MI

❖ Với số đặc trưng chọn lựa là 2500

Mô hình SVM-Multi						
	OCFS	CHI	GSS	IG	OR	MI
Recall	0.955061	0.948313	0.947617	0.94553	0.8288	0.753113
Precision	0.9344415	0.927776	0.917879	0.894634	0.78426	0.6987457
F1	0.9446388	0.937932	0.932511	0.919378	0.805915	0.7249114

Bảng 25: Kết quả trung bình 27 chủ đề với 2500 terms

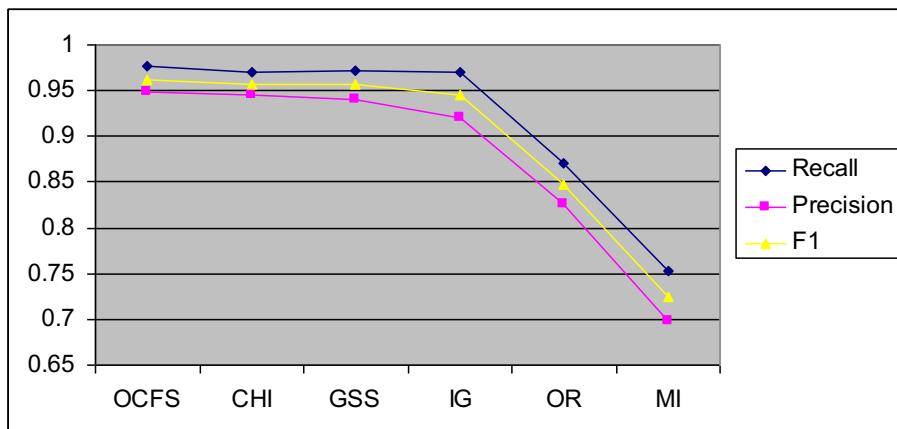


Biểu đồ 12: So sánh kết quả trung bình 27 chủ đề với 2500 terms

❖ Với số đặc trưng chọn lựa là 5000

<i>Mô hình SVM-Multi</i>						
	OCFS	CHI	GSS	IG	OR	MI
Recall	0.976139	0.969608	0.971687	0.970922	0.870539	0.753252
Precision	0.9483923	0.9458282	0.9411046	0.9209292	0.8263579	0.6985996
F1	0.9620656	0.9575705	0.9561513	0.9452651	0.8478733	0.7248971

Bảng 26: Kết quả trung bình 27 chủ đề với 5000 terms

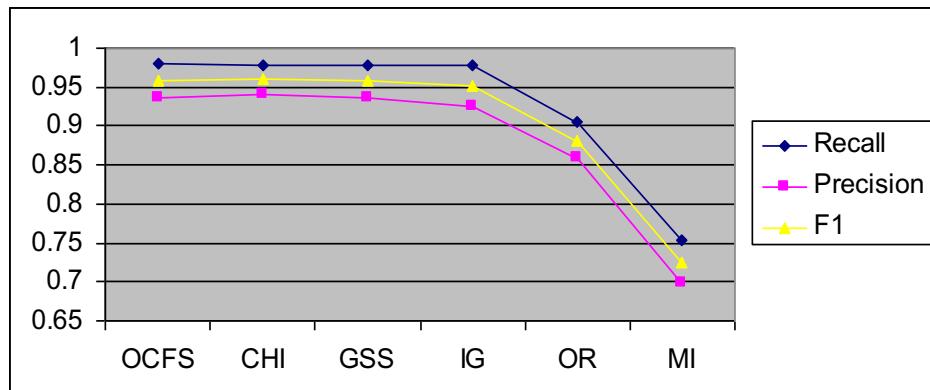


Biểu đồ 13: So sánh kết quả trung bình 27 chủ đề với 5000 terms

❖ Với số đặc trưng chọn lựa là 7500

<i>Mô hình SVM-Multi</i>						
	OCFS	CHI	GSS	IG	OR	MI
Recall	0.980591	0.977878	0.979061	0.97753	0.904835	0.753322
Precision	0.9357295	0.9413957	0.9372536	0.9249439	0.859298	0.6987903
F1	0.9576352	0.9592901	0.9577013	0.9505102	0.8814788	0.7250322

Bảng 27: Kết quả trung bình 27 chủ đề với 7500 terms



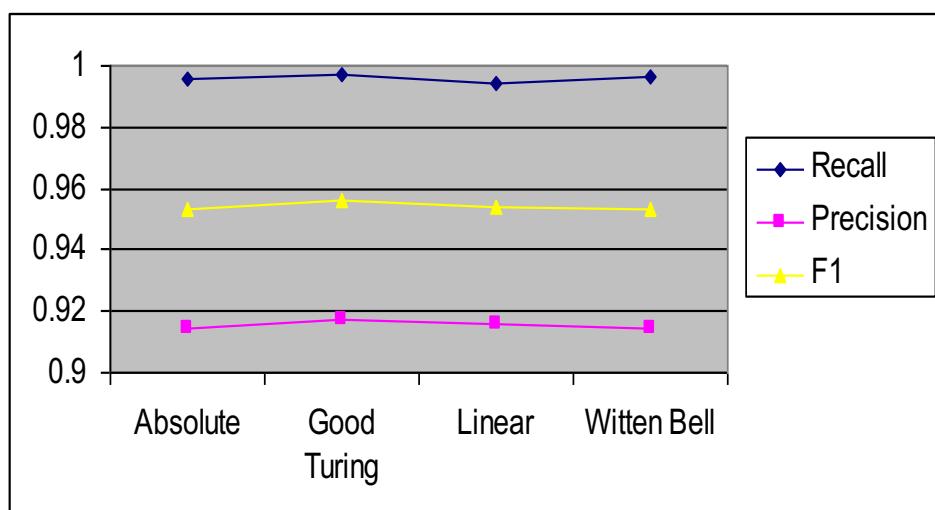
Biểu đồ 14: So sánh kết quả trung bình 27 chủ đề với 7500 terms

5.2.3.2 So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm tròn” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell

❖ Với $N = 2$

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Recall	0.99583	0.99731	0.99446	0.9961
Precision	0.914443	0.917605	0.916064	0.914299
F1	0.953402757	0.955798709	0.953653558	0.95344819

Bảng 28: Kết quả trung bình 27 chủ đề với $N = 2$

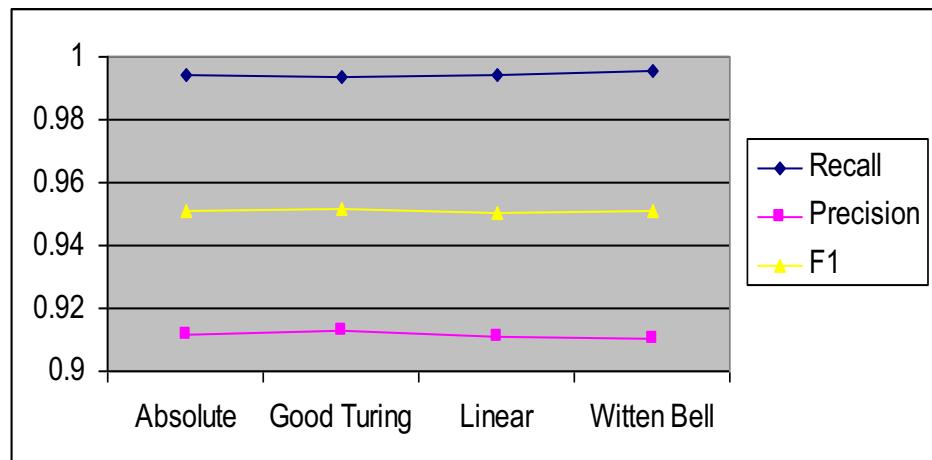


Biểu đồ 15: So sánh kết quả 27 chủ đề với $N = 2$

❖ Với $N = 3$

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Recall	0.99431	0.9936	0.9941	0.9958
Precision	0.911472	0.912966	0.910812	0.91051
F1	0.951090654	0.951577881	0.95063521	0.951247025

Bảng 29: Kết quả trung bình 27 chủ đề với $N = 3$

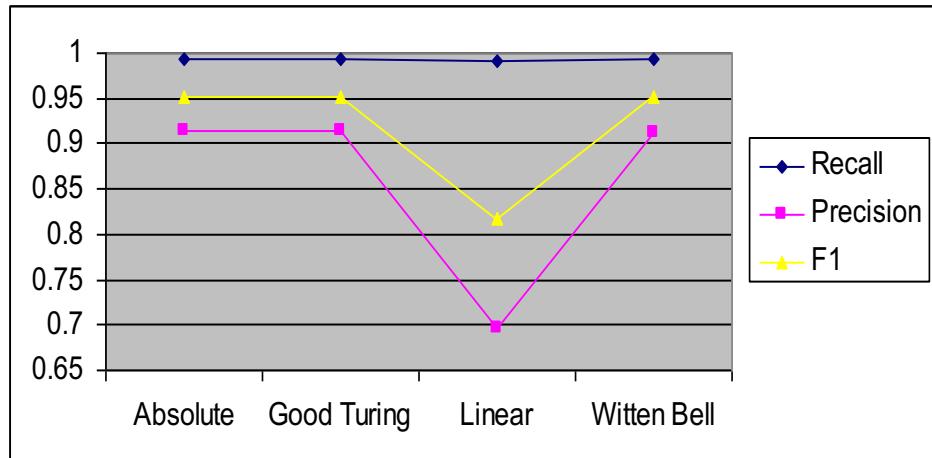


Biểu đồ 16: So sánh kết quả trung bình 27 chủ đề với N = 3

❖ VỚI N = 4

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Recall	0.99324	0.99384	0.99125	0.99267
Precision	0.913342	0.914235	0.695184	0.912467
F1	0.951616881	0.952376937	0.817228709	0.950880296

Bảng 30: Kết quả trung bình 27 chủ đề với N = 4



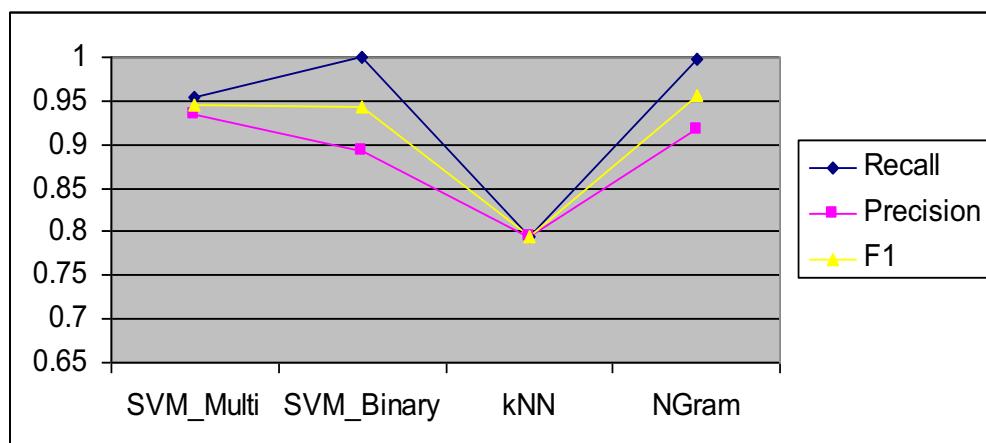
Biểu đồ 17: So sánh kết quả trung bình 27 chủ đề với N = 4

5.2.3.3 So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram

❖ Với số đặc trưng chọn lựa là 2500 và N = 2

So sánh các mô hình	SVM_Multi	SVM_Binary	kNN	NGram
Recall	0.955061	0.999683	0.795181037	0.99731
Precision	0.934441519	0.89379837	0.795181037	0.917605
F1	0.944638753	0.943780119	0.795181037	0.955798709

Bảng 31: Kết quả trung bình 27 chủ đề với 4 mô hình (2500 terms, N = 2)

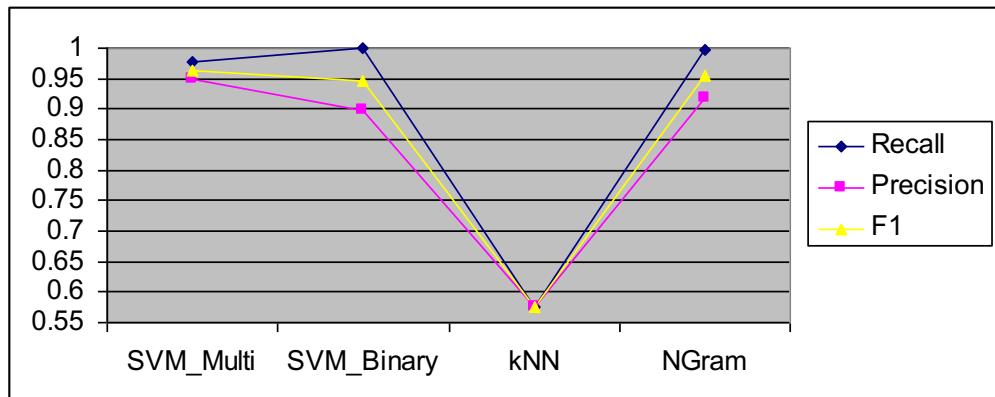


Biểu đồ 18: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (2500 terms, N = 2)

❖ Với số đặc trưng chọn lựa là 5000 và N = 2

So sánh các mô hình	SVM_Multi	SVM_Binary	kNN	NGram
Recall	0.976139	0.999283	0.575839481	0.99731
Precision	0.948392296	0.897244852	0.575839481	0.917605
F1	0.962065631	0.945518967	0.575839481	0.955798709

Bảng 32: Kết quả trung bình 27 chủ đề với 4 mô hình (5000 terms, N = 2)

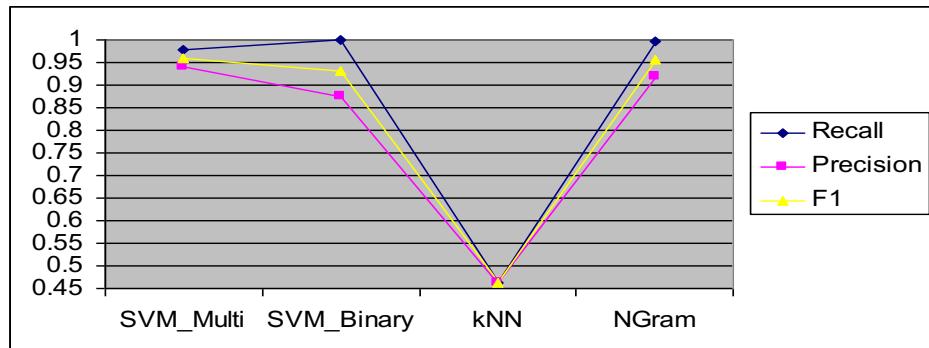


Biểu đồ 19: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (5000 terms, N = 2)

❖ Với số đặc trưng chọn lựa là 7500 và N = 2

So sánh các mô hình				
	SVM_Multi	SVM_Binary	kNN	NGram
Recall	0.977878	0.998957	0.462890481	0.99731
Precision	0.941395741	0.873465074	0.462890481	0.917605
F1	0.959290136	0.932005729	0.462890481	0.955798709

Bảng 33: Kết quả trung bình 27 chủ đề với 4 mô hình (7500 terms, N = 2)

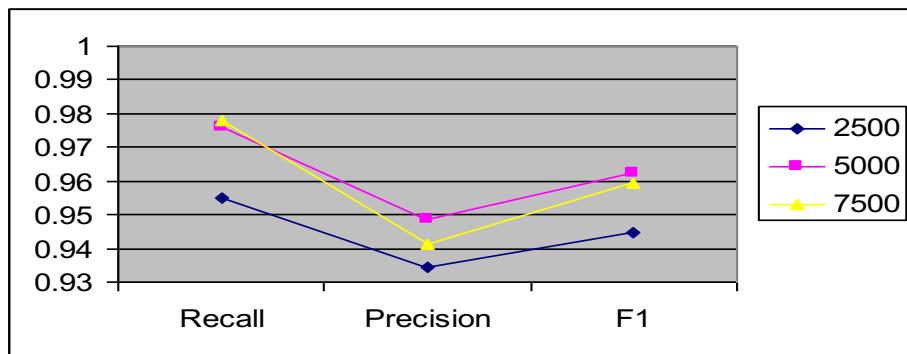


Biểu đồ 20: So sánh kết quả trung bình 27 chủ đề với 4 mô hình (7500 terms, N = 2)

5.2.3.4 So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng chọn lựa với mô hình SVM-Multi

Mô hình SVM-Multi	2500	5000	7500
Recall	0.955061	0.976139	0.977878
Precision	0.934441519	0.948392296	0.941395741
F1	0.944638753	0.962065631	0.959290136

Bảng 34: Kết quả trung bình 27 chủ đề với số lượng đặc trưng khác nhau

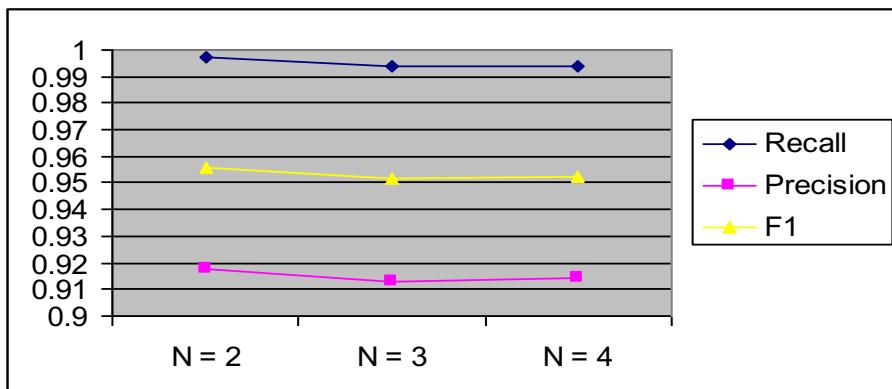


Biểu đồ 21: So sánh kết quả trung bình 27 chủ đề với số đặc trưng khác nhau

5.2.3.5 Mô hình N-gram với phương pháp “làm tròn” (discounting smoothing method) Good Turing

Mô hình N-gram (Good Turing)			
	N = 2	N = 3	N = 4
Recall	0.99731	0.9936	0.99384
Precision	0.917605	0.912966	0.914235
F1	0.955798709	0.951577881	0.952376937

Bảng 35: Kết quả trung bình 27 chủ đề với N khác nhau



Biểu đồ 22: So sánh kết quả trung bình 27 chủ đề với N khác nhau

5.2.3.6 So sánh kết quả kiểm nghiệm giữa hướng tiếp cận của chúng tôi và hướng tiếp cận Naïve Bayes

Trong thí nghiệm này, chúng tôi cho học và kiểm nghiệm trên cùng 1 bộ dữ liệu được cung cấp trong thí nghiệm phân loại văn bản bằng Naïve Bayes (dữ liệu này được mô tả trong 2.2.3.1) và kết quả chỉ so sánh kết quả trên chủ đề cấp 1. Có 2 lý do chúng tôi chỉ so sánh trên chủ đề cấp 1

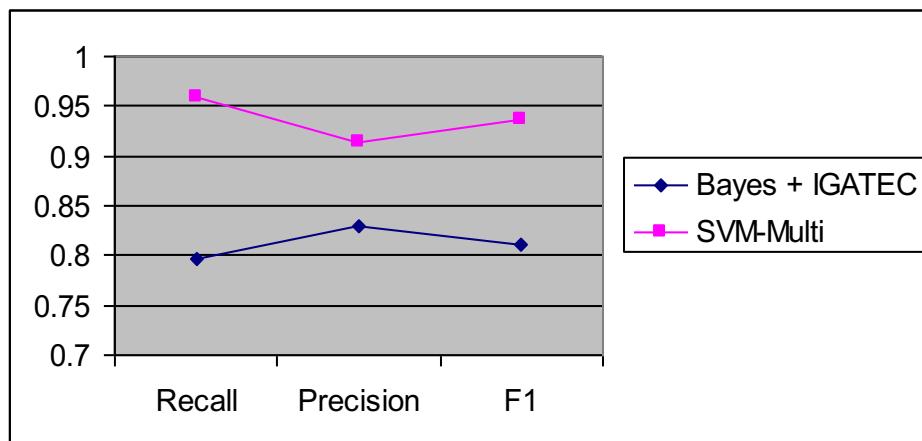
- Với chủ đề cấp 2 chúng tôi hoàn toàn không được cung cấp số liệu kiểm nghiệm của phương pháp Bayes và sự phân chia chủ đề trong tập ngữ liệu ở cấp 2 không hoàn toàn có sự giống nhau.
- Tác giả của chương trình phân loại văn bản bằng phương pháp Naïve Bayes đã đánh giá kết quả thử nghiệm trên chủ đề cấp 1 cao hơn kết quả thử nghiệm trên chủ đề cấp 2

	Bayes + IGATEC	SVM-Multi
Xa hoi	0.784	0.883
Khoa hoc	0.873	0.81
The thao	0.918	0.98
Kinh doanh	0.74	0.98
Trung bình	0.82875	0.91325

Bảng 36: Kết quả chi tiết 4 chủ đề giữa Bayes và SVM

	Bayes + IGATEC	SVM-Multi
Recall	0.7958	0.9587
Precision	0.82875	0.913
F1	0.811940845	0.935292087

Bảng 37: Kết quả trung bình 4 chủ đề giữa Bayes và SVM



Biểu đồ 23: So sánh kết quả trung bình 4 chủ đề giữa Bayes và SVM

5.2.3.7 Nhận xét

Trong hai mức phân loại văn bản, ta thấy rằng kết quả cho được từ việc phân loại trên chủ đề thô thấp hơn so với chủ đề mịn. Kết quả này có thể lý giải như sau:

- Đối với chủ đề mịn, các bài báo được phân chia cụ thể và có tính chính xác cao hơn so với những văn bản nằm trong chủ đề thô có tính tổng quát cao.

- Chủ đề thô bao gồm trong nó nhiều chủ đề mịn, do đó khả năng nhầm lẫn sẽ cao hơn so với chủ đề mịn

Đối với kết quả thử nghiệm trên chủ đề thô ta thấy rằng do tính bao quát của chủ đề thô khá cao nên không thể có kết quả kiểm nghiệm nào đạt đến giá trị tuyệt đối. Trong khi đó, điều này hoàn toàn có thể xảy ra trong chủ đề mịn (ví dụ: cúm gà, tennis). Đây là những chủ đề quá đặc biệt mà các phương pháp phân loại khác nhau đều cho kết quả rất tốt. Tuy vậy, đối với chủ đề mịn vẫn có những chủ đề chưa thực sự phân biệt, tính nhập nhằng vẫn còn cao (ví dụ: cuộc sống đó đây, lối sống, thế giới trẻ, gia đình). Điều này có thể lý giải do tính chất văn bản lấy từ các báo điện tử nên việc phân chia còn phụ thuộc rất lớn vào tính chủ quan của người viết.

Việc so sánh giữa các mô hình tiếp cận phân loại văn bản khác nhau cho ta các kết luận sau:

- **SVM** là phương pháp phân loại tốt nhất hiện nay. Điều này được chứng minh bằng kết quả cho được từ việc phân loại văn bản bằng SVM cao hơn hẳn phương pháp truyền thống được sử dụng trong thí nghiệm là kNN
- **SVM-Binary** cho kết quả thấp hơn **SVM-Multi**. Điều này được giải thích vì lý do: đối với những văn bản có sự nhập nhằng cao, trong quá trình học, **SVM-Multi** đã được tiếp nhận. Trong khi đó với **SVM-Binary**, trong quá trình học, chỉ có hai nhãn là +1 và -1. Do vậy, trong quá trình học, **SVM-Binary** luôn cho kết quả cao khá tuyệt đối. Tuy nhiên khi đi vào quá trình kiểm nghiệm, sự thích nghi của phương pháp **SVM-Multi** sẽ cao hơn hẳn. và **SVM-Binary** sẽ gặp phải hạn chế trong quá trình phân lớp những tài liệu nhập nhằng.
- Trong các phương pháp chọn lựa đặc trưng, phương pháp **OCFS**, **CHI** và **GSS** luôn tỏ ra vượt trội với số lượng đặc trưng chọn thích hợp. Điều này hoàn toàn phù hợp với các kết quả thử nghiệm trên bài toán phân loại văn bản tiếng Anh. Từ kết quả thực nghiệm, chúng tôi nhận thấy rằng, trong các phương pháp chọn lựa đặc trưng, khi số lượng đặc trưng chọn lựa càng lớn, tốc độ của chương trình sẽ càng giảm đồng thời kết quả phân loại càng tốt. Tuy nhiên, khi số lượng đặc trưng vượt quá một giá trị ngưỡng thì tác dụng hoàn toàn ngược. Khi đó số lượng đặc trưng nhiều sẽ tăng cao, điều này dẫn đến kết quả phân loại sẽ giảm.

Chính vì thế, việc chọn số lượng đặc trưng thích hợp rất quan trọng để vừa có thể đảm bảo tính chính xác vừa bảo đảm yêu cầu tốc độ xử lý.

- N-gram là hướng tiếp cận theo mô hình thống kê. Như vậy **N-gram** sẽ không trực tiếp sử dụng kết quả của giai đoạn tách từ như các phương pháp khác. Tuy nhiên kết quả của N-gram lại đạt được xấp xỉ với kết quả phân loại theo hướng truyền thống. Chính kết quả này đã chứng minh được ưu thế của mô hình thống kê và đây cũng mở ra một hướng tiếp cận mới hoàn toàn khả thi.
- Trong mô hình **N-gram**, việc tiếp cận với $N=2$ cho kết quả tốt hơn hẳn các giá trị khác. Điều này có thể lý giải bởi trong tiếng Việt, số lượng từ gồm 2 âm tiết chiếm vai trò và số lượng cao hơn hẳn các từ gồm 1, 3, 4... âm tiết

Cuối cùng, với kết quả đạt được đối với việc phân lớp trên chủ đề mìn, chúng tôi hoàn toàn có thể áp dụng vào dự án **BioCaster**. Đặc biệt với khả năng phân loại chính xác **100%** đối với chủ đề **cúm gà**, việc áp dụng mô hình tìm kiếm văn bản theo chủ đề vào việc lọc văn bản để lấy dữ liệu đầu vào cho hệ thống truy tìm thông tin văn bản là hoàn toàn khả thi. Đây cũng chính là mục đích chúng tôi cần đạt đến khi thực hiện luận văn này.

Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Phân loại văn bản là một bài toán khó. Tuy nhiên, bên cạnh vấn đề khó khăn trong xử lý ngôn ngữ, chúng ta vẫn có thể tìm thấy nhiều điều thú vị. Chính những thú vị này là động lực thôi thúc chúng tôi tham gia nghiên cứu, góp phần giải quyết những vấn đề vướng mắc.

Trong khuôn khổ luận văn này, những vấn đề liên quan đến các bài toán cơ bản như tách từ tiếng Việt, phân loại văn bản tiếng Việt, tìm kiếm văn bản tiếng Việt theo chủ đề đã được chúng tôi tìm hiểu rất công phu cả chiều rộng và chiều sâu vấn đề. Trên những cơ sở nghiên cứu đó, chúng tôi đã cài đặt thành công các phương pháp tách từ và phân loại văn bản trên ngôn ngữ tiếng Việt với độ chính xác rất cao. Đây cũng chính là thành quả rất xứng đáng cho quá trình tìm tòi và thử nghiệm của chúng tôi trong suốt 6 tháng làm việc cật lực. Các kết quả nghiên cứu của chúng tôi đã mở ra nhiều hướng tiếp cận mới để giải quyết cho các bài toán liên quan đến xử lý ngôn ngữ tự nhiên mà kết quả hoàn toàn chấp nhận được. Đặc biệt hơn, thành công của luận văn đã góp phần giải quyết bài toán đầu tiên cho hệ thống **BioCaster**, một hệ thống mang rất nhiều ý nghĩa thiết thực cho cộng đồng.

Tuy nhiên, chúng tôi cũng cần nhấn mạnh rằng, để có thể giải quyết tốt hơn nữa các bài toán liên quan đến xử lý ngôn ngữ tự nhiên trên tiếng Việt, chúng ta cần phải giải quyết tốt các bài toán cơ bản ví dụ như tách từ tiếng Việt. Và điều quan trọng không kém khi sử dụng các phương pháp máy học là chúng ta phải xây dựng được một bộ ngữ liệu hoàn chỉnh và phải được công nhận. Có như thế, chúng ta mới có thể có cơ sở khẳng định cho các kết quả mà chúng ta đạt được ra với thế giới.

6.2 *Hướng phát triển*

Trong bài toán tách từ tiếng Việt, vấn đề tách tên riêng vẫn là một thách thức. Chính vì sự không thống nhất trong cách viết của người Việt nên vấn đề tên riêng càng trở thành một vấn đề thật sự nan giải. Do vậy, để có thể giải quyết tốt bài toán tách từ tiếng Việt, chúng ta phải bắt tay vào giải quyết vấn đề tách tên riêng. Đây chính là hướng phát triển tiếp theo của chúng tôi sau khi bộ công cụ tách từ của chúng tôi đã đạt đến một kết quả rất tốt (**97,72%**).

Trong bài toán phân loại văn bản, vấn đề phát triển tiếp là xây dựng khả năng học tăng cường cho bộ phân loại. Trong thực tế, số chủ đề phân loại rất phong phú, vì vậy khả năng học tăng cường sẽ giải quyết tốt các nhu cầu của mọi người khi áp dụng vào thực tế. Ngoài ra, chúng tôi cũng sẽ đặt vấn đề thử nghiệm áp dụng các phương pháp sử dụng đặc trưng ở cấp cao như ngữ nghĩa (**semantics**) ví dụ **LSI (Latent Semantic Indexing)** và các biến thể của nó, **frame semantics** để nâng cao độ chính xác của bài toán phân loại. Ngoài ra, chúng tôi cũng sẽ tiến hành nghiên cứu cho hệ thống của mình khả năng **học tăng cường** (online learning) thích nghi với mọi điều kiện thay đổi của thực tế.

Mặc dù trên internet có rất nhiều thông tin nhưng thêm vào đó cũng chính là sự đa dạng biểu diễn của tài liệu. Chính vì thế, mục tiêu phát triển tiếp theo của chúng tôi là hoàn thiện hệ thống của mình để có thể truy tìm thông tin online hay nói đúng hơn là một động cơ tìm kiếm cho tiếng Việt với hai tiêu chí: tốc độ nhanh nhất, độ chính xác nhất cao nhất.

TÀI LIỆU THAM KHẢO

- [1]. **Andrei Z. Broder, Marc Najork And Janet L. Wiener.** 2003 Efficient URL Caching for World Wide Web Crawling, *ACM*.
- [2]. **Fabrizio Sebastiani.** Machine Learning in Automated Text Categorization. 2002. *ACM Computing Surveys*, Vol. 34, No. 1, March 2002, pp. 1–47.
- [3]. **Fabrizio SebastianI.** Text Classification for Web Filtering, “Present and Future of Open-Source Content-Based Web filtering”. 2004. *Pisa, IT — 21-22 January, 2004*.
- [4]. **Myers, K., Kearns, M., Singh, S., And Walker, M. A.** 2000. A boosting approach to topic spotting on subdialogues. In *Proceedings of ICML-00, 17th International Conference on Machine Learning (Stanford, CA, 2000)*, 655–662.
- [5]. **Iyer, R. D., Lewis, D. D., Schapire, R. E., Singer, Y., And Singhal, A.** 2000. Boosting for document routing. 2000 . In *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management* (McLean, VA, 2000), 70–77.
- [6]. **Sable, C. L. And Hatzivassiloglou, V.** 2000. Text-based approaches for non-topical image categorization. *Internat. J. Dig. Libr.* 3, 3, 261–275.
- [7]. **Forsyth, R. S.** 1999. New directions in text categorization. In *Causal Models and Intelligent Data Management*, A. Gammerman, ed. Springer, Heidelberg, Germany, 151–185.
- [8]. **Cavnar, W. B. And Trenkle, J. M.** 1994. N-gram based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, 1994), 161–175.
- [9]. **Kessler, B., Nunberg, G., And Schütze, H.** 1997. Automatic detection of text genre. In *Proceedings of ACL-97, 35th Annual Meeting of the Association for Computational Linguistics* (Madrid, Spain, 1997), 32–38.
- [10]. **Larkey, L. S.** 1998. Automatic essay grading using text categorization techniques. In *Proceedings of SIGIR-98, 21st ACM International Conference on*

Research and Development in Information Retrieval (Melbourne, Australia, 1998), 90–95.

- [11]. **Vũ Thụy**. Gán nhãn hình thái từ cho ngữ liệu song ngữ Anh-Việt. *Luận văn cử nhân tin học*. ĐH KHTN, ĐHQG TPHCM 8-2005.
- [12]. **Đinh Điền**. Vấn đề ranh giới từ trong ngữ liệu song ngữ Anh – Việt. *Báo cáo Hội thảo Khoa học “Bảo vệ và Phát triển tiếng Việt”*. Viện ngôn ngữ học, Hội ngôn ngữ học Tp HCM, ĐH KHXN&NV – TPHCM, 12-2002, tr.70 – 78.
- [13]. **Đinh Điền**. Xây dựng và khai thác kho ngữ liệu song ngữ Anh – Việt điện tử. *Luận án Tiến sĩ Ngôn ngữ học so sánh*. ĐH KHXH&NV – TPHCM. 2-2005.
- [14]. **Hoàng Phê**. Từ điển tiếng Việt. *Trung tâm Từ điển học*. NXB Đà Nẵng. 1998.
- [15]. **Chunyu Kit, Haihua Pan, Hongbiao Chen**. Learning Case-based Knowledge for Disambiguating Chinese Word Segmentation: A Preliminary Study. *In COLING2002 workshop: SIGHAN-1*, pp 33 – 39. Taipei. 2002.
- [16]. **Dinh Dien, Hoang Kiem, Nguyen Van Toan**. Vietnamese Word Segmentation. *In Proceedings of NLPPRS'01* (The 6th Natural Language Procesing Pacific Rim Symposium, Tokyo, Japan. 2001, pp 749 – 756. 2001.
- [17]. **Richard Sproat, Chilin Shih**. Corpus – based Methods in Chinese Morphology and Phonology. *Lecture notes for LSA Summer Institute*, Santa Barbara. 2001.
- [18]. **Chooi-Ling Goh, Masayuki Asahara, Yuji Matsumoto. (2004)**. Chinese Word Segmentation by Classification of Characters. *In Proceedings of Third SIGHAN Workshop*.
- [19]. **Chih-Hao Tsai ,” MMSeg: A Word Identification System for Mandarin Chinese Text Based on two Variants of the Maximum Matching Algorithm”** 2000.
- [20]. **Chen, K. J., & Liu, S. H.** (1992). Word identification for Mandarin Chinese sentences. *Proceedings of the Fifteenth International Conference on Computational Linguistics, Nantes: COLING-92*.

- [21]. **Nguyễn Trần Thiên Thanh, Trần Hải Hoàng.** (2005). Tìm hiểu các hướng tiếp cận cho bài toán phân loại văn bản và xây dựng phần mềm phân loại tin tức báo điện tử. *Luận văn cử nhân tin học*. ĐH KHTN, ĐHQG TPHCM 8-2005.
- [22]. **Võ Thị Mỹ Ngọc.** (2002). SVM - Ứng dụng lọc E-mail, *Luận văn thạc sĩ tin học*. ĐH KHTN, ĐHQG TPHCM 8-2002.
- [23]. **Trần Thế Lân.** (2004). Ứng dụng lý thuyết tập thô trong bài toán phân loại văn bản. *Luận văn thạc sĩ tin học*. ĐH KHTN, ĐHQG TPHCM 8-2004.
- [24]. **Tao Liu, Zheng Chen, Benyu Zhang, Wei-ying Ma, Gongti Wu.** (2004). Improving Text Classification using Local Latent Semantic Indexing, Data Mining, 2004. ICDM 2004. *Proceedings, Fourth IEEE International Conference*.
- [25]. **Ciya Liao, Shamim Alpha, Paul Dixon.** Oracle Corporation. (2003). Feature preparation in Text Categorization, *AusDM03 Conference*.
- [26]. **Fuchen Peng, Dale Schuurmans, Shaojun Wang.** (2004). Augmenting Naïve Bayes Classifiers with Statistical Language Models, *Information Retrieval*, 7, 317-345.
- [27]. **Maria Fernanda Caropreso, Stan Matwin, Fabrizio Sebastiani (2001).** A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization, *Text Databases and Document Management: Theory and Practice*, Idea Group Publishing, Hershey, US, pp. 78--102.
- [28]. **Joachims, T.** 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998), 137–142.
- [29]. **Koller, D. And Sahami, M.** 1997. Hierarchically classifying documents using very few words. In *Proceedings of ICML-97, 14th International Conference on Machine Learning* (Nashville, TN, 1997), 170–178.
- [30]. **Larkey, L. S. And Croft, W. B.** 1996. Combining classifiers in text categorization. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval* (Zürich, Switzerland, 1996), 289–297.

- [31]. Lewis, D. D. 1992a. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval* (Copenhagen, Denmark, 1992), 37–50.
- [32]. Lewis, D. D. And Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (Dublin, Ireland, 1994), 3–12.
- [33]. Li, Y. H. And Jain, A. K. 1998. Classification of text documents. *Comput. J.* 41, 8, 537–546.
- [34]. Robertson, S. E. And Harding, P. 1984. Probabilistic automatic indexing by learning from human indexers. *J. Document.* 40, 4, 264–270.
- [35]. Yang, Y. 1994. Expert network: effective and efficient learning from human decisions in text categorisation and retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (Dublin, Ireland, 1994), 13–22
- [36]. Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Inform. Retr.* 1, 1–2, 69–90.
- [37]. Dagan, I., Karov, Y., And Roth, D. 1997. Mistakedriven learning in text categorization. In *Proceedings of EMNLP-97, 2nd Conference on Empirical Methods in Natural Language Processing* (Providence, RI, 1997), 55–63.
- [38]. Ng, H. T., Goh, W. B., And Low, K. L. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval* (Philadelphia, PA, 1997), 67–73.
- [39]. Sch'Utze, H., Hull, D. A., And Pedersen, J. O. 1995. A comparison of classifiers and document representations for the routing problem. In *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval* (Seattle, WA, 1995), 229–237.
- [40]. Wiener, E. D., Pedersen, J. O., And Weigend, A. S. 1995. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual*

- Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, 1995), 317–332.
- [41]. **Lam, S. L. And Lee, D. L.** 1999. Feature reduction for neural network based text categorization. In *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced Systems for Advanced Application* (Hsinchu, Taiwan, 1999), 195–202.
- [42]. **Ruiz, M. E. And Srinivasan, P.** 1999. Hierarchical neural networks for text categorization. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval* (Berkeley, CA, 1999), 281–282.
- [43]. **Weigend, A. S., Wiener, E. D., And Pedersen, J. O.** 1999. Exploiting hierarchy in text catagORIZATION. *Inform. Retr.* 1, 3, 193–216
- [44]. **Yang, Y. And Liu, X.** 1999. A re-examination of text categorization methods. In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval (Berkeley, CA, 1999), 42–49
- [45]. **Fuhr, N. And Buckley, C.** 1991. A probabilistic learning approach for document indexing. *ACM Trans. Inform. Syst.* 9, 3, 223–248.
- [46]. **Cohen, W. W. And Singer, Y.** 1999. Contextsensitive learning methods for text categorization. *ACM Trans. Inform. Syst.* 17, 2, 141–173.
- [47]. **Lewis, D. D. And Ringuette, M.** 1994. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, 1994), 81–93.
- [48]. **Cohen, W. W. And Hirsh, H.** 1998. Joins that generalize: text classification using WHIRL. In *Proceedings of KDD-98, 4th International Conference on Knowledge Discovery and Data Mining* (New York, NY, 1998), 169–173.
- [49]. **Lewis, D. D. And Catlett, J.** 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of ICML-94, 11th International Conference on Machine Learning* (New Brunswick, NJ, 1994), 148–156.

- [50]. Li, Y. H. And Jain, A. K. 1998. Classification of text documents. *Comput. J.* 41, 8, 537–546.
- [51]. Schapire, R. E. And Singer, Y. 2000. BoosTexter: a boosting-based system for text categorization. *Mach. Learn.* 39, 2/3, 135–168.
- [52]. Weiss, S. M., Apt'E, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T., And Hampp, T. 1999. Maximizing text-mining performance. *IEEE Intell. Syst.* 14, 4, 63–69.
- [53]. Mitchell, T.M. 1996. *Machine Learning*. McGraw Hill, New York, NY
- [54]. Fuhr, N. And Pfeifer, U. 1994. Probabilistic information retrieval as combination of abstraction inductive learning and probabilistic assumptions. *ACM Trans. Inform. Syst.* 12, 1, 92–115.
- [55]. Ittner, D. J., Lewis, D. D., And Ahn, D. D. 1995. Text categorization of low quality images. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, 1995), 301–315.
- [56]. Yang, Y. And Chute, C. G. 1994. An example-based mapping method for text categorization and retrieval. *ACMTrans. Inform. Syst.* 12, 3, 252–277.
- [57]. Domingos P and Pazzani M (1997). Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Machine Learning*, 29: 103-130.
- [58]. Friedman N, Geiger D and Goldszmidt M (1997). Bayesian network classifiers. *Machine Learning*.
- [59]. Rish (2001). An empirical study of the naïve bayes classifier. In: *proceedings of IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*.
- [60]. YAMCHA: <http://chazen.org/~taku/software/yamcha/> (17/3/2006)
- [61]. Chooi-Ling Goh, Masayuki Asahara, Yuji Matsumoto. 2003 . Pruning False Unknown Words to Improve Chinese Word Segmentation *In Proceedings of PACLIC 18*, pp. 139-149.
- [62]. Jun Yan-Ning Liu-Benyu Zhang-Shuicheng Yan. (2005) OCFS: Optimal Orthogonal Centroid Feature Selection for Text Categorization, *Microsoft Research Asia, China*.

- [63]. **Yang, Y. and Pedersen ,J,O.** , A comparative Study On Feature Selection in Text Categorization . *In Proceedings of the 14th International Conference on Machine Learning(ICML), (1997)*, 412-420.
- [64]. **Howland, P. and Park, H.** Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (8)*. 995-10006.
- [65]. **M. Jeon, Park, H. and Rosen, J.B.** Dimension Reduction Based on Centroids and Least Squares for Efficient Processing of Text Data, Minneapolis, MN, University of Minnesota.
- [66]. **James E. Gentle, J. Chambers, W. Eddy, W. Haerdle, S. Sheather and Tierney, L.** *Numerical Linear Algebra for Applications in Statistics*. Springer-Verlag, Berlin, 1998.
- [67]. vnQTAG: <http://www.loria.fr/equipes/led/download/source/vnqtag.zip>
(9/3/2006)
- [68]. **Richard Sproat, Thomas Emerson**, “ The First International Chinese Word Segmentation Bakeoff”, *In Proceedings of the second SIGHAN Workshop on Chinese Language Processing, ACL2003* . 2003.
- [69]. **H. Nguyen, H. Nguyen, T. Vu, N. Tran, K. Hoang**. 2005. Internet and Genetics Algorithm-based Text categorization for Documents in Vietnamese. Research, *Innovation and Vision of the Future, the 3rd International Conference in Computer Science*, (RIVF 2005), Can Tho, Vietnam.
- [70]. ⁸**Hoang Cong Duy Vu**, Nguyen Le Nguyen, Dinh Dien, Ngo Quoc Hung “A Vietnamese Word Segmentation Approach using Maximum Matching Algorithms and Support Vector Machines”, *In the 2006 National Conference in Information Technology*, Da Lat, Vietnam, June 2006.
- [71]. ⁹**DIEN Dinh, Nigel Collier, VU Hoang C. D., NGUYEN Nguyen L., HUNG Ngo Q.** "Topic-based Vietnamese Document Filtering in The BioCaster Project". 2006. *In the International Journal of Medical Informatics*, Elsevier, 2006.

⁸ Đây là bài báo chúng tôi vừa báo cáo tại Hội thảo Quốc Gia về CNTT vào 13/6/2006 tại ĐH Đà Lạt

⁹ Đây là bài báo đã submit tại tạp chí quốc tế:

http://www.elsevier.com/wps/find/journaldescription.cws_home/506040/description?navopenmenu=2

TÀI LIỆU THAM KHẢO

Phụ lục 1.Bảng kết quả thử nghiệm trên dữ liệu thô 10 chủ đề

So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 3 phương pháp chọn đặc trưng OCFS, CHI, GSS

Mô hình SVM-Multi			
	OCFS	CHI	GSS
Kinh doanh	0.930061	0.928165	0.925512
Vì tinh	0.966667	0.971491	0.936623
The thao	0.985751	0.986801	0.979901
The gioi	0.948928	0.942674	0.935527
Suc khoe	0.966402	0.969356	0.964925
Phap luat	0.941394	0.941394	0.936906
Doi song	0.79666	0.772593	0.77554
Khoa hoc	0.844943	0.807729	0.828721
Van hoa	0.94752	0.94624	0.94336
Chinh tri Xa hoi	0.878918	0.86175	0.868755
Trung bình	0.9207244	0.9128193	0.909577

Bảng 38: Kết quả chi tiết 10 chủ đề với 2500 terms

Mô hình SVM-Multi			
	OCFS	CHI	GSS
Kinh doanh	0.928355	0.927597	0.924978
Vì tinh	0.973026	0.973246	0.948632
The thao	0.988001	0.987101	0.981211
The gioi	0.955926	0.953544	0.942137
Suc khoe	0.970648	0.968987	0.967254
Phap luat	0.943506	0.944034	0.939066
Doi song	0.811395	0.816306	0.812538
Khoa hoc	0.866412	0.855439	0.860472
Van hoa	0.96112	0.96032	0.95963
Chinh tri Xa hoi	0.912911	0.903528	0.907582
Trung bình	0.93113	0.9290102	0.92435

Bảng 39: Kết quả chi tiết 10 chủ đề với 5000 terms

So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm tròn” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Kinh doanh	0.935557	0.941812	0.938969	0.942191
Vi tinh	0.956579	0.95636	0.956798	0.954825
The thao	0.978551	0.977951	0.979751	0.979451
The gioi	0.974389	0.974241	0.974836	0.974985
Suc khoe	0.958095	0.956618	0.954587	0.959941
Phap luat	0.972281	0.973073	0.972545	0.973601
Doi song	0.89833	0.896857	0.885069	0.895874
Khoa hoc	0.855916	0.861164	0.867844	0.858779
Van hoa	0.96032	0.96032	0.9584	0.9592
Chinh tri xa hoi	0.932866	0.931281	0.929827	0.927448
Trung binh	0.9422884	0.9429677	0.9418626	0.9426295

Bảng 40: Kết quả chi tiết 10 chủ đề với N = 2

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Kinh doanh	0.941433	0.945792	0.941054	0.946171
Vi tinh	0.952851	0.953947	0.953289	0.950877
The thao	0.983351	0.983801	0.984401	0.985301
The gioi	0.974389	0.974389	0.974389	0.973794
Suc khoe	0.961972	0.961233	0.960679	0.96271
Phap luat	0.974657	0.974921	0.973073	0.975977
Doi song	0.894892	0.896857	0.884086	0.892436
Khoa hoc	0.856393	0.864504	0.861641	0.859256
Van hoa	0.96304	0.96336	0.96048	0.96192
Chinh tri xa hoi	0.93432	0.932866	0.931148	0.933924
Trung binh	0.9437298	0.945167	0.942424	0.9442366

Bảng 41: Kết quả chi tiết 10 chủ đề với N = 3

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Kinh doanh	0.940675	0.945224	0.9384	0.946171
Vi tinh	0.954386	0.954386	0.949781	0.952193
The thao	0.983951	0.982751	0.985001	0.984101
The gioi	0.974538	0.974241	0.973198	0.974241
Suc khoe	0.962895	0.961049	0.95791	0.96271
Phap luat	0.974657	0.974921	0.973337	0.975185
Doi song	0.891945	0.89833	0.888016	0.885069
Khoa hoc	0.857824	0.865458	0.844466	0.859733
Van hoa	0.96288	0.9632	0.9568	0.96112
Chinh tri xa hoi	0.937095	0.935774	0.933527	0.938417
Trung binh	0.9440846	0.9455334	0.9400436	0.943894

Bảng 42: Kết quả chi tiết 10 chủ đề với $N = 4$

So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram

So sánh các mô hình				
	SVM Multi	SVM Binary	kNN	NGram
Kinh doanh	0.930061	0.906937	0.777104	0.945224
Vi tinh	0.966667	0.949561	0.95614	0.954386
The thao	0.985751	0.986201	0.979601	0.982751
The gioi	0.948928	0.940143	0.871799	0.974241
Suc khoe	0.966402	0.936865	0.939635	0.961049
Phap luat	0.941394	0.93849	0.767951	0.974921
Doi song	0.79666	0.581532	0.734283	0.89833
Khoa hoc	0.844943	0.825859	0.62834	0.865458
Van hoa	0.94752	0.93808	0.94176	0.9632
Chinh tri Xa hoi	0.878918	0.81364	0.881047	0.935774
Trung binh	0.9207244	0.8817308	0.847766	0.9455334

Bảng 43: Kết quả chi tiết 10 chủ đề với 4 mô hình (2500 terms, $N = 2$)

So sánh các mô hình				
	SVM Multi	SVM Binary	kNN	NGram
Kinh doanh	0.928355	0.904231	0.75214	0.945224
Vi tinh	0.973026	0.95692	0.941682	0.954386
The thao	0.988001	0.989451	0.969701	0.982751
The gioi	0.955926	0.948141	0.867989	0.974241
Suc khoe	0.970648	0.940111	0.919635	0.961049
Phap luat	0.943506	0.941602	0.725651	0.974921
Doi song	0.811395	0.606267	0.712834	0.89833
Khoa hoc	0.866412	0.847328	0.618234	0.865458
Van hoa	0.96112	0.94968	0.91736	0.9632
Chinh tri Xa hoi	0.912911	0.846733	0.810247	0.935774
Trung bình	0.93113	0.8930464	0.8235473	0.9455334

Bảng 44: Kết quả chi tiết 10 chủ đề với 4 mô hình (5000 terms, N = 2)

So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng
chọn lựa với mô hình SVM-Multi

Mô hình SVM-Multi		
	2500	5000
Kinh doanh	0.930061	0.928355
Vi tinh	0.966667	0.973026
The thao	0.985751	0.988001
The gioi	0.948928	0.955926
Suc khoe	0.966402	0.970648
Phap luat	0.941394	0.943506
Doi song	0.79666	0.811395
Khoa hoc	0.844943	0.866412
Van hoa	0.94752	0.96112
Chinh tri Xa hoi	0.878918	0.912911
Trung bình	0.9207244	0.93113

Bảng 45: Kết quả chi tiết 10 chủ đề với số lượng đặc trưng khác nhau

Mô hình N-gram với phương pháp “làm tròn” (discounting smoothing method) Good Turing

<i>Mô hình N-gram (Good Turing)</i>	$N = 2$	$N = 3$	$N = 4$
Kinh doanh	0.941812	0.945792	0.945224
Vì tinh	0.95636	0.953947	0.954386
The thao	0.977951	0.983801	0.982751
The gioi	0.974241	0.974389	0.974241
Suc khoe	0.956618	0.961233	0.961049
Phap luat	0.973073	0.974921	0.974921
Doi song	0.896857	0.896857	0.89833
Khoa hoc	0.861164	0.864504	0.865458
Van hoa	0.96032	0.96336	0.9632
Chinh tri xa hoi	0.931281	0.932866	0.935774
Trung binh	0.9429677	0.945167	0.9455334

Bảng 46: Kết quả chi tiết 10 chủ đề với N khác nhau

Phụ lục 2. Bảng kết quả thử nghiệm trên dữ liệu mìn 27 chủ đề

So sánh kết quả phân loại văn bản bằng mô hình SVM-Multi theo 6 phương pháp chọn đặc trưng OCFS, CHI, GSS, IG, OR, MI

Mô hình SVM-Multi						
	OCFS	CHI	GSS	IG	OR	MI
Am nhac	0.96064	0.95572	0.95941	0.943419	0.664207	0.642066
Cum ga	1	1	0.997375	0.96063	0.968504	0.834646
Giao duc	0.958982	0.951841	0.951841	0.896601	0.851275	0.66289
Du lich	0.953982	0.953982	0.952212	0.911504	0.748673	0.670796
Du hoc	0.979695	0.982234	0.964467	0.961929	0.903553	0.949239
Loi song	0.78972	0.719626	0.742991	0.794393	0.168224	0.471963
Giai tri tin hoc	0.84017	0.818953	0.820368	0.727016	0.408769	0.302687
San pham tin hoc moi	0.961345	0.963025	0.961345	0.919328	0.82521	0.746219
Hackers va Virus	0.974922	0.978056	0.971787	0.940439	0.946708	0.852665
Cuoc song do day	0.814815	0.750617	0.785185	0.8	0.407407	0.402469
San khau dien anh	0.905825	0.883495	0.914563	0.883495	0.713592	0.440777
My thuat	0.965278	0.979167	0.944444	0.881944	0.9375	0.590278
Thoi trang	0.903974	0.89404	0.811258	0.887417	0.887417	0.493377
Bong da	0.997268	0.995902	0.991803	0.967213	0.900273	0.922131
Tennis	1	1	0.992933	0.95053	0.978799	0.915194
Lam dep	0.964626	0.955102	0.97415	0.960544	0.931973	0.921088
Gioi tinh	0.981343	0.977612	0.977612	0.910448	0.876866	0.600746
Hinh su	0.943878	0.954082	0.913265	0.892857	0.913265	0.591837
Kinh doanh quoc te	0.942755	0.928444	0.921288	0.906977	0.697674	0.738819
Duong vao WTO	0.963351	0.973822	0.910995	0.890052	0.95288	0.832461
Chung khoan	0.975	0.978125	0.9625	0.95625	0.921875	0.875
Bat dong san	0.985816	0.985816	0.971631	0.978723	0.932624	0.957447
The gioi tre	0.805263	0.797368	0.784211	0.723684	0.613158	0.392105
Khong gian song	0.931035	0.948276	0.913793	0.931035	0.844828	0.844828
Gia dinh	0.95	0.925	0.939286	0.853571	0.707143	0.832143
Am thuc	0.935	0.9425	0.9425	0.9275	0.925	0.7275
Mua sam	0.845238	0.857143	0.809524	0.797619	0.547619	0.654762
Trung bình	0.93444152	0.92777585	0.91787915	0.894634	0.78425985	0.69874567

Bảng 47: Kết quả chi tiết 27 chủ đề với 2500 terms

<i>Mô hình SVM-Multi</i>						
	OCFS	CHI	GSS	IG	OR	MI
Am nhac	0.97294	0.96925	0.96433	0.96802	0.713407	0.640836
Cum ga	0.994751	0.997375	0.994751	0.992126	0.965879	0.834646
Giao duc	0.964589	0.967422	0.963173	0.949009	0.869688	0.664306
Du lich	0.971681	0.966372	0.961062	0.959292	0.840708	0.670796
Du hoc	0.974619	0.979695	0.956853	0.977157	0.939086	0.949239
Loi song	0.906542	0.836449	0.873832	0.859813	0.336449	0.471963
Giai tri tin hoc	0.930693	0.910891	0.905233	0.838755	0.473833	0.302687
San pham tin hoc moi	0.941176	0.946218	0.951261	0.932773	0.880672	0.746219
Hackers va Virus	0.952978	0.959248	0.965517	0.924765	0.974922	0.852665
Cuoc song do day	0.901235	0.876543	0.879012	0.869136	0.496296	0.402469
San khau dien anh	0.93301	0.921359	0.930097	0.920388	0.728155	0.441748
My thuat	0.972222	0.972222	0.965278	0.930556	0.916667	0.590278
Thoi trang	0.923841	0.927152	0.854305	0.933775	0.887417	0.493377
Bong da	0.999317	0.998634	0.996585	0.991803	0.95082	0.922131
Tennis	1	1	0.982332	0.971731	0.971731	0.915194
Lam dep	0.971429	0.967347	0.967347	0.957823	0.955102	0.921088
Gioi tinh	0.981343	0.981343	0.973881	0.958955	0.891791	0.600746
Hinh su	0.938776	0.943878	0.94898	0.933673	0.928571	0.586735
Kinh doanh quoc te	0.969589	0.958855	0.958855	0.948122	0.744186	0.738819
Duong vao WTO	0.95288	0.968586	0.958115	0.858639	0.95288	0.832461
Chung khoan	0.971875	0.971875	0.96875	0.971875	0.9625	0.875
Bat dong san	0.98227	0.98227	0.98227	0.975177	0.932624	0.957447
The gioi tre	0.863158	0.831579	0.847368	0.794737	0.605263	0.392105
Khong gian song	0.931035	0.948276	0.965517	0.931035	0.965517	0.844828
Gia dinh	0.864286	0.928571	0.878571	0.771429	0.796429	0.832143
Am thuc	0.9475	0.945	0.9475	0.935	0.9525	0.7275
Mua sam	0.892857	0.880952	0.869048	0.809524	0.678571	0.654762
Trung bình	0.9483923	0.94582822	0.94110456	0.92092919	0.82635793	0.69859956

Bảng 48: Kết quả chi tiết 27 chủ đề với 5000 terms

<i>Mô hình SVM-Multi</i>						
	OCFS	CHI	GSS	IG	OR	MI
Am nhac	0.96679	0.96925	0.96187	0.97171	0.805658	0.640836
Cum ga	0.994751	0.994751	0.994751	0.992126	0.973753	0.834646
Giao duc	0.96034	0.966006	0.963173	0.961756	0.889518	0.664306
Du lich	0.969912	0.966372	0.964602	0.961062	0.886726	0.670796
Du hoc	0.961929	0.956853	0.967005	0.969543	0.954315	0.949239
Loi song	0.873832	0.859813	0.883178	0.864486	0.490654	0.471963
Giai tri tin hoc	0.92645	0.933522	0.923621	0.871287	0.601132	0.304102
San pham tin hoc moi	0.927731	0.92437	0.929412	0.941176	0.902521	0.744538
Hackers va Virus	0.943574	0.946708	0.949843	0.918495	0.959248	0.852665
Cuoc song do day	0.906173	0.898765	0.898765	0.896296	0.595062	0.404938
San khau dien anh	0.934951	0.927184	0.938835	0.929126	0.769903	0.44466
My thuat	0.951389	0.972222	0.958333	0.9375	0.965278	0.590278
Thoi trang	0.930464	0.92053	0.86755	0.923841	0.940397	0.493377
Bong da	0.999317	0.999317	0.997951	0.995219	0.959016	0.921448
Tennis	1	1	0.985866	0.971731	0.971731	0.915194
Lam dep	0.967347	0.965986	0.971429	0.967347	0.964626	0.921088
Gioi tinh	0.962687	0.970149	0.970149	0.962687	0.902985	0.600746
Hinh su	0.933673	0.933673	0.938776	0.938776	0.943878	0.586735
Kinh doanh quoc te	0.964222	0.9678	0.964222	0.949911	0.78712	0.740608
Duong vao WTO	0.926702	0.95288	0.942408	0.853403	0.937173	0.832461
Chung khoan	0.96875	0.971875	0.971875	0.9625	0.978125	0.875
Bat dong san	0.971631	0.978723	0.975177	0.975177	0.953901	0.957447
The gioi tre	0.865789	0.85	0.860526	0.852632	0.660526	0.392105
Khong gian song	0.913793	0.913793	0.913793	0.913793	0.948276	0.844828
Gia dinh	0.778571	0.875	0.825	0.728571	0.842857	0.828571
Am thuc	0.9425	0.945	0.9425	0.93	0.95	0.73
Mua sam	0.821429	0.857143	0.845238	0.833333	0.666667	0.654762
Trung bình	0.9357295	0.9413957	0.9372536	0.9249439	0.859298	0.6987903

Bảng 49: Kết quả chi tiết 27 chủ đề với 7500 terms

So sánh kết quả phân loại văn bản bằng mô hình N-gram theo 4 phương pháp “làm trơn” (discounting smoothing methods) Absolute, Good Turing, Linear, Witten Bell

Mô hình N-gram	Absolute	Good Turing	Linear	Witten Bell
Am Nhac	0.851169	0.851169	0.856089	0.858549
Cum ga	0.973753	0.973753	0.984252	0.973753
Giao duc	0.881188	0.879774	0.876945	0.879774
Du lich	0.976991	0.973451	0.973451	0.969912
Du hoc	0.959391	0.956853	0.959391	0.961929
Loi song	0.67757	0.71028	0.668224	0.640187
Giai tri tin hoc	0.909477	0.90099	0.892504	0.889675
San pham tin hoc moi	0.737815	0.773109	0.778151	0.781513
Hackers va Virus	0.978056	0.981191	0.981191	0.981191
Cuoc song do day	0.881481	0.876543	0.88642	0.876543
San khau dien anh	0.987379	0.985437	0.974757	0.980583
My thuat	0.833333	0.840278	0.833333	0.840278
Thoi trang	0.917219	0.910596	0.897351	0.913907
Bong da	0.989754	0.988388	0.990437	0.991803
Tennis	0.992933	0.992933	0.992933	0.992933
Lam dep	0.964626	0.955102	0.95102	0.964626
Gioi tinh	0.932836	0.932836	0.929105	0.940298
Hinh su	0.938776	0.933673	0.938776	0.923469
Kinh doanh quoc te	0.883721	0.889088	0.887299	0.899821
Duong vao WTO	0.968586	0.968586	0.973822	0.973822
Chung khoan	0.965625	0.971875	0.965625	0.971875
Bat dong san	0.985816	0.989362	0.98227	0.978723
The gioi tre	0.95	0.95	0.944737	0.947368
Khong gian song	0.931035	0.948276	0.965517	0.931035
Gia dinh	0.814286	0.832143	0.828571	0.817857
Am thuc	0.95	0.9525	0.9525	0.9475
Mua sam	0.857143	0.857143	0.869048	0.857143
Trung binh	0.914442926	0.917604778	0.916063667	0.914298778

Bảng 50: Kết quả chi tiết 27 chủ đề với N =2

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Am Nhac	0.851169	0.854859	0.851169	0.869619
Cum ga	0.981627	0.981627	0.981627	0.984252
Giao duc	0.899576	0.898161	0.886846	0.905233
Du lich	0.976991	0.975221	0.957522	0.966372
Du hoc	0.954315	0.951777	0.959391	0.969543
Loi song	0.658879	0.672897	0.626168	0.616822
Giai tri tin hoc	0.91372	0.903819	0.896747	0.908062
San pham tin hoc moi	0.732773	0.756303	0.776471	0.766387
Hackers va Virus	0.971787	0.971787	0.971787	0.971787
Cuoc song do day	0.871605	0.874074	0.871605	0.859259
San khau dien anh	0.990291	0.988349	0.979612	0.986408
My thuat	0.854167	0.854167	0.840278	0.833333
Thoi trang	0.903974	0.890728	0.884106	0.890728
Bong da	0.991803	0.991803	0.990437	0.992486
Tennis	0.992933	0.992933	0.992933	0.996466
Lam dep	0.967347	0.957823	0.95102	0.965986
Gioi tinh	0.929105	0.929105	0.940298	0.936567
Hinh su	0.933673	0.933673	0.928571	0.923469
Kinh doanh quoc te	0.876565	0.86941	0.876565	0.880143
Duong vao WTO	0.963351	0.963351	0.973822	0.968586
Chung khoan	0.975	0.96875	0.975	0.971875
Bat dong san	0.98227	0.98227	0.98227	0.98227
The gioi tre	0.939474	0.942105	0.947368	0.936842
Khong gian song	0.931035	0.931035	0.931035	0.931035
Gia dinh	0.778571	0.8	0.807143	0.775
Am thuc	0.9425	0.945	0.955	0.95
Mua sam	0.845238	0.869048	0.857143	0.845238
Trung binh	0.911471815	0.912965741	0.91081237	0.910509926

Bảng 51: Kết quả chi tiết 27 chủ đề với N = 3

<i>Mô hình N-gram</i>				
	Absolute	Good Turing	Linear	Witten Bell
Am Nhac	0.851169	0.852399	0.864699	0.875769
Cum ga	0.981627	0.981627	0.984252	0.984252
Giao duc	0.893918	0.898161	0.892504	0.899576
Du lich	0.978761	0.971681	0.955752	0.971681
Du hoc	0.959391	0.954315	0.964467	0.969543
Loi song	0.658879	0.668224	0.014019	0.626168
Giai tri tin hoc	0.91372	0.902405	0.881188	0.909477
San pham tin hoc moi	0.744538	0.761345	0.791597	0.776471
Hackers va Virus	0.971787	0.968652	0.971787	0.971787
Cuoc song do day	0.876543	0.879012	0.85679	0.871605
San khau dien anh	0.986408	0.985437	0.984466	0.988349
My thuat	0.854167	0.847222	0	0.840278
Thoi trang	0.89404	0.89404	0.890728	0.89404
Bong da	0.991803	0.991803	0.99112	0.993169
Tennis	0.992933	0.992933	0.992933	0.996466
Lam dep	0.964626	0.956463	0.970068	0.964626
Gioi tinh	0.932836	0.932836	0.970149	0.936567
Hinh su	0.933673	0.933673	0.005102	0.923469
Kinh doanh quoc te	0.876565	0.88551	0.88551	0.880143
Duong vao WTO	0.963351	0.963351	0.968586	0.968586
Chung khoan	0.975	0.971875	0.975	0.971875
Bat dong san	0.98227	0.98227	0.971631	0.98227
The gioi tre	0.95	0.944737	0.952632	0.923684
Khong gian song	0.931035	0.931035	0	0.931035
Gia dinh	0.782143	0.814286	0	0.778571
Am thuc	0.95	0.95	0.035	0.95
Mua sam	0.869048	0.869048	0	0.857143
Trung binh	0.913341889	0.914234815	0.695184444	0.912466667

Bảng 52: Kết quả chi tiết 27 chủ đề với N = 4

So sánh kết quả phân loại văn bản với 4 mô hình khác nhau: SVM-Multi, SVM-Binary, kNN, N-gram

So sánh các mô hình	SVM_Multi	SVM_Binary	kNN	NGram
Am nhac	0.96064	0.95572	0.920049	0.851169
Cum ga	1	0.994751	0.958005	0.973753
Giao duc	0.958982	0.915014	0.804533	0.879774
Du lich	0.953982	0.918584	0.943363	0.973451
Du hoc	0.979695	0.961929	0.939086	0.956853
Loi song	0.78972	0.649533	0.228972	0.71028
Giai tri tin hoc	0.84017	0.763791	0.683168	0.90099
San pham tin hoc moi	0.961345	0.919328	0.919328	0.773109
Hackers va Virus	0.974922	0.918495	0.874608	0.981191
Cuoc song do day	0.814815	0.666667	0.679012	0.876543
San khau dien anh	0.905825	0.92233	0.965049	0.985437
My thuat	0.965278	0.951389	0.847222	0.840278
Thoi trang	0.903974	0.89404	0.850993	0.910596
Bong da	0.997268	0.996585	0.992486	0.988388
Tennis	1	1	0.943463	0.992933
Lam dep	0.964626	0.946939	0.904762	0.955102
Gioi tinh	0.981343	0.902985	0.839552	0.932836
Hinh su	0.943878	0.954082	0.846939	0.933673
Kinh doanh quoc te	0.942755	0.878354	0.772809	0.889088
Duong vao WTO	0.963351	0.931937	0.827225	0.968586
Chung khoan	0.975	0.96875	0.925	0.971875
Bat dong san	0.985816	0.98227	0.868794	0.989362
The gioi tre	0.805263	0.731579	0.471053	0.95
Khong gian song	0.931035	0.896552	0.689655	0.948276
Gia dinh	0.95	0.735714	0.632143	0.832143
Am thuc	0.935	0.93	0.845	0.9525
Mua sam	0.845238	0.845238	0.297619	0.857143
Trung binh	0.934441519	0.89379837	0.795181037	0.917604778

Bảng 53: Kết quả chi tiết 27 chủ đề với 4 mô hình (2500 terms, N = 2)

So sánh các mô hình				
	SVM Multi	SVM Binary	kNN	NGram
Am nhac	0.97294	0.947109	0.920049	0.851169
Cum ga	0.994751	0.992126	0.905512	0.973753
Giao duc	0.964589	0.896601	0.57932	0.879774
Du lich	0.971681	0.945133	0.782301	0.973451
Du hoc	0.974619	0.946701	0.913706	0.956853
Loi song	0.906542	0.67757	0.009346	0.71028
Giai tri tin hoc	0.930693	0.855728	0.605375	0.90099
San pham tin hoc moi	0.941176	0.878992	0.894118	0.773109
Hackers va Virus	0.952978	0.902821	0.824451	0.981191
Cuoc song do day	0.901235	0.718518	0.25679	0.876543
San khau dien anh	0.93301	0.930097	0.976699	0.985437
My thuat	0.972222	0.9375	0.479167	0.840278
Thoi trang	0.923841	0.880795	0.543046	0.910596
Bong da	0.999317	0.995902	0.994536	0.988388
Tennis	1	1	0.809187	0.992933
Lam dep	0.971429	0.940136	0.760544	0.955102
Gioi tinh	0.981343	0.936567	0.350746	0.932836
Hinh su	0.938776	0.938776	0.387755	0.933673
Kinh doanh quoc te	0.969589	0.898032	0.601073	0.889088
Duong vao WTO	0.95288	0.921466	0.638743	0.968586
Chung khoan	0.971875	0.959375	0.75625	0.971875
Bat dong san	0.98227	0.964539	0.659574	0.989362
The gioi tre	0.863158	0.831579	0.028947	0.95
Khong gian song	0.931035	0.87931	0.137931	0.948276
Gia dinh	0.864286	0.675	0.139286	0.832143
Am thuc	0.9475	0.93	0.5575	0.9525
Mua sam	0.892857	0.845238	0.035714	0.857143
Trung binh	0.948392296	0.897244852	0.575839481	0.917604778

Bảng 54: Kết quả chi tiết 27 chủ đề với 4 mô hình (5000 terms, N = 2)

So sánh các mô hình				
	SVM_Multi	SVM_Binary	kNN	NGram
Am nhac	0.96925	0.926199	0.892134	0.851169
Cum ga	0.994751	0.989501	0.803019	0.973753
Giao duc	0.966006	0.892351	0.534107	0.879774
Du lich	0.966372	0.929204	0.591239	0.973451
Du hoc	0.956853	0.93401	0.867326	0.956853
Loi song	0.859813	0.560748	0.009346	0.71028
Giai tri tin hoc	0.933522	0.834512	0.507582	0.90099
San pham tin hoc moi	0.92437	0.84874	0.878908	0.773109
Hackers va Virus	0.946708	0.899687	0.804294	0.981191
Cuoc song do day	0.898765	0.711111	0.165432	0.876543
San khau dien anh	0.927184	0.929126	0.992349	0.985437
My thuat	0.972222	0.902778	0.153212	0.840278
Thoi trang	0.92053	0.844371	0.325099	0.910596
Bong da	0.999317	0.995219	0.994536	0.988388
Tennis	1	1	0.657911	0.992933
Lam dep	0.965986	0.931973	0.561326	0.955102
Gioi tinh	0.970149	0.906716	0.13806	0.932836
Hinh su	0.933673	0.928571	0.071429	0.933673
Kinh doanh quoc te	0.9678	0.890877	0.432937	0.889088
Duong vao WTO	0.95288	0.890052	0.415026	0.968586
Chung khoan	0.971875	0.946875	0.678521	0.971875
Bat dong san	0.978723	0.953901	0.405354	0.989362
The gioi tre	0.85	0.797368	0.028947	0.95
Khong gian song	0.913793	0.87931	0.093793	0.948276
Gia dinh	0.875	0.592857	0.033571	0.832143
Am thuc	0.945	0.9175	0.426871	0.9525
Mua sam	0.857143	0.75	0.035714	0.857143
Trung binh	0.941395741	0.873465074	0.462890481	0.917604778

Bảng 55: Kết quả chi tiết 27 chủ đề với 4 mô hình (7500 terms, N = 2)

**So sánh kết quả phân loại văn bản khác nhau theo số lượng đặc trưng
chọn lựa với mô hình SVM-Multi**

Mô hình SVM-Multi	2500	5000	7500
Am nhac	0.96064	0.97294	0.96925
Cum ga	1	0.994751	0.994751
Giao duc	0.958982	0.964589	0.966006
Du lich	0.953982	0.971681	0.966372
Du hoc	0.979695	0.974619	0.956853
Loi song	0.78972	0.906542	0.859813
Giai tri tin hoc	0.84017	0.930693	0.933522
San pham tin hoc moi	0.961345	0.941176	0.92437
Hackers va Virus	0.974922	0.952978	0.946708
Cuoc song do day	0.814815	0.901235	0.898765
San khau dien anh	0.905825	0.93301	0.927184
My thuật	0.965278	0.972222	0.972222
Thoi trang	0.903974	0.923841	0.92053
Bong da	0.997268	0.999317	0.999317
Tennis	1	1	1
Lam dep	0.964626	0.971429	0.965986
Gioi tinh	0.981343	0.981343	0.970149
Hinh su	0.943878	0.938776	0.933673
Kinh doanh quoc te	0.942755	0.969589	0.9678
Duong vao WTO	0.963351	0.95288	0.95288
Chung khoan	0.975	0.971875	0.971875
Bat dong san	0.985816	0.98227	0.978723
The gioi tre	0.805263	0.863158	0.85
Khong gian song	0.931035	0.931035	0.913793
Gia dinh	0.95	0.864286	0.875
Am thuc	0.935	0.9475	0.945
Mua sam	0.845238	0.892857	0.857143
Trung bình	0.934441519	0.948392296	0.941395741

Bảng 56: Kết quả chi tiết 27 chủ đề với số lượng đặc trưng khác nhau

Mô hình N-gram với phương pháp “làm tròn” (discounting smoothing method) Good Turing

<i>Mô hình N-gram (Good Turing)</i>			
	N = 2	N = 3	N = 4
Am Nhac	0.851169	0.854859	0.852399
Cum ga	0.973753	0.981627	0.981627
Giao duc	0.879774	0.898161	0.898161
Du lich	0.973451	0.975221	0.971681
Du hoc	0.956853	0.951777	0.954315
Loi song	0.71028	0.672897	0.668224
Giai tri tin hoc	0.90099	0.903819	0.902405
San pham tin hoc moi	0.773109	0.756303	0.761345
Hackers va Virus	0.981191	0.971787	0.968652
Cuoc song do day	0.876543	0.874074	0.879012
San khau dien anh	0.985437	0.988349	0.985437
My thuat	0.840278	0.854167	0.847222
Thoi trang	0.910596	0.890728	0.89404
Bong da	0.988388	0.991803	0.991803
Tennis	0.992933	0.992933	0.992933
Lam dep	0.955102	0.957823	0.956463
Gioi tinh	0.932836	0.929105	0.932836
Hinh su	0.933673	0.933673	0.933673
Kinh doanh quoc te	0.889088	0.86941	0.88551
Duong vao WTO	0.968586	0.963351	0.963351
Chung khoan	0.971875	0.96875	0.971875
Bat dong san	0.989362	0.98227	0.98227
The gioi tre	0.95	0.942105	0.944737
Khong gian song	0.948276	0.931035	0.931035
Gia dinh	0.832143	0.8	0.814286
Am thuc	0.9525	0.945	0.95
Mua sam	0.857143	0.869048	0.869048
Trung binh	0.917604778	0.912965741	0.914234815

Bảng 57: Kết quả chi tiết 27 chủ đề với N khác nhau