**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

🙣🕮🙡

Logo

Description automatically generated

**ĐỒ ÁN CUỐI KỲ**

Môn học: **XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**Đề tài: PHƯƠNG PHÁP ENSEMBLE HAI GIAI ĐOẠN CHO BÀI TOÁN PHÂN TÍCH CẢM XÚC BÌNH LUẬN TIẾNG VIỆT**

|  |  |
| --- | --- |
| Giảng viên: | **ThS. Nguyễn Trọng Chỉnh** |
|  | **GV. Đặng Văn Thìn** |

Lớp: Xử lý ngôn ngữ tự nhiên – **CS211.N21**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| **Nguyễn Huỳnh Vương Quốc** | **-** | **20521813** |
| **Lê Tuấn Cường** | **-** | **20520146** |

***TP. Hồ Chí Minh****, Thứ Hai, 03 Tháng Bảy 2023*

**MỤC LỤC**

[MỤC LỤC 2](#_Toc139289789)

[NHẬN XÉT CỦA GIẢNG VIÊN 3](#_Toc139289790)

[CHƯƠNG I: GIỚI THIỆU 4](#_Toc139289791)

[1. Giới thiệu bài toán: 4](#_Toc139289792)

[2. Ứng dụng của bài toán phân tích cảm xúc: 4](#_Toc139289793)

[CHƯƠNG II: PHÂN TÍCH HÌNH THÁI 6](#_Toc139289794)

[1. Phân tích ngữ liệu 6](#_Toc139289795)

[***1.1*** ***Nguồn ngữ liệu cho việc phân tích hình thái*** 6](#_Toc139289796)

[***1.2*** ***Ngữ liệu cho việc phân tích hình thái*** 6](#_Toc139289797)

[2. Tách từ 7](#_Toc139289798)

[***2.1*** ***Thủ công*** 7](#_Toc139289799)

[***2.2*** ***Thuật toán Maximum Matching*** 9](#_Toc139289800)

[***2.3*** ***Sử dụng thư viện*** 15](#_Toc139289801)

[***2.4*** ***So sánh & đánh giá*** 17](#_Toc139289802)

[CHƯƠNG III: BÀI TOÁN ỨNG DỤNG 19](#_Toc139289803)

[1. Phát biểu bài toán: 19](#_Toc139289804)

[2. Mô tả ngữ liệu của bài toán 19](#_Toc139289805)

[***2.1*** ***Phân tích thăm dò dữ liệu*** 20](#_Toc139289806)

[***2.2*** ***Tiền xử lý dữ liệu*** 23](#_Toc139289807)

[3. Phương pháp giải quyết bài toán 24](#_Toc139289808)

[***3.1*** ***Phương pháp xử lý dữ liệu không cần bằng*** 24](#_Toc139289809)

[***3.2*** ***Trích xuất đặc trưng*** 28](#_Toc139289810)

[***3.3*** ***Các mô hình học máy và học sâu sử dụng*** 29](#_Toc139289811)

[CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM 32](#_Toc139289812)

[1. Cài đặt thực nghiệm 32](#_Toc139289813)

[2. Đánh giá kết quả và so sánh 33](#_Toc139289814)

[***2.1*** ***Mô hình phân lớp nhị phân (ở giai đoạn 1)*** 33](#_Toc139289815)

[***2.2*** ***Mô hình phân lớp các nhãn mang tính tiêu cực (ở giai đoạn 2)*** 33](#_Toc139289816)

[***2.3*** ***Mô hình phân lớp các nhãn mang tính tích cực (ở giai đoạn 2)*** 34](#_Toc139289817)

[***2.4*** ***Kết quả của pipeline dự đoán*** 34](#_Toc139289818)

[***2.5*** ***So sánh với các mô hình baseline*** 35](#_Toc139289819)

[CHƯƠNG V: KẾT LUẬN 37](#_Toc139289820)

[TÀI LIỆU THAM KHẢO 38](#_Toc139289821)

# **NHẬN XÉT CỦA GIẢNG VIÊN**

# **CHƯƠNG I: GIỚI THIỆU**

## **Giới thiệu bài toán:**

Phân tích cảm xúc (Sentiment analysis) là nhằm phát hiện ra thái độ mang tính lâu dài, màu sắc tình cảm, khuynh hướng niềm tin vào các đối tượng hay người nào đó. Các vấn đề xung quanh đến việc phân tích cảm xúc:

* Làm thế nào mà máy có thể hiểu đó là ý kiến chủ quan hay thiên về cảm xúc
* bản thân?
* Làm thế nào để máy phân rõ 3 loại cảm xúc: tích cực (positive) tiêu cực (negative) và trung tính (neutral)?
* Làm thế nào để máy có thể giải quyết các cảm xúc chủ quan?
* Làm thế nào để máy tính điểm cho một quan điểm, ý kiến?
* Làm thế nào để máy biết được cường độ của cảm xúc?
* Làm thế nào để biết sự khác nhau giữa fact (sự thật) và opinion (ý kiến viết ra)? Nhiều khi bề ngoài gồm những cụm từ khen nhưng thật sự lại đang xỏ xiên.

A picture containing human face, screenshot

Description automatically generated

Hình 1. Phân tích cảm xúc từ văn bản *(Nguồn: http://aylien.com/).*

## **Ứng dụng của bài toán phân tích cảm xúc:**

Các vấn đề xung quanh đến việc phân tích cảm xúc, việc phân tích cảm xúc trong văn bản được ứng dụng trong hàng loạt các vấn đề như: quản trị thương hiệu doanh nghiệp, thương hiệu sản phẩm, quản trị quan hệ khách hàng, khảo sát ý kiến xã hội học, phân tích trạng thái tâm lý con người…

Một ví dụ cụ thể tại Việt Nam là vụ việc “con ruồi trong chai number one” của doanh nghiệp Tân Hiệp Phát, gây ảnh hưởng xấu đến hình ảnh của Tân Hiệp Phát và việc tiêu thụ sản phẩm nước uống tăng lực number one của doanh nghiệp này. Xét về luật pháp thì Tân Hiệp Phát là đúng nhưng không khéo léo trong việc xử lý quan hệ với khách hàng, gây bất bình trên mạng xã hội, đó lại là bài toán quản gây bất bình trên mạng xã hội, đó lại là bài toán quản trị quan hệ với khách hàng mà doanh nghiệp phải giải quyết. Mà ai biết được các thông tin bất lợi về Tân Hiệp Phát này có được thúc đẩy bởi các đối thủ cạnh tranh hay không? Điều này đòi hỏi phải có một công cụ hỗ trợ đắc lực, mà chỉ có áp dụng công nghệ thông tin mới giải quyết được, chứ không lực lượng con người nào có thể làm xuể. Rút kinh nghiệm từ Tân Hiệp Phát thì các doanh nghiệp lớn của Việt Nam hiện nay cũng đã đặt hàng các doanh nghiệp công nghệ thông tin giải quyết vấn đề này. Giải pháp công nghệ hiện nay được gọi là "lắng nghe mạng xã hội", tức là các doanh nghiệp CNTT mua các dữ liệu thời gian thực (real time) từ các công ty mạng xã hội về để xử lý các thông tin liên quan đến doanh nghiệp hay các sản phẩm mà doanh nghiệp đó kinh doanh, nhằm phát hiện và ngăn chặn sớm sự lan rộng các thông tin bất lợi trên mạng xã hội, có hình thức đính chính phản hồi đến các khách hàng của mình, đồng thời thương lượng, ngăn chặn tận gốc những người tạo ra các nội dung đó. Điều cốt yếu của giải pháp này chính là phân tích cảm xúc của các dòng trạng thái trên mạng xã hội nhằm lọc ra các thông tin bất lợi để xử lý.

# **CHƯƠNG II: PHÂN TÍCH HÌNH THÁI**

## **Phân tích ngữ liệu**

### ***1.1 Nguồn ngữ liệu cho việc phân tích hình thái***

Việc tìm ngữ liệu cho bài toán phân tích hình thái sẽ ảnh hưởng trực tiếp đến bài toán phân tích cảm xúc. Vì vậy, để ngữ liệu có thể dùng chung và trực quan nhất, nhóm em đã quyết định sử dụng ngữ liệu tìm được từ [website mua vé xem phim của momo](https://momo.vn/cinema). Đây là một website giúp đặt vé xem phim, đồng thời cũng có thể để lại bình luận tại phim đó. Ngữ liệu là những câu bình luận thu được từ nhiều bộ phim khác nhau hiện đang công chiếu tại Việt Nam.

### ***1.2 Ngữ liệu cho việc phân tích hình thái***

Ngữ liệu của phân tích hình thái bao gồm 100 câu *(cau.txt)* được gán nhãn 1 trong 3 nhãn tương ứng với từng 3 sắc thái:

* **Tích cực**: các lời khen, lời nói mang tính chất ủng hộ, các lời khuyên mang tính chất định hướng tốt…
* **Tiêu cực**: những bình luận chê bai, chửi rủa, lời khuyên mang tính chất định hướng nên tránh né…
* **Trung tính**: ngoài 2 phân loại trên những bình luận mang tính chất không khen cũng không chê sẽ nằm trong nhãn này.

Các nhãn được đánh dấu và lưu vào trong file *(label.txt)* tương ứng với:

* 1: Tích cực (gồm 45 nhãn).
* -1: Tiêu cực (gồm 44 nhãn).
* 0: Trung tính (gồm 11 nhãn).

A screenshot of a computer program

Description automatically generated with low confidence

Hình 2. Ngữ liệu trong file *cau.txt*.

A picture containing text, screenshot

Description automatically generated

Hình 3. Ngữ liệu trong file *labels.txt*.

## **Tách từ**

### ***2.1 Thủ công***

Việc tách từ Tiếng Việt được quy định như sau:

* Giữa các từ đơn hoặc từ ghép hoặc giữa từ đơn và từ ghép, cách nhau bởi dấu cách (‘ ’).
* Giữa các tiếng của từ ghép ngăn cách với nhau với dấu gạch chân (‘\_’).
* Các dấu câu cũng xem như là một từ đơn nên ngăn cách trước sau bởi dấu cách (‘ ’).

**Ví dụ** **cho câu sau**: “Cảnh phim đẹp, nội dung lôi cuốn người xem”. Sau khi tách từ kết quả sẽ là: (với cách khoảng màu đỏ là dấu cách).

A picture containing text, font, screenshot, typography

Description automatically generated

Hình 4: Kết quả tách từ thủ công.

Cách tách từ thủ công sẽ dựa trên từ điển Tiếng Việt VLSP, thực hiện tra trên [website](https://vlsp.hpda.vn/) online của Giáo sư Hồ Bảo Tú. Sau khi thực hiện tách từ, ta được file kết quả tách từ *(hinhthai.txt)* có mẫu như sau:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 5: File hinhthai.txt (kết quả sau khi tách từ).

A picture containing text, screenshot, plot, diagram

Description automatically generated

Hình 6: Số lượng câu theo độ dài của số từ.

* **Các khó khăn cần lưu ý (khi thực hiện tách từ, nhóm đã gặp phải một số từ khác thường và được phân loại thành theo 3 nguyên do sau):**

1. *Các từ có ý nghĩa nhưng lại không có trong từ điển VLSP:*
   * Từ **'phân cảnh'** câu 8 không có trong từ điển nhưng nếu tách ra thì từ **'phân'** không có nghĩa trong VLSP.
   * Từ **'đen thui thùi lùi'** câu 38 cách chơi chữ mang ý nghĩa rất đen.
2. *Các từ là từ mới hoặc từ vựng vùng miền:*
   1. Từ **'giả trân'** câu 38 là cụm từ mới được xuất hiện do kết hợp từ từ **'giả'** và từ **'trân'** và được giới trẻ sử dụng khá phổ biến.
   2. Từ **'dữ thần'** câu 69 là từ vựng Tây Nam Bộ mang nghĩa là rất nhiều, rất lớn, đến mức ngoài sự tưởng tượng của mọi người.
   3. Từ **'tuyệt cú mèo'** câu 74 là từ vựng Nam Bộ mang nghĩa tuyệt vời, rất tuyệt, đạt đến mức coi như lý tưởng, không còn chê chỗ nào được nữa.
3. *Các từ là từ tiếng Anh vay mượn:*
   1. Từ **'ok'** câu 2 là từ vay được dùng với nghĩa là được hoặc tạm được.
   2. Từ **'relax**' câu 45 là từ tiếng Anh với nghĩa là thư giãn.
   3. Từ **'Marketing'** câu 89 là tên riêng tiếng Anh của phòng Quảng cáo, truyền thông.

Đối với nguyên nhân 1) và 2), ta có thể phân thành các từ ghép và giữa các tiếng của từ ghép dùng dấu gạch dưới (‘\_’). Đối với nguyên do 3) phân loại như một từ đơn bình thường và cần chú ý khi gán nhãn các từ này.

### ***Thuật toán Maximum Matching***

Phương pháp khớp tối đa (Maximum Matching) hay còn gọi là Left-Right Maximum (LRMM). Theo phương pháp này, ta sẽ duyệt một ngữ hoặc câu từ trái sang phải và chọn từ có nhiều âm tiết nhất có mặt trong từ điển, rồi cứ thế tiếp tục cho từ kế tiếp cho đến hết câu.

* **Các bước thực hiện thuật toán như sau:**

**B1**: Dùng một từ điển D chứa toàn bộ từ trong ngôn ngữ đang xét. Có thể chia từ điển D thành các từ điển nhỏ hơn như:

* + D-1: chứa các từ 1 âm tiết.
  + D-2: chứa các từ 2 âm tiết.
  + D-3: chứa các từ 3 âm tiết.
  + D-4: chứa các từ 4 âm tiết.

**B2**: Chọn từ có độ dài lớn nhất trong câu cần tách theo thứ tự từ trái qua phải. Theo thống kê, từ tiếng Việt thường xuất hiện có độ dài cao nhất là 4. Nên ta sẽ xét 4 từ liên tiếp xem có thuộc vào D-4.

**B3**: Nếu từ đó xuất hiện trong từ điển thì chọn, ngược lại giảm độ dài từ và kiểm tra lại.

**B4**: Tiếp tục thực hiện việc chọn từ kế tiếp.

* **Ví dụ cho câu**: “Cốt truyện hay, mới mẻ, thú vị, dễ tiếp cận.”
* Bắt đầu từ ‘cốt’ xét 4 từ liên tiếp: ‘cốt’, ‘truyện’, ‘hay’, ‘,’ (lưu ý: các dấu câu sẽ được nhận dạng như 1 từ đơn).
* “cốt truyện hay” không thuộc vào từ điển D-4. Giảm số lương từ xuống 3.
* “cốt truyện hay” không thuộc vào từ điển D-3. Giảm số lương từ xuống 2.
* “cốt truyện” thuộc vào từ điển D-2. Vậy ta được từ ‘Cốt truyện’.
* Chọn từ tiếp theo là ‘hay’ và thực hiện xét 4 từ liên tiếp như trên cho đến khi hết.

Ta được kết quả cuối cùng: *“Cốt\_truyện hay, mới\_mẻ, thú\_vị, dễ tiếp\_cận”*.

* **Cài đặt code:** Ở thuật toán này, từ điển được sử dụng là từ điển 74k.txt gồm 73901 từ [nguồn từ điển](https://github.com/vntk/dictionary).

A picture containing text, screenshot, font

Description automatically generated

Hình 7: Hàm chia từ điển thành các từ điển nhỏ hơn.

Việc chia nhỏ từ điển sẽ giúp tăng tốc độ, khi máy tính thực hiện việc kiểm tra xem từ đó có nằm trong từ điển hay không? Ở đây, việc xác định một từ thuộc từ điển nào, được thực hiện bằng cách đếm số dấu cách (‘ ’) có trong từ đó.

* + 0 dấu cách: Từ điển 1 từ, D-1
  + 1 dấu cách: Từ điển 2 từ, D-2
  + 2 dấu cách: Từ điển 3 từ, D-3
  + 3 dấu cách: Từ điển 4 từ, D-4

Do ta đã xác định từ tiếng Việt thường xuất hiện có độ dài cao nhất là 4 nên ta chỉ cần lấy từ điển 4 từ trở xuống. Ngoài ra, ta có thể bỏ từ điển 1 từ, do nếu từ không thuộc D-4, D-3, D-2 thì chắc chắc sẽ là từ 1 âm tiết. Hơn nữa các dấu câu theo quy ước là từ đơn nên sẽ gây ra sai sót khi so sánh với từ điển D-1.

A picture containing text, screenshot, font

Description automatically generated

Hình 8: Hàm tách câu thành các âm tiết riêng biệt.

Để thực hiện phương pháp tách từ, đầu tiên ta cần tách câu thành các âm tiết riêng biệt. Nếu chỉ tách bởi dấu cách, cách trường hợp dấu nằm ngay sau chữ sẽ không được tách. *Ví dụ: ‘Phim hay, đáng xem.’ Khi tách bằng dấu cách (‘ ’) sẽ trở thành: [‘Phim’, ‘hay,’, ‘đáng’, ‘xem.’]*. Như vậy khi phân tích sẽ không phân tích được. Vì vậy, ta sẽ sử dụng regex để tách các trường hợp này theo thứ tự:

* Cách từ nối đặc biệt: như a-pác-thai, phát-xít, …
* Các số: như 5, 10.5, 10%...
* Các từ bình thường: như ‘đẹp’, ‘nam’, ‘hoa’, …
* Các từ còn lại không phải chữ hoặc số: sẽ là các dấu câu.

A screen shot of a computer program

Description automatically generated with low confidence

Hình 9: Hàm tách từ - Maximum Matching.

Thuật toán được bắt đầu bằng việc phân tách câu thành các âm tiết bằng hàm **syllablize()** và tách từ điển **split\_grams()**. Tiếp theo, khởi tại **index** bắt đầu vị trí thực hiện thuật toán. Khởi tạo độ dài chuỗi từ ban đầu **length\_words** là 4. Thực hiện vòng lặp sau cho đến khi *index có giá trị lớn độ dài mảng các âm tiết*:

* Nếu **length\_words là 4** 🡪 Xét các từ từ vị trí hiện tại (index) đến (index + 4) có nằm trong từ điển 4:
* Đúng:
  + Thêm từ vào mảng.
  + Cập nhật **index = index + 4**.
* Sai:
  + Giảm **length\_words** đi 1 đơn vị.

Nếu **length\_words là 3**: Xét các từ từ vị trí hiện tại (index) đến (index + 3) có nằm trong từ điển 3:

* Đúng:
  + Thêm từ vào mảng.
  + Cập nhật **index =index + 3**, **length\_words = 4**.
* Sai:
  + Giảm **length\_words** đi 1 đơn vị.
* Nếu **length\_words == 2** 🡪 Xét các từ từ vị trí hiện tại (index) đến (index + 2) có nằm trong từ điển 2:
* Đúng:
  + Thêm từ vào mảng.
  + Cập nhật **index =index + 2**, **length\_words = 4.**
* Sai:
  + Giảm **length\_words** đi 1 đơn vị.
* Cập nhật từ đơn vào mảng nếu không thoả các điều trên và cập nhật **index =index** **+ 1**, **length\_words = 4**.

Hàm trên, khi chạy vòng lặp nhiều lần, việc chia lại cùng một từ điển là không cần thiết. Nên ta có thể tách từ điển split\_grams() trước khi gọi thuật toán, và truyền vào các từ điển D-2, D-3, D-4 để có thể cải thiện tốc độ thực thi.

### ***Sử dụng thư viện***

#### **a. Underthesea**

Thư viện **Underthesea** là một bộ công cụ NLP tiếng Việt. Underthesea là một bộ dữ liệu mô-đun Python mã nguồn mở và các hướng dẫn hỗ trợ nghiên cứu và phát triển trong Xử lý ngôn ngữ tự nhiên bằng tiếng Việt. Thư viện cung cấp API cực kỳ dễ dàng để nhanh chóng áp dụng các mô hình NLP được đào tạo trước vào văn bản tiếng Việt, chẳng hạn như phân đoạn từ, gắn thẻ một phần của bài phát biểu (PoS), nhận dạng thực thể được đặt tên (NER), phân loại văn bản và phân tích cú pháp phụ thuộc.

Underthesea cung cấp một function giúp tách từ tiếng Việt là word\_tokenize(). Hàm này sử dụng **Trường xác xuất có điều kiện (CRFs)** cho thuật toán tách từ.

Kết quả sau khi tách có dạng danh sách các từ như: [‘Cảnh’, ‘phim’, ‘đẹp’, ‘,’, ‘nội dung’, ‘lôi cuốn’, ‘người’, ‘xem’, ‘.’]. Vì vậy, để thống nhất kết quả giữa các phương pháp sẽ chuyển kết quả về dạng chuỗi, và giữa các từ ghép là dấu gạch dưới.

* “Cảnh phim đẹp, nội\_dung lôi\_cuốn người xem”.

A screen shot of a computer code

Description automatically generated with medium confidence

Hình 10: Sử dụng thư viện Underthesea.

#### **b. pyvi**

pyvi là một thư viện xử lý ngôn ngữ tự nhiên tiếng Việt, bao gồm các chức năng như:

* **Tách từ**: Chia một câu thành các từ đơn vị, bao gồm cả từ ghép, từ láy, từ viết tắt, …
* **Gán nhãn từ loại**: Gắn mỗi từ với một nhãn chỉ loại từ của nó, ví dụ danh từ, động từ, tính từ, …
* **Bỏ dấu**: Xóa các dấu thanh của các nguyên âm trong tiếng Việt, ví dụ a, ă, â, e, ê, i, o, ô, ơ, u, ư, y.
* **Thêm dấu**: Thêm các dấu thanh cho các nguyên âm trong tiếng Việt dựa trên ngữ cảnh của câu.

pyvi cũng sử dụng thuật toán **Trường xác xuất có điều kiện (CRFs)** để huấn luyện và dự đoán các chức năng trên.

Kết quả sau khi tách có dạng chuỗi các từ đúng định dạng cần:

* ‘Cảnh phim đẹp, nội\_dung lôi\_cuốn người xem.’

A picture containing text, screenshot, font, electric blue

Description automatically generated

Hình 11: Sử dụng thư viện pyvi.

### ***So sánh & đánh giá***

Do việc tách từ tương tự như việc quyết định đặt dấu gạch chân ‘\_’ hay dấu cách ‘ ’ vào giữa các âm tiết. Nên ta sẽ đánh giá kết quả tách từ dựa vào việc đặt dấu ‘\_’ hay ‘ ’ vào một vị trí là đúng hay sai.

* **Ví dụ:**
* **Prediction**: “Phim không làm tôi thất\_vọng, đáng để dẫn bạn bè gia\_đình đi xem.”
* **True**: “Phim không làm tôi thất\_vọng, đáng để dẫn bạn\_bè gia\_đình đi xem.”

Ma trận nhầm lẫn:

|  |  |
| --- | --- |
| TP = 2 | TN = 0 |
| FN = 1 | FP = 13 |

Bảng 1. Ma trận nhầm lẫn.

* **Kết quả:**

**A picture containing text, screenshot, font, number

Description automatically generated**

Hình 12: Kết quả đánh giá.

* **Nhận xét:** Trong 3 cách trên thì thư viện Underthesea với phương thức Conditional Random Field (CRF) cho kết quả tốt hơn so vưới 2 cách còn lại. Pyvi tuy cũng sử dụng Conditional Random Field (CRF) nhưng kết quả không tốt bằng underthesea. Và thuật toán Maximum matching cho kết quả thấp nhất.
* **So sánh & đánh giá:**

|  |  |  |
| --- | --- | --- |
|  | **Ưu điểm** | **Nhược điểm** |
| **Maximum Matching** | **-** Không cần phải huấn luyện mô hình, chỉ cần có một bộ từ điển đầy đủ và cập nhật.  - Độ phức tạp tính toán thấp hơn | - Không thể xử lý được các trường hợp ngoại lệ, ví dụ như các từ mới, các từ viết tắt, hay các từ ghép.  - Gặp khó khăn khi xử lý các ngôn ngữ có cấu trúc phức tạp hơn so với tiếng Việt, ví dụ như tiếng Anh, tiếng Trung, hay tiếng Nhật. |
| **Thư viện Underthesea** | - Cả 2 thuật toán đều dựa trên mô hình xác suất có điều kiện, để tách từ. Thuật toán CRF có thể học được các mối quan hệ giữa các đặc trưng của dữ liệu và các nhãn, cũng như giữa các nhãn liền kề trong chuỗi nên sẽ cho kết quả tốt hơn.  - CRF còn có thể áp dụng cho nhiều ngôn ngữkhác như tiếng Anh, tiếng Trung, tiếng Nhật | **-** Cần phải có một bộ dữ liệu đã được gắn nhãn sẵn để huấn luyện mô hình, cũng như phải xây dựng và lựa chọn các đặc trưng phù hợp cho bài toán.  - Thuật toán CRF có độ phức tạp tính toán cao hơn |
| **Thư viện pyvi** |

Bảng 2. Ưu và nhược điểm của các thuật toán tách từ.

# **CHƯƠNG III: BÀI TOÁN ỨNG DỤNG**

## **Phát biểu bài toán:**

Bài toán phân tích cảm xúc thuộc dạng bài toán phân tích ngữ nghĩa văn bản. Vì vậy, ta cần phải xây dựng một mô hình để hiểu được ý nghĩa của câu văn, đoạn văn để quyết định xem câu văn đó hoặc đoạn văn đó mang màu sắc cảm xúc chủ đạo nào. Phát biểu theo góc nhìn của *máy học (Machine Learning)* thì phân tích cảm xúc là bài toán phân lớp cảm xúc dựa trên văn bản ngôn ngữ tự nhiên:

* Đầu vào của bài toán là một câu hay một đoạn văn bản.
* Đầu ra là các giá trị xác suất (điểm số) của N lớp cảm xúc mà ta cần xác định.

Một ví dụ về bài toán phân tích cảm xúc trong tiếng Việt đã được trình bày thông thường là kết quả của thực nghiệm của việc phân tích cảm xúc văn bản với 3 lớp cảm xúc là: tiêu cực (negative), tích cực (positive) và trung tính (neural). Quá trình này được mô tả ở Hình 13.

A picture containing text, screenshot, font, circle

Description automatically generated

Hình 13: Quy trình thực hiện.

## **Mô tả ngữ liệu của bài toán**

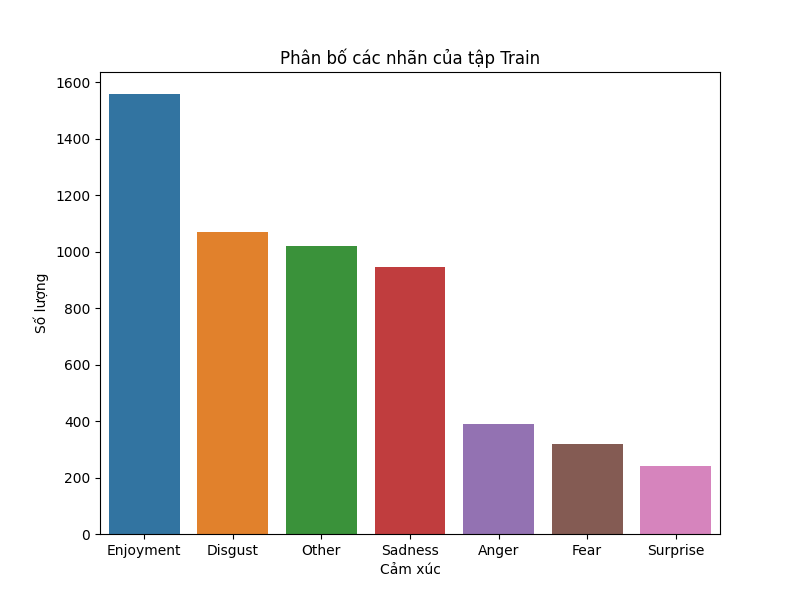
Trong báo cáo đồ án lần này, chúng em sử dụng tập dữ liệu *UIT-VSMEC (Vietnamese Social Media Emotion Corpus)* được xây dựng vào năm 2019. Đây là bộ dữ liệu về phân tích cảm xúc cho tiếng Việt đầu tiên. Bộ dữ liệu này gồm 6927 câu bình luận tiếng Việt được thu thập và gán nhãn với 6 nhãn chính bao gồm:

* **Enjoyment**: Đây là những cảm xúc khác nhau, từ sự thích thú khi giúp đỡ người khác, cảm giác phấn khởi ấm áp mà mọi người trải nghiệm khi họ nhìn thấy lòng tốt. *Ví dụ: “Nháy mắt thôi cũng đáng yêu, kkk”.*
* **Sadness**: Đây là những cảm xúc chứa đựng sự thất vọng và tuyệt vọng. *Ví dụ: “Lúc đấy khổ lắm… kỷ niệm :( “*
* **Fear**: Đây là những cảm xúc lo lắng và kinh hãi. Các trạng thái thay đổi từ lo lắng - dự đoán khả năng xảy ra nguy hiểm, căng thẳng, sợ hãi đến tuyệt vọng. *Ví dụ: “Chuyện này làm tao nổi hết da gà”.*
* **Anger**: Đây là những cảm xúc bực tức hoặc cảm giác khó chịu khi có ai đó cố ý cản trở việc gì đó của chúng ta. Nó chứa đựng cả sự khó chịu và giận dữ và thay đổi từ sự thất vọng là phản ứng đối với những thất bại lặp đi lặp lại để vượt qua một trở ngại, sự bực tức. *Ví dụ: “Biến đi!”*.
* **Surprise**: Đây là những cảm xúc do sự việc bất ngờ, điều gì đó khó tin và có thể khiến bạn bị sốc. Đây là cảm xúc ngắn hạn chỉ diễn ra trong vài giây. Và nó qua đi khi chúng ta hiểu chuyện gì đang xảy ra, và nó có thể trở thành sợ hãi, tức giận, nhẹ nhõm hay không gì cả... Ví dụ: *“Trên đời còn tồn tại thứ này sao?”.*
* **Disgust**: Đây là những cảm xúc thể hiện sự không thích, không hài lòng và cảm thấy ghê tởm. Cường độ của chúng thay đổi từ xung lực tránh thứ gì đó kinh tởm hoặc ác cảm, phản ứng với mùi vị, mùi, đồ vật hoặc ý tưởng xấu. *Ví dụ: “Làm bạn với mấy thể loại này nhục cả người”.*
* **Other**: Dành cho những bình luận không thể hiện cảm xúc nào ở trên hoặc những bình luận không chứa bất kỳ cảm xúc nào. Ví dụ: *“Mình đã xem rất nhiều phim có kịch bản như này rồi nên thấy cũng bình thường!”.*

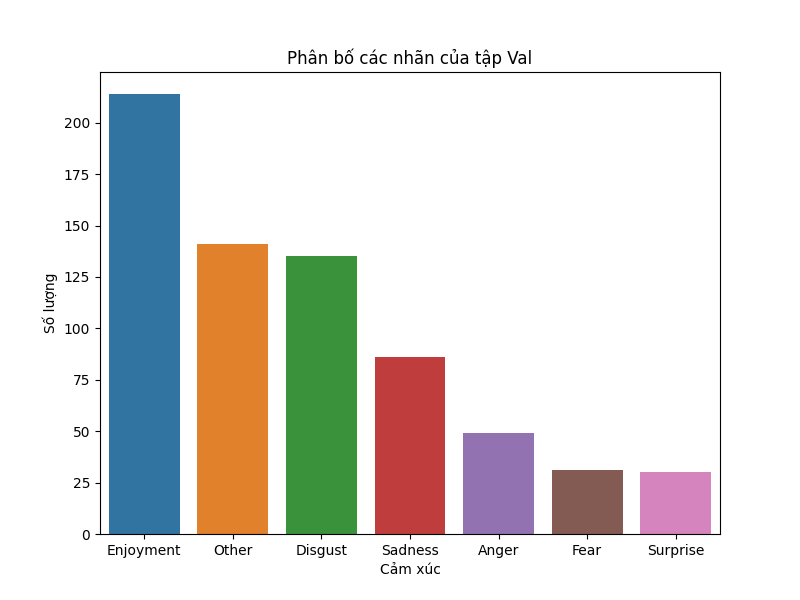
### ***2.******1 Phân tích thăm dò dữ liệu***

Sau khi qua thực hiện qua các bước xem xét để hiểu về dữ liệu, chúng em thấy rằng bộ dữ liệu được chia làm 3 tập: train, validation và test. Cả 3 tập dữ liệu này đều không có giá trị NULL. Tập train gồm có 5548 bình luận, tập validation gồm có 686 bình luận và tập test gồm có 693 bình luận.

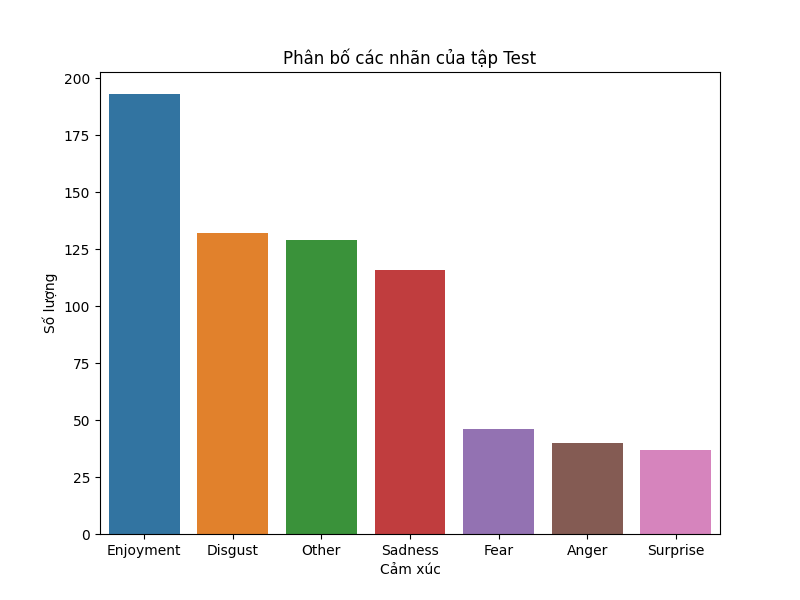
Các hình 14, 15 và 16 dưới đây là các biểu đồ phân bố các nhãn của cả 3 tập dữ liệu.



Hình 14: Phân bố các nhãn trên tập training.



Hình 15: Phân bố các nhãn trên tập validation.



Hình 16: Phân bố các nhãn trên tập test.

Nhìn sơ bộ các phân bố nhãn của các bình luận chúng ta có thể thấy rằng, giữ các cảm xúc có sự chênh lệch lớn. Các cảm xúc có số lượng chiếm đa số thường là các cảm xúc mang các yếu tố tích cực vì thông thường ở trên các nền tảng mạng xã hội lớn như facebook, youtube, instagram… thường sẽ có những tiêu chuẩn đánh giá về bình luận cũng như các bình luận sẽ được kiểm duyệt. Chính điều này làm mất cân bằng trong cả ba tập của bộ dữ liệu, gây ảnh hưởng và làm khó khăn trong việc huấn luyện mô hình. Các thách thức lớn trong trường hợp huấn luyện trên tập dữ liệu không đồng đều có thể kể đến như: hiệu suất đánh giá không chính xác, một hình bị chênh lệch precision về các lớp có số lượng lớn hơn, overfitting hay các thông tin quan trọng của dữ liệu.

### ***Tiền xử lý dữ liệu***

Đối với các bài toán xử lý ngôn ngữ tự nhiên, tiền xử lý dữ liệu hay các bước phân tích hình thái ngôn ngữ là hết sức quan trọng. Vì vậy, để đảm bảo và gia tăng độ chính xác cho mô hình chúng ta cần phải có một quy trình tiền xử lý đủ tốt trước khi đem vào huấn luyện. Với bài toán phân tích cảm xúc này, chúng em sẽ thực hiện các bước tiền xử lý dữ liệu văn bản như sau:

* Đưa tất cả các bình luận về dạng chữ thường, không in hoa (lower case).
* Xóa tất cả các ký tự đặc biệt như %, $, #, @… (các ký tự này có thể có ảnh hưởng đến mô hình tuy nhiên để đơn giản hóa cho việc xử lý thì chúng ta cần bỏ những ký tự này đi).
* Xóa tất cả các khoảng trắng thừa.
* Tách từ (tokenization).

Để trực quan cho quá trình này, dưới là bảng so sánh giữa các câu bình luận chưa được tiền xử lý và sau khi được tiền xử lý.

|  |  |
| --- | --- |
| Sentence | Cleaned sentence |
| ko phải con mình , mà xem còn thấy đau như vậy huống gì người trong cuộc . thật là phẫn nộ mà . cơ quan chức năng làm việc quá chậm trễ , đến giờ mà vẫn chưa tìm ra người chịu trách nhiệm . 😠😠😠😠😠😠 | ko phải con mình mà xem còn thấy đau như vậy huống gì người trong cuộc thật là phẫn nộ mà cơ quan chức năng làm việc quá chậm trễ đến giờ mà vẫn chưa tìm ra người chịu trách nhiệm |
| uớc gì sau này về già vẫn có thể như cụ này :)) | uớc gì sau này về già vẫn có thể như cụ này |

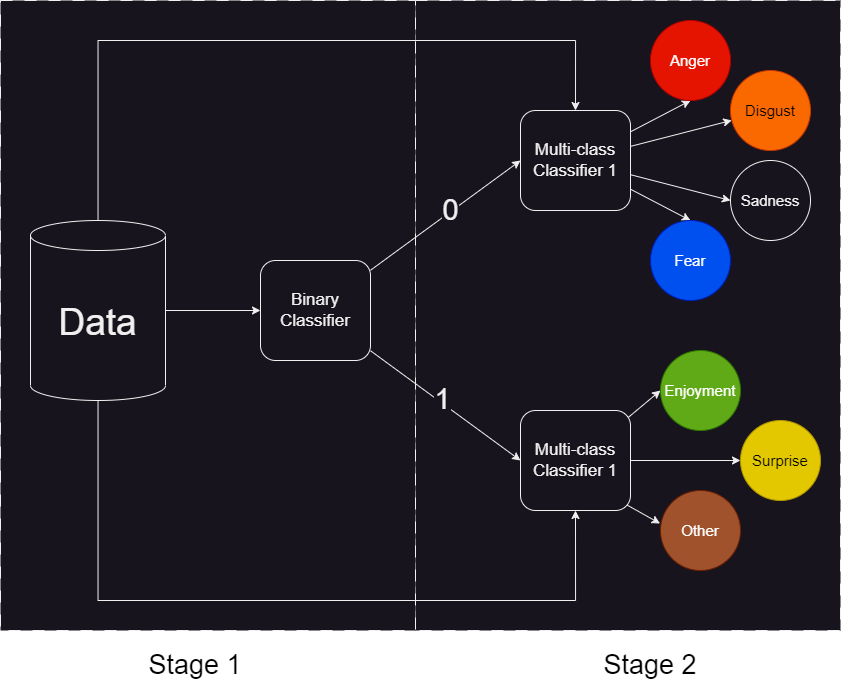
Bảng 3. So sánh giữa dữ liệu đã và chưa tiền xử lý.

## **Phương pháp giải quyết bài toán**

### ***3.1 Phương pháp xử lý dữ liệu không cần bằng***

Với mong muốn tăng độ chính xác của mô hình, giảm sự ảnh hưởng của việc sử dụng dữ liệu không đồng đều nhãn, chúng em sẽ thực nghiệm triển khai một pipeline học máy theo dạng ống nước gồm 2 giai đoạn như sau:

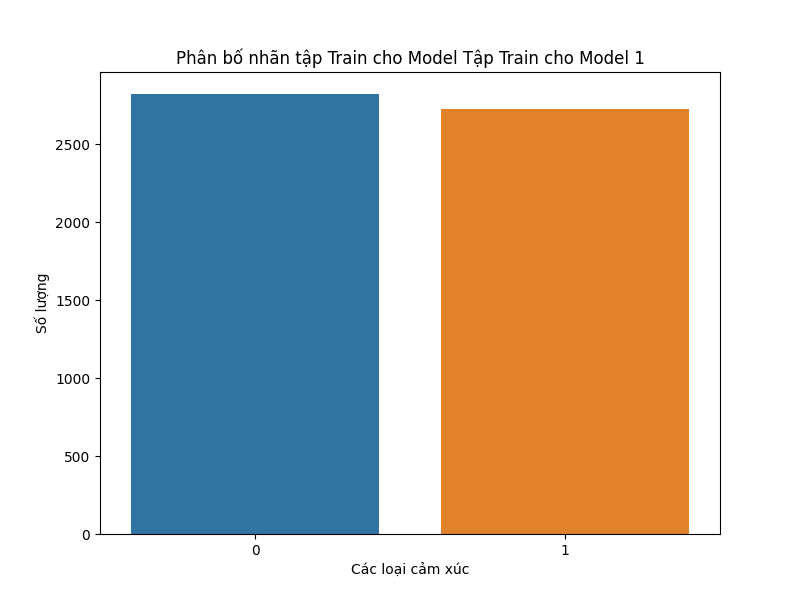
* **Giai đoạn 1**: Phân lớp nhị phân các câu bình luận để trả lời cho câu hỏi liệu bình luận đó là bình luận thuộc dạng tiêu cực, có tính chất tiêu cực hay bình luận đó thuộc dạng tích cực, không rõ bản chất.
* **Giai đoạn 2**: Giai đoạn này gồm có hai mô hình học máy. Mô hình đầu tiên dùng để trả lời câu hỏi bình luận này tiêu cực như thế nào mức nào nếu kết quả trả về của giai đoạn 1 là tiêu cực. Mặc khác, mô hình thứ hai dùng để trả lời câu hỏi bình luận này tích cực như thế nào nếu kết quả trả về của giai đoạn 1 là tích cực.



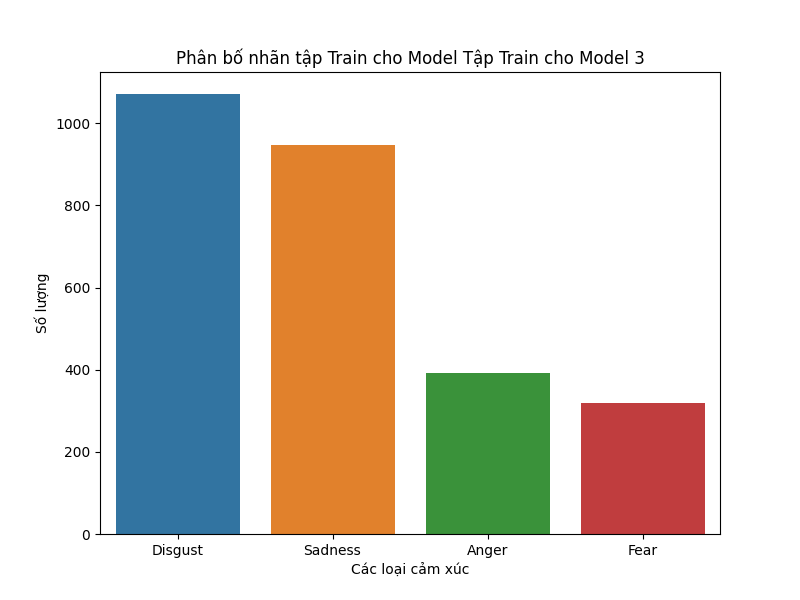
Hình 17. Pipeline xây dựng.

Hình 17 mô tả quá trình này. Ở giai đoạn 1, vì bộ dữ liệu này gồm có 7 nhãn khác nhau tương ứng với 7 cảm xúc. Nên chúng ta cần phân chia 7 nhãn cảm xúc này thành hai nhóm mang trạng thái tiêu cực và tích cực. Nếu xét ở khía cạnh nhận thức, chúng ta có thể thấy rằng 4 cảm xúc là *Anger, Disgust, Sadness* và *Fear* mức độ liên quan đến nhau cao và thường mang ý nghĩa tiêu cực nhiều hơn. Ngược lại, cả 3 cảm xúc *Enjoyment, Surprise* và *Other* đều ở dạng không cảm xúc, không rõ hoặc mang nghĩa tích cực. Hay nói cách khác, cả 3 cảm xúc này là phủ định của nhóm 4 cảm xúc trước đó. Vì thế, ở giai đoạn 1, chúng em chỉ cần huấn luyện mô hình phân lớp nhị phân trên bộ dữ liệu được phân chia làm hai phần tích cực và tiêu cực để giảm thiểu số nhãn và mong muốn nâng cao hiệu suất tổng thể của mô hình.

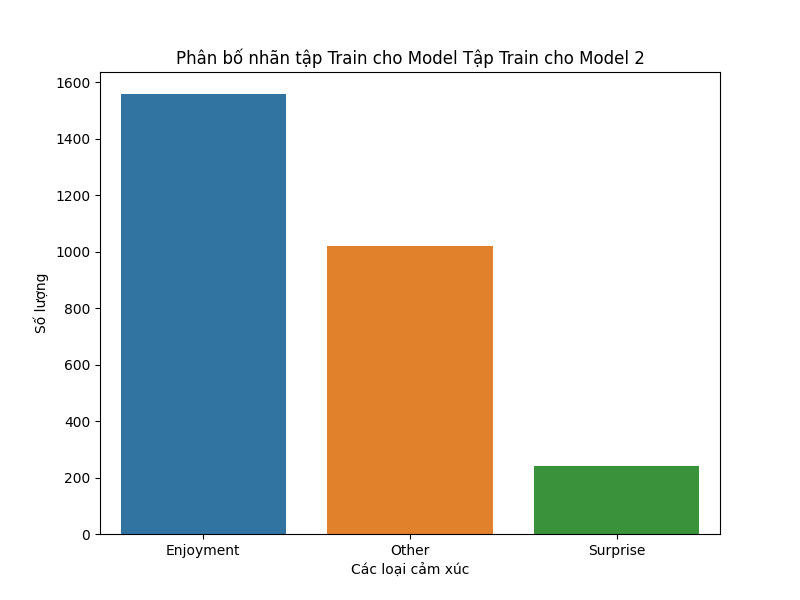
Ở gian đoạn 2, giai đoạn này gồm 2 mô hình học máy khác nhau để xác định rõ nếu tiêu cực hay tích cực thì cụ thể đó sẽ là yếu tố nào.



Hình 18. Phân bố nhãn trên tập dữ liệu training cho giai đoạn 1 (mô hình phân lớp nhị phân).



Hình 19. Phân bố nhãn trên tập dữ liệu traning cho giai đoạn 2 (mô hình thứ 1 – phân loại cảm xúc tiêu cực).



Hình 20. Phân bố nhãn trên tập dữ liệu training cho giai đoạn 2 (mô hình thứ 2 – phân loại cảm xúc tích cực hoặc không rõ bản chất).

Qua các thể hiện trên hình 18, 19 và 20, dữ liệu sau khi phân chia đã được cân bằng hơn. Đối với mô hình phân lớp nhị phân (giai đoạn 1), dữ liệu huấn luyện đã gần như cân bằng nhau. Đối với 2 mô hình phân lớp đa nhãn (giai đoạn 2), mặc dù dữ liệu huấn luyện trên cả vẫn còn tương đối mất cân bằng tuy nhiên số lượng nhãn đã được giảm xuống.

Mục đích cuối cùng cho việc phân chia quá trình này thành hai giai đoạn là để giảm thiểu tối đa ảnh hưởng của dữ liệu mất cân bằng và số nhãn mà các mô hình phải phân lớp. Với mong muốn nâng cao hiệu suất tổng thể của cả mô hình, ở từng giai đoạn, có rất nhiều mô hình học máy và học sâu khác nhau vì vậy chúng ta cần thử nhiều mô hình cặp mô hình khác nhau để tìm ra cặp tốt nhất.

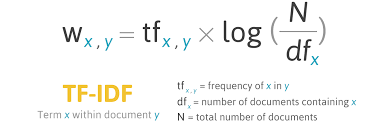
Ngoài ra, nếu chỉ xây dựng duy nhất mỗi pipeline này, chúng ta sẽ rất khó đánh giá rằng liệu pipeline này đang có hiệu suất tốt hoặc không. Vì vậy, để so sánh kết quả rõ ràng nhất chúng ta cần sử dụng thêm các mô hình baseline phân lớp đa lớp cho tất cả các nhãn.

### ***3.2 Trích xuất đặc trưng***

#### **a. TF-IDF**

Đối với các mô hình học máy đơn giản, chúng em sử dụng phương pháp trích xuất đặc trưng dựa vào thống kê TF-IDF. TF-IDF (Term Frequency-Inverse Document Frequency) là một phương pháp tính toán trọng số cho các từ trong một văn bản dựa trên tần suất xuất hiện của từ đó trong văn bản và trong toàn bộ tập dữ liệu. TF-IDF được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên và trích xuất thông tin văn bản. Trọng số TF-IDF cho một từ trong một văn bản được tính bằng cách kết hợp hai thành phần:

* **Term Frequency (TF)**: Đây là đo lường tần suất xuất hiện của từ trong văn bản. Giá trị TF cao cho biết từ đó xuất hiện nhiều trong văn bản.
* **Inverse Document Frequency (IDF)**: Đây là đo lường độ quan trọng của từ đó trong toàn bộ tập dữ liệu. Giá trị IDF cao cho biết từ đó xuất hiện ít trong các văn bản khác.



Hình 21. Công thức tính TF-IDF.

#### **b. PhoW2V**

PhoW2V (Phonetic Word2Vec) là một phương pháp nhúng từ dựa trên âm vị học được phát triển để xử lý văn bản tiếng Việt. Được xây dựng dựa trên kiến trúc Word2Vec, PhoW2V giúp biểu diễn từ vựng thành các vector số để mô hình có thể hiểu được ý nghĩa và mối quan hệ giữa các từ. PhoW2V sử dụng kỹ thuật tokenization âm vị học để chia nhỏ từ thành các âm tiết. Thay vì xử lý từ vựng bằng cách phân tách thành các từ riêng biệt, PhoW2V tách chúng thành các âm tiết. Điều này giúp bắt được những đặc điểm ngữ âm đặc trưng trong tiếng Việt và giữ được sự tương quan giữa các từ dựa trên phân tích âm vị học.

### ***3.3 Các mô hình học máy và học sâu sử dụng***

#### **a. Logistic Regression**

Logistic Regression là một thuật toán phân loại đa lớp được sử dụng rộng rãi trong học máy. Mặc dù ban đầu được thiết kế cho bài toán phân loại nhị phân, nhưng Logistic Regression có thể mở rộng để xử lý phân loại đa lớp bằng cách sử dụng kỹ thuật "One-vs-Rest" hoặc "One-vs-One". Thuật toán này dựa trên mô hình hồi quy tuyến tính và sử dụng hàm softmax để tính xác suất thuộc vào từng lớp. Logistic Regression đơn giản và dễ hiểu, đặc biệt hiệu quả khi các lớp không phức tạp và tương đối tách biệt. Support Vector Machine là một thuật toán phân loại đa lớp mạnh mẽ trong học máy. SVM có thể được sử dụng để phân loại dữ liệu vào nhiều lớp thông qua các biến thể của thuật toán, bao gồm "One-vs-One" và "One-vs-Rest". SVM tạo ra các siêu phẳng hoặc siêu mặt phân cách giữa các lớp dữ liệu bằng cách tối đa hoá khoảng cách giữa các điểm gần nhất từ các lớp khác nhau. Thuật toán này có khả năng xử lý cả bài toán phân loại tuyến tính và phi tuyến tính.

#### **b. Support Vector Machine**

Support Vector Machine là một thuật toán phân loại đa lớp mạnh mẽ trong học máy. SVM có thể được sử dụng để phân loại dữ liệu vào nhiều lớp thông qua các biến thể của thuật toán, bao gồm "One-vs-One" và "One-vs-Rest". SVM tạo ra các siêu phẳng hoặc siêu mặt phân cách giữa các lớp dữ liệu bằng cách tối đa hoá khoảng cách giữa các điểm gần nhất từ các lớp khác nhau. Thuật toán này có khả năng xử lý cả bài toán phân loại tuyến tính và phi tuyến tính. Decision Tree là một thuật toán phân loại đa lớp phổ biến trong học máy. Nó dựa trên cấu trúc cây quyết định, trong đó mỗi nút đại diện cho một quyết định hoặc một thuộc tính để phân loại dữ liệu. Các quyết định được đưa ra tại mỗi nút dựa trên giá trị của thuộc tính và mục tiêu phân loại. Decision Tree có thể được sử dụng trực tiếp cho bài toán phân loại đa lớp và có khả năng xử lý cả dữ liệu rời rạc và liên tục.

#### **c. Decision Tree**

Decision Tree là một thuật toán phân loại đa lớp phổ biến trong học máy. Nó dựa trên cấu trúc cây quyết định, trong đó mỗi nút đại diện cho một quyết định hoặc một thuộc tính để phân loại dữ liệu. Các quyết định được đưa ra tại mỗi nút dựa trên giá trị của thuộc tính và mục tiêu phân loại. Decision Tree có thể được sử dụng trực tiếp cho bài toán phân loại đa lớp và có khả năng xử lý cả dữ liệu rời rạc và liên tục. Random Forest là một thuật toán phân loại đa lớp dựa trên việc kết hợp nhiều cây quyết định (Decision Trees). Thuật toán này tạo ra một tập hợp các cây quyết định ngẫu nhiên và sau đó kết hợp kết quả của các cây này để đưa ra dự đoán phân loại. Random Forest giúp giảm tình trạng overfitting và cung cấp độ chính xác cao hơn so với một cây quyết định đơn lẻ.

#### **d. Random Forest**

Random Forest là một thuật toán phân loại đa lớp dựa trên việc kết hợp nhiều cây quyết định (Decision Trees). Thuật toán này tạo ra một tập hợp các cây quyết định ngẫu nhiên và sau đó kết hợp kết quả của các cây này để đưa ra dự đoán phân loại. Random Forest giúp giảm tình trạng overfitting và cung cấp độ chính xác cao hơn so với một cây quyết định đơn lẻ.

#### **e. K Nearest Neighbors**

K Nearest Neighbor là một thuật toán phân loại đa lớp dựa trên việc so sánh khoảng cách giữa các điểm dữ liệu. KNN xác định lớp cho một điểm dữ liệu mới bằng cách xem xét lớp của K điểm gần nhất trong tập dữ liệu huấn luyện. Lớp của điểm mới được quyết định bởi đa số các điểm gần nhất. KNN phụ thuộc vào giá trị của K và các phép đo khoảng cách để xác định lớp phân loại. Naïve Bayes là một thuật toán phân loại đa lớp dựa trên nguyên tắc của định lý Bayes. Nó dựa trên giả định Naïve Bayes rằng các đặc trưng đầu vào độc lập với nhau. Thuật toán này tính toán xác suất của mỗi lớp dựa trên xác suất của các đặc trưng đầu vào và sử dụng định lý Bayes để đưa ra dự đoán phân loại.

#### **f. Naïve Bayes**

Naïve Bayes là một thuật toán phân loại đa lớp dựa trên nguyên tắc của định lý Bayes. Nó dựa trên giả định Naïve Bayes rằng các đặc trưng đầu vào độc lập với nhau. Thuật toán này tính toán xác suất của mỗi lớp dựa trên xác suất của các đặc trưng đầu vào và sử dụng định lý Bayes để đưa ra dự đoán phân loại.

#### **g. TextCNN**

TextCNN (Convolutional Neural Network for Text Classification) là một mô hình học sâu sử dụng trong xử lý ngôn ngữ tự nhiên để phân loại văn bản vào nhiều lớp khác nhau. Kiến trúc của TextCNN dựa trên sự kết hợp giữa các lớp tích chập và lớp pooling để học và tóm tắt thông tin từ văn bản. Khi áp dụng TextCNN cho một tác vụ phân loại đa lớp, đầu tiên, các từ trong văn bản được biểu diễn dưới dạng vectơ từ, thường được tạo bằng các phương pháp như Word2Vec hoặc GloVe. Các vectơ từ này đại diện cho ý nghĩa của từ dựa trên ngữ cảnh trong tập dữ liệu huấn luyện. Sau đó, các đặc trưng đã được tóm tắt sẽ được đưa vào các lớp kết nối đầy đủ (fully connected layers) để học các quy tắc phân loại cuối cùng. Các lớp kết nối đầy đủ sẽ sắp xếp các đặc trưng lại và áp dụng các hàm kích hoạt để đưa ra dự đoán cho từng lớp. TextCNN có khả năng học được các đặc trưng cấu trúc trong văn bản và xử lý hiệu quả các bài toán phân loại đa lớp. Điều này làm cho nó trở thành một lựa chọn phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên.

#### **h. LSTM (Long short-term Memory)**

LSTM là một kiến trúc mạng nơ-ron gia đình của mạng nơ-ron tái phát (Recurrent Neural Network - RNN), được sử dụng phổ biến trong các bài toán xử lý ngôn ngữ tự nhiên và dữ liệu chuỗi. LSTM được thiết kế để xử lý vấn đề biến mất đối tượng (vanishing gradient) trong RNN bằng cách sử dụng các cổng (gate) để kiểm soát việc truyền ngược và lưu trữ thông tin quan trọng trong quá trình chuỗi. LSTM có khả năng xử lý các chuỗi dài và giữ được thông tin quan trọng từ quá khứ để dự đoán phân loại cho tương lai.

# **CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM**

## **Cài đặt thực nghiệm**

Đối với bước tiền xử lý dữ liệu, chúng em sử dụng công cụ **VNCoreNLP** để tách từ, đây là một công cụ tách từ cho tiếng Việt mang lại hiệu suất cao.

Đối với bước trích xuất đặc trưng, chúng em sử dụng 2 phương pháp là TF-IDF và PhoW2V. Số lượng từ tối đa trong từ điển của TF-IDF là 5000 từ. Từ điển này bao gồm 5000 từ phổ biến nhất trong toàn bộ tập dữ liệu. Chúng em sử dụng phương pháp phân tích câu thành các từ riêng lẻ để tính tần suất và trọng số của từng từ. Đối với PhoW2V, chúng em sử dụng ma trận pretrained embedding với embedding size là 300 được embedding ở cấp độ từ (word).

Đối với các mô hình học máy, chúng em sẽ huấn luyện bằng thư viện *sklearn* và *tensorflow.* Các hyperparameters của các mô hình học máy được thiết lập như hình 22.

A screen shot of a computer code

Description automatically generated with low confidence

Hình 22. Thiết lập của các mô hình học máy.

Đối với các mô hình học sâu sẽ có thiết lập như sau:

* *TextCNN*: Gồm 2 lớp convolution 2D với filter\_size lần lượt là 2 và 3, sử dụng phương pháp khởi tạo ‘normal’, activation function là hàm ‘elu’. Sau đó, 2 lớp này được lần lượt qua các 2 lớp max pooling 2D khác nhau và được nối lại trước khi qua lớp fully connected để phân lớp.
* *LSTM*: Gồm 2 lớp LSTM với số lượng units là 64.

Về hàm mất mát, cả 2 mô hình học sâu này sử dụng Binary Crossentropy hoặc Categorical Crossentropy tùy thuộc vào mong muốn ở đó chúng ta sử dụng để phân lớp nhị phân hay đa lớp.

## **Đánh giá kết quả và so sánh**

### ***2.1 Mô hình phân lớp nhị phân (ở giai đoạn 1)***

Dưới đây là bảng 4 – là kết quả thực nghiệm của mô hình phân lớp nhị phân (ở giai đoạn 1). Chúng ta có thể thấy rằng Logistic Regression và SVM có độ chính xác cao nhất với lần lượt là 0,76 và 0,75.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Accuracy | F1-score | Precision | Recall |
| **Logistic Regression** | **0.76** | **0.77** | **0.76** | **0.78** |
| **SVM** | **0.75** | **0.76** | 0.75 | **0.77** |
| Decision Tree | 0.61 | 0.61 | 0.64 | 0.57 |
| Random Forest | 0.73 | 0.73 | **0.76** | 0.71 |
| kNN | 0.69 | 0.69 | 0.72 | 0.67 |
| Naïve Bayes | 0.73 | 0.73 | 0.75 | 0.72 |
| TextCNN | **0.75** | 0.75 | 0.75 | 0.75 |
| LSTM | 0.74 | 0.74 | 0.75 | 0.74 |

Bảng 4. Kết quả mô hình phân lớp nhị phân (giai đoạn 1).

### ***2.2 Mô hình phân lớp các nhãn mang tính tiêu cực (ở giai đoạn 2)***

Dưới đây là bảng 5 – là kết quả thực nghiệm của mô hình phân lớp các cảm xúc tiêu cực (giai đoạn 2). Chúng ta có thể thấy rằng LSTM và SVM có độ chính xác cao nhất với lần lượt là 0,69 và 0,67.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Accuracy | F1-score | Precision | Recall |
| Logistic Regression | 0.66 | 0.64 | 0.67 | 0.66 |
| **SVM** | **0.67** | **0.65** | **0.7** | **0.67** |
| Decision Tree | 0.56 | 0.57 | 0.57 | 0.56 |
| Random Forest | 0.65 | 0.62 | 0.64 | 0.65 |
| kNN | 0.63 | 0.62 | 0.62 | 0.63 |
| Naïve Bayes | 0.61 | 0.54 | 0.59 | 0.61 |
| TextCNN | 0.66 | 0.64 | 0.66 | 0.66 |
| **LSTM** | **0.69** | **0.67** | **0.69** | **0.69** |

Bảng 5. Kết quả mô hình phân lớp đa lớp mang tính chất tiêu cực (giai đoạn 2).

### ***2.3 Mô hình phân lớp các nhãn mang tính tích cực (ở giai đoạn 2)***

Dưới đây là bảng 6 – là kết quả thực nghiệm của mô hình phân lớp các cảm xúc tích cực (giai đoạn 2). Chúng ta có thể thấy rằng LSTM và TextCNN có độ chính xác cao nhất với lần lượt là 0,69 và 0,68.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Accuracy | F1-score | Precision | Recall |
| Logistic Regression | 0.65 | 0.63 | 0.65 | 0.65 |
| SVM | 0.67 | 0.65 | 0.68 | 0.67 |
| Decision Tree | 0.57 | 0.57 | 0.57 | 0.57 |
| Random Forest | 0.67 | 0.66 | 0.68 | 0.67 |
| kNN | 0.68 | 0.66 | **0.69** | **0.68** |
| Naïve Bayes | 0.64 | 0.59 | 0.58 | 0.64 |
| TextCNN | **0.68** | **0.67** | **0.70** | **0.68** |
| LSTM | **0.69** | **0.68** | **0.69** | **0.69** |

Bảng 6. Kết quả mô hình phân lớp đa lớp mang tính chất tích cực (giai đoạn 2).

### ***Kết quả của pipeline dự đoán***

Để xây dựng mô hình pipeline ống nước này, chúng em sẽ chọn ra 2 mô hình có kết quả cao nhất của từng loại và đánh giá kết quả đầu ra. Ở đây, các mô hình được chọn ra như sau:

* Models 1 (phân lớp nhị phân – giai đoạn 1): Logistic Regression và SVM.
* Models 2 (phân lớp đa lớp tiêu cực – giai đoạn 2): LSTM và SVM.
* Models 3 (phân lớp đa lớp tích cực – giai đoạn 2): LSTM và TextCNN.

Sau khi chọn ra các mô hình tốt nhất ở từng giai đoạn, chúng em sẽ sử dụng pipeline này để dự đoán và đánh giá hiệu suất trên kết quả đầu. Kết quả đánh giá của pipeline được thể hiện qua bảng 7. Chúng ta có thể thấy rằng, pipeline đạt kết quả cao nhất đến từ pipeline đầu tiên SVM + SVM +TextCNN, cho độ chính xác là 0,55.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pipeline | Accuracy | F1-score | Precision | Recall |
| **SVM + SVM + TextCNN** | **0.55** | **0.54** | **0.56** | **0.55** |
| SVM + SVM + LSTM | 0.53 | 0.53 | 0.54 | 0.53 |
| SVM + LSTM + TextCNN | 0.54 | 0.53 | 0.54 | 0.54 |
| SVM + LSTM + LSTM | 0.53 | 0.52 | 0.53 | 0.53 |
| LR + SVM + TextCNN | 0.53 | 0.53 | 0.55 | 0.53 |
| LR + SVM + LSTM | 0.52 | 0.51 | 0.53 | 0.52 |
| LR + LSTM + TextCNN | 0.53 | 0.52 | 0.53 | 0.53 |
| LR + LSTM + LSTM | 0.51 | 0.51 | 0.51 | 0.51 |

Bảng 7. Kết quả của pipeline.

### ***2.5 So sánh với các mô hình baseline***

Để so sánh kết quả của pipeline bên trên thì chúng ta cần các mô hình baseline tương ứng để so sánh. Dưới đây là bảng 8 – là kết quả thực nghiệm của các mô hình phân lớp đa lớp baseline (gồm 7 nhãn cảm xúc) được thực nghiệm để so sánh với pipeline đã xây dựng bên trên. Mô hình đạt kết quả cao nhất là mô hình LSTM, đạt được độ chính xác là 0,54.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Accuracy | F1-score | Precision | Recall |
| Logistic Regression | 0.53 | 0.52 | **0.56** | 0.53 |
| SVM | **0.54** | 0.53 | 0.58 | 0.54 |
| Decision Tree | 0.38 | 0.38 | 0.38 | 0.38 |
| Random Forest | 0.49 | 0.48 | 0.52 | 0.49 |
| kNN | 0.48 | 0.47 | 0.49 | 0.48 |
| Naïve Bayes | 0.46 | 0.39 | 0.41 | 0.46 |
| TextCNN | **0.54** | **0.54** | 0.54 | **0.54** |
| LSTM | **0.54** | **0.54** | **0.56** | **0.54** |

Bảng 8. Kết quả mô hình phân lớp đa lớp mang tính chất tích cực (giai đoạn 2).

Theo dữ liệu từ bảng 7 và bảng 8, chúng ta có một số nhận xét như sau:

* Kết quả của pipeline xây dựng có đôi phần nhỉnh hơn các mô hình baseline đôi chút. Pipeline đạt hiệu suất cao nhất nhỉnh hơn 1% về độ chính xác và 1% về recall. Mặc dù kết quả này vẫn còn chưa thật sự chênh lệch rõ rệt có lẽ một phần là do nhãn của bộ dữ liệu quá nhiều cũng gây khó khăn cho việc dự đoán.
* Pipeline có kết quả cao nhất là **SVM + SVM + TextCNN** với độ chính xác 55%.
* Việc tách từ giúp cho các mô hình được huấn luyện một cách nhanh chóng.

# **CHƯƠNG V: KẾT LUẬN**

Phân tích cảm xúc (Sentiment Analysis) là một công cụ hỗ trợ doanh nghiệp và tổ chức có được những hiểu biết sâu sắc về những gì khách hàng đang nghĩ về sản phẩm, dịch vụ hoặc thương hiệu. Bằng cách phân tích lượng lớn dữ liệu văn bản, doanh nghiệp có thể nắm bắt được những thông tin ‘ẩn sâu’ trong lời nói của khách hàng và hỗ trợ đưa ra quyết định dựa trên “dữ liệu định hướng”.

Trong báo cáo này, chúng em đã xây dựng được pipeline sử dụng 2 giai đoạn khác nhau để dự đoán cảm xúc theo 7 nhãn khác nhau. Để giảm thiểu ảnh hưởng của dữ liệu không đồng đều nhãn, chúng ta đã phân chia 7 nhãn cảm xúc thành hai nhóm: tiêu cực và tích cực. Giai đoạn 1 tập trung vào phân lớp nhị phân trên dữ liệu đã được phân chia, trong khi giai đoạn 2 sử dụng hai mô hình học máy khác nhau để xác định mức độ tích cực hoặc tiêu cực của câu bình luận. Pipeline có kết quả cao nhất đạt độ chính xác là 55%. Tuy nhiên, cần lưu ý rằng việc dự đoán cảm xúc từ dữ liệu văn bản là một thách thức phức tạp, do sự đa dạng và sự tương đối của cảm xúc. Việc nâng cao độ chính xác của mô hình cần sự cải tiến trong việc tiền xử lý dữ liệu, trích xuất đặc trưng và chọn lựa mô hình phân loại phù hợp.

Mặc dù pipeline đề xuất có nhỉnh hơn mô hình baseline truyền thông tuy nhiên mức độ chênh lệch này chưa cao như mong đợi, nhưng qua bài báo cáo này chúng em đã tìm ra được một hướng đi khác có thể cải thiện hiệu suất cho tổng thể của mô hình.

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | P. T. Q. Điểm, VinBigData, [Online]. Available: https://vinbigdata.com/chatbot/ky-thuat-phan-tich-quan-diem-sentiment-analysis.html. |
| [2] | "Phân tích phản hồi khách hàng hiệu quả với Machine learning(Vietnamese Sentiment Analysis)," [Online]. Available: https://viblo.asia/p/phan-tich-phan-hoi-khach-hang-hieu-qua-voi-machine-learningvietnamese-sentiment-analysis-Eb85opXOK2G. |
| [3] | Việt, "Tổng quan về Phân tích cảm xúc trong Tiếng," [Online]. Available: https://hoctructuyen123.net/tong-quan-ve-phan-tich-cam-xuc-trong-tieng-viet/. |
| [4] | "Phân Tích Cảm Xúc Tiếng Việt," [Online]. Available: https://streetcodevn.com/blog/sav. |
| [5] | A. V. Ho, N. H. C. Duong, N. H. Danh, P. V. T. Linh, N. V. Duc, N. V. Kiet and N. L. T. Ngan, "Emotion Recognition for Vietnamese Social Media Text," 2020. |
| [6] | N. Q. Dat, "PhoW2V: Pre-trained Word2Vec syllable and word embeddings for Vietnamese," 2021. [Online]. Available: https://github.com/datquocnguyen/PhoW2V. |
| [7] | A. T. Nguyen, M. H. Dao and D. Q. Nguyen, "A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese," 2020. |