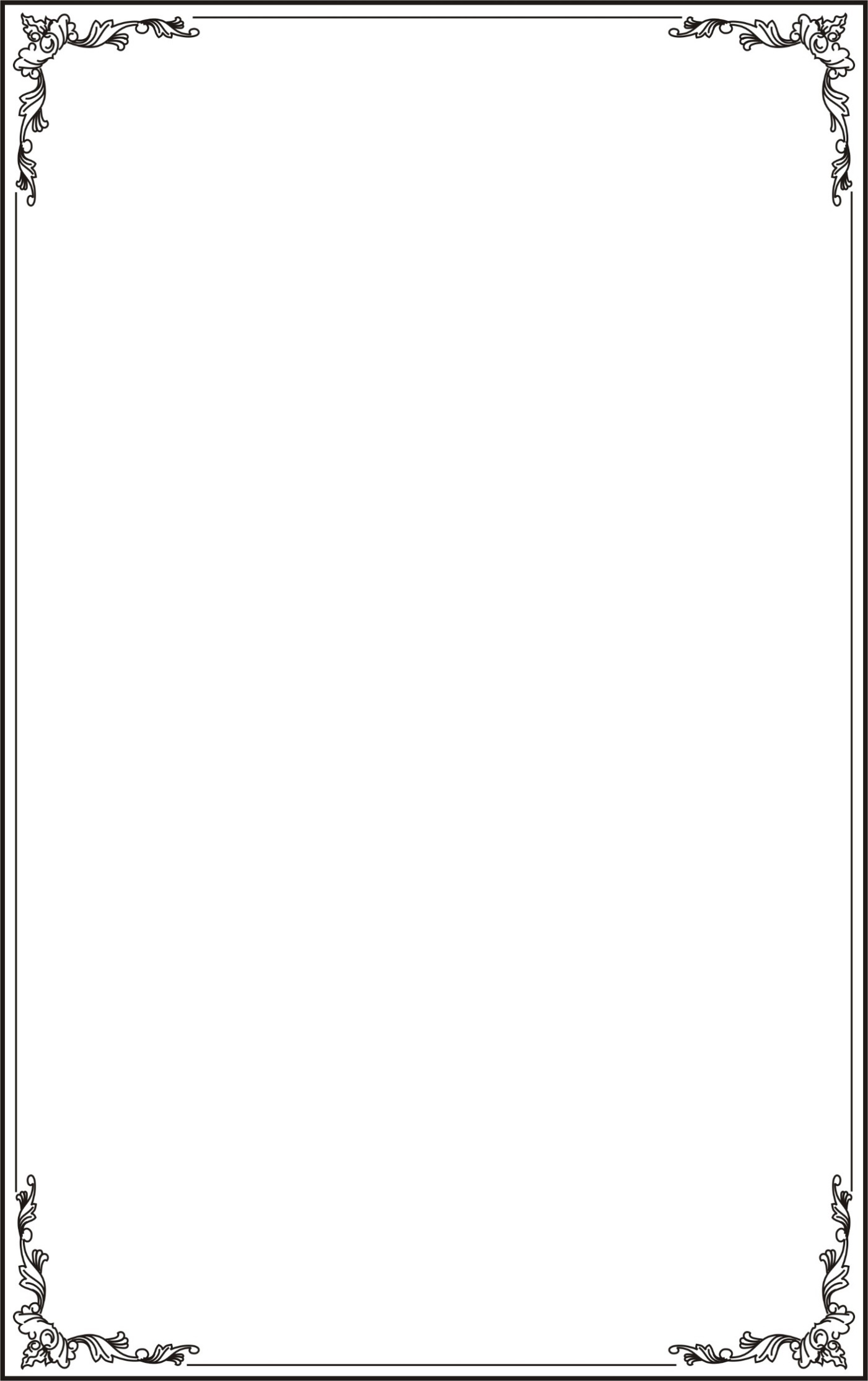
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆNCÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO PROJECT 1**

**CRAWL DỮ LIỆU VÀ TÁCH TỪ**

**Giáo viên hướng dẫn : TS. Nguyễn Kiêm Hiếu**

**Sinh viên thực hiện : Nguyễn Lưu Nhật**

**MSSV : 20163041**

**Hà Nội, ngày 07 tháng 05 năm 2019**

**­­**

**Phần mở đầu :**

Với sự phát triển nhanh chóng của công nghệ thông tin, nguồn thông tin trực tuyến (online) dưới dạng văn bản xuất hiện càng ngày càng nhiều. Việc khám phá tri thức tiềm ẩn từ kho dữ liệu văn bản là cần thiết cho việc quản lý, khai thác hiệu quả nguồn thông tin văn bản khổng lồ này

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Và bước đầu tiên và không thể thiếu trong việc xử lý ngôn ngữ tự nhiên là **tiền xử lý**. Tách từ là một bước quan trọng không thể thiếu trong tiền xử lý, nhằm xác định được ranh giới các từ có trong văn bản.

Trong bài tập lớn này, em sẽ crawl data từ trang web <https://vnexpress.net/> , rồi dùng data đã thu thập được đem đi tách từ, và thống kê lại kết quả thu được.

Mục lục

[**I.** **Tổng quan về tách từ tiếng việt** 4](#_Toc8064457)

[**1.** **Về tách từ tiếng việt** 4](#_Toc8064458)

[**2.** **Tách từ tiếng việt dựa trên từ điển** 4](#_Toc8064459)

[**II.** **Crawl data** 6](#_Toc8064460)

[**1.** **Crawler là gì?** 6](#_Toc8064461)

[**2.** **Crawl dữ liệu với Scrapy** 7](#_Toc8064466)

[**a) Giới thiệu** 7](#_Toc8064467)

[**b) Cách thức hoạt động** 7](#_Toc8064469)

[**3 . Chi tiết về cách crawl với scrapy** 9](#_Toc8064478)

[**III .** **Tách từ tiếng việt dựa trên từ điển với phương pháp MaximumMatching** 12](#_Toc8064480)

[**1.Từ điển tiếng Việt** 12](#_Toc8064481)

[**2. Phương pháp MaximumMatching** 12](#_Toc8064482)

[**a) Yêu cầu** 12](#_Toc8064483)

[**b) Tư tưởng** 12](#_Toc8064484)

[**c) Mã giả** 13](#_Toc8064485)

[**IV . Kết quả thu được** 14](#_Toc8064486)

[**1.** **Crawl data** 14](#_Toc8064487)

[**2.** **Tách từ bằng phương pháp MaximumMatching** 14](#_Toc8064488)

[**3.** **Thống kê kết quả** 14](#_Toc8064489)

[**4.** **Độ chính xác của phương pháp Maximum Matching** 16](#_Toc8064490)

[**5.** **Đánh giá phương pháp Maximum Matching** 16](#_Toc8064492)

[**a) Ưu điểm** 16](#_Toc8064493)

[**b) Nhược điểm** 16](#_Toc8064494)

[**V. Kết luận và hướng phát triển** 17](#_Toc8064495)

[**VI . Tài liệu tham khảo** 17](#_Toc8064496)

1. **Tổng quan về tách từ tiếng việt**
2. **Về tách từ tiếng việt**

Tách từ là một quá trình xử lý nhằm mục đích xác định ranh giới của các từ trong câu văn, cũng có thể hiểu đơn giản rằng tách từ là quá trình xác định các từ đơn, từ ghép… có trong câu. Đối với xử lý ngôn ngữ, để có thể xác định cấu trúc ngữ pháp của câu, xác định từ loại của một từ trong câu, yêu cầu nhất thiết đặt ra là phải xác định được đâu là từ trong câu. Vấn đề này tưởng chừng đơn giản với con người nhưng đối với máy tính, đây là bài toán rất khó giải quyết.

Chính vì lý do đó tách từ được xem là bước xử lý quan trọng đối với các hệ thống Xử Lý Ngôn Ngữ Tự Nhiên, đặc biệt là đối với các ngôn ngữ thuộc vùng Đông Á theo loại hình ngôn ngữ đơn lập, ví dụ: tiếng Trung Quốc, tiếng Nhật, tiếng Thái, và tiếng Việt. Với các ngôn ngữ thuộc loại hình này, ranh giới từ không chỉ đơn giản là những khoảng trắng như trong các ngôn ngữ thuộc loại hình hòa kết như tiếng Anh…, mà có sự liên hệ chặt chẽ giữa các tiếng với nhau, một từ có thể cấu tạo bởi một hoặc nhiều tiếng. Vì vậy đối với các ngôn ngữ thuộc vùng Đông Á, vấn đề của bài toán tách từ là khử được sự nhập nhằng trong ranh giới từ.

Những âm tiết được kết hợp để tạo thành các từ khác nhau tùy thuộc vào ngữ cảnh của văn bản. Để nhận dạng đúng ranh giới của các từ (tách từ) phục vụ cho các bài toán phân tích dữ liệu văn bản nhƣ: gom nhóm, phân lớp văn bản, các nhà khoa học đã đề xuất nhiều phƣơng pháp tách từ. Dựa trên đặc điểm của ―từ, kết hợp với cách tiếp cận khác nhau, các phương pháp tách từ này có thể chia thành ba nhóm chính: dựa trên từ điển (dictionary-based), dựa trên thống kê (statistic-based) và phương pháp lai (hybrid).

1. **Tách từ tiếng việt dựa trên từ điển**

Hai phương pháp kinh điển của hướng tiếp cận dựa trên từ điển là: Longest Matching (LM) và Maximal Matching (MM).

Phương pháp LM duyệt câu đầu vào tuần tự từ trái qua phải và chọn từ dài nhất nếu từ đó có trong từ điển. Rõ ràng là phương pháp này rất đơn giản nhưng bị phân lớp sai trong nhiều trường hợp nhập nhằng. Ví dụ câu “Đó là cách để truyền thông tin”, nếu áp dụng phương pháp LM thì câu này sẽ bị phân tách sai thành “Đó là cách để truyền\_thông tin”.

Phương pháp MM sẽ tạo ra tất cả các phân đoạn có thể cho một câu bất kỳ, sau đó câu 9 được phân đoạn đúng được chọn là câu chứa ít từ nhất. Giống như phương pháp trên phương pháp này cũng có yếu điểm là không thể đưa ra phân đoạn đúng trong trường hợp nhiều kết quả phân đoạn lại chứa cùng một số lượng từ ít nhất. Ví dụ câu “Học sinh học sinh học” có hai ứng cử là “Học\_sinh học sinh\_học” và “Học sinh\_học sinh\_học”. Trong những trường hợp này, ta cần áp dụng các phương pháp học máy trên một cơ sở dữ liệu lớn để xác định được phân đoạn đúng.

Hiệu quả của các phương pháp dựa trên từ điển như Longest Matching và Maximal Matching phụ thuộc phần lớn vào độ bao phủ của từ điển. Tuy nhiên, trên thực tế không tồn tại một từ điển hoàn thiện nào có khả năng bao phủ hết các mục từ của một ngôn ngữ bất kỳ bởi vì những từ mới luôn luôn xuất hiện.

# **II. Crawl data**

## **Crawler là gì?**

## Crawler là 1 từ để ám chỉ các công cụ (phần mềm, modules, plugins… hay đơn giản chỉ là 1 funtion nho nhỏ) có chức năng chính là tự động phân tích dữ liệu từ nguồn nội dung sau đó bóc tách những thông tin cần thiết theo tiêu chí mà nó được lập trình viên hệ thống thiết lập.

## Lấy tin tự động hiểu đơn giản nghĩa là sau khi đã đầu tư và phát triển được 1 bot hay 1 tập đoàn bots (crawler) tương ứng với từng chức năng. Rồi bạn cho nó hoạt động, nếu hoạt động tốt nó sẽ lấy tất cả những thông tin bạn cần từ các nguồn thông tin sau đó xử lý lưu trữ lại cho mục đích của bạn (làm dữ liệu phân tích, đăng tải lên website, chia sẻ lên data center…) thì đó là lấy tin tự động.

## Ví dụ:

## Thu thập tất cả lời bài hát từ trang <https://www.nhaccuatui.com/>

* Thu thập hình ảnh từ các trang mạng xã hội như <https://www.facebook.com/>

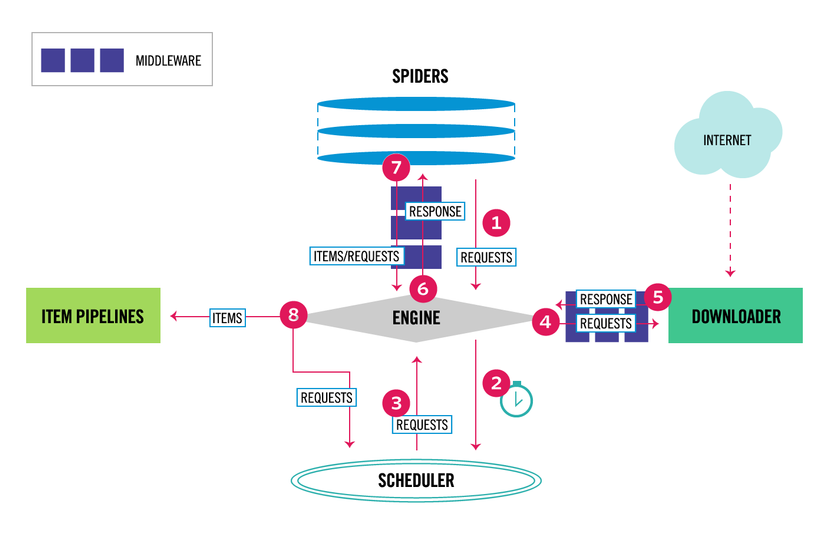
Trong bài tập lớn này, e sẽ sử dụng thư viện Scrapy trong python để crawl dữ liệu từ các bài báo ở trên trang VnExpress : <https://vnexpress.net/>

## **Crawl dữ liệu với Scrapy**

### **a) Giới thiệu**

## Scrapy là một khung thu thập dữ liệu web miễn phí và mã nguồn mở được viết bằng Python. Được thiết kế ban đầu để quét web, nó cũng có thể được sử dụng để trích xuất dữ liệu bằng API hoặc làm trình thu thập dữ liệu web có mục đích chung.

### **b) Cách thức hoạt động**



## ***i. Thành phần***

### ***Scrapy Engine*** *:* Scrapy Engine có trách nhiệm kiểm soát luồng dữ liệu giữa tất cả các thành phần của hệ thống và kích hoạt các sự kiện khi một số hành động xảy ra

### ***Scheduler*** : Giống như một hàng đợi (queue), scheduler sắp xếp thứ tự các URL cần download

### ***Dowloader*** : Thực hiện dowload trang web và cung cấp cho engine

### ***Spiders*** : là class được viết bởi người dùng, chúng có trách nhiệm bóc tách dữ liệu cần thiết và tạo các url mới để nạp lại cho scheduler qua engine.

### ***Item Pipeline*** : Những dữ liệu được bóc tách từ spiders sẽ đưa tới đây, Item pipeline có nhiệm vụ xử lý chúng và lưu vào cơ sở dữ liệu

### ***Các Middlewares*** : Là các thành phần nằm giữa Engine với các thành phần khác, chúng đều có mục địch là giúp người dùng có thể tùy biến, mở rổng khả năng xử lý cho các thành phần. VD: sau khi dowload xong url, bạn muốn tracking, gửi thông tin ngay lúc đó thì bạn có thể viết phần mở rộng và sửa lại cấu hình để sau khi Dowloader tải xong trang thì sẽ thực hiện việc tracking.

## ***ii. Luồng dữ liệu***

1. Khi bắt đầu crawl một website, Engine sẽ xác định tên miền và tìm vị trí của spider đó và yêu cầu spider đó tìm các urls đầu tiên để crawl
2. Engine nhận danh sách các urls đầu tiên từ spider, gửi cho Scheduler để sắp xếp
3. Engine yêu cầu danh sách cách urls tiếp theo từ Scheduler
4. Engine nhận danh sách các url tiếp theo từ Scheduler vào gửi đến Dowloader (requests)
5. Downloader nhận request và thực hiện việc tải trang, sau khi tải xong sẽ tạo một response và gửi lại Engine
6. Respone từ Dowloader sẽ được Engine đẩy qua Spiders để xử lý
7. Tại Spiders, khi nhận được response, chúng bóc tách thông tin từ response (tilte, content, author, date publish...) và những url có khả năng để crawl và đẩy lại cho Engine (requests)
8. Ở bước này, Engine nhận được kết quả từ Spiders sẽ thực hiện 2 công việc: đẩy những dữ liệu đã được bóc tách tới Item Pipeline để xử lý và lưu vào Databases, đẩy những url mới (requests) mới về Scheduler và quay về bước 3

## **3 . Chi tiết về cách crawl với scrapy**

Ví dụ, tạo một spider tên là stack\_spider.py  trong thư mục spiders. Thư mục này khá đặc biệt, bởi nó là nơi chúng ta đưa ra các chỉ định cho Scrapy biết chính xác chúng ta muốn thu thập dữ liệu gì.

Bắt đầu bằng một class kế thừa từ class Spider của Scrapy và chúng ta sẽ thêm vào các thuộc tính cần thiết.

from scrapy import Spider

class StackSpider(Spider):

name = "stack"

allowed\_domains = ["stackoverflow.com"]

start\_urls = [ "http://stackoverflow.com/questions?pagesize=50&sort=newest", ]

Những thuộc tính ở đây thể hiện ý nghĩa qua chính tên của mình.

* name định nghĩa tên của Spider.
* allowed\_domains chứa URL gốc của trang Web bạn muốn scrape.
* start\_urls là danh sách các URL để Spider bắt đầu quá trình scraping. Tất cả mọi dữ liệu sẽ được Spider download từ các URL ở trong start\_urls này.

## ***XPath selector***

Một điều rất quan trọng, đó là [Scrapy sử dụng XPath selector](http://doc.scrapy.org/en/latest/topics/selectors.html) để trích xuất dữ liệu từ các trang Web. Nói một cách khác, đó là chúng ta có thể chọn lọc ra một thành phần chính xác trên một trang Web bằng cách sử dụng [XPath](http://www.w3.org/TR/xpath/).

Ví dụ đối với trang Vnexpress thì dùng :

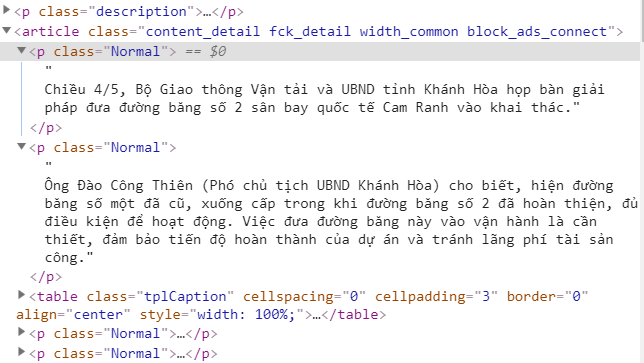
+) xpath('//\*[@class="sidebar\_1"]//\*[@class="title\_news" and not(i)]/a[1]/@href')

để lấy link của từng bài báo



+) xpath('/html/body/section[2]/section[1]/section[1]/p[1]/text()')

để lấy nội dung của các bài báo đó



Việc này sẽ đảm bảo dữ liệu lấy ra đã loại bỏ hầu hết các Html tag không quan trọng với quá trình tách từ và giúp quá trình crawl dữ liệu trở nên đơn giản hơn.

# **III . Tách từ tiếng việt dựa trên từ điển với phương pháp MaximumMatching**

## **1.Từ điển tiếng Việt**

* Khoảng 35.000 từ, với các thông tin hình thái, ngữ pháp, ngữ nghĩa dùng cho xử lý ngôn ngữ tự nhiên;
* Chứa đựng đủ lớp từ thông dụng trong tiếng Việt hiện đại;
* Mô hình từ điển theo chuẩn quốc tế nhằm sử dụng/trao đổi được giữa nhiều hệ thống, thuận tiện cho nghiên cứu đối sánh các ngôn ngữ, dễ cập nhật khi cần thiết;
* Mã hoá chuẩn XML.

## **2. Phương pháp MaximumMatching**

### **a) Yêu cầu**

- Từ điển

- Chuỗi đầu vào đã tách các dấu câu và âm tiết ( thường thì sử dụng Regex để xử lý )

### **b) Tư tưởng**

- Dựa trên thuật toán tham lam

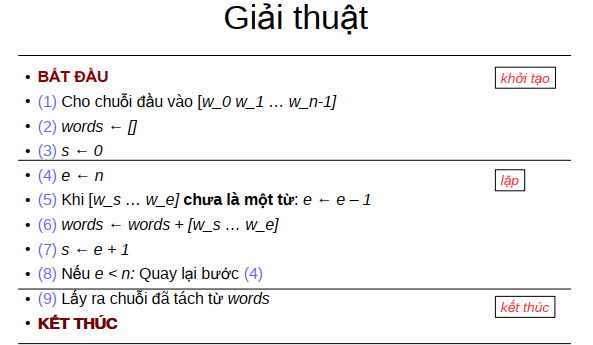
- Đi từ trái sáng phải hoặc từ phải sang trái, lấy các từ dài nhất có thể, dừng lại khi duyệt hết

- Độ phức tạp tính toán: O(n!.v)

N : số âm tiết trong chuỗi

V : số từ trong từ điển

### **c) Mã giả**



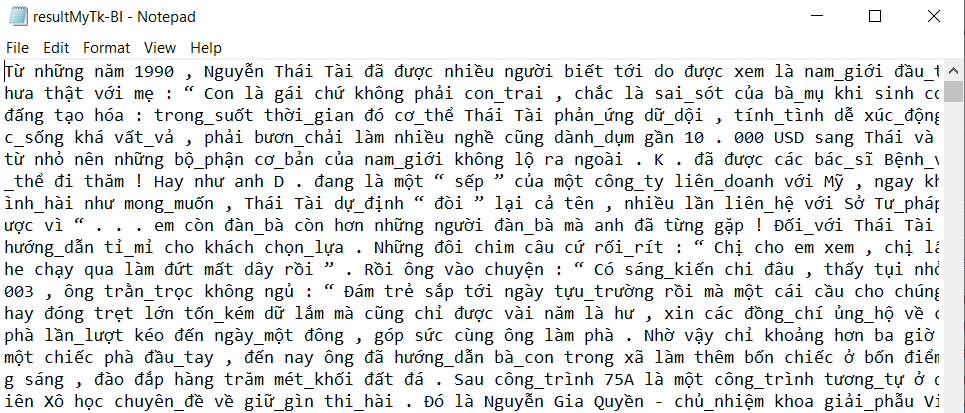
# **IV . Kết quả thu được**

1. **Crawl data**

Crawl được 355.000 bài báo từ trang VnExpress, mỗi bài báo ở trên 1 dòng, và gom lại vào file test2.txt, tổng dung lượng là 824 MB.

1. **Tách từ bằng phương pháp MaximumMatching**

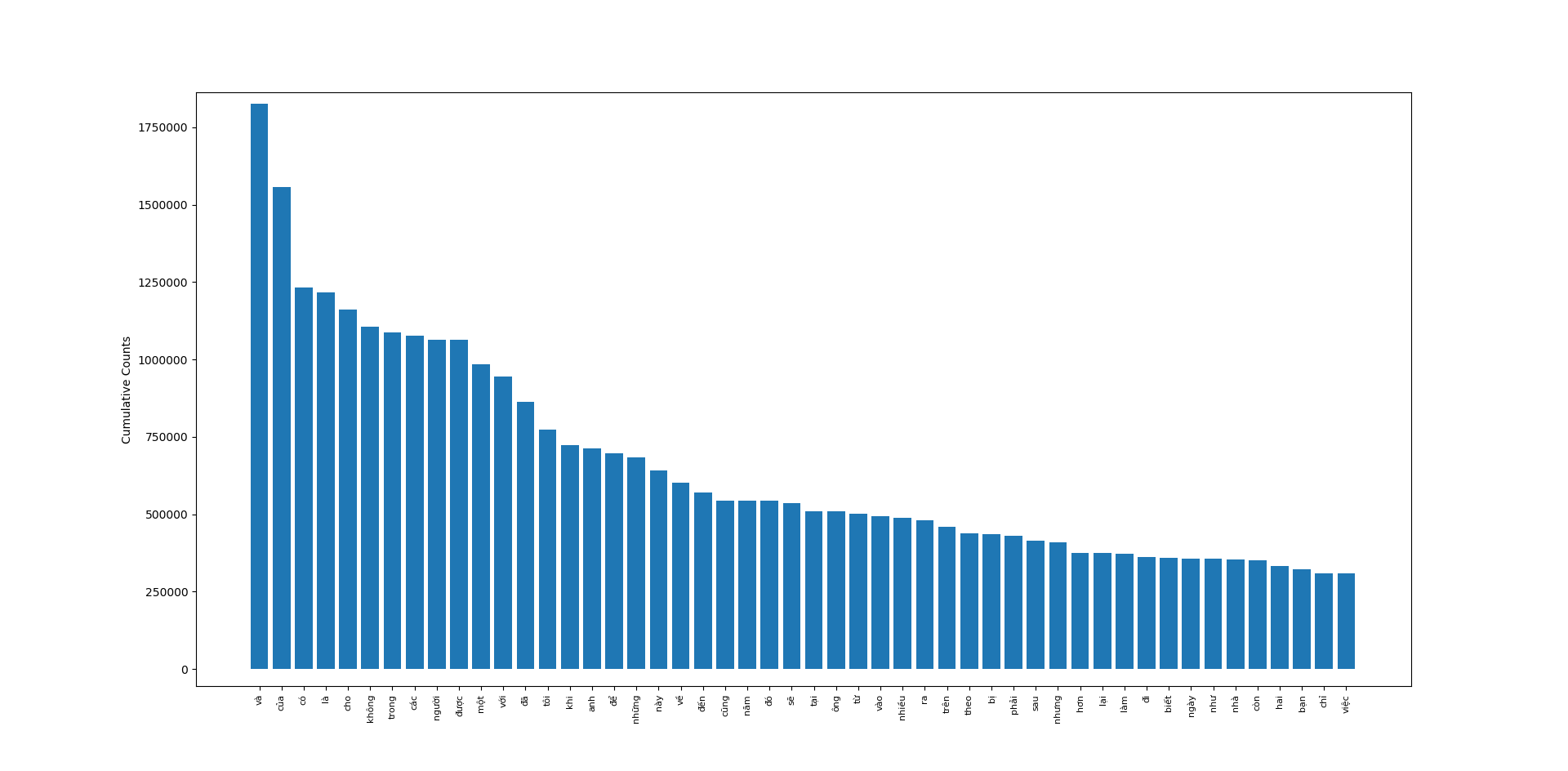
Thu được file result đã được tách từ



1. **Thống kê kết quả**

Dưới đây là biểu đồ top 50 từ xuất hiện nhiều nhất trong các bài báo trên trang VnExpress :

|  |  |
| --- | --- |
| Từ | Số lần xuất hiện |
| và | 1826778 |
| của | 1556629 |
| có | 1233729 |
| ……………… | ……………… |
| có\_thể | 278137 |



* Tổng số từ khác nhau thu được : 357473 từ
* Các từ đơn âm tiết xuất hiện nhiều nhất là : và, của , có, là , cho, không…

Đây hầu hết là các stop-word, là những từ xuất hiện với tần suất cực nhiều trong ngôn ngữ tự nhiên, tuy nhiên lại không mang nhiều ý nghĩa.

* Các từ đa âm tiết xuất hiện nhiều nhất là : có\_thể, gia đình, công\_ty, thời\_gian, chúng\_tôi, tuy\_nhiên…

Đây hầu như đều là các từ xuất hiện ít, mang ý nghĩa key-word trong việc xác định nội dung, í nghĩa . Tần suất xuất hiện của chúng thấp hơn rất nhiều lần so với top các từ đơn âm tiết xuất hiện nhiều nhất.

1. **Độ chính xác của phương pháp Maximum Matching**

## Kết quả khi test với Test-BI.txt là : 209383/222004 = 94.315%

1. **Đánh giá phương pháp Maximum Matching**

### **a) Ưu điểm**

- Cài đặt đơn giản

- Độ phức tạp tính toán hợp lý

### **b) Nhược điểm**

- Phụ thuộc vào từ điển

- Chưa giải quyết được vấn đề nhập nhằng

# **V. Kết luận và hướng phát triển**

Tách từ là bước tiền xử lý không thể thiếu trong xử lý ngôn ngữ tự nhiên, có nhiều phương pháp để tách từ trong Tiếng Việt, trong đó đơn giản nhất là tách từ bằng từ điển với phương pháp Maximum Matching, tuy nhiên phương pháp này có nhiều hạn chế.

Qua project lần này, em đã rút ra được nhiều kinh nghiệm về crawl dữ liệu, xử lý tách từ trong Tiếng Việt.

Theo dự định, em sẽ phát triển thêm ở phần thuật toán tách từ, áp dụng các phương pháp khác hiệu quả hơn như phương pháp học máy và phương pháp thống kê để tăng độ chính xác.

# **VI . Tài liệu tham khảo**

1. Slide Xử lý ngôn ngữ tự nhiên ( TS. Nguyễn Kiêm Hiếu -ĐH BK HN)
2. <https://www.wikipedia.org/>
3. <https://kipalog.com/posts/Gioi-thieu-tien-xu-ly-trong-xu-ly-ngon-ngu-tu-nhien>
4. <https://viblo.asia/p/gioi-thieuhuong-dan-ve-crawler-voi-scrapy-framework-ByEZkWoEZQ0>
5. <https://vlsp.hpda.vn/demo/?page=resources&lang=en>