**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

─────── \* ───────



**TÀI LIỆU BÁO CÁO**

**Môn học:** **Project III**

**Đề tài:**

Tóm tắt văn bản Tiếng Việt sử dụng Machine Learning

Sinh viên: Đàm Trọng Tuyên

Mã SV: 20173463

GVHD: Th.S. GVC. Nguyễn Hồng Phương

Hà Nội, tháng 12 năm 2020

Nội dung

[1. Giới thiệu đề tài 2](#_Toc58788628)

[1.1 Tóm tắt văn bản là gì? 2](#_Toc58788629)

[1.2 Ý nghĩa của việc tóm tắt văn bản 2](#_Toc58788630)

[1.3 Các hướng tiếp cận 2](#_Toc58788631)

[1.4 Các phương pháp đánh giá 3](#_Toc58788632)

[2 Chi tiết thuật toán 3](#_Toc58788633)

[2.1 Thuật toán phân cụm Kmeans 3](#_Toc58788634)

[1.1 Thuật toán LSA (Latent Semantic Analysis) 4](#_Toc58788635)

[2.2 Thuật toán TextRank 5](#_Toc58788636)

[3 Kết quả cài đặt 6](#_Toc58788637)

[3.1 Cài đặt thuật toán 6](#_Toc58788638)

[3.2 Cài đặt giao diện web 6](#_Toc58788639)

[4 Chuẩn bị dữ liệu cho bài toán tóm tắt văn bản 7](#_Toc58788640)

[4.1 Về ý tưởng 7](#_Toc58788641)

[4.2 Phương pháp 7](#_Toc58788642)

[4.3 Kết quả 8](#_Toc58788643)

[5 Đánh giá, hướng phát triển 8](#_Toc58788644)

[5.1 Đánh giá 8](#_Toc58788645)

[5.2 Khó khăn 8](#_Toc58788646)

[5.3 Định hướng phát triển 9](#_Toc58788647)

[6 Tham khảo 9](#_Toc58788648)

# Giới thiệu đề tài

Trong thời đại dữ liệu bùng nổ hiện nay, mỗi phút, mỗi giây lại có một lượng thông tin khổng lồ được sinh ra. Thông tin có thể đến từ nhiều nguồn khác nhau: con người, các thiết bị IoT, các công ty thu thập dữ liệu… dưới nhiều hình thức khác nhau: dạng văn bản, dạng ảnh, dạng video… trong đó văn bản là một dạng dữ liệu bùng nổ nhanh chóng, đến từ các bài viết trên mạng xã hội Facebook, trên các trang thương mại điện tử, các bài báo và tạp chí trên Internet. Việc lưu trữ và tìm kiếm văn bản ngày càng gặp khó khăn, với một lượng văn bản lớn như vậy chúng ta phải lưu trữ làm sao, thiết kế thuật toán tìm kiếm làm sao để có thể tìm kiếm với thời gian nhanh nhất và chính xác nhất. Không những vậy, làm sao để chắt lọc ra những thông tin đặc trưng, quan trọng nhất từ một văn bản điều đó giúp người đọc tiết kiệm thời gian, tập trung vào những gì quan trọng nhất. Các thuật toán tóm tắt văn bản – Text Summarization ra đời nhằm giải quyết vấn đề đó.

## Tóm tắt văn bản là gì?

Tóm tắt văn bản là rút gọn văn bản thành một văn bản có độ dài ngắn hơn nhưng vẫn giữ được các nội dung chính và ý nghĩa của văn bản.

## Ý nghĩa của việc tóm tắt văn bản

* Giúp phân loại văn bản với văn bản có kích thước chênh lệch quá nhiều (ví dụ độ dài của văn bản dao động từ 5-100 câu 🡺 rút gọn các văn bản quá dài về 5-10 câu) hay văn bản quá dài🡺 chỉ lấy ý chính phân loại.
* Tóm tắt tin tức 🡺giới thiệu cho người đọc những gì quan trọng và “tinh túy” nhất từ bài viết.
* Tạo headline tự động: sử dụng trong các hệ thống tổng hợp tin tức tự động, tự tạo ra tiêu đề cho bài viết đầu vào.
* Ứng dụng cho các bài toán tìm kiếm: khi người dùng tìm kiếm 1 cụm từ thì chỉ tìm kiếm trong dữ liệu đã được tóm tắt, vì người dùng thường tìm kiếm những thứ có ý nghĩa và quan trọng.

## Các hướng tiếp cận

* Extraction based: tư tưởng của các thuật toán dựa trên trích dẫn là chọn ra top những câu có độ quan trọng cao nhất, mang ý nghĩa tổng quan cho toàn thể văn bản đầu vào
* Abtraction based: phương pháp này tiếp cận dựa trên sự phát triển của các thuật toán Deep Learning, điển hình là các mô hình sinh – Generative Model. Mục tiêu của thuật toán là tự động sinh ra văn bản tóm tắt dựa trên văn bản đầu vào.

## Các phương pháp đánh giá

* Đánh giá về cú pháp, ngữ nghĩa của văn bản tóm tắt: đối với hướng tiếp cận theo mô hình sinh, cần đánh giá xem văn bản tóm tắt có nghĩa không, đúng cú pháp, ngữ pháp hay không…
* Đánh giá về ngữ nghĩa: cần đánh giá xem văn bản tóm tắt có giữ được ý chính, những chủ đề quan trọng nhất của văn bản đầu vào. Về phương diện ngữ nghĩa, ta chia phương pháp đánh giá làm 2 loại:
  + Dữ liệu có nhãn- mỗi văn bản đã có 1 hoặc nhiều bài tóm tắt: một số phương pháp được sử dụng để đánh giá độ tương đồng ngữ nghĩa giữa đầu ra của thuật toán và văn bản được tóm tắt bằng tay như: N-gram Co-occurrence Statistics – ROUGE, Longest Common Subsequence hay Unit Overlap.
  + Dữ liệu không có nhãn: cách đánh giá trên dữ liệu này chủ yếu dựa trên độ tương đồng ngữ nghĩa giữa văn bản đầu vào và văn bản đầu ra của mô hình. Phương pháp content-based dựa trên độ tương đồng ngữ nghĩa trực tiếp giữa văn bản tóm tắt và văn bản được tóm tắt. Main Topic Similarity dựa trên chính mô hình chủ đề LSA, đánh giá xem liệu văn bản tóm tắt có giữ được những chủ đề chính của văn bản đầu vào.

# Chi tiết thuật toán

## Thuật toán phân cụm Kmeans



Mục tiêu thuật toán phân cụm

* Kmeans là một thuật toán học máy không giám sát – Unsupervised Learning. Đầu vào của thuật toán phân cụm nói chung là các điểm dữ liệu và đầu ra là các điểm với cụm tương ứng của nó sao cho các điểm thuộc cùng một cụm có nhiều mối quan hệ, liên kết với nhau hơn nhiều so với các điểm khác cụm.
* Với thuật toán Kmeans ta cần truyền thêm tham số là số cụm. Kmeans sử dụng Euclid để xác định sự giống nhau giữa hai điểm dữ liệu, hai điểm càng gần nhau thì độ tương tự giữa hai điểm này càng cao.
* Dựa trên ý tưởng như vậy, Kmeans sử dụng hàm lỗi là tổng khoảng cách từ 1 điểm tới tâm cụm tương ứng của nó và tối ưu dựa trên thuật toán lặp. Với ý tưởng đơn giản, dễ triển khai nhưng lại đem lại kết quả tốt trên nhiều bài toán. Hiện tại Kmeans vẫn luôn nằm trong top những thuật toán Machine Learning phổ biến nhất.

(Chi tiết thuật toán Kmeans có thể xem tại: [machinelearningcoban.com](https://machinelearningcoban.com/2017/01/01/kmeans/))

## Thuật toán LSA (Latent Semantic Analysis)

* LSA là một thuật toán thuộc mô hình chủ đề, phân loại topic. Ý tưởng của thuật toán dựa trên phép phân tích suy biến ma trận biểu diễn văn bản đầu vào (A) để xác định không gian tuyến tính con của A mà vẫn giữ được hầu hết những biến động của ma trận A. Nói cách khác, thuật toán này đi giảm chiều ma trận dữ liệu đầu vào sử dụng phép phân tích ma trận.
* Gỉa sử ma trận biểu diễn mối quan hệ giữa document-word là Amn∈Rm×n. Về cơ bản phép phân tích suy biến sẽ biến đổi ma trận gốc A thành tích của các ma trận suy biến trái U, ma trận đường chéo S và ma trận suy biến phải V theo công thức:

A picture containing diagram

Description automatically generated

Chi tiết thuật toán LSA

* Trong đó t là tham số số lượng topics được lựa chọn. Khi đó:
* Ma trận U∈Rm×t là ma trận document-topic.
* Ma trận V∈Rn×t là ma trận term-topic.
* Mỗi dòng của U sẽ biểu diễn phân phối của văn bản và mỗi dòng của V là phân phối của mỗi từ theo t topics ẩn.
* Các cột của U và V đại diện cho các topics.
* Để tìm ra 2 văn bản có giống nhau hoặc 2 từ giống nhau chúng ta sẽ tính tương quan của các véc tơ dòng tương ứng với các văn bản của ma trận U hoặc các véc tơ dòng tương ứng với từ của ma trận V.

Phép phân tích này giảm được khá nhiều chiều đối với trường hợp bộ văn bản lớn. Tuy nhiên nó vẫn tồn tại những nhược điểm đó là:

* Các tính toán hoàn toàn dựa trên phân tích ma trận.
* Các topic được xác định dựa trên các phép chiếu tuyến tính nên mối quan hệ giữa topic-document và topic-term là tuyến tính.
* Chi phí tinh toán của mô hình rất tốn kém nếu kích thước dữ liệu lớn

(Chi tiết thuật toán xem tại: [wikipedia](https://en.wikipedia.org/wiki/Latent_semantic_analysis))

## Thuật toán TextRank

* TextRank được bắt nguồn từ thuật toán PageRank, thuật toán nhằm xếp hạng các trang web của Google.
* Thuật toán này chỉ định giá trị nhất định cho mỗi thành phần của một tập hợp các văn bản liên kết với nhau, ví dụ như World Wide Web.
* Mục đích "đo" tầm quan trọng tương đối của các liên kết trong tập hợp đó.
* Áp dụng cho bất kỳ tập hợp văn bản nào có trích dẫn đối ứng và liên kết cụ thể.
* Giá trị (weight) mà nó gán cho bất kỳ thành phần E được gọi là PageRank của E và ký hiệu là PR(E).

Diagram

Description automatically generated

Thuật toán Google pagerank

Giá trị **Pagerank**hình thành từ thuật toán toán học dựa trên webgraph: các trang world wide web được coi như các đỉnh và các đường link là các cạnh. Khi hình thành webgraph người ta có tính đến những trang của các cơ quan có thẩm quyền như cnn.com hay usa.gov. Giá trị xếp hạng cho thấy tầm quan trọng của từng trang cụ thể. Mỗi đường link tới trang web sẽ được tính như 1 sự hỗ trợ làm tăng thêm giá trị Pagerank.  
  
Giá trị Pagerank của trang được định nghĩa đệ quy và phụ thuộc vào số lượng và giá trị của các trang mà có link dẫn đến trang đó (incoming links).Một trang web có chứa nhiều link liên kết từ các trang web có giá trị PageRank cao thì giá trị *PageRank*của trang đó cũng sẽ cao.Có rất nhiều bài viết đã được xuất bản ra công chúng dựa trên nghiên cứu gốc của Page và Brin.

Dựa trên ý tưởng này, thuật toán TextRank ra đời nhằm xếp hạng các câu trong một văn bản dựa trên độ tương tự giữa các câu. Mỗi đỉnh sẽ là một câu, cạnh nối là độ tương tự giữa hai câu. Từ đó xác định được các đỉnh với rank cao nhất-các câu quan trọng nhất.

# Kết quả cài đặt

## Cài đặt thuật toán

Với hướng tiếp cận extraction-based, với đơn vị extract là câu ta sử dụng 3 phương pháp mã hóa cho câu là CountVectorize, Tf-idf Vectorize và Word2Vec; Mỗi câu sẽ xem như một điểm dữ liệu đầu vào sử dụng cho mỗi thuật toán. Dữ liệu đầu vào là không có nhãn nên sẽ sử dụng phương pháp đánh giá đơn giản dựa trên nội dung-Content Based.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **vectorize method** | **summarize method** | **min** | **max** | **mean** | **std** |
| **0** | count | clustering | 0.549474 | 0.971858 | 0.828509 | 0.072546 |
| **1** | count | lsa | **0.635774** | **0.978254** | **0.863468** | 0.05693 |
| **2** | count | textrank | 0.606043 | 0.959311 | 0.843669 | 0.05943 |
| **3** | tfidf | clustering | 0 | 0.965379 | 0.809225 | 0.100389 |
| **4** | tfidf | lsa | 0.347293 | 0.961283 | 0.760018 | 0.092178 |
| **5** | tfidf | textrank | 0.59064 | 0.959311 | 0.842451 | 0.062583 |
| **6** | word2vec | clustering | 0.594531 | 0.96886 | 0.831716 | 0.067225 |
| **7** | word2vec | lsa | 0.635774 | 0.978254 | 0.861422 | 0.058301 |
| **8** | word2vec | textrank | 0.633165 | 0.976642 | 0.848594 | 0.06024 |

Kết quả cài đặt đánh giá trên bộ 1000 bài viết ([test set](https://github.com/trongtuyen99/viet_summarizer/tree/main/src/data)).

## Cài đặt giao diện web

Graphical user interface, application

Description automatically generated

Ô điền văn bản cần tóm tắt

Chọn thuật toán tóm tắt

Tổng quan giao diện web summarizer.

Graphical user interface, text, application

Description automatically generated

Danh sách các câu được chọn. Lượng thông tin giữ lại đánh giá theo content-based.

Văn bản sau khi tóm tắt

Giao diện tóm tắt văn bản

# Chuẩn bị dữ liệu cho bài toán tóm tắt văn bản

Việc sử dụng học sâu cho một bài toán đỏi hỏi một lượng dữ liệu rất lớn. Vì vậy việc thu thập, xử lý và lưu trữ dữ liệu là đặc biệt quan trọng. Do vấn đề về thời gian cũng như việc thiếu tài nguyên cho việc training mô hình nên hiện tại đề tài chỉ tập trung vào việc thu thập dữ liệu.

## Về ý tưởng

Hầu hết các trang báo hiện tại đều có một phần “sapo” cho bài báo tương ứng. Đoạn Sapo này có thể được viết bởi 1 hoặc nhiều câu văn hoàn chỉnh khác nhau. Những câu văn này có thể ngắn, có thể dài nhưng nó phải mang tính khái quát để người đọc hiểu được nội dung phần thông tin mà bạn cung cấp phía dưới ([Sapo Là Gì?](https://vietbaixuyenviet.com/sapo-la-gi/) ). Tuy phần sapo không phải cũng như không tóm tắt đầy đủ nội dung của bài viết nhưng đa số sapo thể hiện được nội dung của bài viết (trực tiếp hoặc gián tiếp). Do vậy ta có thể sử dụng dữ liệu nội dung bài viết – sapo của bài viết tương ứng để làm dữ liệu đầu vào cho bài toán tóm tắt văn bản. Với hy vọng mô hình có thể tự động tạo sapo cho bài viết dựa vào nội dung bài viết.

## Phương pháp

Đề tài tập trung crawl dữ liệu từ trang [soha.vn](https://soha.vn/) do số lượng bài viết lớn, uy tín và các bài viết đều có phần sapo tương ứng.

Về công cụ, Scrapy (python) là một công cụ rất mạnh với khả năng song song hóa cao, khả năng xử lý lỗi hiệu quả. Dựa trên cấu trúc cây XML, mỗi phần của bài viết (sapo, title, nội dung…) sẽ có một địa chỉ XPath tương ứng từ đó Scrapy sẽ giúp ta lấy được các thành phần này.

Đầu vào của Scrapy là danh sách các đường dẫn của bài viết nên việc thu thập đường dẫn cũng là một phần quan trọng. Để thuận tiện, ta sử dụng đường dẫn của các bài báo chứa ngay trong các bài đang crawl (sử dụng cấu trúc tập hợp để loại bỏ các đường dẫn đã crawl), để extract ra các đường dẫn này ta sử dụng biểu thức chính quy (regex). Phương pháp này giúp ta “duyệt theo chiều rộng” các bài viết trong cơ sở dữ liệu của trang web này với mỗi node là một bài viết tương ứng.

## Kết quả

Sau khoảng 12 tiếng crawl, với 1 trang web khởi tạo đã duyệt qua khoảng 300000 link, 196497 link thành công (các link không có nội dung hoặc sapo được coi là không thành công).

Chart, icon

Description automatically generated

Chi tiết top 10 thể loại có nhiều bài viết nhất

Chi tiết dữ liệu đã crawl: [link google drive](https://drive.google.com/file/d/1SuYzCb8kpNQRkM24nLdStGp1c9iwPBnQ/view?usp=sharing)

# Đánh giá, hướng phát triển

## Đánh giá

Nhìn chung, trong ba phương pháp tóm tắt văn bản sử dụng thuật toán machine learning ở trên ta đều cho được kết quả trong thời gian rất ngắn (4 mico second cho 1 văn bản khoàng 30 câu). Với thuật toán hiện tại phương pháp mã hóa tf-idf tỏ ra không phù hợp, dễ dàng nhận thấy một từ khi xuất hiện trong nhiều câu có thể nó chính là chủ đề chính, từ quan trọng chứ không phải stops word, từ không quan trọng.

Trong ba phương pháp trên LSA tỏ ra hiệu quả hơn cả, chất lượng văn bản tóm tắt cũng ổn định hơn. Nhưng phương pháp này tồn tại một vấn đề, là vấn đề chung của Extraction-based đó là “chọn bao nhiêu câu là đủ?”. Có thể cải tiến theo cách chọn sao cho trị riêng của ma trận phân tích không nhỏ hơn một ngưỡng nào đó so với giá trị lớn nhất của trị riêng.

## Khó khăn

Các khó khăn gặp phải với bài toán tóm tắt văn bản bằng cách trích dẫn:

* Số lượng câu trích dẫn là bao nhiêu
* Đâu là cách đánh giá tốt nhất cho văn bản đầu ra
* Cần cải tiến các thuật toán Machine learning sao cho phù hợp với bài toán

Các khó khăn nếu áp dụng phương pháp abtraction-based:

* Làm sao thu thập lượng lớn dữ liệu có nhãn
* Thuật toán nào là phù hợp
* Tài nguyên cho việc training model Deep Learning
* Làm sao để đánh giá ngữ pháp cũng như ngữ nghĩa văn bản đầu ra

## Định hướng phát triển

Các hướng phát triển của đề tài:

* Tối ưu model Machine Learning: sử dụng thuật toán phân cụm khác (ví dụ DBSCAN), chọn số lượng câu trích dẫn phù hợp với từng văn bản đầu vào.
* Sử dụng hàm tính độ tương tự khác cho thuật toán Text Rank
* Cài đặt các thuật toán Deep Learning, dựa trên các mô hình sinh, Auto Encoder. Cần thu thập, xử lý dữ liệu trước và tìm hiểu các phương pháp đánh giá mô hình.

# Tham khảo

* [Automatic Text Summarization with Machine Learning — An overview](https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25)
* [A Quick Introduction to Text Summarization in Machine Learning](https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f)
* [Text Summarization in Machine Learning](https://techinsight.com.vn/en/text-summarization-in-machine-learning/)
* [Evaluation Measures for Text Summarization](https://www.researchgate.net/publication/220106310_Evaluation_Measures_for_Text_Summarization)
* [Evaluation Measures For Text](http://www.cai.sk/ojs/index.php/cai/article/viewFile/37/24)
* [Summarization Evaluation: An Overview](http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings2/sum-mani.pdf)

Chi tiết source code xem tại: [github](https://github.com/trongtuyen99/viet_summarizer)