**ĐẠI HỌC THỦ DẦU MỘT**



**ĐỒ ÁN MÔN HỌC  
NHẬP MÔN LẬP TRÌNH KHOA HỌC DỮ LIỆU**

**XÂY DỰNG HỆ THỐNG TÌM KIẾM BẰNG THUẬT TOÁN TF-IDF**

Sinh viên thực hiện: **HOÀNG KIM TUYẾN**

Mã số sinh viên: **1824801040043**

Lớp**: D18HT01**

***Bình Dương, tháng 12 năm 2020***

1. **LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của ThS. Nguyễn Thế Bảo. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây.

Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình.** Trường Đại học Thủ Dầu Một không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*Bình Dương, ngày 04 tháng 12 năm 2020*

*Người thực hiện  
 (ký tên và ghi rõ họ tên)*

**MỤC LỤC**

[PHẦN 1 3](#_Toc57918703)

[1.1. Mục đích đề tài 3](#_Toc57918704)

[1.2. Câu hỏi nghiên cứu 4](#_Toc57918705)

[1.4. Thu thập dữ liệu 5](#_Toc57918706)

[1.5. Xây dựng thuật toán 6](#_Toc57918707)

[ Xử lí *StopWord* 6](#_Toc57918708)

[ TF-IDF 8](#_Toc57918709)

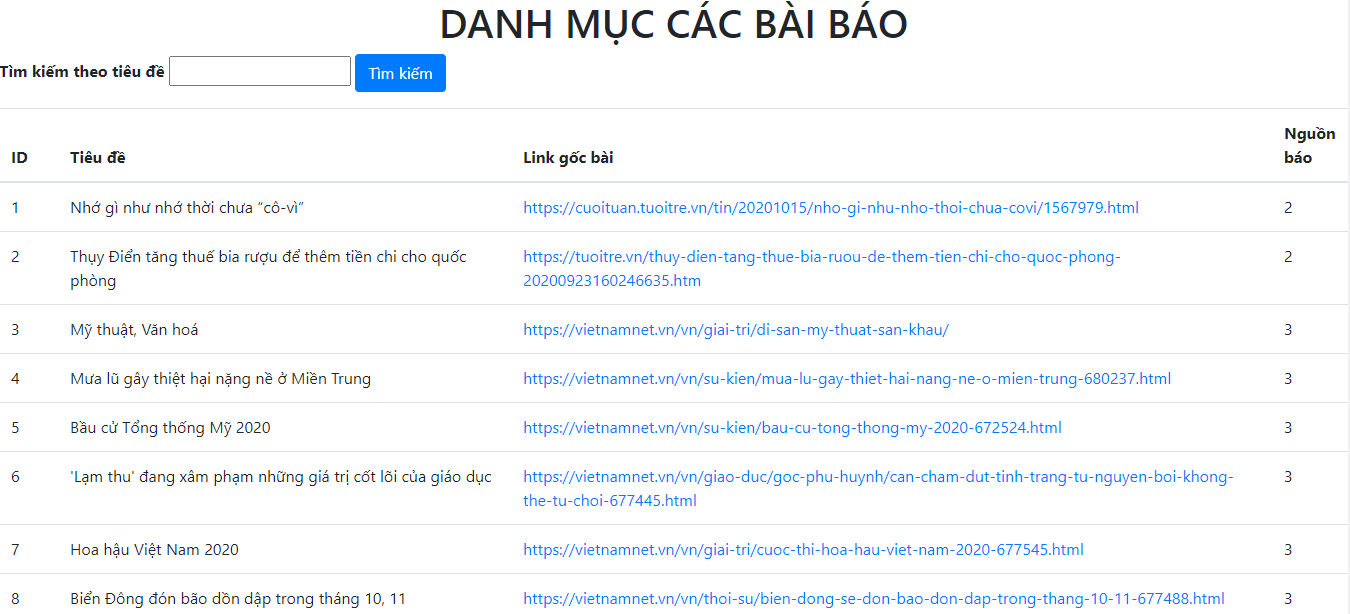
[ Tìm kiếm 11](#_Toc57918710)

[DANH MỤC TÀI LIỆU THAM KHẢO 13](#_Toc57918711)

# PHẦN 1

## 1.1. Mục đích đề tài

Xây dựng hệ thống tìm kiếm bằng thuật toán TF\_IDF



Giao diện tìm kiếm

## 1.2. Câu hỏi nghiên cứu

* Sử dụng thư viện trong Python để thu thập và lưu trữ thông tin trên các mặt báo lớn như baomoi.com, dantri.com.vn, vnexpress.net, . . .
* Các trường dữ liệu thu thập *thông tin tuyển dụng* gồm:
* Tiêu đề, mô tả
* Link bài viết
* Hình ảnh

## 1.4. Thu thập dữ liệu

Sử dụng các thư viện **newspaper**, phần mềm **sqlite3** để thu thập thông tin các thư viện **nltk** để xử lí ngôn ngữ tự nhiên.

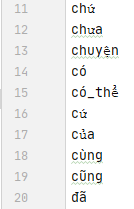
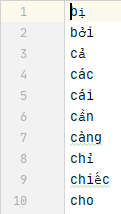
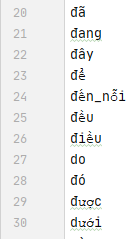
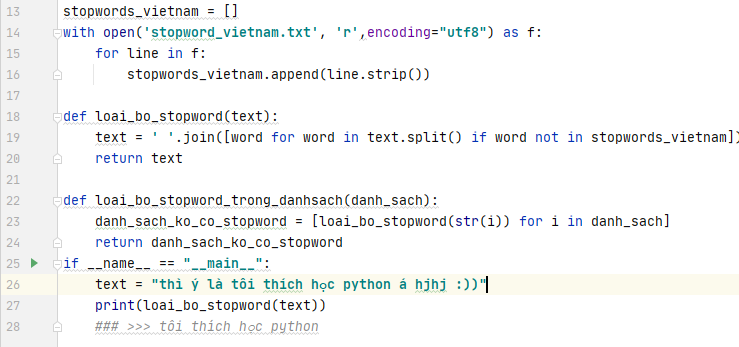
Tiến hành thu thập dữ liệu



Tiến hành thu thập dữ liệu

* Sử dụng các hàm trong thư viện **BeautifulSoup,** **newspaper**  để tìm ra các khối dữ liệu sau đó tiến hành xử lí, loại bỏ các khối *html*, *dấu cách*, . . . và lưu vào **database(sqlite3)**

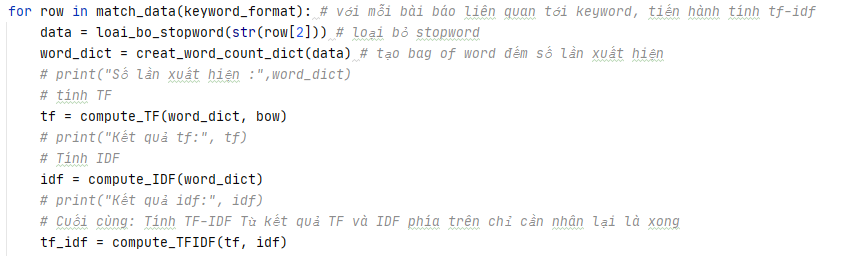
## 1.5. Xây dựng thuật toán

* Xử lí *StopWord*
* StopWords là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên, tuy nhiên lại không mang nhiều ý nghĩa. Ở tiếng việt StopWords là những từ như: để, này, kia... Tiếng anh là những từ như: is, that, this...

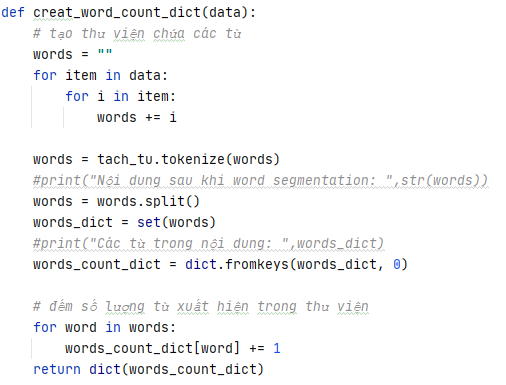
Xử lí, loại bỏ StopWord

Danh sách StopWord sưu tầm trên wikipedia

* Sau khi thu thập stopword, tiến hành loại bỏ chúng trong tiêu đề, mô tả của dữ liệu



Thực hiện loại bỏ stopword trong mô tả của bài báo(dòng 2)

* 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. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp.
* Bag of Words (Bow)
* Bag of Words là một thuật toán hỗ trợ xử lý ngôn ngữ tự nhiên và mục đích của BoW là phân loại text hay văn bản. Ý tưởng của BoW là phân tích và phân nhóm dựa theo “Bag of Words”(corpus). Với test data mới, tiến hành tìm ra số lần từng từ của test data xuất hiện trong "bag".

Tạo bag of word

* TF-IDF
* tf–idf, viết tắt của thuật ngữ tiếng Anh term frequency – inverse document frequency, của một từ là một con số thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.
* TF- term frequency – tần số xuất hiện của 1 từ trong 1 văn bản. Cách tính:

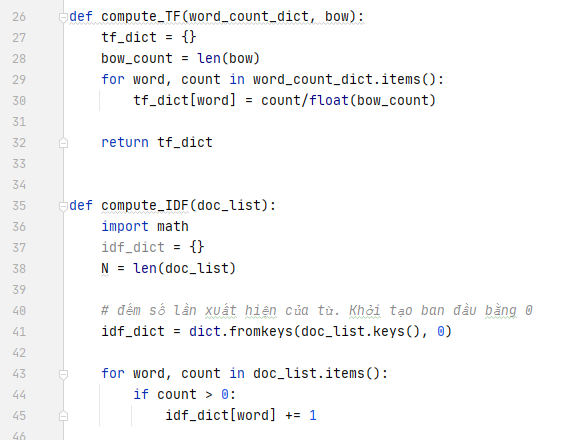


Trong đó:

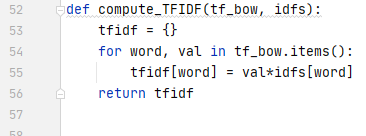
* f(t,d) - số lần xuất hiện từ t trong văn bản d
* max{f(w,d):w∈d} - số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.
* IDF – inverse document frequency. Tần số nghịch của 1 từ trong tập văn bản (corpus).
* Tính IDF để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản. Cách tính:



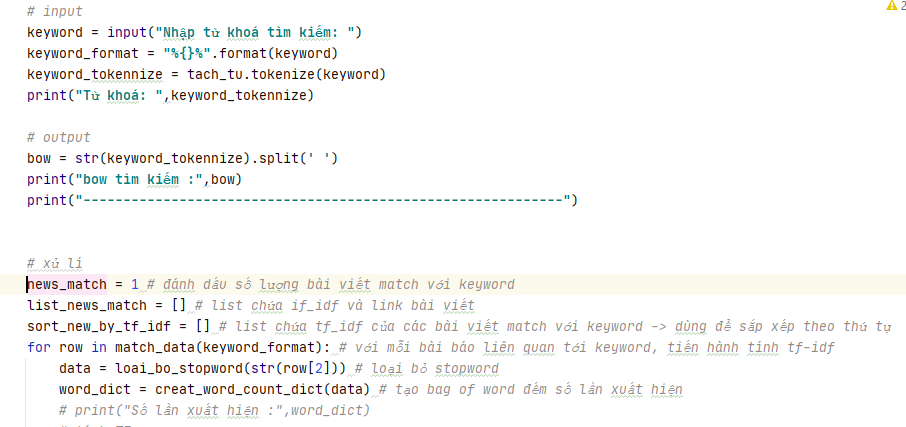
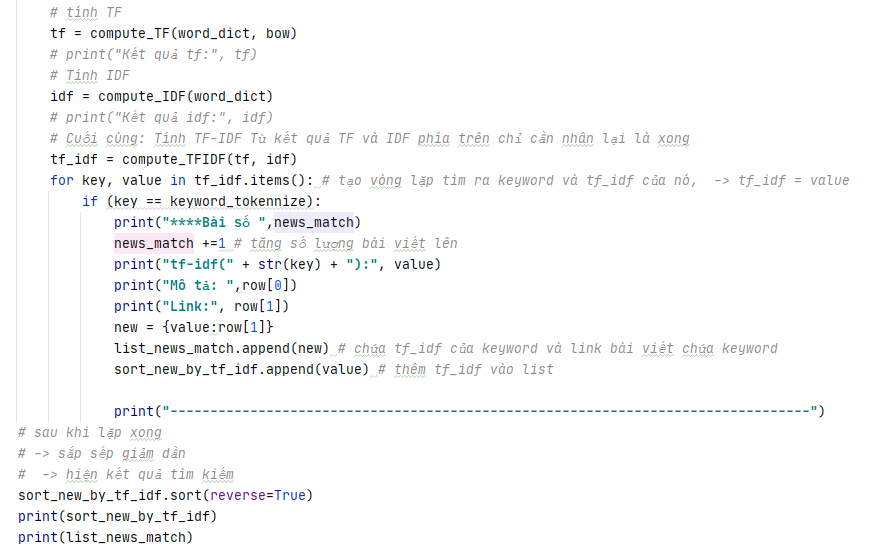
Trong đó:

* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.
* Cụ thể, chúng ta có công thức tính tf-idf hoàn chỉnh như sau: tfidf(t, d, D) = tf(t, d) x idf(t, D)
* Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).
* Tiến hành cài đặt tf, idf, tf-idf :

Tính tf, idf

* Tính tf-idf bằng cách nhân tf với idf lại với nhau:

Tính tf-idf

* Tìm kiếm

Hàm tìm kiếm khi nhâp vào từ khoá tìm kiếm

Hàm tìm kiếm khi nhâp vào từ khoá tìm kiếm



Kết quả tìm kiếm bằng TF\_IDF

Những bài báo nào có giá trị TF\_IDF cao hơn thì có liên quan nhiều với từ khoá tìm kiếm, còn những bài báo nào có giá trị TF\_IDF thấp hoặc bằng 0 thì không có liên quan tới từ khoá tìm kiếm.  
Dựa vào đó, sau khi tính được giá trị TF\_IDF, tiến hành sắp xếp các bài cáo có giá trị TF\_IDF lớn nhất lên đầu kết quả tìm kiếm.

# DANH MỤC TÀI LIỆU THAM KHẢO

<https://www.nltk.org/book/ch01.html>

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

<https://newspaper.readthedocs.io/en/latest/>

<https://viblo.asia/p/xu-ly-ngon-ngu-tu-nhien-voi-python-p4-WAyK8RymlxX>

<https://codetudau.com/machine-learning-nlp-scikit-learn/index.html>

<https://codetudau.com/bag-of-words-tf-idf-xu-ly-ngon-ngu-tu-nhien/index.html>

<https://www.sqlitetutorial.net/>

<https://maelfabien.github.io/machinelearning/NLP_2/#2-term-frequency-inverse-document-frequency-tf-idf>