**TEXTRANK**

**Hàm trích xuất từ khoá:** *extract\_key\_phrases(text)*

Input: Văn bản cần trích xuất từ khoá (text).

Output: Một set các từ khoá quan trọng của văn bản (text).

1. Thư viện đi kèm:

* Nltk
* Networkx
* Itertools
* Editdistance

1. Các bước hoạt động:
   * Dùng thư viện nltk để tách văn bản thành các tokens bao gồm các: từ, cụm từ, dấu câu,… Và lưu vào biến *word\_tokens*.
   * Gắn nhãn (tag) từ loại cho các token vào biến *tagged*.
   * Lọc tìm các từ có khả năng là từ khoá bằng hàm *filter\_for\_tags(tagged)*:
     + Danh từ, cụm danh từ: có tag *NN* và *NNP*;
     + Tính từ: có tag JJ.
   * Loại bỏ các dấu câu khỏi *tagged* bằng hàm *normalize(tagged).*
   * Sắp xếp các từ, cụm từ duy nhất (loại bỏ các lần xuất hiện trùng lặp) vào một list tên *word\_set\_list* .
   * Dựng một biểu đồ bằng hàm build\_graph(). Trong đó mỗi đỉnh thể hiện một từ khoá của văn bản, mỗi cạnh thể hiện minimum edit distance của các từ.

Hàm *build\_graph(word\_set\_list):*

* + - Tạo một biểu đồ bằng *networkx.graph()*
    - So sánh edit distance của từng cặp từ trong *word\_set\_list* và lưu thành *weight*
    - Add các cặp từ và *weight* của chúng vào *graph.*
    - Lưu *graph.*
  + Tính toán tầm quan trọng của mỗi từ khoá trong văn bản bằng hàm *networkx.pagerank(graph)* rồi sắp xếp lại theo thứ tự giảm dần và lưu vào biến *keyphrases*.
  + Chọn ra top các từ khoá có tầm quan trọng cao nhất (1/3 số đỉnh đầu tiên trong biểu đồ).
  + Xét lại các cụm từ (ghép lại từ nhiều từ) được chọn làm keyphrases trong văn bản, nếu có 2 từ nằm cạnh nhau trong văn bản mà được chọn làm từ khoá thì ghép chúng lại với nhau.
  + Cuối cùng, trả về set các từ, cụm từ khoá.

**Hàm trả về dãy từ khoá:** *return\_key\_phrases(text, delimiter)*

Input:

* Văn bản cần trích xuất từ khoá (text).
* Dấu phân cách (delimiter) giữa các từ khoá. Mặc định là dấu phẩy: “, “

Output: Một chuỗi các từ khoá quan trọng của văn bản (text) được cách nhau bởi dấu phân cách (delimiter).