Sử dụng thuật toán N-Gram để phân tích văn bản text ra keyword hộ trợ người dung không phải nhâp keywords (viết thành dạng lib càng tốt).

Đầu vào là text văn bản, đâu ra là 1 list<String> keywords.

Lưu ý cần có 1 bước tiền xử lý văn bản text trước khi đưa vào thuật toán N-Gram. (<https://spark.apache.org/docs/latest/ml-features.html>).

+ phân tích:

Văn bản hiện thời nằm ở các định dạng khác nhau . không quản lý , không search được .

Cần convert về dạng file text văn bản (.txt) lưu vào trong data để search.

Để hổ trợ search nhanh và chính xác nên biến nội dung text thành các keyword lưu trong data .

Các keyword có thể được manager nhập vào nhưng để thuận tiện cho người dung hệ thống sẽ tự sinh ra các keyword để người dung lựa chọn.

Sử dụng N-gram để biến nội dung text của văn bản thành các keyword. Với các tham số có thể hiệu chỉnh được như N , tỷ lệ xuất hiện của 1 từ , mức độ hiệu chỉnh của 1 văn bản.

N trong n-gram chạy từ 1-5 nhưng nên chọn trong khoảng 2-3.

Tỷ lệ xuất hiện của 1 từ nên lớn hơn 1.

Mức độ hiệu chỉnh văn bản : 1 văn bản có 1000 từ thì chỉ nên lấy tầm 10-15 từ làm keyword. 1 văn bản có 100 từ thì nên lấy 40- 50 từ làm keyword. Tức là 1 văn bản càng nhiều từ thì lấy ít keyword nhưng tỷ lệ xuất hiện cao, và ngược lại.

Các tham số này được lưu trong 1 file text có thể edit được = notepab.

**++ lưu ý là cần loại bỏ các từ vô nghĩa trước khi sử dụng thuật toán N-gram để tạo ra keyword. Vi du:(**[**https://spark.apache.org/docs/latest/ml-features.html**](https://spark.apache.org/docs/latest/ml-features.html)**).**