Chương 3: Giai đoạn Keyword Extractor

3.1 Các công cụ

3.1.1 Môi trường

- Programming Language: Python v3.10.8
- Package installer: Pip v22.3.1
- Text Editor: Visual Studio Code v1.74.1

3.1.2 Các công cụ trích xuất từ khoá

- rake-nltk v1.0.6 (https://github.com/csurfer/rake-nltk): Cài đặt của thuật toán RAKE bằng Python và NLTK.
- pytextrank v3.2.4 (https://github.com/DerwenAI/pytextrank): Cài đặt của thuật toán TextRank bằng Python và spaCy pipeline extension.
- yake v0.4.8 (https://github.com/LIAAD/yake): Cài đặt của thuật toán YAKE bằng Python.
- keybert v0.7.0: (https://github.com/MaartenGr/KeyBERT): Sử dụng framework máy học BERT và pretrained data all-MiniLM-L6-v2 rồi sử dụng độ tương tự cosin (cosine similarity) để xử lý.

3.1.3 Các công cụ hỗ trợ khác

- Microsoft Excel: Để lập bảng đánh giá, so sánh từng công cụ trích xuất từ khoá.

3.2 Thuật toán

3.2.1 Sơ lược về các thuật toán Keyword Extraction

- Được sử dụng để nhận diện những cụm từ mà mô tả được nội dung của văn bản.
- Là vấn đề trong những lĩnh vực text mining, trích xuất thông tin (information extraction), thu thập thông tin (information retrieval) và xử lý ngôn ngữ tự nhiên (NLP).
- Những phương pháp chính được sử dụng để tạo nên keyword extraction bao gồm giám sát (supervised), bán giám sát (semi-supervised) hoặc không giám sát (supervised).
- Phương pháp không giám sát còn được chia ra cụ thể hơn là sử dụng thống kê, sử dụng ngôn ngữ học hay đồ thị hay các phương pháp tổng hợp kết hợp vài hoặc tất cả các phương pháp trên.

3.2.2 Thuật toán RAKE (Rapid Automatic Keyword Extraction)

- Phương pháp không giám sát (unsupervised).
- Không phụ thuộc vào từ điển, văn bản khác.
- Dựa trên các phương pháp thống kê (statistical methods).
- **Nguyên lý thuật toán**
- Hệ thống được chia làm 3 thành phần chính gồm: (1) Tiền xử lý văn bản và tìm ra ứng cử viên từ khoá, (2) Chấm điểm từng ứng viên, (3) Sáp nhập từ khoá, cụ thể như sau:
    - Văn bản đầu vào được chuyển về dưới dạng một mảng các string các ứng viên từ khoá dựa vào vị trí của các ký tự đặc biệt (như dấu chấm, dấu phẩy, xuống dòng…) hay những stop words (như and, the, of…)
    - Mỗi từ khoá được tính điểm là tổng điểm các từ đơn thành phần của nó, điểm của từ đơn được tính dựa trên word frequency freq(w), word degree deg(w) và tỉ lệ deg(w)/freg(w). Trong đó deg(w) ưu tiên những từ đơn thường xuất hiện nhiều và xuất hiện trong trong những ứng viên dài hơn, freg(w) ưu tiên tuần suất xuất hiện của từ đơn bất kể nó đi chung

với từ nào khác và tỉ lệ deg(w)/freg(w) ưu tiên những từ đa số chỉ xuất hiện trong những ứng viên dài.

- Vì RAKE tách từ khoá bởi các stop words, những từ khoá sau khi trích xuất không chứa những từ nối thành phần (như axis of evil) nên ta cần bước (3) sáp nhập từ khoá. Bước này được thực hiện bằng cách tìm các cặp ứng viên đứng liền kề nhau ít nhất hai lần và một ứng viên mới được hình thành có điểm là tổng điểm của các từ đơn trong nó.

Nguyên lý được tham khảo từ nguồn:

https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents

3.2.3 Thuật toán TextRank

- Phương pháp không giám sát (unsupervised).
- Không phụ thuộc vào từ điển, văn bản khác.
- Dựa trên đồ thị (graph-based).

- Nguyên lý thuật toán
- Hệ thống được chia làm 3 thành phần chính gồm: (1) Tiền xử lý văn bản và phân loại từ, (2) Khởi tạo đồ thị và tính toán, (3) Hậu xử lý và kết quả, cụ thể như sau:
- Văn bản được tách ra thành từng từ đơn, phân loại từng từ đơn (danh từ, động từ, trạng từ…) bằng các bộ lọc.
- Để tránh làm cho kích thước đồ thị quá to, chỉ bỏ những từ đơn là danh từ và động từ
- Quy ước G = (V, E) là đồ thị với tập hợp các đỉnh V và tập hợp cách cạnh E, trong đó E là tập hợp con của tập hợp V x V. Với mỗi đỉnh V_i, đặt $In(V_i)$ là các đỉnh hướng đến V_i và đặt $Out(V_i)$ là các đỉnh mà V_i hướng đến, d là hệ số từ 0 đến 1 với mục đích tạo ra xác suất nhảy từ một đỉnh đến một đỉnh khác ngẫu nhiên. Điểm của một đỉnh được quy ước bởi công thức:

$$s(v_i) = (1-d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(v_j)|} S(V_j)$$

- Công thức trên được chạy đến khi hội tụ, thường là sau khoảng 20 – 30 lần ở mức cao nhất là 0.0001
- Những từ khoá có điểm tốt nhất được lưu giữ lại để hậu xử lý, những từ đơn được đánh giá cao nếu đứng kề nhau thì sẽ được gộp lại thành từ khoá lớn hơn.

Nguyên lý được tham khảo từ nguồn:

https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf

3.2.4 Thuật toán YAKE! (Yet Another Keyword Extractor)

- Phương pháp không giám sát (unsupervised)
- Không phụ thuộc vào từ điển, văn bản khác
- Dựa trên các phương pháp thống kê (statistical methods)

- **Nguyên lý thuật toán**

- Hệ thống được chia làm 6 thành phần chính gồm: (1) Tiền xử lý văn bản (Text pre-processing), (2) Thu thập tính năng (feature extraction), (3) Chấm điểm cụm đơn lẻ (Individual terms score), (4) Khởi tạo danh sách các ứng viên (Candidate keywords list generation), (5) Khử lặp dữ liệu (Data Deduplication), (6) Xếp hạng (Ranking) cụ thể như sau:
    - Tách văn bản thành từng từ được ngăn cách bởi dấu cách hoặc một vài ký tự đặc biệt (như dấu chấm, dấu phẩy, xuống dòng…) hay những stop words (như and, the, of…)
    - Casing: phân loại từ theo tính chất (danh từ, động từ, trạng từ…)
    - Word Positional: đánh giá cao những từ nằm ở ví trí phía trên của văn bản.
    - Word Frequency: đánh giá tần suất xuất hiện của từ.
    - Word Relatedness to Context: tính toán số lượng của những từ xuất hiện để bổ nghĩa cho từ đang xét.
    - Word DifSentence: tính toán số lượng, tần suất của một từ xuất hiện ở nhiều câu khác nhau
    - Mỗi tiêu chí trên được cho một số điểm là S(w).
    - Ta có công thức tính điểm:

$$S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) * (1 + \sum w \in kw S(w))}$$

    - Trong đó, S(kw) là điểm của một ứng viên keyword, S(w) là các điểm của từng chữ trong ứng viên đó với các tiêu chí ở trên. Ta sẽ lấy tích số điểm của một cụm rồi chia cho tổng số điểm của một cụm để có được điểm trung bình từng chữ trên ứng viên keyword đó. Kết quả được chia tiếp cho TF(kw) (term frequency) là tần xuất xuất hiện của cụm từ - để phạt những từ xuất hiện ít hơn.
    - Để khử lặp dữ liệu, YAKE sử dụng khoảng cách Levenshtein (Levenshtein distance)

Nguyên lý được tham khảo từ nguồn:

https://asset-pdf.scinapse.io/prod/2790109590/2790109590.pdf

3.2.5 Phương pháp kết hợp kết quả của BERT và độ tương tự cosin (cosine similarity)

- Hệ thống thực hiện theo thứ tự là đưa lần lượt từng câu của văn bản vào BERT (Bidirectionally Transformer) là một mô hình học sâu hai hướng.
- BERT sẽ cho ra những vector tương ứng với từng từ đơn và có vector của từ ghép là trung bình cộng của các vector từ đơn con của nó.
- Từ đó sử dụng công thức $Sim_i = cos(w_i, W)$. Trong đó Sim_i là độ tương tự cosin của vector của từ i và cả câu.

Nguyên lý được tham khảo từ nguồn:

Sharma, P., & Li, Y. (2019). Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling.

3.2. Bộ dữ liệu thực nghiệm

Ta có những tiêu chí:

- Độ dài ít nhất 250 từ.
- Tiếng Anh và tiếng Việt là chủ đạo.
- Nhiều danh từ, danh từ riêng, động từ gây nhiễu.

- Từ khoá có thể không lặp lại nhiều lần trong văn bản.
- Độ dài từ khoá có thể ngắn một từ, cũng có thể dài năm từ.
- Xen lẫn từ khoá là ngôn ngữ khác (Tiếng Nhật, tiếng Nga và phiên âm) giữa ngôn ngữ chủ đạo.

3.3. Độ đo

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

$$accuracy = \frac{True\ Positives\ +\ True\ Negatives}{True\ Positives\ +\ True\ Negatives\ +\ False\ Positives\ +\ False\ Negatives}$$

$$F1\ measure = 2\frac{precision\ .\ recall}{precision\ +\ recall}$$

$$F0.5\ measure = \frac{(1 + 0.5^2)\ .\ precision\ .recall}{0.5^2\ .\ precision\ +\ recall}$$

- **precision** là tỉ lệ kết quả trả về chính xác trên kết quả trả về.
- **recall** là tỉ lệ kết quả trả về chính xác trên số các yếu tố cần trả về.
- **accuracy** là tỉ lệ kết quả trả về chính xác cộng và kết quả loại ra chính xác trên toàn bộ văn bản.
- **F1 measure** là kết quả kết hợp giữa precision và recall với trọng số hai kết quả ngang nhau.
- **F0.5 measure** là kết quả kết hợp giữa precision và recall với trọng số precision cao hơn.

Lý do chọn F0.5 measure là yếu tố chính để chọn ra công cụ tốt nhất cho đồ án vì:

- Mục đích chính của công cụ trích xuất từ khoá trong đồ án là để cho bước tiếp theo để đưa lên search engine nên kết quả cần precision cao để khi search không bị nhiễu.
- Khi sử dụng search engine, việc xác định được các từ khoá quan trọng nhất quan trọng hơn có nhiều từ khoá ít giá trị trong việc tìm ra kết quả cần tìm
- Accuracy sẽ dễ gây kết quả gây hiểu lầm khi số lượng phần tử tập hợp sai nhiều hơn số lượng phần tử của tập hợp đúng (ví dụ phần tử sai chiếm 95% và phần tử đúng chiếm 5% thì nếu kết quả trích xuất toàn phần tử sai thì sẽ được điểm accuracy là 95%)

3.4 Kết quả thực nghiệm

	ACCURACY			
	RAKE-nltk	TextRank	YAKE	keyBERT
Tom Lehrer	77%	77%	78%	70%
HCMUS	75%	65%	82%	64%
Diophantos	57%	55%	72%	61%
2ndPianoCon	76%	77%	80%	81%
Gennady Korotkevich	77%	84%	86%	82%
Kaguya-sama	86%	70%	81%	82%
Fourier Transform	55%	73%	77%	74%
Math Neurons	68%	86%	89%	81%
HDFS	53%	68%	69%	67%
AVERAGE	69%	73%	79%	74%

	PRECISION			
	RAKE-nltk	TextRank	YAKE	keyBERT
Tom Lehrer	78%	83%	91%	66%
HCMUS	71%	72%	88%	63%
Diophantos	53%	52%	80%	59%
2ndPianoCon	72%	74%	82%	71%
Gennady Korotkevich	81%	91%	92%	78%
Kaguya-sama	84%	80%	79%	77%
Fourier Transform	36%	60%	67%	59%
Math Neurons	49%	79%	81%	63%
HDFS	38%	61%	68%	56%
AVERAGE	63%	72%	81%	66%

	RECALL			
	RAKE-nltk	TextRank	YAKE	keyBERT
Tom Lehrer	70%	63%	59%	75%
HCMUS	82%	47%	73%	66%
Diophantos	63%	49%	54%	52%
2ndPianoCon	55%	53%	56%	76%
Gennady Korotkevich	81%	83%	84%	98%
Kaguya-sama	87%	48%	81%	87%
Fourier Transform	46%	56%	66%	70%
Math Neurons	56%	76%	85%	95%
HDFS	37%	46%	37%	62%
AVERAGE	64%	58%	66%	76%

	F1-MEASURE			
	RAKE-nltk	TextRank	YAKE	keyBERT
Tom Lehrer	74%	72%	72%	70%
HCMUS	76%	57%	80%	64%
Diophantos	58%	50%	64%	56%
2ndPianoCon	62%	62%	67%	74%
Gennady Korotkevich	81%	86%	88%	87%
Kaguya-sama	85%	60%	80%	82%
Fourier Transform	41%	58%	66%	64%
Math Neurons	53%	78%	83%	76%
HDFS	37%	53%	48%	59%
AVERAGE	63%	64%	72%	70%

F0.5-MEASURE

	RAKE-nltk	TextRank	YAKE	keyBERT
Tom Lehrer	76%	78%	82%	68%
HCMUS	73%	65%	85%	64%
Diophantos	55%	51%	73%	58%
2ndPianoCon	68%	69%	75%	72%
Gennady Korotkevich	81%	89%	91%	81%
Kaguya-sama	85%	70%	80%	79%
Fourier Transform	38%	59%	66%	61%
Math Neurons	50%	79%	82%	68%
HDFS	38%	57%	58%	57%
AVERAGE	63%	69%	77%	68%

Kết luận:

- Từ kết quả ta có thuật toán YAKE có precision và F0.5 measure cao hơn hẳn so với những công cụ khác, có accuracy và F1 measure cao nhất và recall cao thứ hai trong cả 4 công cụ nên nhóm đã chọn công cụ YAKE làm công cụ chính cho đồ án.