GK
DS204

Scientific article:

QuoteR: A Benchmark of Quote Recommendation for Writing

Group 14
20521609 – 20521614 -20521998

1. Introduction

- Giới thiệu về task quote recommendation
- Đề tài ra 3 đóng góp:
  - Xây dựng bộ dữ liệu mở với kích thướng lớn là Quote Recommendation Dataset
  - Đánh giá toàn diện và công bằng của các phương pháp đề xuất trích dẫn hiện có.
  - Đề xuất một mô hình đề xuất trích dẫn vượt trội hơn tất cả các phương pháp trước đó

2. Related Work

2.1 Quote Recommendation

- Learning to recommend quotes for writing.
- A neural network approach to quote recommendation in writings.
- Quote recommendation for dialogs and writings.
- Quote recommendation in dialogue using deep neural network.
- Continuity of topic, interaction, and query: Learning to quote in online conversations.
- Quotation recommendation and interpretation based on transformation from queries to quotations

2.2 Content-based Recommendation

- Contentbased recommendation systems.
- Recommending citations for academic papers.
- Context-aware citation recommendation.
- Neural-based chinese idiom recommendation for enhancing elegance in essay writing.

3. Task Formulation

- Đưa ra định nghĩa cho task đề xuất trích dẫn (quote recommendation) và giới thiệu một số khái niệm cơ bản, phần lớn dựa theo các nghiên cứu trước đó (Tan et al., 2015).
- Với một đoạn văn bản chứa một trích dẫn q, phần văn bản xuất hiện trước trích dẫn là ngữ cảnh bên trái cl, phần văn bản xuất hiện sau trích dẫn là ngữ cảnh bên phải cr. Sự kết hợp của hai ngữ cảnh này tạo thành ngữ cảnh trích dẫn c = [cl ; cr]. Giả sử tập hợp các trích dẫn gồm tất cả các trích dẫn ứng cử viên đã biết Q = {q1, · · · , q|Q|}.
- Nhiệm vụ của mô hình là tính toán một điểm số cho mỗi trích dẫn ứng cử viên trong Q ứng với ngữ cảnh c và đưa ra một danh sách trích dẫn theo thứ tự giảm dần điểm số.

4. Dataset Construction

4.1 The English part

- **Nguồn data:** Wikiquote
- Trích xuất hơn 60.000 trích dẫn tiếng Anh để tạo thành bộ trích dẫn
- Để có được ngữ cảnh thực tế của các trích dẫn,sử dụng ba bộ sưu tập văn bản
    - Bộ Gutenberg Project
    - Bộ BookCorpus
    - Bộ sưu tập OpenWebText

=>Tổng kích thước của văn bản thô của ba bộ sưu tập văn bản này đạt 48,8 GB.

- Lấy 40 từ trước và sau mỗi trích dẫn làm ngữ cảnh trái và phải tương ứng, kết hợp ngữ cảnh và trích dẫn tạo thành cặp ngữ cảnh-trích dẫn. Loại bỏ các cặp trùng và lọc bỏ các trích dẫn xuất hiện ít hơn 5 lần trong các bộ sưu tập văn bản.
- Để tránh sự mất cân bằng của tập dữ liệu, ngẫu nhiên chọn 200 cặp ngữ cảnh-trích dẫn cho một trích dẫn xuất hiện hơn 200 lần và loại bỏ các cặp ngữ cảnh-trích dẫn khác. => thu được 126.713 cặp ngữ cảnh-trích dẫn liên quan đến 6.108 trích dẫn khác nhau, tạo thành phần tiếng Anh của QuoteR.

4. Dataset Construction

4.2 The Standard Chinese Part

- Nguồn: Juzimi
- Số lượng: 32000
  - Để thu được ngữ cảnh, sử dụng hai bộ văn bản
  - Các câu trả lời trong một trang web hỏi đáp tiếng Trung
- Một bộ sách lớn đã được xây dựng đặc biệt và bao gồm hơn 8.000 cuốn sách điện tử tiếng Trung miễn phí.

=> Tổng kích thước của hai bộ văn bản này là khoảng 32 GB.

- Xử lý tương tự English Part (có sự điều chỉnh) =>thu được 40.842 cặp ngữ cảnh-trích dẫn liên quan đến 3.004 trích dẫn.

4. Dataset Construction

4.3 The Classical Chinese Part

- Nguồn: Gushiwenwang, Juzimi
- Số lượng: 17000
- Xử lý tương tự Standard Chinese Part => thu được 116.537 cặp văn bản-trích dẫn của 4.438 trích dẫn.

****Chia dữ liệu thành các tập Train, Validation, Test với tỉ lệ 8:1:1**

Part	Train	Validation	Test	Total
English	101,171/6,008	12,771/6,108	12,771/6,108	126,713/6,108
sChinese	32,472/2,904	4,185/3,004	4,185/3,004	40,842/3,004
cChinese	93,031/4,338	11,753/4,438	11,753/4,438	116,537/4,438

Đánh giá dữ liệu:

- Lấy mẫu 100 cặp văn bản-trích dẫn, yêu cầu 3 annotator xác định độ phù hợp của mỗi trích dẫn với văn bản tương ứng.

=> Kết quả cuối cùng được tính dựa trên nguyên tắc bỏ phiếu. (Tương ứng 99/98/94 cặp văn bản-trích dẫn được coi là phù hợp)

5.1. Basic Framework

Learning Representations of Contexts

1. **Firstly**, They insert an additional separator token between [c_l;c_r] before feeding into BERT:

$$\mathbf{h}^c_{[\mathrm{C}]}, \cdots = \mathrm{BERT}^c([\mathrm{C}], c_l, [\mathrm{S}], c_r), \quad (2)$$

- With every **Contexts**, they use [C] (CLS) as the representation of the **Context**. **c=$h^c_{[c]}$**

=> **That solution not suitable for this task**. Because It only used to classify the relation between the two segments *(the next sentence prediction (NSP) pre-training task)*.

2. **After that**, They use other pre-training task of BERT, **masked language modeling (MLM), Aimed at predicting masked tokens.**

- Inspired by the MLM pre-training task, **they propose another way to learn the context representation by inserting an additional [MASK] token:**

$$\mathbf{h}^c_{[\mathrm{C}]}, \cdots, \mathbf{h}^c_{[\mathrm{M}]}, \cdots = \mathrm{BERT}^c([\mathrm{C}], c_l, [\mathrm{M}], c_r), \quad (3)$$

- With every **Contexts**, they use [M] (MASK) as the representation of the **Context**. **c=$h^c_{[M]}$**

5.1. Basic Framework

Learning Representations of Quotes

$$\mathbf{h}^q_{[\mathrm{C}]}, \mathbf{h}^q_1, \cdots, \mathbf{h}^q_m = \mathrm{BERT}^q([\mathrm{C}], x_1, \cdots, x_m), \quad (1)$$

- With every **Quotes**, they use q = $\mathbf{h}^c_{[M]}$ ([CLS]) as the representation of the quote
=> All quotes form: **Q**= **[q_1, ... , $q_{|Q|}$]**

Calculating Rank Scores of Candidate Quotes

$$\mathbf{p} = \mathrm{softmax}(\mathbf{Q}^\top \mathbf{c}), \quad (4)$$

- **p** is **a normalized probability vector** whose **i-th element** is the rank score of the **i-th quote**.

5.2. Training Strategy

- **Problem**
  - **cross-entropy loss** to **train simultaneously** the **quote** and **context encoders**
    - For each context, the quote encoder needs to be updated for every quote.
    - The huge imbalance between positive and negative samples (one vs. several thousands).
- **Solution**
  - A simple solution is **freeze the quote encoder during training** => untrained quote encoder would **decrease** their **task performance => they don't use.**
  - **They adopt the negative sampling strategy in training**
  - Pseudo-rank score:

$$p^* = \frac{e^{\mathbf{q}\cdot\mathbf{c}}}{e^{\mathbf{q}\cdot\mathbf{c}} + \sum_{q^* \in \mathbb{N}(q)} e^{\mathbf{q}^*\cdot\mathbf{c}}}, \qquad (5)$$

    - N(q) is the set of quotes selected as negative samples
  - Then the training loss is the cross-entropy based on the pseudo-rank score:

$$\mathcal{L} = -\log(p^*).$$

5.2. Training Strategy

- **Problem**
  - Context encoder may be under-trained

  (The context encoder needs to process lots of contexts and thus requires more training than the quote encoder).
- **Solution**
  - **They adopt a two-stage training strategy.**
  - After the simultaneous training of quote and context encoders in the first stage.
  - Continue to train the context encoder while freezing the quote encoder in the second stage - The training loss of the second stage: cross-entropy loss among all quotes.

5.3. Incorporation of Sememes

- **Examples Sememes:**
  - Sememes là các đơn vị ý nghĩa nhỏ nhất của ngôn ngữ, tương tự như các nguyên tử trong hóa học. Chúng được sử dụng để giải thích ý nghĩa của các từ và cụm từ trong ngôn ngữ.
    - Ví dụ về sememes cho từ "chó":
      - Động vật: sememe mô tả loài động vật
      - Động, có bốn chân: sememe mô tả cấu trúc bộ phận của loài động vật này
      - Có lông: sememe mô tả tính chất của long
  - Với những sememe này, chúng ta có thể định nghĩa từ "chó" như sau: "Một loài động vật có bốn chân và lông".

5.3. Incorporation of Sememes

- **They propose to incorporate sememe knowledge into quote representation learning**
  - Inspired by the studies on “incorporating sememes into recurrent neural networks” and “Enhancing transformer with sememe knowledge” => They **adopt a similar way to incorporate sememes into the quote encoder.**
  - They simply **add** the **average embedding of a word’s sememes** to EVERY **token embedding of the word** in BERT.
  - Explain:
    - **a** quote that is divided into **n** tokens => . With xi convert follow fomular:

$$\mathbf{x}_i \rightarrow \mathbf{x}_i + \frac{\alpha}{|\mathbb{S}(w)|} \sum_{s_j \in \mathbb{S}(w)} \mathbf{s}_j, \ \forall i = 1, \cdots, n \quad (6)$$

    - S(w) is the sememe set of the word w (ex: “chó” có S(w) = “Động vật”, “Động, có bốn chân”, “Có lông”).
    - Sj: average embedding of a word’s sememes.
    - α is a hyper-parameter controlling the weight of sememe embeddings.

6.1. Approaches for Comparison

- **Three groups of approaches for comparison:**
  - The first group consists of two methods **(widely serve as BASELINE in previous studies)**
    - **CRM** (context-aware relevance model)
    - **LSTM**
  - The second group **(REPRESENTATIVE APPROACHES PROPOSED in previous studies)**
    - **top-k RM** (namely top-k rank multiplication)
    - **NNQR**
    - **N-QRM**
    - **Transform**
  - The third group **(two BERT-based approaches that are usually use in sentence matching and sentence pair classification (like QuoteR task))**
    - **BERT-Sim**
    - **BERT-Cls**

6.2. Evaluation Metrics

1. Recall@K

$$\text{Recall@}k = |G|/|C_{test}|$$

2. Mean reciprocal rank (MRR)

$$\text{MRR} = \frac{1}{|C_{test}|} \sum_{s \in C_{test}} \frac{1}{rank\,(s)}$$

3. Normalized discounted cumulative gain (NDCG@K)

$$\text{NDCG@K} = Z_K \sum_{i=1}^{K} \frac{2^{r(i)} - 1}{\log_2(i+1)},$$

4. Median Rank
5. Mean Rank
6. Rank Variance

=> **The higher** Recall@K, MRR, NDCG@K and **the lower** R̃, R⁻ and σR are **the better a model**.

Example Metrics

Ví dụ k = 5:

- **Context**: 2 context
- **Gold qoute:** [1 0 0 0 0], [1 0 0 0 0]
  - **Predict 1:** [0 0 0 1 0], [0 0 0 0 0] (ví dụ Gold qoute ở vị trí *12*).
  - **Predict 2:** [0 0 1 0 0], [0 0 0 0 1]
  - **Predict 3:** [1 0 0 0 0], [1 0 0 0 0]

Tính độ đo:

- **Recall@5:** (trong paper nó nhân 100. Đơn vị là percentage)
  - **Predict 1:** $\frac{1}{1}+\frac{0}{1}=0.5$
  - **Predict 2:** $\frac{1}{1}+\frac{1}{1}=1$
  - **Predict 3:** $\frac{1}{1}+\frac{1}{1}=1$
- **MRR: (ko quan tâm tới K)**
  - **Predict 1:** $\frac{1}{2}\left(\frac{1}{4}+\frac{1}{12}\right)\cong 0.17$
  - **Predict 2:** $\frac{1}{2}\left(\frac{1}{3}+\frac{1}{5}\right)\cong 0.27$
  - **Predict 3:** $\frac{1}{2}\left(\frac{1}{1}+\frac{1}{1}\right)\cong 1$

Example Metrics

- NDCG@5:

- **Predict 1:**
  - $1\left(\frac{2^0-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^0-1}{\log_2(1+3)}\right)+1\left(\frac{2^1-1}{\log_2(1+4)}\right)+1\left(\frac{2^0-1}{\log_2(1+5)}\right)\cong 0.43$
  - $1\left(\frac{2^0-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^0-1}{\log_2(1+3)}\right)+1\left(\frac{2^0-1}{\log_2(1+4)}\right)+1\left(\frac{2^0-1}{\log_2(1+5)}\right)\cong 0$
  - **NDCG@5 =** $\frac{0.43}{2}=0.215$
- **Predict 2:**
  - $1\left(\frac{2^0-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^1-1}{\log_2(1+3)}\right)+1\left(\frac{2^0-1}{\log_2(1+4)}\right)+1\left(\frac{2^0-1}{\log_2(1+5)}\right)\cong 0.5$
  - $1\left(\frac{2^0-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^0-1}{\log_2(1+3)}\right)+1\left(\frac{2^0-1}{\log_2(1+4)}\right)+1\left(\frac{2^1-1}{\log_2(1+5)}\right)\cong 0.39$
  - **NDCG@5 =** $\frac{0.5+0.39}{2}=0.445$
- **Predict 3:**
  - $1\left(\frac{2^1-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^0-1}{\log_2(1+3)}\right)+1\left(\frac{2^0-1}{\log_2(1+4)}\right)+1\left(\frac{2^0-1}{\log_2(1+5)}\right)=1$
  - $1\left(\frac{2^1-1}{\log_2(1+1)}\right)+1\left(\frac{2^0-1}{\log_2(1+2)}\right)+1\left(\frac{2^0-1}{\log_2(1+3)}\right)+1\left(\frac{2^0-1}{\log_2(1+4)}\right)+1\left(\frac{2^0-1}{\log_2(1+5)}\right)=1$
  - **NDCG@5 =** $\frac{1+1}{2}=1$

6.4. Main Results

Part	English				Standard Chinese				Classical Chinese			
Model	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100
CRM	0.192	0.193	599/1169/1408	16.51/23.66/32.78	0.397	0.407	13/325/584	33.60/49.32/61.70	0.198	0.203	166/548/811	14.52/28.79/44.51
LSTM	0.321	0.320	30/334/727	27.23/40.78/62.47	0.292	0.290	48/338/574	24.78/37.71/58.06	0.247	0.245	56/341/633	20.08/33.23/56.96
top-k RM	0.422	0.431	6/548/1243	35.99/53.31/66.20	0.480	0.494	3/377/774	40.17/60.67/72.26	0.294	0.299	48/511/980	23.54/39.58/56.90
NNQR	0.318	0.319	31/359/773	26.78/41.10/61.29	0.271	0.271	54/348/595	22.94/35.72/57.18	0.272	0.270	41/310/620	22.03/36.59/60.63
N-QRM	0.365	0.368	28/777/1465	32.24/44.41/58.26	0.343	0.347	55/575/890	30.20/41.22/54.15	0.287	0.288	98/917/1373	24.88/35.02/49.49
Transform	0.561	0.568	1/241/749	50.11/65.88/79.98	0.512	0.519	2/271/576	45.50/60.31/72.83	0.449	0.453	5/269/663	39.01/55.78/73.58
BERT-Sim	0.526	0.529	2/487/1064	49.38/58.05/67.75	0.500	0.508	2/229/511	44.47/59.07/72.21	0.439	0.443	7/320/711	38.85/53.04/68.32
BERT-Cls	0.310	0.329	7/134/453	18.15/57.11/82.05	0.378	0.395	5/152/413	26.88/57.90/78.38	0.330	0.345	8/**135/377**	21.93/54.27/78.75
Ours	**0.572**	**0.580**	1/**123/433**	50.74/**69.03/83.84**	**0.541**	**0.548**	2/**139/370**	**47.91/64.97/79.35**	**0.484**	**0.490**	3/146/422	**41.67/60.78/79.38**

- Phương pháp cho kết quả tốt nhất
- hai Mô hình BERT, đặc biệt là BERT-Sim, cho hiệu suất khá cao

⇒ Tầm quan trọng của **powerful sentence encoder**

- Classical Chinese (tiếng Trung cổ) **hiệu suất kém hơn** có thể do **được huấn liện bằng Stadard Chinese (tiếng Trung chuẩn) trước.**

6.4. Main Results

Ablation study

Part	English				Standard Chinese				Classical Chinese			
Model	MRR	NDCG	$\tilde{R}$ /$\bar{R}$ /σ_R	Recall@1/10/100	MRR	NDCG	$\tilde{R}$ /$\bar{R}$ /σ_R	Recall@1/10/100	MRR	NDCG	$\tilde{R}$ /$\bar{R}$ /σ_R	Recall@1/10/100
Ours	**0.572**	**0.580**	**1/123/433**	50.74/**69.03/83.84**	**0.541**	**0.548**	**2/139/370**	**47.91/64.97/79.35**	**0.484**	**0.490**	3/146/422	**41.67/60.78/79.38**
-Sememe	0.568	0.574	1/145/492	**51.05**/67.07/82.34	0.535	0.543	2/160/402	47.62/63.66/77.68	0.475	0.481	3/152/435	40.93/60.26/78.39
-ReTrain	0.299	0.307	12/176/503	20.46/47.89/75.74	0.255	0.260	20/210/435	16.87/42.94/68.43	0.265	0.269	17/184/450	17.87/43.56/72.89
-SimTrain	0.529	0.532	2/467/1060	49.31/58.97/69.48	0.519	0.526	2/204/489	46.00/62.03/75.34	0.465	0.470	4/310/713	41.40/55.53/70.09

- -Sememe gây ra sự suy giảm hiệu suất nhất quán

⇒ Vai trò của sememes trong **cải thiện mã hóa trích dẫn (quote encoding)**

- Hiệu suất của -ReTrain khá kém

⇒ Sự **cần thiết** của việc **train riêng context encoder** sau khi **train đồng thời**

- -SimTrain kém hơn -Sememe

⇒ Tính hữu ích của việc đào tạo đồng thời hai bộ mã hóa

6.5 Quote Recommendation with Left Context Only

- Theo công trình trước đó, đề xuất trích dẫn **chỉ dựa trên văn bản bối cảnh bên trái** có thể hữu ích hơn.

Part	English				Standard Chinese				Classical Chinese			
Model	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\tilde{R}/\bar{R}/\sigma_R$	Recall@1/10/100
CRM	0.154	0.156	353/948/1297	11.88/21.78/33.66	0.292	0.296	124/401/524	25.28/35.39/48.43	0.141	0.146	276/587/763	9.88/19.75/34.57
LSTM	0.272	0.271	89/552/992	23.38/33.87/51.12	0.210	0.208	146/483/662	18.26/27.67/45.50	0.182	0.178	117/465/750	13.87/25.44/47.80
top-k RM	0.360	0.366	30/833/1497	31.20/44.55/56.80	0.350	0.358	38/620/926	29.77/44.40/55.53	0.276	0.280	77/645/1088	22.61/36.16/52.57
NNQR	0.267	0.266	98/592/1043	22.82/33.48/50.28	0.224	0.223	145/495/683	17.16/27.67/45.81	0.189	0.187	98/441/766	14.18/26.86/50.29
N-QRM	0.270	0.272	156/1145/1735	23.40/33.18/46.54	0.266	0.270	287/778/946	21.27/30.63/42.32	0.215	0.215	356/1232/1505	17.72/27.13/40.73
Transform	0.438	0.443	6/429/1036	38.47/53.43/68.65	0.371	0.374	29/465/748	32.54/44.83/58.04	0.331	0.334	29/435/842	27.76/42.87/60.85
BERT-Sim	0.399	0.401	44/839/1407	36.95/44.75/54.32	0.364	0.370	41/431/695	31.71/44.28/56.18	0.310	0.313	56/522/902	26.32/39.05/54.56
BERT-Cls	0.265	0.275	15/**237/640**	16.75/45.37/71.77	0.213	0.220	24/318/646	12.47/40.53/64.67	0.204	0.208	25/253/568	11.50/38.27/66.73
Ours	**0.456**	**0.462**	**4**/254/685	**39.62/56.21/73.26**	**0.413**	**0.419**	**7/97/186**	**34.64/53.29/75.91**	**0.409**	**0.411**	**9/196/419**	**35.22/51.47/70.82**

- Đánh giá với **Left Context Only** phương pháp của họ vẫn **tốt nhất**
- **Hiệu suất** của tất cả các phương pháp **giảm**

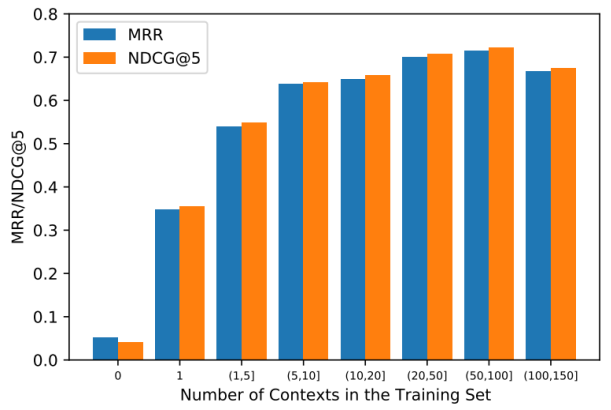
⇒ **Bên trái** và **bên phải** cung cấp thông tin **quan trọng**

6.6 Effect of Occurrence Frequency

- Nghiên cứu tác động của tần suất xuất hiện **trích dẫn vàng**
  - **Có tác động lớn** đến hiệu suất
  - Phổ biến nhất **không** có hiệu suất **tốt nhất**

  ⇒ Mang ý nghĩa phong phú và trích dẫn trong nhiều ngữ cảnh khác nhau
  - Hiệu xuất của **chưa được xem trước** rất hạn chế

  => **Điểm yếu** của mô hình và sẽ được nghiên cứu sau



6.7 Effect of Negative Sample Number

- Nghiên cứu tác động của số lượng mẫu tiêu cực (#NS)
- Tăng số lượng mẫu tiêu cực (từ 4 đến 19) có thể cải thiện hiệu suất => **Train đầy đủ hơn**
- Tiếp tục tăng => Hiệu suất **dao động** hoặc **giảm** => Mất cân bằng giữa **tích cực** và **tiêu cực**

6.8 Human Evaluation

- **Vấn đề:** Có thể có các trích dẫn khác phù hợp với ngữ cảnh truy vấn ngoài trích dẫn vàng.
- Chọn ngẫu nhiên 50 văn bản trong tập test tiếng Trung chuẩn và liệt kê top 10 đề xuất
- Gán nhãn bởi 3 người bản ngữ và quyết định cuối cùng là bỏ phiếu.
- So sánh:
  - **Human Evaluation**: NDCG@5 = 0,661, Recall@1/10 = 0,50/0,92
  - **Machine Evaluation**: NDCG@5 = 0,439, Recall@1/10 = 0,36/0,64

⇒ Hiệu suất thực tế bị đánh giá thấp hơn rất nhiều.

6.9 Case Study

- Trích dẫn vàng được xếp hạng thứ hai
- Trích dẫn đầu tiên có cùng ý nghĩa trích dẫn vàng.
  Trích dẫn thứ ba và tư cũng liên quan đến ngữ cảnh.

⇒ Tính hiệu quả của mô hình

Rank	Quote	Score
1	sufficient for the day is its own trouble	0.723
2	**sufficient unto the day is the evil thereof**	0.124
3	you can never plan the future by the past	0.060
4	tomorrow will be a new day	0.025
5	the darkest hour is just before the dawn	0.008

Thanks for listening