ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

VÒNG VĨNH PHÚ

Xây dựng cơ sở tri thức cho truy vấn văn bản tiếng việt

Building a Knowledge Graph via Vietnamese Relation Extraction

KHOÁ LUẬN TỐT NGHIỆP

Tp. Hồ Chí Minh - 2023

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

VÒNG VĨNH PHÚ

Xây dựng cơ sở tri thức cho truy vấn văn bản tiếng việt

Building a Knowledge Graph via Vietnamese Relation Extraction

KHOÁ LUẬN TỐT NGHIỆP

CHUYÊN NGÀNH PHƯƠNG PHÁP TOÁN TRONG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC
PGS.TS.NGUYỄN THANH BÌNH

Tp. Hồ Chí Minh - 2023

MỤC LỤC

DANH SÁCH HÌNH VẼ

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Khoa học Tự Nhiên, Khoa Toán - Tin học và Bộ Môn Phương pháp toán trong tin đã có sự sắp xếp các môn học vào chương trình giảng dạy một cách có khoa học và logic qua đó em đã có kiến thức chuyên môn tốt để có thể hoàn thành khoá luận ngày hôm nay. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn - Thầy Nguyễn Thanh Bình đã dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Cũng như các bạn cùng khoá 2019 đã giúp đỡ em trong suốt quá trình làm khoá luận này. Trong thời gian học tại trường và tham gia khoá luận, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Chuyên ngành Phương pháp toán trong tin là một chuyên ngành rất thực tế và vô cùng bổ ích. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong cô xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỞ ĐẦU

1. Lý do chọn đề tài

- Với sự phát triển ngày càng tăng của dữ liệu, nhu cầu truy xuất và xử lý văn bản trở nên ngày càng quan trọng. Trong bối cảnh này, mục tiêu chính của đề tài này là xây dựng một bộ dữ liệu biểu đồ tri thức nhằm tăng cường mô hình truy vấn văn bản cho ngôn ngữ tiếng Việt. Qua đó, xây dựng được một biểu đồ tri thức có liên kết và lưu trữ thông tin truy vấn thông qua các mối quan hệ ngữ nghĩa, tạo ra một cấu trúc tổ chức thông tin hữu ích.

2. Mục đích nghiên cứu

- Tìm ra hướng tiếp cận cho các bài toán thuộc lĩnh vực xử lý ngôn ngữ tự nhiên, xây dựng được bộ dữ liệu dễ dàng truy vấn trong bộ dữ liệu lớn

3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Định nghĩa, mô hình, phương pháp tối ưu hàm số mất mát.

- Phạm vi nghiên cứu: khoá luận này tập trung nghiên cứu về xây dựng mô hình triết xuất thông tin liên hệ giữa chủ thể và đối tượng trong câu.

4. Phương pháp nghiên cứu

- Tìm hiểu các tài liệu liên quan đến mô hình xử lý ngôn ngữ tự nhiên.

- Phân tích dữ liệu hiện có và xử lý để đưa ra thông tin tốt nhất cho mô hình học và dự đoán.

- Quan sát để đưa ra các tiêu chí đánh giá mô hình nhằm cải thiện tính chính xác của mô hình.

5. Ý nghĩa khoa học và thực tiễn của đề tài

- Xây dựng bộ dữ liệu biểu đồ tri thức cho ngôn ngữ tiếng Việt không chỉ mang lại lợi ích trong việc truy vấn và xử lý văn bản, mà còn cung cấp một

cơ sở kiến thức vững chắc cho các ứng dụng liên quan.

- Mặc dù mô hình vẫn chưa hoàn thiện do thiếu sót về kinh nghiệm cũng như dữ liệu, nhưng nó đã đem lại kết quả tích cực trong việc nhận diện các thực thể và tạo ra các bộ ba thông tin truy vấn và giúp ích rất nhiều trong lĩnh vực xử lý tự nhiên.

6. Cấu trúc bài khoá luận

- Bài khoá luận này bao gồm có 4 phần chính :

- **Các kiến thức liên quan :** là các kiến thức đã nhóm nghiên cứu trước đó và các kiến thức liên quan đến mô hình mà được đề cập trong bài khoá luận.

- **Giới thiệu :** là các vấn đề và phương pháp đề xuất phát triển cũng như giới thiệu mô hình được đề cập trong khoá luận.

- **Dataset** Giới thiệu thông tin các bộ dữ liệu sử dụng trong việc huấn luyện phát triển mô hình.

- **Kết quả và thực nghiệm** Giới thiệu kết quả mà mô hình đã làm được trong thực tế.

CHƯƠNG 1:

KIẾN THỨC LIÊN QUAN

1.1 Cơ sở tri thức

1.1.1 Cơ sở tri thức

Cơ sở tri thức (Knowledge base) là một tập hợp các thông tin có tổ chức được lưu trữ và sử dụng để cung cấp kiến thức và thông tin cho hệ thống hoặc người dùng. Nó là một nguồn dữ liệu chứa các kiến thức, thông tin và sự hiểu biết về một lĩnh vực cụ thể.

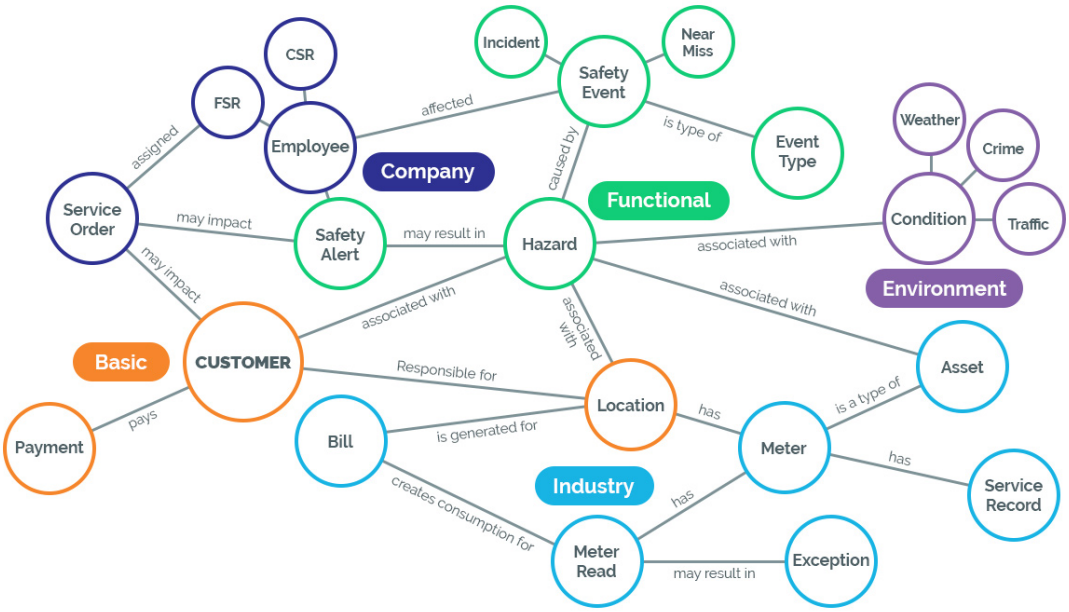
Mục tiêu của việc xây dựng một cơ sở tri thức là thu thập, tổ chức và duy trì kiến thức và thông tin có giá trị để sử dụng trong việc giải quyết vấn đề, truy vấn thông tin, hỗ trợ quyết định và tạo ra giá trị.

Một cơ sở tri thức chất lượng cao cung cấp nguồn thông tin đáng tin cậy và hữu ích, giúp người dùng nắm bắt kiến thức cần thiết và giải quyết các vấn đề một cách chính xác.

1.1.2 Đồ thị tri thức

Đồ thị tri thức[7] (Knowledge graph) là một cấu trúc dữ liệu mạnh mẽ và linh hoạt được sử dụng để biểu diễn tri thức cũng như mối quan hệ giữa các yếu tố trong một lĩnh vực cụ thể. Nó là một sự mở rộng của cơ sở tri thức, trong đó "tri thức" được biểu diễn dưới dạng các "nút" (nodes) và các "liên kết" (edges) để tạo ra một mạng lưới phức tạp của các thực thể và mối quan hệ giữa chúng.

Mỗi nút trong đồ thị tri thức đại diện cho một thực thể cụ thể, chẳng hạn như người, địa điểm, sự kiện, sản phẩm hoặc ý tưởng. Các liên kết giữa các nút mô tả các mối quan hệ hoặc sự liên kết giữa chúng.



Hình 1.1: Knowledge graph

Đồ thị tri thức có thể được xây dựng và mở rộng thông qua việc tổng hợp thông tin từ nhiều nguồn khác nhau, bao gồm cơ sở dữ liệu, trang web, sách báo ngoài ra các phương pháp tự động trích xuất thông tin và xử lý ngôn ngữ tự nhiên cũng thường được sử dụng để trích xuất dữ liệu và xây dựng đồ thị tri thức.

Đồ thị tri thức là một cách hiệu quả để tổ chức và truy vấn tri thức, giúp tăng cường khả năng hiểu và tương tác với dữ liệu phức tạp, và đóng vai trò quan trọng trong việc phát triển các ứng dụng thông minh và trí tuệ nhân tạo.

1.2 BERT

BERT[4] (Bidirectional Encoder Representations from Transformers) là một kiến trúc mạng nơ-ron được giới thiệu bởi Devlin et al., 2018. BERT có khả năng "hiểu" các mối quan hệ ngữ nghĩa giữa các từ trong một câu bằng cách xem xét ngữ cảnh của từ đó trong toàn bộ câu. Điều này được thực hiện bằng cách sử dụng mô hình transformer, một kiến trúc mạng nơ-ron sử

dụng self-attention mechanism để tạo ra các biểu diễn ngữ nghĩa phong phú cho các từ trong câu

Một điểm đặc biệt của BERT là khả năng xử lý hai chiều (bidirectional), tức là nó có thể "nhìn thấy" cả phần trước và phần sau của một từ trong ngữ cảnh của nó. Điều này cho phép BERT hiểu được cấu trúc câu ngôn ngữ tự nhiên một cách toàn diện hơn.

Transformers và cơ chế Attention

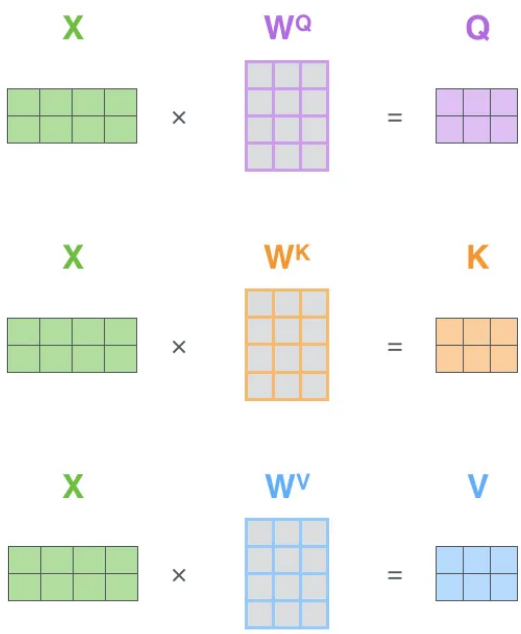
Cơ chế attention[16] là "Chú ý" từ quan trọng nhất trong câu nói. Cơ chế Attention cho phép mạng nơ-ron tập trung vào các phần quan trọng của đầu vào mà nó đang xử lý.

Trong mạng transformer, attention được sử dụng để tạo ra biểu diễn tập trung (contextual representation) của từng từ trong câu. Ta có thể hiểu như sau:

- Đầu vào đầu tiên là các vector embeddings. Nhân mỗi vector embedding đầu vào với 3 ma trận trọng số W_q, W_k, W_v để tạo ra 3 vector q, k, v.
- Vector q và k được dùng để tính trọng số khuếch đại thông tin cho các từ trong câu và vector v là vector biểu diễn của các từ trong câu.

Ví dụ ta có 2 vector embeddings(tương ứng với 2 từ đầu vào “Vui”, “vẻ”) là x_1, x_2. Nhân 2 vector này với 3 ma trân Wq, Wk, Wv ta được tập các vector: $\{q_1, q_2\}$, $\{k_1, k_2\}$, $\{v_1, v_2\}$. Để tính toán vector biểu diễn cho từ “Vui”. Đầu tiên ta cần tính trọng số khuếch đại thông tin cho mỗi từ(gọi là Attention), Attention cho từ "Vui"(a_1) và từ "vẻ"(a_2) được tính theo công thức sau:

$$a_1 = \text{softmax}(q_1 * k_1/\sqrt{(d)})$$
$$a_2 = \text{softmax}(q_2 * k_2/\sqrt{(d)})$$



Hình 1.2: Ma trận trọng số Attention

Trong đó d là số chiều của vector k. Cuối cùng vector biểu diễn cho từ "vui" được tính theo công thức:

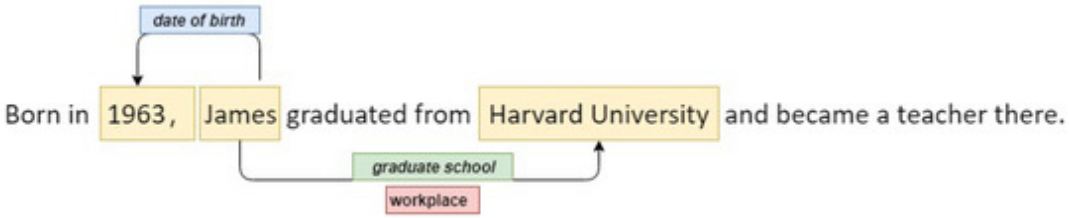
$$z_1 = a_1 * v_1 + a_2 * v_2$$

Tương tự với cơ chế này ta có multi-head Attention[16]. Ý nghĩa của cơ chế multi-head này là để tăng thêm phần chắc chắn trong việc quyết định thông tin nào cần khuếch đại, thông tin nào cần bỏ qua.

Multi-head Attention được thiết kế với 8 lớp self-attention kiến trúc giống hệt nhau nhưng trọng số của 3 ma trận W_q, W_k, W_v khác nhau. Việc tính toán của 8 layer này được thực hiện song song. Các vector biểu diễn qua mỗi lớp self-attention sẽ được nối lại với nhau sau đó được nhân với một ma trận trọng số W_o để nén thông tin từ 8 vector(8 vector này cùng biểu diễn cho 1 từ) thành một vector duy nhất.

1.3 Trích xuất mối quan hệ

Trích xuất mối quan hệ [17] (Relation Extraction) là quá trình xác định và trích xuất các mối quan hệ giữa các thực thể trong văn bản tự nhiên. Nhiệm vụ của Relation Extraction là tìm ra các mối quan hệ ngữ nghĩa, logic hoặc ngữ cảnh giữa các cặp thực thể (Subject, Object, Relation) trong một câu hoặc đoạn văn.



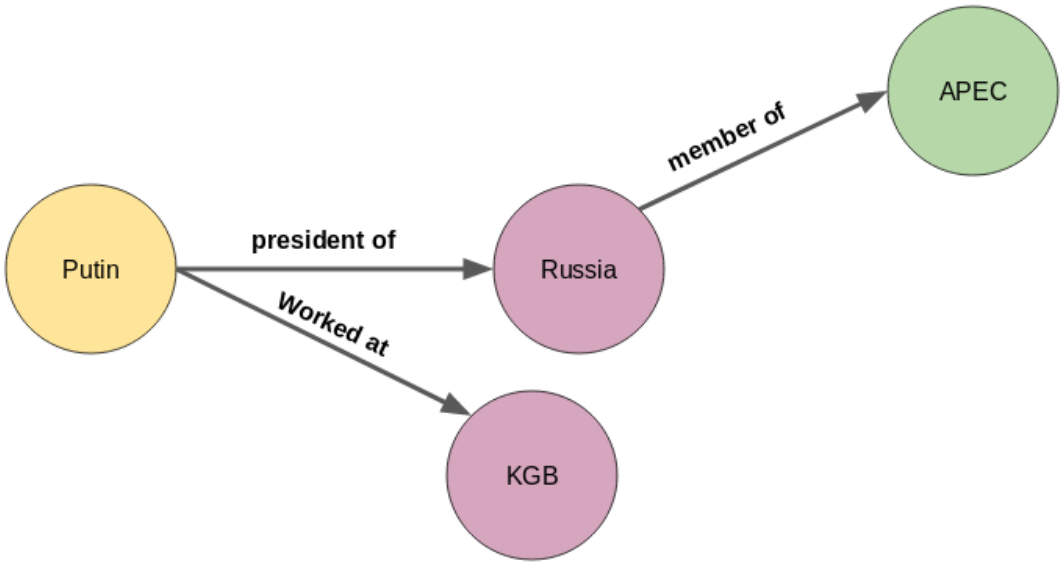
Hình 1.3: Relation Extraction

Để xác định mối quan hệ Relation extraction thường được chia nhỏ thành 2 mô hình chính

- **Named Entity Recognition (NER):** Mô hình sẽ xác định các entity trong câu hoặc đoạn văn và lấy ra cặp subject và object tương ứng
- **Classification:** Mô hình sẽ phân loại cặp thực thể đã được nhận diện trong mô hình NER để tìm ra nhóm quan hệ phù hợp cho cặp chủ thể được xác định

Ngoài NER ta cũng có thể thay thế nó bằng mô hình Word Segmentation để trích xuất ra cặp chủ thể cần xác định nhưng vì hiệu quả của mô hình không cao trong bài luận văn này nhóm sẽ không đề cập tới mô hình này.

Có thể thấy mô hình Relation Extraction là một mô hình gồm có 2 công việc nhỏ đó là nhận diện được cặp chủ thể và phân loại quan hệ của cặp từ trong câu từ đó cho ra được 1 bộ liên hệ nhau từ điều này mô hình sẽ cho ra được các bộ thông tin liên kết giữa các chủ thể (nodes) thông qua các quan hệ (edges).



Hình 1.4: Relation Graph

1.4 Mô hình rút trích quan hệ Casrel

Casrel[17] là mô hình trích xuất bộ ba quan hệ (Subject, Relation, Object) có trong một câu nhằm phục vụ cho quá trình xây dựng bộ dữ liệu biểu đồ tri thức. Trong các mô hình rút trích quan hệ mô hình Casrel[17] đã cho thấy được bản thân có kết quả ấn tượng khi được so sánh với các mô hình khác như NovelTagging[21], CopyR[20] và GraphRel[5].

Method	NYT			WebNLG		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
NovelTagging (Zheng et al., 2017)	62.4	31.7	42.0	52.5	19.3	28.3
CopyR$_{OneDecoder}$ (Zeng et al., 2018)	59.4	53.1	56.0	32.2	28.9	30.5
CopyR$_{MultiDecoder}$ (Zeng et al., 2018)	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel$_{1p}$ (Fu et al., 2019)	62.9	57.3	60.0	42.3	39.2	40.7
GraphRel$_{2p}$ (Fu et al., 2019)	63.9	60.0	61.9	44.7	41.1	42.9
CopyR$_{RL}$ (Zeng et al., 2019)	77.9	67.2	72.1	63.3	59.9	61.6
CopyR$^{*}_{RL}$	72.8	69.4	71.1	60.9	61.1	61.0
CASREL$_{random}$	81.5	75.7	78.5	84.7	79.5	82.0
CASREL$_{LSTM}$	84.2	83.0	83.6	86.9	80.6	83.7
CASREL	**89.7**	**89.5**	**89.6**	**93.4**	**90.1**	**91.8**

Hình 1.5: Kết quả mô hình Casrel

CasRel được đề xuất bởi vì có sự sáng tạo trong mô hình, thay vì coi các

mối quan hệ là các nhãn rời rạc trên các cặp thực thể, Casrel mô hình hóa các mối quan hệ dưới dạng các hàm ánh xạ chủ thể tới đối tượng. Trích xuất đối với Casrel quan hệ là một quá trình gồm hai bước: đầu tiên xác định tất cả các Subject có thể có trong một câu; sau đó đối với mỗi Object, áp dụng các Label theo Relation cụ thể để xác định đồng thời tất cả các mối quan hệ có thể và Object tương ứng.

1.5 Truy vấn câu hỏi

Question Answering (QA) là một lĩnh vực trong xử lý ngôn ngữ tự nhiên (NLP) có nhiệm vụ tự động trả lời các câu hỏi của người dùng dựa trên một nguồn tri thức cụ thể. Mục tiêu của QA là hiểu câu hỏi từ người dùng và cung cấp câu trả lời chính xác cũng như cụ thể từ nguồn tri thức.

Thông thường, QA có 2 mô hình chạy liên tiếp, chúng hỗ trợ nhau trong quá trình truy vấn và trả lời câu hỏi:

- **Truy vấn văn bản[14] (Information Retrieval)** là mô hình đánh giá các văn bản nó nghĩ rằng sẽ liên quan tới câu hỏi và xếp hạng các văn bản liên quan theo các tiêu chí tương đồng của các vector, từ đó chọn ra đáp án tốt nhất từ các văn bản truy vấn này.

- **Mô hình trích xuất đáp án[3] (Reader Model)** là mô hình trích xuất đáp án từ văn bản và câu hỏi được cung cấp cho mô hình, mô hình thường được xây dựng dưới mô hình học có giám sát (Supervised Learning) để có thể tìm ra các đáp dựa trên những thông tin đã cho.

1.5.1 Truy vấn văn bản

TF-IDF

Phương pháp TF-IDF[12] (Term Frequency-Inverse Document Frequency) là một kỹ thuật được sử dụng trong xử lý ngôn ngữ tự nhiên và truy vấn

thông tin để đánh giá sự quan trọng của một từ trong một văn bản. Bằng cách tính toán tỷ lệ giữa tần suất xuất hiện của một từ trong một văn bản (TF) và tần suất nghịch đảo của từ đó trong tập hợp các văn bản (IDF).

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

- *Term Frequency* đo lường tần suất của một từ xuất hiện trong tài liệu. Vì mỗi tài liệu có độ dài khác nhau, có thể xảy ra trường hợp một từ xuất hiện nhiều lần hơn trong các tài liệu dài hơn so với các tài liệu ngắn hơn. Do đó, tần suất xuất hiện của từ thường được chia cho độ dài của tài liệu để chuẩn hóa.

  $$tf(t, D) = \log(1 + freq(t, d))$$

  Trong đó t là 1 n-gram và d là tài liệu thuộc bộ dữ liệu D. $freq(t, d)$ được coi là tần suất của t xuất hiện trong tài liệu d chia cho toàn bộ số từ trong tài liệu d.

  Ví dụ. Ta có câu sau "Nếu ta chăm chỉ, ta sẽ có một cuộc sống tốt hơn" với unigram t (n-gram = 1) = "ta" thì $freq(t, d) = \frac{2}{12}$ suy ra $tf = \log(1 + \frac{2}{12}) \approx 0.06$.

- *Inverse Document Frequency* đo lường mức độ quan trọng của một từ. Trong quá trình tính toán TF-IDF, tất cả các từ được coi là có cùng mức độ quan trọng. Tuy nhiên, đã biết rằng một số từ có thể xuất hiện nhiều lần nhưng không quan trọng. Do đó, chúng ta cần giảm trọng số của các thuật ngữ phổ biến trong khi tăng trọng số của các thuật ngữ hiếm.

  $$idf(t, D) = \log(\frac{|D|}{\{d \in D : t \in d\}})$$

  Với $|D|$ là số lượng tài liệu trong toàn bộ tập tài liệu chia cho $\{d \in D : t \in d\}$ là số lượng tài liệu trong toàn bộ tập tài liệu chứa từ t.

Ví dụ. Ta có 10 văn bản và tần suất xuất hiện của từ "ta" trong tập dữ liệu là 3 ta có $idf = \log(\frac{10}{3}) \approx 0.52$

Có thể thấy TF-IDF đánh giá mức độ quan trọng của một từ bằng cách tính toán giá trị cho các từ xuất hiện nhiều trong văn bản hiện tại và đồng thời tính toán sự xuất hiện của từ trong toàn bộ tập văn bản. Kết quả là, các từ quan trọng trong một văn bản sẽ có trọng số cao.

Ví dụ. Trọng số từ "ta" trên 2 ví dụ trên là $tfidf(t, d, D) = 0.6 \times 0.52 \approx 0.3$ vậy mức độ quan trọng của từ "ta" trên toàn bộ văn bản là 0.3

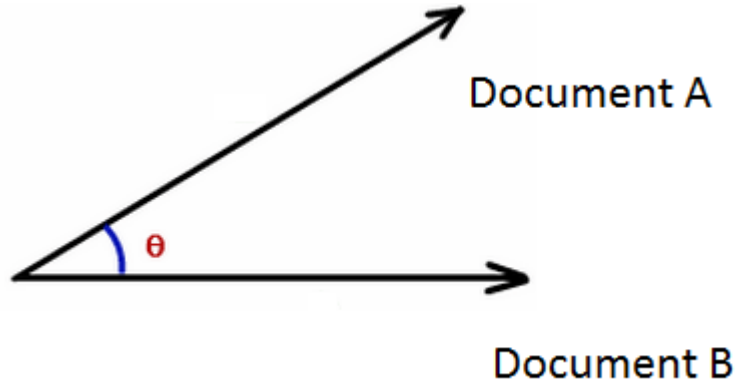
Phương pháp TF-IDF có nhiều ứng dụng, trong đó truy vấn câu hỏi là một trong những ứng dụng của TF-IDF. Khi áp dụng phương pháp này trong truy vấn câu hỏi, chúng ta tính toán giá trị TF-IDF cho từng từ trong câu hỏi và so sánh với giá trị TF-IDF của các từ trong các văn bản khác nhau. Bằng cách này, chúng ta có thể xác định các văn bản liên quan nhất đến câu hỏi dựa trên sự tương đồng về mức độ quan trọng của các từ.

Truy vấn văn bản bằng Cosine similarity

Cosine similarity[10] là một phương pháp dựa trên việc tính toán góc cosin của hai vector biểu diễn văn bản. Mỗi thành phần của vector đại diện cho một thuộc tính hoặc một đặc trưng của văn bản, ví dụ như tần suất xuất hiện của các từ khác nhau trong văn bản (TF-IDF). Giá trị cosine similarity là một số trong khoảng từ -1 đến 1, cho biết mức độ tương đồng giữa hai văn bản. Giá trị càng gần 1 thể hiện sự tương đồng cao hơn giữa hai văn bản , trong khi giá trị càng gần -1 thể hiện sự khác biệt hay trái ngược nhau giữa hai văn bản. Ta có thể tính Cosine similarity như sau :

$$similarity(q, d_i) = \cos(\theta) \ = \frac{q \cdot d_i}{\|q\|\|d_i\|}$$

Trong đó q là văn bản ta muốn query và d_i là văn bản i trong của tập dữ liệu thì $\cos(\theta)$ chính là similarity score trong khoảng $[-1, 1]$



Hình 1.6: Cosine Similarity

Việc truy vấn văn bản bằng cosine similarity có nhiều ứng dụng, như tìm kiếm thông tin, phân loại văn bản, và gợi ý nội dung. Bằng cách so sánh cosine similarity giữa câu truy vấn và các văn bản trong tập dữ liệu, chúng ta có thể xác định các văn bản tương đồng nhất với câu truy vấn và sắp xếp chúng theo độ tương đồng.

1.5.2 Mô hình triết xuất đáp án

Mô hình triết xuất đáp án[3] (Reader Model) giải quyết nhiệm vụ đọc hiểu - trích xuất câu trả lời cho một câu hỏi cụ thể từ một văn bản ngữ cảnh đã cho dựa vào mô hình mạng nơ-ron.

Trong bài nghiên cứu trước, nhóm có sử dụng mô hình MRCQuestionAnswering là mô hình được phát triển trên XLM-RoBerta[18]. XLM-RoBerta[18] là một mô hình pre-train transformers model với một bộ dữ liệu lớn sau sự phát triển của BERT[4].

Đối với mô hình MRCQuestionAnswering đã được cải tiến phù hợp với ngôn ngữ tiếng việt để dễ dàng triết xuất các thông tin từ văn bản cũng như câu hỏi có cấu trúc tiếng việt cụ thể tác giả đã thay thế tách nhỏ các từ như XLM-RoBerta thay vào đó kết hợp nó lại bằng phương pháp cộng sau khi

encode bằng BERT model

1.6 Rút trích từ khóa

Rút trích từ khóa[1] (Keyword Extraction) là quá trình xác định và trích xuất các từ hoặc cụm từ quan trọng và mang tính đại diện trong một văn bản. Mục tiêu của Keyword Extraction là tìm ra những từ khóa quan trọng.

Keyword Extraction có thể được thực hiện bằng nhiều phương pháp khác nhau, bao gồm các phương pháp dựa trên tần suất xuất hiện, ngữ cảnh, phân tích ngữ nghĩa. Đối với khoá luận lần này nhóm quyết định sử dụng kỹ thuật YAKE

YAKE

YAKE[2] (Yet Another Keyword Extractor) là một phương pháp rút trích từ khóa. Nó được thiết kế để xác định các từ khóa quan trọng từ một đoạn văn bản mà không cần sự giám sát hoặc các nguồn dữ liệu bên ngoài.

YAKE[2] sử dụng một kỹ thuật gọi là "statistical language model" để đánh giá tính quan trọng của từng từ trong văn bản.

Quá trình rút trích từ khóa bằng YAKE[2] được chia thành các giai đoạn sau:

- Tiền xử lý văn bản: Đầu tiên, văn bản đầu vào được tiền xử lý để loại bỏ các ký tự đặc biệt, các từ không cần thiết và thực hiện việc tách từ.
- Xác định các thuộc tính của từ: Ta sẽ quyết định xem từ khóa có độ dài tối đa bao nhiêu từ. Chẳng hạn bạn chỉ chấp nhận các ứng viên có số từ tối đa là 3, khi đó ta sẽ có các ứng viên 1-gram, 2-gram và 3-gram bằng cách sử dụng cửa sổ trượt. Tuy nhiên, ta không lấy tất cả các ứng viên này. Các ứng viên từ khóa mà từ bắt đầu hoặc kết thúc của nó có trong danh sách stopword sẽ bị loại bỏ.

- Tính toán điểm quan trọng: Sử dụng các thuộc tính của từ, YAKE tính toán điểm quan trọng cho mỗi từ trong văn bản. Các từ có điểm cao được coi là từ khóa quan trọng. Các tiêu chí chấm điểm như sau:

  – **Yếu tố viết hoa/thường** : Thước đo này xem xét yếu tố viết hoa (title case, upper case) của từ. Các từ viết hoa chữ cái đầu hoặc từ viết tắt (Ex: ASEAN) thường đóng vai trò quan trọng hơn.

  Để đánh giá yếu tố này, chúng ta sẽ đếm số lần xuất hiện của từ w được viết hoa chữ cái đầu (bỏ qua nếu nó đứng đầu câu). Đồng thời, ta cũng đếm số lần xuất hiện của từ w này ở trạng thái viết hoa toàn bộ (upper case).

  $$case(w) = \frac{\max(count(w_{capital}), count(w_{acronym})}{1 + \log(count(w))}$$

  – **Vị trí của từ** : Vị trí của từ trong văn bản cũng là một yếu tố đánh giá mức độ quan trọng của nó trong văn bản. Điều này dựa trên thực tế là phần đầu của văn bản thường khái quát nội dung toàn bài và chứa những từ khóa quan trọng. Do đó, nếu 1 ứng cử viên xuất hiện ở phần đầu của bài viết thì khả năng nó là từ khóa sẽ cao hơn. Đầu tiên, chúng ta sẽ lấy ra vị trí của tất cả các câu (sentence) có chứa từ $Sen(w)$. Tiếp theo, ta sẽ tính toán đặc trưng vị trí của từ bằng cách lấy vị trí trung vị (median) và tính với công thức sau:

  $$pos(w) = \log(\log(3 + median(sen(w))))$$

  – **Tần suất của từ** : Tần suất của các từ được tính và chuẩn hóa bởi 1-standard deviation so với giá trị trung bình

  $$freq(w) = \frac{count(w)}{mean(count(w)) + std(count(w))}$$

– **Ngữ cảnh của từ** Đặc trưng ngữ cảnh của từ (word relatedless) giúp xác định mức độ liên quan của một từ với ngữ cảnh của nó. Ta sẽ đếm xem có bao nhiều từ khác nhau đứng bên cạnh (đằng trước, đằng sau) của mỗi từ. Nếu có 1 từ nào đó thường xuyên đứng cạnh nhiều từ khóa khác nhau thì có thể nó là một stopword.

$$rel(w) = 1 + (WL + WR) * \frac{count(w)}{\max(count)} + (PL + PR)$$

Trong đó:

WL là (Số lượng từ khác nhau ở bên trái) / (tổng số từ xuất hiện bên trái)

WR là (Số lượng từ khác nhau ở bên phải) / (tổng số từ xuất hiện bên phải)

PL (Tổng số từ bên trái) / (max count)

PR (Tổng số từ bên phải) / (max count)

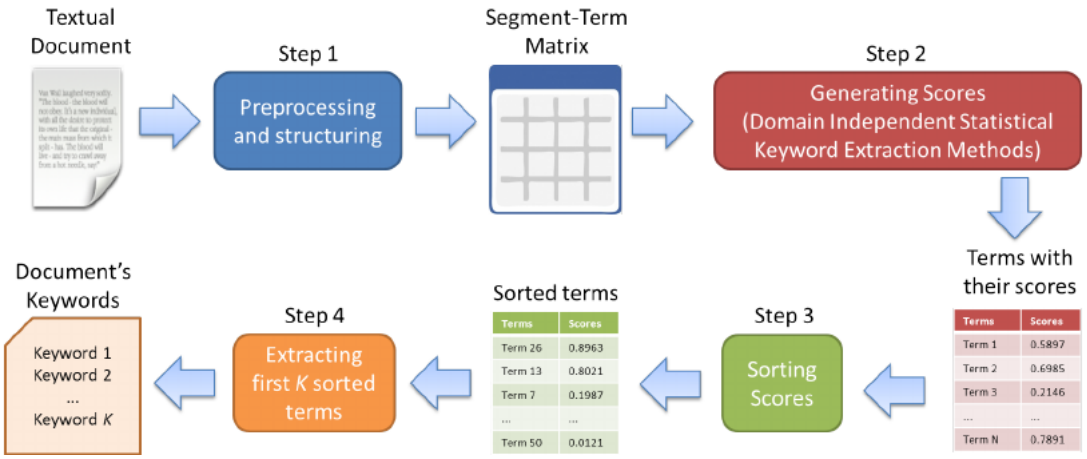
– **Từ trong các câu khác nhau** : Đặc trưng này xác định tần suất một từ xuất hiện ở trong các câu (sentence) khác nhau. Một từ xuất hiện trong nhiều câu khác nhau thì sẽ có điểm cao hơn.

$$diff(w) = \frac{number\ of sentences\ w\ occurs\ in}{total\ sentences}$$

5 đặc trưng trên được kết hợp lại theo công thức dưới đây để cho ra điểm số cuối cùng của mỗi từ

$$score(w) = \frac{rel * pos}{case + \frac{freq}{rel} + \frac{diff}{rel}}$$

- Dựa trên điểm quan trọng, YAKE chọn ra một tập hợp các từ khóa có tính quan trọng cao nhất từ văn bản.



Hình 1.7: Keyword Extraction

Một trong những ưu điểm của YAKE được nhóm lựa chọn vì phương pháp YAKE có ưu điểm là không yêu cầu dữ liệu huấn luyện và có khả năng xử lý các đoạn văn bản tự nhiên. Điều này có nghĩa là không cần phải có một tập dữ liệu lớn và đã được gán nhãn để huấn luyện mô hình.

Tuy nhiên, cần lưu ý rằng YAKE có thể không phù hợp cho các tác phẩm văn bản lớn hoặc các tài liệu chuyên ngành đặc thù. Ngoài ra, việc định cấu hình và điều chỉnh các tham số trong YAKE cũng có thể ảnh hưởng đến kết quả rút trích từ khóa.

1.7 Khoảng cách Levenshtein

Trong lý thuyết thông tin, và khoa học máy tính, khoảng cách Levenshtein[6] là một thước đo chuỗi để đo sự khác biệt giữa hai chuỗi. khoảng cách Levenshtein giữa hai từ là số lần chỉnh sửa một ký tự tối thiểu (chèn, xóa hoặc thay thế) cần thiết để thay đổi từ này thành từ khác.

Công thức Levenshtein[6] được biểu diễn như sau:

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + \delta(a_i, b_j) \end{cases} & \text{if } \min(i,j) > 0 \end{cases}$$

Trong đó:

- $\text{lev}_{a,b}(i, j)$ là khoảng cách Levenshtein giữa hai chuỗi a và b có độ dài lần lượt là i và j.
- $\delta(a_i, b_j)$ là hàm đánh giá khác nhau giữa hai ký tự a_i và b_j. Nó bằng 0 nếu $a_i = b_j$ và bằng 1 nếu $a_i \neq b_j$.

1.7.1 Fuzzy Wuzzy

Fuzzy Wuzzy[8] là một thư viện Python được sử dụng để đo lường độ tương đồng (similarity) giữa hai chuỗi văn bản. Fuzzy Wuzzy[8] cung cấp các công cụ và thuật toán phức tạp để so sánh các chuỗi văn bản dựa trên sự tương đồng về ngữ nghĩa (semantic similarity) thay vì chỉ dựa trên sự trùng khớp đúng (exact matching). Thư viện Fuzzy Wuzzy sử dụng các phương pháp xử lý ngôn ngữ tự nhiên (NLP) như đồng âm (phonetic matching), chuẩn hoá (string normalization), và thuật toán Levenshtein (distance) để tính toán độ tương đồng giữa các chuỗi văn bản. Điều này giúp Fuzzy Wuzzy có thể xử lý những trường hợp mà các thuật toán so sánh chuỗi thông thường không thể làm được, như khi có các sai sót chính tả, từ viết sai hoặc các biến thể từ ngữ.

CHƯƠNG 2:

GIỚI THIỆU

2.1 Vấn đề truy vấn văn bản

Trong các bài nghiên cứu gần đây nhóm có triển khai mô hình truy vấn câu hỏi với 2 bộ dữ liệu tiếng Việt là SquAD[13] và wikiQA[19] với bộ dữ liệu đã được tách mỗi đoạn văn trong bộ dữ liệu thành các đoạn nhỏ hơn để giảm thiểu tính toán trên toàn bộ văn bản khi đó SquAD[13] là một bộ dữ liệu lớn với 45.052 câu và wikiQA[19] với kích thước dữ liệu nhỏ có 115 câu:

- WikiQA : 115 câu
- SQuAD : 45.052 câu

Dataset	**Exact match**	**F1 score**
WikiQA	0.2321	0.5212
SQuAD	0.2900	0.4407

Bảng 2.1: Previous result

So sánh kết quả thực nghiệm của mô hình trên ta thấy SquAD[13] có tuy Extract match cao hơn SquAD[13] nhưng F1 score có phần thấp hơn, lý giải cho vấn đề này mô hình QA của SQuAD[13] có dữ liệu khá lớn tuy rằng với dữ liệu lớn thì việc triết xuất đáp án trong câu Extract match là rất tốt do dữ liệu đã đủ nhiều và học được các feature phù hợp cho mô hình để trích xuất chính xác từ văn bản truy vấn đúng. Nhưng đối với F1 score ta có thể được tính bằng công thức $\frac{2 \times P \times R}{P + R}$ trong đó P là Precision và R là Recall, nói cách khác F1-score là trung bình điều hòa của precision và recall có thể dễ dàng nhận thấy SquAD[13] đang có precision và recall thấp hơn WikiQA[19]

vì bộ dữ liệu SquAD[13] quá lớn làm cho mô hình truy vấn thiếu chính xác do chọn sai văn bản truy vấn vì có quá nhiều câu gây nhiễu trong khi đó wikiQA[19] lại có số lượng ít hơn cho ra kết quả truy vấn tốt hơn cho thấy việc tìm kiếm văn bản khá là chính xác và hiệu quả.

Qua đó cho ta thấy ta đang đánh đổi giữa độ chính xác của câu trả lời và khả năng truy vấn bộ văn bản.

2.2 Phương pháp đề xuất phát triển

Để cải thiện cho những hạn chế đó, trong bài luận này nhóm tập trung phát triển mô hình có thể giải quyết việc lưu trữ thông tin dữ liệu lớn. Kết hợp với một số kỹ thuật như trích xuất từ khoá từ câu hỏi để xác định chủ thể, từ đó tìm ra các thực thể và các quan hệ để trích xuất thông tin cần thiết để truy vấn văn bản trở nên tối ưu hơn và chính xác hơn.

2.2.1 Trích xuất quan hệ với BERT

Đối với trích xuất quan hệ mô hình được chia thành 2 phần đó chính là xác định Subject và xác định Relation cho mỗi Object tương ứng:

BERT Encoder

Một trong những điều quan trong bất kỳ mô hình học sâu nói chung và xử lý ngôn ngữ tự nhiên nói riêng đều cũng phải có đó chính là phải trích xuất được các feature cần thiết cho việc huấn luyện.

Với sự phổ biến và khả năng của BERT[4] nhóm quyết định sử dụng BERT[4] để trích xuất các đặc trưng từ bằng cách sử dụng mô hình BERT Encoder. Mô hình sẽ giúp ta trích xuất các thông tin từ các token thành giá trị cần thiết trong việc tính toán trọng số các từ quan trọng trong câu từ đó cho ra các Feature thiết yếu để có thể tính toán cho mô hình Relation Extraction được đề cập tiếp theo

Relation Extraction

Sau khi đã trích xuất được thông tin các thông tin từ BERT[4] ta sẽ sử dụng nó như một Feature quan trọng trong mô hình Relation Extraction

Mục tiêu chính ở mục này đó chính là tìm ra ứng với mỗi subject ta sẽ tìm ra cho nó một hoặc nhiều object sao cho cặp subject, object này có quan hệ với nhau

$$f(s, r) \rightarrow o.$$

Để giải quyết vấn đề này mô hình được chia thành 2 mô hình đơn được gọi là Subject Tagger và Object Tagger để có thể tìm ra 2 chủ thể là Subject và Object. Mô hình được dựa theo mô hình Casrel[17] nhưng khác với mô hình Casrel, mô hình được sử dụng BERT-Multilingual để Encode. Mô hình BERT-Multilingual được tối ưu tham số cho 104 quốc gia trong đó có Việt Nam.

Subject Tagger Subject Tagger được thiết kế để nhận ra tất cả các subject có thể có trong câu đầu vào. Chính xác hơn, nó sử dụng hai bộ phân loại nhị phân giống hệt nhau để phát hiện vị trí bắt đầu và kết thúc của các đối tượng tương ứng bằng cách gán cho mỗi token các giá trị (0/1) tuỳ thuộc vào vị trí token hiện tại. Theo đó ứng với mỗi token ta có 2 hàm số sau:

$$p_i^{start_s} = \sigma(w_{start} X_i + b_{start})$$

$$p_i^{end_s} = \sigma(w_{end} X_i + b_{end})$$

Trong đó:

- $p_i^{start\_s}$ và $p_i^{end\_s}$ là 2 giá trị xác suất "start" và "end" của token thứ i trong câu. Token sẽ được gán là 1 nếu vượt ngưỡng xác suất nhất định và 0 nếu là giá trị còn lại.

- X_i là giá trị token sau khi được BERT encode.
- W_{start} và W_{end} chính là các trọng số cần học (Trainable Weight) và b chính là chỉ số bias của mô hình.
- Cuối cùng σ là activation Sigmoid để tăng độ phức tạp của mô hình.

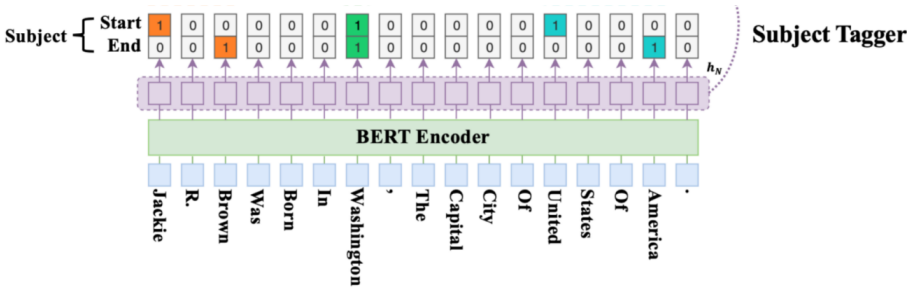
Đối với Subject Tagger hàm mục tiêu chính là maximizes hàm log-Likelihood được sinh bởi các subject trong câu

$$\sum_{s \in T_j} \log(P_\theta(s|X)) \tag{2.1}$$

$$\text{Với } P_\theta(s|X) = \prod_{t \in \{start\_s, end\_s\}} \prod_{i=1}^{L} (P_i^t)^{I\{y_i^t=1\}} (1 - P_i^t)^{I\{y_i^t=0\}}$$

Trong đó

- s subject trong bộ $\{s, v, o\}$ được định nghĩa bởi $T_j = \{s, v, o\}$ với j là một câu thuộc bộ dữ liệu.
- L là độ dài của câu.
- $I\{z\} = 1$ khi z đúng và 0 đối với trường hợp còn lại.
- y_i^{start} là giá trị nhị phân của của hàm start và y_i^{end} là giá trị nhị phân của của hàm end tại token thứ i trong câu.
- θ là các tham số (Parameter) mà hàm mục tiêu cần học bao gồm $\{W_{start}, W_{end}, b_{start}, b_{end}\}$.



Hình 2.1: Subject Tagger

Object Tagger Về cơ bản Object Tagger sử dụng cấu trúc tương đồng với Subject Tagger. Điểm khác biệt giữa 2 mô hình này nằm ở đầu vào và mục tiêu của mô hình nếu như Subject Tagger đầu vào chỉ là các giá trị được encode từ BERT Encoder thì được lấy thêm các thông tin từ các feature subject mà Subject Tagger nhân diện trước đó được kết hợp với các thông tin từ BERT Encoder và đồng thời Object Tagger xác định Relation với Object tương ứng. Mô hình sẽ dựa theo các đặc trưng các ứng viên và phân loại cho cặp (Subject, relation) này cho một Object Candidate.

Ví dụ. Mối quan hệ "Birth_place" được xác định cho "Jackie R. Brown" và Object candidate "Washington" nên sẽ tìm đánh 1 cho token ở vị trí start và token ở vị trí end của Object Candidate "Washington" được thể hiện ở 2.2

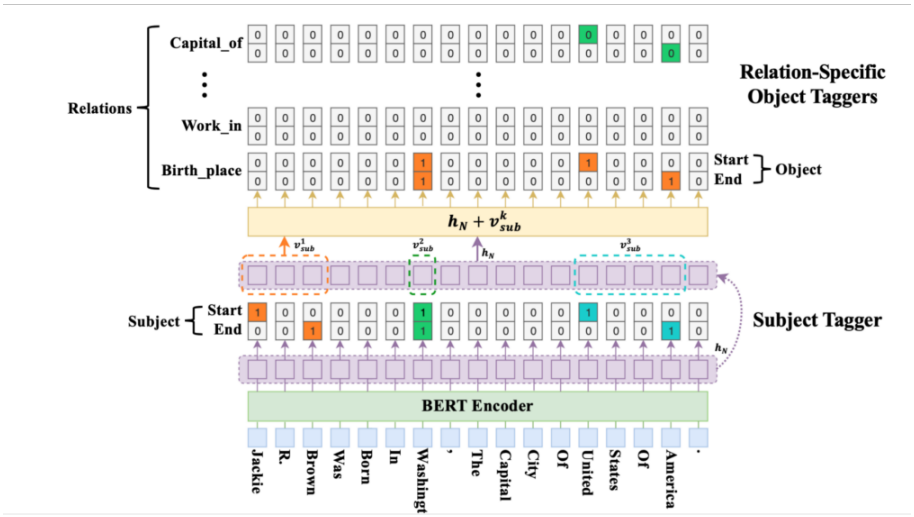
Theo đó ứng với mỗi token ta có 2 hàm số sau:

$$p_i^{start\_o} = \sigma(w_{start}^r(X_i + v_{sub}^k) + b_{start}^r)$$

$$p_i^{end\_o} = \sigma(w_{end}^r(X_i + v_{sub}^k) +_{end}^r)$$

Trong đó:

- Tương tự như Subject Tagger $p_i^{start\_o}$ và $p_i^{end\_o}$ là 2 giá trị xác suất "start" và "end" của token thứ i trong câu.
- $\{W_{start}^r, W_{end}^r, b_{start}^r, b_{end}^r\}$ là các tham số cần học (Trainable Parameter).



Hình 2.2: Object Tagger

- v^k_{sub} là vector của subject thứ k được nhận diện bởi Subject Tagger.

Vì có cấu trúc tương tự như Subject Tagger nên Object Tagger cũng sử dụng tương tự hàm mục tiêu với Subject Tagger nhưng có chút khác biệt để phù hợp với việc dự đoán và phân loại mỗi Object cho Relation Nhưng ta có thêm một hàm mục tiêu đối với object là "null" ($o_\oslash$) cũng tương tự như Object Tagger nhưng $y_i^{start\_o_\oslash}$ và $y_i^{end\_o_\oslash}$ sẽ bằng 0 trên toàn bộ i.

Ví dụ. Mối quan hệ "Work_in" không tồn tại giữa đối tượng được phát hiện "Jackie R. Brown" và Object candidate "Washington". Do đó, việc chọn Object cho quan hệ "Work_in" sẽ không xác định cho "Washington", tức là, đầu ra của cả vị trí bắt đầu và kết thúc đều 0 cũng được thể hiện ở 2.2

$$\sum_{r \in T_j|s} \log(P_\phi(o|s,X)) + \sum_{r \in R \setminus T_j|s} \log(P_\phi(o_\oslash|s,X)) \tag{2.2}$$

$$\text{Với } P_\phi(o|s,X) = \prod_{t \in \{start\_o, end\_o\}} \prod_{i=1}^{L} (P_i^t)^{I\{y_i^t=1\}} (1-P_i^t)^{I\{y_i^t=0\}}$$

Và cuối cùng, dựa theo công thức 2.1 và 2.2 ta có hàm mục tiêu $J(\Theta)$ cho tổng mô hình theo công thức sau:

$$\sum_{j=1}^{|D|}[\sum_{s\in T_j}\log(P_\theta(s|X)) + \sum_{r\in T_j|s}\log(P_\phi(o|s,X)) + \sum_{r\in R\setminus T_j|s}\log(P_\phi(o_\oslash|s,X))]$$

Trong đó

- $|D|$ là toàn bộ dữ liệu được huấn luyện.
- r là quan hệ trong tập quan hệ R của bộ dữ liệu
- $\Theta = \{\theta, \{\phi_r\}_{r\in R}\}$.

Do giới hạn của phần cứng mô hình được huấn luyện với các tham số phù hợp với phần cứng môi trường cho phép. Cụ thể mô hình được train với 10 Epochs, độ dài tối đa mà BERT xử lý (BERT_MAX_LEN) sẽ là 512 và thuật toán tối ưu được sử dụng là Adam Stochastic và được "xáo trộn" (Shuffle) trên các mini-batch

CHƯƠNG 3:

DATASET

3.1 NYT

NYT[15] là một dữ liệu mở được thu thập từ báo The New York Times. Tập dữ liệu cung cấp thông tin đa dạng về sự kiện.

Mỗi bài báo trong tập dữ liệu được chú thích bằng các thông tin như tác giả, ngày xuất bản, danh sách thực thể (như tên riêng, địa điểm, tổ chức) và các mối quan hệ giữa chúng nhằm để nghiên cứu huấn luyện các mô hình relation extraction bằng tiếng anh với 1.18M câu và 24 quan hệ. Nhưng bộ dữ liệu được sử dụng để huấn luyện trong khoá luận đã được tối giản và giảm bớt còn 56195 câu cho train set, 5000 cho test set và 5000 cho validation set.

Nhờ kích thước lớn và đã được xử lý một cách cụ thể bộ dữ liệu NYT dễ dàng đạt được các kết quả đáng kể trong việc huấn luyện mô hình triết xuất quan hệ.

Dataset	**Precision**	**Recall**	**F1-Score**
NYT	89.7	89.5	89.6

Nhưng trong khoá luận lần này ta sẽ tập trung nhiều hơn vào mô hình tiếng việt để có thể truy vấn nên cụ thể mô hình sẽ được huấn luyện trên bộ dữ liệu dưới đây.

3.2 VLSP 2020

Tuy Dataset không được public nhưng nhóm đã xin phép và được phép sử dụng cho mục đích nghiên cứu và thử nghiệm của tổ chức VLSP[9] (Vietnamese Language and Speech Processing).

Bộ dữ liệu được sử dụng lại và phát triển từ nhiệm vụ VLSP-2018 Named Entity Recognition for Vietnamese[11] (VNER 2018), được sưu tầm từ các báo điện tử đăng tải trên mạng. Được chú thích bằng ba loại thực thể: LOCATION (LOC), ORGANIZATION (ORG) và PERSON (PER).

Dataset đã được nhóm đánh label thủ công để thử nghiệm cho mô hình Relation Extraction cho tiếng Việt nên còn chưa ổn định cũng như không phải chính thức từ VSLP. Quan hệ được quyết định bởi entity subject của nó với entity object bao gồm :

- LOC/LOC, LOC/PER, LOC/ORG
- PER/PER, PER/ORG, PER/LOC
- ORG/ORG, ORG/PER, ORG/LOC

Do chưa dataset chưa hoàn thiện nên đối với dataset nay sẽ tập trung nhiều hơn về dữ liệu liên hệ giữa Subject và Object trong câu.

Đối với dataset VLSP có tổng cộng 4178 trong đó 3153 câu cho training, 421 câu cho testing và 631 câu cho validation.

3.2.1 Tiền xử lý dữ liệu

Do dữ liệu được lấy từ dữ liệu báo chí nên bộ dữ liệu này thường gặp các ký tự ngoài mong muốn làm ảnh hưởng đến mô hình triết xuất nên nhóm có xử lý dữ liệu này bằng các phương pháp loại bỏ các thành phần như "Các ký tự đặc biệt, dấu cách bị thừa, các ký tự đặc biệt của HTML,...". Qua đó mô hình triết xuất quan hệ có tăng độ chính xác đáng kể so với ban đầu

CHƯƠNG 4:

KẾT QUẢ VÀ THỰC NGHIỆM

4.1 Thiết lập môi trường thực nghiệm

Với mô hình trích xuất quan hệ mô hình sẽ được đánh giá dựa trên các chỉ số (**Golden numbers, Correct numbers, Predict numbers**) các giá trị này lần lượt là số lượng cặp quan hệ cần trích xuất (**Golden numbers**), số lượng cặp quan hệ mô hình dự đoán đúng trên toàn bộ dữ liệu (**Correct numbers**), và số lượng cặp quan hệ mô hình đã dự đoán (**Predict numbers**) được dùng để tính 2 giá trị *Precision* (P) và *Recall* (R) và *F1-Score* của mô hình trích xuất.

$$P = \frac{Correct\ Numbers}{Predict\ numbers}$$
$$R = \frac{Correct\ numbers}{Golden\ numbers}$$

Với mô hình truy vấn câu hỏi sẽ có **Extract Match** để đánh giá độ chính xác kết quả đáp án. Extract Match đòi hỏi kết quả của mô hình dự đoán đúng chính xác không được thừa hay thiếu kí tự nào trong đáp án.

4.2 Kết quả Đánh giá

4.2.1 Mô hình trích xuất quan hệ

Với các tham số và tiền xử lý mô hình đã đề cập các mục trên mô hình có cho ra kết quả của của 421 câu từ tập test như sau:

- Golden numbers : 3383 cặp quan hệ.
- Correct numbers : 2035 cặp quan hệ.

- Predict numbers: 3343 cặp quan hệ.

Qua số liệu trên ta có thể thấy mô hình cũng cho kết quả lần lượt với các chỉ số Precision, Recall và F1-Score lần lượt là:

- Precision : 60.87.
- Recall : 60.15.
- F1-Score: 60.51.

Nhận xét : Mô hình hiện tại chưa ổn định khi mô hình chưa đủ tổng quát khi chỉ đạt khoảng 60% trên test set nhưng ở góc nhìn khác bộ dữ liệu huấn luyện còn có nhiều vấn đề cũng như các label chưa được chuẩn hoá nên việc mô hình dự đoán không tốt vẫn là phần cần và nên cải thiện mô hình và thu thập dữ liệu nhiều hơn trong tương lai.

4.2.2 Mô hình truy vấn câu hỏi với trích xuất quan hệ

Trong thử nghiệm này ta sẽ đánh giá giữa mô hình việc không sử dụng bộ dữ liệu mà mô hình trích xuất quan hệ đã trích xuất

Dữ liệu

Trong mục này dữ liệu được thử nghiệm đó là dữ liệu SquAD là bộ dữ liệu được dịch từ bộ câu hỏi standford. Bộ dự liệu được dịch bằng mô hình ngôn ngữ của VinAI. Nên có thể sẽ có nhiều câu trả lời sẽ không khớp với câu từ trong văn bản được truy vấn nên chỉ lấy 100 câu hỏi được kiểm tra thủ công nhằm đảm bảo tính khả thi của mô hình.

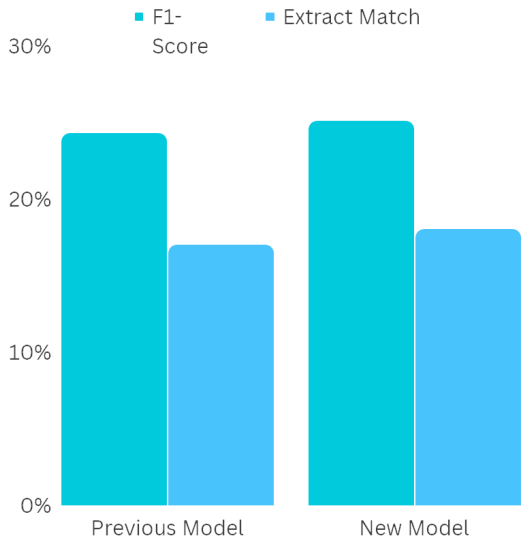
Môi trường thí nghiệm

Đối với các văn bản được truy vấn bằng Cosine similarity số lượng tối đa (Top K) truy vấn sẽ là 10 văn bản có số điểm cao nhất.

Để có thể lấy ra các thực thể liên quan mà được đề cập tới câu hỏi ta dùng mô hình rút trích từ khoá bằng thuật toán YAKE với tham số sẽ sử dụng là n-gram = 1, cũng như sử dụng từ khoá quan trọng nhất là từ khoá và sử dụng Fuzzywuzzy để tìm kết quả tương đối để truy vấn thực thể trong biểu đồ tri thức được xây dựng từ mô hình trích xuất quan hệ.

Kết quả

Sau khi chạy 100 câu hỏi từ bộ dữ liệu SquAD[13] ta có kết quả sau



Hình 4.1: Metric Evaluation

Qua kết quả cho thấy mô hình đã có cải thiện một phần nào đó so với mô hình không có trích xuất quan hệ.

Nhận xét: Trong quá trình kiểm tra giữa 2 mô hình tuy rằng việc sử dụng mô hình trích xuất quan hệ có kết quả khá tốt nhưng vì dữ liệu huấn luyện bị giới hạn bởi Location, Person, Organization nên có nhiều thực thể khác mô hình không thể nhận diện được và thường xuyên bỏ trống làm cho rất nhiều mô hình không thể truy vấn. Ngoài ra mô hình YAKE là mô hình phụ thuộc nhiều vào các tham số cũng như dữ liệu đầu vào nên nhiều câu hỏi phức tạp hoặc quá chung chung làm mô hình khó trích xuất được từ khoá phù hợp để tìm thấy các thực thể phù hợp để truy vấn thông tin.

KẾT LUẬN

Trong khoá luận này, nhóm đã thực hiện được các công việc sau đây.

1. Xây dựng được biểu đồ tri thức việc trích xuất quan hệ các thực thể trong câu.
2. Cải thiện thêm được các hạn chế mà nghiên cứu trước chưa hoàn thiện.

Trong tương lai, sau khi khoá luận này kết thúc để đảm bảo tính khả thi và độ hữu ích cung cấp một công cụ hữu ích và linh hoạt cho việc xử lý ngôn ngữ tự nhiên, giúp người dùng truy xuất thông tin một cách hiệu quả và chính xác. Trong tương lai, nhóm có thể sẽ tiếp tục xây dựng và phát triển mô hình này với một bộ dữ liệu được chỉnh chu và cải thiện hơn bằng cách thu thập một bộ dữ liệu phong phú và đa dạng hơn, bao gồm nhiều nguồn thông tin khác nhau như luật, báo chí và các nguồn dữ liệu lớn khác. Bằng việc có một bộ dữ liệu chỉnh chu hơn và phong phú hơn, cũng có thể hy vọng mô hình có được ứng dụng trong nhiều lĩnh vực khác nhau. Ví dụ áp dụng mô hình truy vấn thông tin trong các lĩnh vực như luật pháp, báo chí hoặc tìm kiếm thông tin từ các nguồn dữ liệu lớn.

Mặc dù đã cố gắng hết sức nhưng do thời gian và khả năng có hạn nên khoá luận cũng có nhiều thiếu sót cũng như có nhiều kết quả chưa quá xuất sắc. Rất mong được sự đóng góp của quý thầy cô, các bạn để luận văn được hoàn chỉnh hơn.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. A text feature based automatic keyword extraction method for single documents. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *Advances in Information Retrieval*, pages 684–691, Cham, 2018. Springer International Publishing.

[2] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[3] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. *CoRR*, abs/1905.05733, 2019.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy, July 2019. Association for Computational Linguistics.

[6] Rishin Haldar and Debajyoti Mukhopadhyay. Levenshtein distance technique in dictionary lookup methods: An improved approach. *CoRR*, abs/1101.1232, 2011.

[7] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[8] Krishna Kalyanathaya, Akila D., and Suseendran G. A fuzzy approach to approximate string matching for text retrieval in nlp. *Journal of Computational Information Systems*, 15:26–32, 05 2019.

[9] Vu Tran Mai, Hoang-Quynh Le, Duy-Cat Can, Thi Minh Huyen Nguyen, Tran Ngoc Linh Nguyen, and Thanh Tam Doan. Overview of VLSP RelEx shared task: A data challenge for semantic relation extraction from Vietnamese news. In *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*, pages 92–98, Hanoi, Vietnam, December 2020. Association for Computational Lingustics.

[10] Vishaka Arjun Mandge and Meenakshi A. Thalor. Revolutionize cosine answer matching technique for question answering system. In *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 335–339, 2021.

[11] Huyen T M Nguyen, Quyen T Ngo, Luong X Vu, Vu M Tran, and Hien T T Nguyen. Vlsp shared task: Named entity recognition. *Journal of Computer Science and Cybernetics*, 34(4):283–294, Jan. 2019.

[12] Shahzad Qaiser and Ramsha Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181, 07 2018.

[13] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[14] Álvaro Rodrigo, Joaquín Pérez-Iglesias, Anselmo Peñas, Guillermo Garrido, and Lourdes Araujo. A question answering system based on information retrieval and validation. 12 2010.

[15] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[17] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, Online, July 2020. Association for Computational Linguistics.

[18] Shu-Yi Xie, Jian Ma, Haiqin Yang, Lian-Xin Jiang, Yang Mo, and Jianping Shen. PALI at semeval-2021 task 2: Fine-tune xlm-roberta for word in context disambiguation. *CoRR*, abs/2104.10375, 2021.

[19] Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[20] Xiaofeng Yu and Wai Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Coling 2010: Posters*, pages 1399–1407, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[21] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada, July 2017. Association for Computational Linguistics.