ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN
Tel. +84 236 3736.949, Fax. +84 236 3842.771
Website: http://www.dut.udn.vn/KhoaCNTT, E-mail: cntt@dut.udn.vn

BÁO CÁO MÔN HỌC

LẬP TRÌNH TÍNH TOÁN

ĐỀ TÀI:

PHƯƠNG PHÁP XÁC ĐỊNH ĐỘ TƯƠNG ĐỒNG NGỮ NGHĨA GIỮA HAI BẢN VĂN VÀ ỨNG DỤNG VÀO SEMANTIC SEARCH

SINH VIÊN : **Thái Thị Thu Loan – 102180254**
Ngô Văn Anh Tuấn – 102180278
LỚP : **18TCLC-NHẬT**
CBHD : **Ph.D Nguyễn Văn Hiệu**

Mục lục

Đà Nẵng, 01/2020

BẢNG ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH PHÂN CÔNG CÔNG VIỆC CỦA CÁC THÀNH VIÊN NHÓM

I. Danh sách thành viên:

1. Thái Thị Thu Loan *(18TCLC_Nhat, MSSV: 102180254)*
2. Ngô Văn Anh Tuấn *(18TCLC_Nhat, MSSV: 102180278)*

II. Đánh giá mức độ hoàn thành công việc

Điểm	**Bảng đánh giá mức độ hoàn thành công việc**		**Mức độ hoàn thành**
	Thái Thị Thu Loan	Đọc tài liệu, code, viết báo cáo và làm slide	50% trên tổng công việc cả nhóm
	Ngô Văn Anh Tuấn	Đọc tài liệu, code và viết báo cáo	50% trên tổng công việc cả nhóm

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

BẢNG KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

LỜI CẢM ƠN

Trước tiên với tình cảm chân thành và sâu sắc nhất, cho phép chúng em được bày tỏ lòng biết ơn đến tất cả mọi thành viên trong nhóm đã cùng nhau vất vả để có thể hoàn tất quá trình nghiên cứu. Đặc biệt là thầy Nguyễn Văn Hiệu của Khoa Công nghệ thông tin đã luôn nhiệt tình hướng dẫn động viên để chúng em hoàn thành tốt công việc.

Trong quá trình nghiên cứu cũng như làm bài báo cáo, khó tránh khỏi sai sót, rất mong thầy, cô thông cảm và bỏ qua. Đồng thời do hạn chế về kiến thức và kinh nghiệm thực tiễn, những khuất mắc bỡ ngỡ còn đó. Chúng em rất mong nhận được những ý kiến, đóng góp quý báu của thầy cô và các bạn để đề tài nghiên cứu này được hoàn thiện.

Một lần nữa xin chúc các thầy cô thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

MỞ ĐẦU

I. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

Trong thời đại của cách mạng công nghệ 4.0, Internet ngày càng trở nên phổ biến và đóng vai trò quan trọng đối với đời sống của mỗi người. Con người ngày càng phụ thuộc vài Internet trong các hoạt động hàng ngày, và cứ mỗi ngày trôi qua thì số lượng người sử dụng Internet lại càng tăng lên.

Theo thống kê cho thấy, tại Việt nam có 64 triệu người trên tổng số 97 triệu dân sử dụng Internet, tăng đến 28% so với năm 2017. Cùng với sự phát triển của công nghệ, khối lượng dữ liệu khổng lồ phát triển không ngừng và nhu cầu tìm kiếm ngày càng tăng của người dùng, việc truy hồi thông tin (Information retrieval - IR) đang ngày càng trở nên quan trọng và nhận được nhiều chú ý từ các chuyên gia trong lĩnh vực công nghệ thông tin. Những nghiên cứu trong lĩnh vực này đã mở ra nhiều phương pháp giúp con người tìm được những thông tin cần thiết trong kho dữ liệu đồ sộ với thời gian tìm kiếm ngày một ngắn hơn. Tuy nhiên, sự mở rộng nhanh chóng của web và sự đa dạng hóa thông tin tăng đáng kể độ khó trong việc tuy hồi thông tin.

Semantic search đã thể hiện tiềm năng to lớn trong việc cải thiện hiệu năng của việc truy hồi. Khác với những kĩ thuật truyền thống chỉ tập trung vào tần số xuất hiện của các từ khóa, semantic search thiên về cố gắng hiểu ý nghĩa đằng sau các tài liệu tìm được và truy vấn của người dùng thông qua việc thêm các semantic tag vào văn bản, với mục đích hệ thống hóa và khái niệm hóa đối tượng chứa trong tài liệu.

Hiện nay chúng ta mới chỉ ở bước đầu trong việc nghiên cứu và phát triển semantic search. Tuy rằng đến bây giờ những công cụ tìm kiếm truyền thống như Google, Yahoo vẫn đang được sử dụng bởi đại đa số nhưng không thể phủ nhận rằng Semantic search nắm giữ tiềm năng to lớn và chắc chắn sẽ mang lại lợi ích không nhỏ cho con người trong việc tìm kiếm và truy hồi thông tin nếu được phát triển xa hơn.

Chính vì thế, đề tài mà chúng tôi nghiên cứu chính là: Xác định độ tương đồng ngữ nghĩa giữa các câu sử dụng Semantic Search.

II. MỤC ĐÍCH VÀ Ý NGHĨA

1. Mục đích

- Hiểu được ngữ cảnh của một câu rõ nhất có thể, từ đó tìm kiếm được các câu có ngữ nghĩa tương tự với câu đó.
- Xử lý ngôn ngữ tự nhiên để lọc được nhiều dữ liệu hơn, ít spam hơn, hiểu rõ hơn về mục đích tìm kiếm của người dùng và dựa vào ý nghĩa theo ngữ cảnh để cung cấp các kết quả có liên quan.
- Xác định và loại bỏ nội dung có chất lượng thấp.

2. Ý nghĩa

- Sử dụng semantic search có thể tối đa hóa khả năng người dùng có được trải nghiệm tìm kiếm tốt nhất có thể, từ đó thu hút thêm lượng người dùng cho các công cụ tìm kiếm.
- Mở rộng và ứng dụng semantic search trong các lĩnh vực như tóm tắt văn bản, hệ thống đề xuất, …

III. PHƯƠNG PHÁP THỰC HIỆN

- Phương pháp phân tích tổng hợp từ tài liệu.
- Phương pháp thống kê, điều tra.
- Phương pháp phân tích thiết kế hệ thống.
- Phương pháp thử nghiệm, đánh giá kết quả.
- Phương pháp tổng hợp và phân tích lý thuyết.

IV. BỐ CỤC ĐỒ ÁN

Đồ án bao gồm 4 phần:

Phần MỞ ĐẦU sẽ khái quát về vấn đề mà nhóm đang nghiên cứu cũng như mục đích và ý nghĩa của đề tài.

Phần CƠ SỞ LÝ THUYẾT sẽ trình bày lý thuyết được sử dụng để giải quyết bài toán đặt ra.

Phần MÔ TẢ THIẾT KẾ sẽ phân tích thiết kế của toàn bộ hệ thống.

Phần TRIỂN KHAI VÀ ĐÁNH GIÁ sẽ nêu kết quả của hệ thống sau khi chạy thử và đưa ra các đánh giá đối với hệ thống.

Phần KẾT LUẬN CHUNG VÀ HƯỚNG PHÁT TRIỂN sẽ kết luận lại những gì nhóm đã gặt hái được trong khi thực hiện đề tài và hướng phát triển tiếp theo.

CỞ SỞ LÝ THUYẾT

I. CẤU TẠO CỦA MỘT CÂU

Từ và ngữ sẽ tạo nên thành phần câu – những yếu tố cấu thành nên một câu hoàn chỉnh. Phạm trù ngữ pháp thành phần câu trong hệ thống phân loại lấy cấu trúc chủ - vị làm cơ sở phân biệt các thành phần câu: thành phần chính và thành phần phụ. Thành phần phụ có loại chứa trong thành phần nòng cốt, có loại đứng ngoài hay biệt lập với nòng cốt câu.

Các phạm trù ngữ pháp thành phần câu:

- Thành phần chính: Chủ ngữ và vị ngữ
- Thành phần phụ:
    + Định ngữ
    + Trạng ngữ
    + Hô ngữ

Tuy nhiên, nếu categorize các cụm từ, từ đó theo loại ngữ pháp, chúng ta sẽ không hiểu được ngữ nghĩa nhiều, chỉ biết được câu nói đó có những thành phần gì. Vì vậy, chúng cần được categorize thành từng nhóm "ngữ nghĩa".

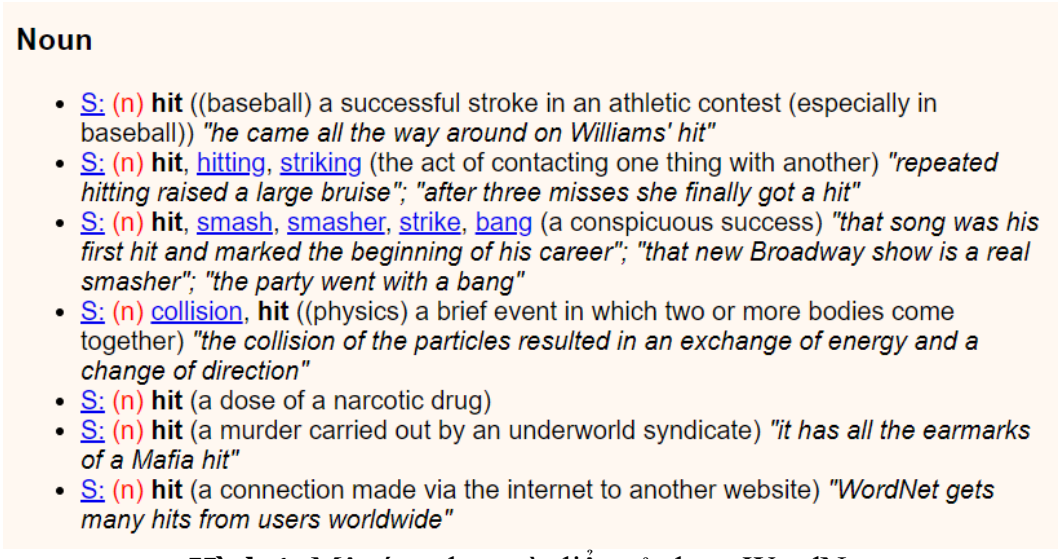
Xác định rõ được các thành phần câu và categorize các thành phần ấy thành từng nhóm có thể giúp chúng ta xác định được ngữ nghĩa của của câu đó.

II. MẠNG NGỮ NGHĨA – WORDNET

1. Định nghĩa

WordNet là dự án từ điển đơn ngữ (tiếng Anh) thiên về ngữ nghĩa, do Princeton University phát triển. WordNet đã tạo ra một tập hợp từ vựng đồ sộ, theo đó các từ được sắp xếp trong dãy của những tập hợp đồng nghĩa, giúp cho việc xác định nghĩa của từ và để phân biệt được nghĩa đang xét với các nghĩa khác. Nguyên lí tổ chức chung của WordNet là mạng lưới quan hệ ngữ nghĩa. Đó là quan hệ đồng nghĩa (synonym): dog – domestic dog; quan hệ trái nghĩa (antonym): rich – poor; quan hệ trên dưới (hyponym): maple – tree, parent – father; quan hệ chỉnh thể – bộ phận (meronym): body – limb; quan hệ kéo theo (entailment): snore – sleep (cho động từ); v.v.

Một điều đáng ý là một từ có thể có nhiều nghĩa. Ví dụ như từ "*hit*" được cở sở dữ liệu WordNet định nghĩa như sau:



Noun

- S: (n) **hit** ((baseball) a successful stroke in an athletic contest (especially in baseball)) *"he came all the way around on Williams' hit"*
- S: (n) **hit**, hitting, striking (the act of contacting one thing with another) *"repeated hitting raised a large bruise"; "after three misses she finally got a hit"*
- S: (n) **hit**, smash, smasher, strike, bang (a conspicuous success) *"that song was his first hit and marked the beginning of his career"; "that new Broadway show is a real smasher"; "the party went with a bang"*
- S: (n) collision, **hit** ((physics) a brief event in which two or more bodies come together) *"the collision of the particles resulted in an exchange of energy and a change of direction"*
- S: (n) **hit** (a dose of a narcotic drug)
- S: (n) **hit** (a murder carried out by an underworld syndicate) *"it has all the earmarks of a Mafia hit"*
- S: (n) **hit** (a connection made via the internet to another website) *"WordNet gets many hits from users worldwide"*

Hình 1: Một ứng dụng từ điển sử dụng WordNet

Để tổ chức dữ liệu được chặt chẽ hơn, WordNet đã sử dụng khái niệm synset (synonyms set). Thông tin rõ hơn sẽ được trình bày bên dưới.

WordNet bao gồm 4 bộ dữ liệu riêng biệt, một bộ của danh từ, một bộ của động từ, một bộ của tính từ và trạng từ. Tính đến phiên bản 3.0, bộ dữ liệu WordNet dành cho tiếng anh đã có khoảng 117000 danh từ, 11400 động từ, 22000 tính từ và 4600 trạng từ.

2. Ứng dụng

a. Đo khoảng cách về nghĩa của các từ

Ta có thể ứng dụng các mối quan hệ ý nghĩa giữa các từ được định nghĩa sẵn trong WordNet như một đơn vị đo khoảng cách về ngữ nghĩa giữa các từ. Số từ trung gian để có thể từ từ này đến từ kia sẽ là *khoảng cách giữa hai từ*. Khoảng cách giữa các từ sẽ được đo bằng độ dài đường đi ngắn nhất trong WordNet.

Ví dụ: Chỉ xét mối quan hệ hyponym, ta có "cat" và "dog" là hyponym của "animal", như vậy khoảng cách giữa "cat" và "animal" là 1 còn giữa "cat" và "dog" là 2 ("cat" – "animal" – "dog").

Các từ đồng nghĩa trong WordNet sẽ có khoảng cách là 0.

Việc xây dựng độ đo như vậy có ý nghĩa rất quan trọng trong bài toán tìm từ thay thế trong văn bản.

Ví dụ: Cho câu "I have a cat" và cần thay từ "cat" thành từ khác sao cho nghĩa ít biến đổi nhất. Theo độ đo của WordNet thì ta thấy việc thay "cat" thành "animal" sẽ phù hợp hơn "dog" vì độ đo giữa "cat" và "animal" ngắn hơn. Thực tế đã kiểm nghiệm điều đó, rõ ràng là câu "I have an animal" gần nghĩa với câu "I have a cat" hơn là câu "I have a dog".

b. **Kiểm tra tính đúng đắn trong câu**

Chúng ta đã biết phân tích cú pháp văn phạm cho ta cấu trúc cú pháp của câu. Tuy nhiên phân tích văn phạm chỉ có thể kiểm tra tính đúng về văn phạm chứ không kiểm tra tính đúng về ngữ nghĩa. Xét câu "the table eats the chicken", ta nhận thấy câu này hoàn toàn đúng về ngữ pháp. Tuy nhiên rõ ràng là cái bàn (table) không thể "ăn" được con gà (chicken), thay vào đó nếu đổi thành "the dog eats the chicken" thì sẽ hợp lý hơn. Vậy làm sao biết được "table" hay "dog" có thể "eats the chicken"?

Bằng cách sử dụng quan hệ hyponym – hypernym trong WordNet. Giả sử ra có một heuristic là chỉ có "animal" mới có thể thực hiện động từ "eat" (tức là chỉ có động vật mới có thể ăn.Như vậy, để kiểm tra một vật có biết ăn hay không ta sẽ kiểm tra xem nó có phải "animal" hay không bằng cách duyệt các hypernyms của nó. Bằng cách duyệt ngược về các hypernyms, ta dễ dàng kiểm tra được là "dog" có thể thực hiện hành động "eat" còn "table" thì không thể.

Tương tự, ta có thể thêm các ràng buộc về ngữ nghĩa để kiểm tra tính đúng đắn về ngữ nghĩa trong câu.

3. **Khái niệm synsets**

WordNet được xây dựng xoay quanh một khái niệm cơ bản là các synset (synonyms set), là tập hợp các từ đồng nghĩa, sao cho, các từ này có thể thay thế cho nhau trong một ngữ cảnh nói. Chúng ta có thể hiểu là mỗi synset sẽ chứa đựng một ngữ cảnh nói trong đó, và tất cả các từ trong synset sẽ thể hiện một ý nghĩa đúng nhất chỉ trong ngữ cảnh đó. Lưu ý rằng, một từ có thể xuất hiện trong nhiều synset, cũng

bởi vì chúng có thể là từ nhiều nghĩa, hoặc là chúng tuy có cùng một nghĩa nhưng tại nhiều ngữ cảnh mà thể hiện nhiều cung bậc ý nghĩa khác nhau. Dựa vào những thông tin mà WordNet cung cấp, có lẽ như vậy cũng đủ để chúng ta suy ra rằng số lượng synsets tồn tại sẽ tương ứng với số lượng ý nghĩa mà chúng ta có thể diễn đạt bằng chỉ một từ (một từ có thể có nhiều hơn một tiếng), tất nhiên với giả sử là WordNet đã thống kê đầy đủ mọi từ và mọi ý nghĩa mà nó có thể mang theo.

4. **Cách tổ chức synsets**

Các synset trong WordNet có những con trỏ, trỏ đến các synset khác. Mỗi con trỏ sẽ mang một mối quan hệ, thể hiện mối quan hệ của synset đang trỏ và synset bị trỏ.

Trong các mối quan hệ, chúng ta quan tâm đến hai quan hệ là hypernymy và hyponymy.

Hypernym là một trường từ vựng của hyponym. Nhờ vậy, chúng ta có thể nói một hyponym là một dạng của hypernym. Ví dụ:

- "Bồ câu", "Đại bàng", "Hải yến" là hyponym của hypernim "chim"
- "Vàng", "Xanh", "Đỏ" là hyponym của hypernym "màu sắc"

Nhờ mối quan hệ này, nếu tại một synset, chúng ta cứ đi theo con trỏ hypernymy lên mãi thì sẽ có lúc chúng ta sẽ phải dừng lại, không còn đi lên được nữa. Hiểu theo trực giác, một trường từ vựng không thể thuộc trường từ vựng con của nó. Nói cách khác, đây là đồ thị không chứa chu trình, khiến nó trở thành một đồ thị hình cây. Dựa vào điều này, chúng ta có thể sử dụng phương pháp đếm cạnh để đo độ tương đồng của hai từ.

III. ĐỘ TƯƠNG ĐỒNG CÂU

1. **Độ tương đồng**

Trong toán học, một độ đo là một hàm số cho tương ứng với một "chiều dài", một "thể tích" hoặc một "xác suất" với một phần nào đó của một tập hợp cho sẵn. Nó là một khái niệm quan trọng trong giải tích và trong lý thuyết xác suất.

Rất khó để đo sự giống nhau, sự tương đồng. Sự tương đồng là một đại lượng (con số) phản ánh cường độ của mối quan hệ giữa hai đối tượng hoặc hai đặc trưng. Đại lượng này thường ở trong phạm vi từ -1 đến 1 hoặc 0 đến 1. Như vậy, một độ đo tương đồng có thể coi là một loại scoring function (hàm tính điểm).

Ví dụ, trong mô hình không gian vector, ta sử dụng độ đo cosine để tính độ tương đồng giữa hai văn bản, mỗi văn bản được biểu diễn bởi một vector. Phân loại độ đo tương đồng, ở đây có thể liệt kê ra một số độ đo như độ đo tương đồng giữa các từ, độ đo tương đồng giữa các văn bản, độ đo tương đồng giữa nhiều ảnh, độ đo tương đồng giữa các ontology, …

2. **Độ tương đồng câu**

Xét ví dụ gồm 2 câu: "*I am male*" và "*I am female*". Ta có thể thấy hai câu trên nếu xét về mặt ngữ pháp thì có sự tương đồng cao. Tuy nhiên nếu xét về mặt ngữ nghĩa thì đây là hai câu có nghĩa đối lập. Vì vậy, khi xét độ tương đồng của câu, thì ta cần phải xét độ tương đồng về mặt ngữ nghĩa để có thể đưa ra được một đáp án chính xác. Vì vậy, chúng ta cần tìm ra một độ đo để có thể tính được độ tương đồng của chúng.

Bài toán tính độ tương đồng câu được phát biểu như sau: Với hai câu s1 và s2, ta cần tìm được một giá trị của hàm S(s1, s2) với $S \in [0.0, 1.0]$. Hàm S(s1, s2) được gọi là độ tương đồng giữa 2 câu s1 và s2. Giá trị này càng cao thì sự tương đồng về mặt ngữ nghĩa giữa hai câu này càng lớn và ngược lại.

Độ tương đồng ngữ nghĩa là một giá trị tin cậy phản ánh mối quan hệ ngữ nghĩa giữa hai câu. Trên thực tế, khó có thể lấy một giá trị có chính xác cao bởi vì ngữ nghĩa chỉ được hiểu đầy đủ trong một ngữ cảnh cụ thể.

3. **Phương pháp để đo động tương đồng câu**

Hiện nay có hai phương pháp chính để đo độ tương đồng câu là phương pháp thống kê và phương pháp xử lý ngôn ngữ tự nhiên.

a. **Phương pháp thống kê**

Với phương pháp thống kê, có một số phương pháp sử dụng các độ đo dựa vào tần số xuất hiện của các từ trong câu, nổi bật là phương pháp sử dụng độ đo cosin. Ví dụ như Brown Corpus.

Ưu điểm: Xử lý nhanh và tốn ít chi phí.

Nhược điểm: Chưa đảm bảo độ chính xác cao về mặt ngữ nghĩa.

b. **Phương pháp xử lí ngôn ngữ tự nhiên**

Một số cách tiếp cận đặc trưng của phương pháp này là sử dụng phân tích cấu trúc ngữ pháp, sử dụng mạng ngữ nghĩa đối với từ, ví dụ như sử dụng WordNet.

Ưu điểm: Xét về mặt ngữ nghĩa thì độ chính xác cao hơn phương pháp thống kê.

Nhược điểm: Xử lý chậm hơn, tốn nhiều chi phí hơn.

Phương pháp mà chúng tôi đã triển khai và sắp sửa trình bày là một phương pháp tổng hợp từ hai phương pháp trên, với mục đích là phát huy tối đa ưu điểm của chúng.

IV. TÍNH CHẤT CÁC HÀM TRUYỀN

1. **Ảnh hưởng của độ dài:**

Đối với mạng lưới hierarchical semantic như trong hình 2, độ dài đường đi giữa hai từ w1 và w2 có thể được xác định từ một trong ba trường hợp sau:

- *w1* và *w2* ở trong cùng một "tập từ đồng nghĩa" (synset).
- *w1* và *w2* không ở trong cùng một "tập từ đồng nghĩa", nhưng "tập từ đồng nghĩa" cho w1 và w2 chứa một hoặc nhiều từ giống nhau. Ví dụ trong hình 2, tập từ đồng nghĩa cho *'boy'* và *'girl'* chứa một từ chung *'child'*.
- *w1* và *w2* không nằm trong cùng một tập từ đồng nghĩa và tập từ đồng nghĩa cho cả hai không chứa từ nào giống nhau.

Trong trường hợp 1, *w1* và *w2* có cùng ngữ nghĩa, chúng ta gán độ đài đường đi giữa *w1* và *w2* về 0. Trường hợp 2 chỉ ra rằng *w1* và *w2* có ngữ nghĩa giống nhau ở vài khía cạnh, chúng ta gán độ đài đường đi giữa *w1* và *w2* về 1. Đối với trường hợp 3, chúng ta đếm độ dài đường đi thật giữa w1 và w2.

Để thỏa mãn các điều kiện bắt buộc mà chúng ta sẽ thảo luận ở phần 3.2.1 thì hàm của độ dài đường đi sẽ có dạng hàm mũ.

2. **Ảnh hưởng của độ sâu:**

Synset trong Cây trường từ vựng Wordnet có xu hướng có ngữ cảnh chi tiết hơn nếu nó xuất hiện ở tầng có độ sâu càng thấp. Yếu tố độ sâu của một synset trong Cây sẽ có ảnh hưởng đến Độ tương đồng.

Ví dụ, xét hai từ cùng với nghĩa của nó sau đây: "*builder*" có nghĩa là một công nhân với ngữ cảnh nói là trong xây dựng, "*worker*" có một nghĩa là một người công nhân với ngữ cảnh nói chung, nên "*worker*" có mối quan hệ hypernym với "*builder*" nên "*builder*" sẽ nằm ở tầng sâu hơn. Vì vậy, theo trực giác, nếu một synset nằm ở tầng càng sâu thì mức độ ý nghĩa nó mang theo càng chi tiết.

V. DUNG LƯỢNG THÔNG TIN (INFORMATION CONTENT)

Triển khai việc đo đạc độ tương đồng ngữ nghĩa bao gồm 2 nhiệm vụ liên quan đến việc chuẩn bị các nguồn thông tin (Informantion Sources) được sử dụng trong việc hình thành các vector ngữ nghĩa và vector từ ngữ.

Đầu tiên, việc tìm kiếm trên mạng ngữ nghĩa được thể hiện bằng độ dài đường đi ngắn nhất giữa các synsets chứa các từ được so sánh và độ sâu của synset đầu tiên, bao gồm các synsets tương ứng với các từ được so sánh.

Thứ hai, thực hiện việc tính toán thống kê các thông tin cần thiết từ Brown Corpus.

1. **Tìm kiếm trên WordNet**

Các synsets trên WordNet được thiết kế theo cấu trúc cây phân cấp, từ nhiều thuật ngữ cụ thể ở các cấp thấp hơn đến một vài thuật ngữ chung ở trên cùng. Hệ thống phân cấp từ vựng được kết nối bằng cách theo vết các trường từ vựng (superordinate terms) trong các mối quan hệ "is a" hoặc "is a kind of" (IS-A). Để xây dựng đường đi giữa hai từ,

mỗi từ sẽ "leo lên" cây từ vựng cho đến khi hai đường đi của hai từ này giao nhau. Các synset tại điểm giao của hai dường đi này được gọi là tập bao hàm (subsumer), đường đi kết nối giữa hai từ sẽ được xác định dựa vào các subsumer. Độ dài của đường dẫn được tính bằng cách đếm các liên kết synset dọc theo đường dẫn giữa hai từ. Độ sâu của subsumer sẽ được tính bằng cách đếm các cấp, từ cấp của subsumer cho đến đỉnh của cây phân cấp từ vựng.

Nếu một từ là đa nghĩa (có nhiều nghĩa), nhiều đường dẫn có thể tồn tại giữa hai từ. Tuy nhiên, chỉ có con đường ngắn nhất được sử dụng để tính toán độ tương đồng ngữ nghĩa giữa các từ.

Ngoài ra, còn có một cơ chế được sử dụng để xử lý trường hợp ngoại lệ: Các từ không có trong WordNet. Nếu từ không có trong WordNet, thì việc tìm kiếm sẽ hông được tiến hành và tính tương tự của từ sẽ được gán bằng giá trị 0. Một cảnh báo về tính không hợp lệ của từ sẽ được hiện ra để nhắc nhở người dùng. Ngoài ra, vấn đề này còn có thể được giải quyết nếu từ còn thiếu tồn tại trong cơ sở dữ liệu từ vựng khác thông qua tổng hợp kiến thức (knowledge fusion).

2. **Thống kê từ Brown Corpus**

Xác suất của một từ w trong Brown corpus được tính toán một cách đơn giản theo công thức sau:

$$\hat{p}(w) = \frac{n+1}{N+1}$$

Trong đó, N là số lượng từ có trong Brown corpus, n là tần suất của từ w trong Brown corpus (các giá trị này được cộng thêm 1 để tránh xuất hiện kết quả vô định đối với logarit).

Dung lượng thông tin của w trong Brown corpus có công thức như sau:

$$I(w) = -\frac{\log p(w)}{\log(N+1)} = 1 - \frac{\log(n+1)}{\log(N+1)}$$

Vì thế, $I \in [0,1]$

MÔ TẢ THIẾT KẾ

I. PHÂN TÍCH BÀI TOÁN

Nhập vào một câu từ bàn phím, với một cơ sở dữ liệu cho trước *(ngôn ngữ: Tiếng Anh)*, hãy in ra các câu trong bộ dữ liệu có độ tương đồng cao về mặt ngữ nghĩa với câu này.

INPUT: Một câu bất kỳ được nhập từ bàn phím (*ngôn ngữ: Tiếng Anh*).

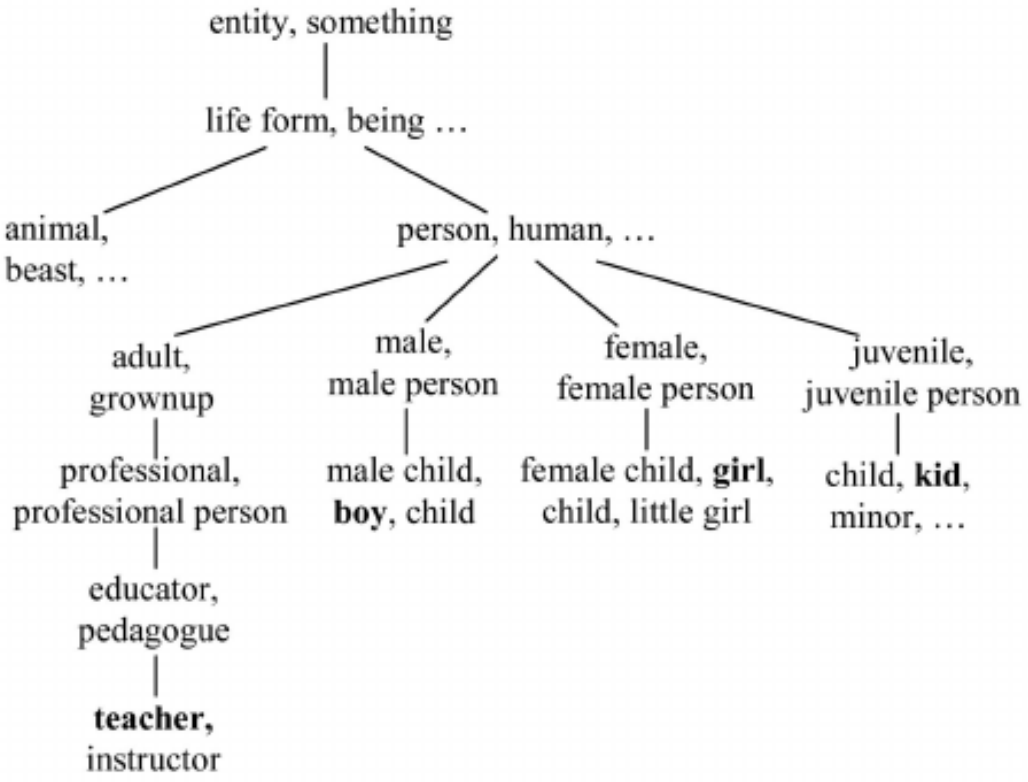
OUTPUT: Một hoặc nhiều câu trong bộ dữ liệu có độ tương đồng cao về mặt ngữ nghĩa với câu này.

Thuật toán chính của bài toán này là Đo độ tương đồng về mặt ngữ nghĩa giữa hai câu (*ngôn ngữ: Tiếng Anh*).

II. ĐO ĐỘ TƯƠNG ĐỒNG GIỮA HAI TỪ VÀ HỆ THỐNG CÂY PHÂN CẤP DỮ LIỆU (XÂY DỰNG TỪ WORDNET)

1. Hệ thống cây phân cấp dữ liệu

Từ bộ dữ liệu của WordNet, có thể xây dựng được một hệ thống phân cấp dữ liệu được biểu diễn bằng cây, trong đó các nút sẽ có mối quan hệ cha/con với nhau, mỗi nút sẽ có một cha duy nhất.



Hình 2: Mô tả mối quan hệ hypernymy/hyponymy giữa các synsets[*]

2. **Đo độ tương đồng về mặt ngữ nghĩa giữa hai từ**

   Ý tưởng là với hai từ, chúng ta tìm ra được hai tập hợp synset, và rồi tìm trường từ vựng mà chứa hai synset đó (hoặc chứa trường từ vựng của hai synset đó). Tổng số cạnh từ nó đến cả hai synset đang xét sẽ góp phần vào trong thang đo Tương đồng ngữ nghĩa.

   a. **Tìm công thức phù hợp**

      Thông qua quan sát cho được các nhận xét sau:

      - Tổng số cạnh mà hai synset "đến được nhau" sẽ ảnh hưởng chỉ số tương đồng. Gọi L là số cạnh này.
      - Vì một synset càng ở sâu dưới cây, càng thể hiện một ngữ cảnh chi tiết hơn, nên độ sâu của synset chung (trường từ vựng chung) sẽ ảnh hưởng đến Chỉ số tương đồng. Gọi H là độ sâu của synset mà bao quát hai synset đang xét.
      - Ngoài ra, lượng thông tin của một từ sẽ tác động đến Chỉ số tương đồng.

      Với hai từ là w1, w2. Độ tương đồng của chúng sẽ là một hàm f với hai tham số là L và H:

      $$s(w_1, w_2) = f(L, H) \qquad (1)$$

      Hàm (1) có thể được viết lại như sau:

      $$s(w_1, w_2) = f_1(L).f_2(H) \qquad (2)$$

   b. **Đặc trưng của hàm f()**

      Hàm phải cho kết quả [0.0, 1.0], với độ tương đồng 0.0 nghĩa là hoàn toàn không tương đồng, với độ tương đồng 1.0 là hoàn toàn giống nhau.

      Hàm f có miền kết quả hữu hạn trong khi nguồn tin có miền giá trị là vô hạn, khiến hàm f của chúng ta không thể là một hàm tuyến tính mà phải là hàm lũy thừa.

c. **Đóng góp của L**

Vì hàm L là một hàm mũ (như đã nêu ở chương IV của Cơ sở lý thuyết), hàm của L có thể được xây dựng như sau:

$$f_1(L) = e^{-\alpha L} \quad (3)$$

Trong đó, α là một hằng số. Giá trị của f_1 (L) sẽ nằm trong khoảng từ 0 đến 1.

d. **Đóng góp của H**

Kết hợp với chương IV của Cơ sở lý thuyết, hàm của H sẽ có công thức:

$$f_2(h) = \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}}$$

với $\beta > 0$. Nếu $\beta \rightarrow \infty$, độ sâu của từ trong mạng ngữ nghĩa sẽ là không xác định.

Tóm lại, ta lập được công thức đo độ tương đồng giữa hai từ:

$$s(w_1, w_2) = e^{-\alpha L} \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}}$$

với α ϵ [0,1] và β ϵ [0,1] là các tham số đo sự đóng góp của đường đi ngắn nhất - độ dài và chiều sâu. Giá trị tối ưu của α và β phụ thuộc vào knowledge base được sử dụng và có thể được xác định bằng cách sử dụng một tập hợp các từ đi kèm với đánh giá sự tương đồng với con người (human similarity ratings). Đối với WordNet, những tham số tối ưu là α=0.2 và β=0.45.

III. XÂY DỰNG TỆP CƠ SỞ DỮ LIỆU

Chúng tôi có hai tệp các câu mẫu để thực hiện phép chọn câu có độ tương đồng giống nhất với câu nhập vào. Tệp thứ nhất được trích dẫn từ Brown Corpus. Tệp thứ hai chứa những câu thành ngữ và câu trích dẫn của tiếng Anh, được tổng hợp bởi project Gutenberg.

1. **Brown Corpus**

Brown Corpus là một tuyển tập bản văn chung, chứa hơn 500 bản văn ngôn ngữ tiếng Anh, có tổng cộng xấp xỉ 1000000 từ, được tổng

hợp từ những ấn phẩm in được phát hành ở Hoa Kỳ năm 1961. Các lĩnh vực bao gồm từ thể thao, tin tức đến chính trị. **Brown Corpus** là một bộ sưu tập bản văn rất có giá trị cho những mục đích liên quan đến thống kê xác suất của từ.

2. **Project Gutenberg**

   Là một thư viện gồm nhiều eBook, là một trong những nguồn bản văn nổi tiếng thích hợp cho các dự án lập trình ngôn ngữ học.

   eBook mà chúng tôi sử dụng làm nguồn văn bản sẽ là *"A Dictionary of English Proverbs"*.

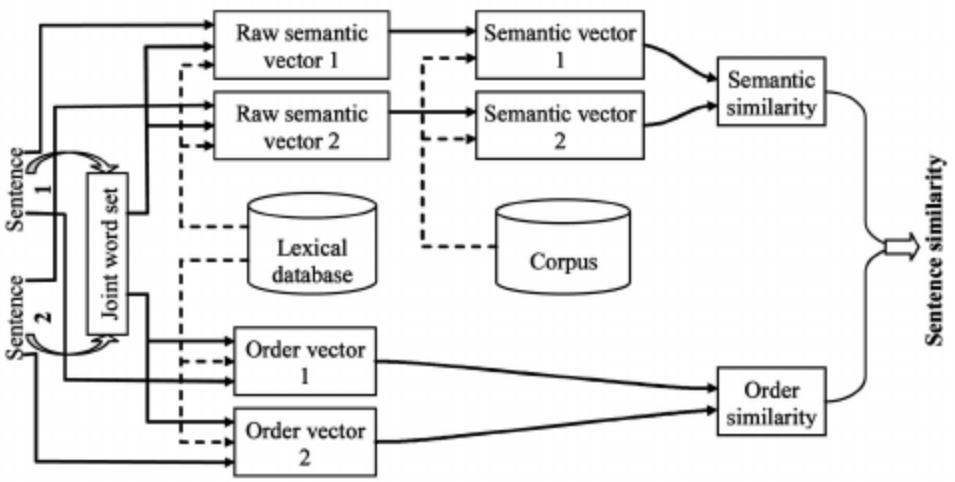
   Một vài câu trích dẫn từ tệp:

   ```
   "Flowers are the pledge of fruit."

   "It is one thing to flourish and another to fight."

   "No flying without wings."

   "Happy is he who knows his follies in his youth."

   "No one is a fool always; every one sometimes."

   "A fool may ask more questions in half an hour than a wise man"

   "A fool may give a wise man counsel."
   ```

   Tệp chứa: `1883 sentences ~ 14903 words.`

IV. PHÂN TÍCH THUẬT TOÁN CHÍNH: ĐO ĐỘ TƯƠNG ĐỒNG VỀ MẶT NGỮ NGHĨA GIỮA HAI CÂU

1. Sơ đồ khối



Hình 3: Sơ đồ thuật toán[*]

2. Phân tích thuật toán chính: Đo độ tương đồng về mặt ngữ nghĩa giữa hai câu

a. Tiền xử lý

- Tách mỗi câu thành một danh sách các từ tố (token): Mỗi câu được tách ra thành một danh sách từ.
- Xác định một tập từ chung cho hai câu. Tập từ chung này bao gồm tất cả những từ phân biệt có trong hai câu đó. Mỗi từ trong tập từ chung chỉ xuất hiện duy nhất một lần.

  Cho 2 câu T_1 và T_2, tập từ chung của 2 câu này là:

  $$T = T_1 \cup T_2$$

  Tập từ chung T bao gồm các từ riêng biệt thuộc 2 câu T_1 và T_2. Vì sự đa dạng về mặt hình thái của từ có thể làm cho một từ xuất hiện trong một câu với các hình thức khác nhau truyền đạt các ý nghĩa cụ thể trong các bối cảnh cụ thể, chúng tôi chính hình thái của từ xuất hiện trong câu. Ví dụ, boy và boys, woman và women sẽ được xem như 4 từ riêng biệt và tất cả sẽ đưa vào tập từ chung.

  Cụ thể hơn, tập từ chung của 2 câu:

T_1: RAM keeps things being worked with.

T_2: The CPU uses RAM as a short-term memory store.

là: T – {RAM keeps things being worked with The CPU uses as a short-term memory store}.

b. **Tính độ tương đòng giữa hai từ**

Dựa vào tập từ chung đồng thời sử dụng WordNet, ta sẽ ước tính được độ tương đồng về ngữ nghĩa cho các từ trong mỗi câu với tập từ chung.

Bởi vì tập từ chung T có nguồn gốc hoàn toàn từ 2 câu T_1 và T_2, nên nó nhỏ gọn và hoàn toàn không có thông tin dư thừa. Tập từ chung này có thể ddược xem là thông tin ngữ nghĩa cho các câu cần được so sánh. Mỗi câu dễ dàng được biểu diễn lại bằng cách sử dụng tập từ chung như sau: Mỗi vector thừa kế từ tập từ chung sẽ được gọi là vector từ vựng ngữ nghĩa, được kí hiệu là $\check{s}(i = 1,2, \dots, m)$, được xác định bằng độ tương đồng ngữ nghĩa giữa 2 từ tương ứng trong câu.

Lấy T_1 làm ví dụ:

Trường hợp 1: Nếu w_i tồn tại trong câu T_1, $\check{s}_i$ sẽ có giá trị là 1.

Trường hợp 2: Nếu w_i không tồn tại trong câu T_1, tính giá trị tương đồng ngữ nghĩa giữa $w_i$và từng từ trong câu T_1, sử dụng công thức đã được nêu trong chương II. Sau đó, từ có độ tương đồng ngữ nghĩa cao nhất trong T_1 với w_i với giá trị tương đồng ngữ nghĩa ç sẽ được chọn. Nếu ç vượt được ngưỡng cho trước thì $\check{s}_i$ sẽ có giá trị là ç, nếu không thì $\check{s}_i = 0$.

c. **Tính độ tương đồng ngữ nghĩa cho hai câu**

Người ta chứng minh được rằng, một từ xuất hiện với tần suất càng cao (trong một văn thể) thì mang thông tin càng ít. Lượng thông tin có trong một từ phụ thuộc vào xác suất của nó trong một văn thể. Mỗi ô được tính bằng lượng tin liên quan $I(w_i)$ và $I(\widetilde{w}_i)$. Cuối cùng, giá trị của mỗi phần tử trong vector ngữ nghĩa được tính bằng:

$$s_i = \check{s}.I(w_i).I(\widetilde{w}_i)$$

Trong đó, w_i là từ trong tập từ chung, $\widetilde{w}_i$ là từ tương đồng với nó trong câu.

Độ tương đồng ngữ nghĩa giữa hai câu được định nghĩa bằng hàm cosine giữa hai vector:

$$S_S = \frac{s_1.s_2}{\|s_1\|.\|s_2\|}$$

d. **Tính độ tương đồng trật tự từ**

Giả sử 2 câu T_1 và T_2 có các từ giống nhau tại các vị trí giống nhau, 2 từ trong câu T_1 có vị trí ngược lại với 2 từ như vậy trong câu T_2. Ví dụ:

T_1: *A quick brown dog jumps over the lazy fox.*

T_2: *A quick brown fox jumps over the lazy dog.*

Bởi vì 2 câu này chứa các từ giống hệt nhau, bất kì phương pháp nào dựa trên "bag of words" đều sẽ đưa ra kết quả là hai câu này là hai câu hoàn toàn giống nhau. Tuy nhiên, rõ ràng là hai câu này chỉ tương tự nhau ở một mức độ nào đó. Sự khác nhau giữa hai câu T_1 và T_2 là kết quả của sự khác nhau giữa trật tự từ của hai câu ấy. Vì vậy, phương pháp tính toán độ tương đồng giữa hai câu cần tính đến sự tác động của trật tự từ trong câu đấy.

Đối với cặp câu ví dụ T_1 và T_2, tập từ chung là:

T = {*A quick brown dog jumps over the lazy fox*}

Chúng ta gán các index duy nhất cho mỗi từ trong hai câu T_1 và T_2. Index đơn giản là số thứ tự mà từ xuất hiện trong câu. Ví dụ index = 4 là của từ dog và 6 là của từ over trong T_1. Trong việc tính toán độ tương đồng trật tự từ, một vector trật tự từ r, được tạo nên cho hai câu T_1 và T_2, dựa trên tập từ chung T.

Lấy T_1 làm ví dụ, đối với mỗi từ w_i trong T, chúng ta sẽ cố gắng tìm từ giống hoặc gần giống nhất trong câu T_1 theo nguyên tắc sau:

- Nếu từ giống hệt xuất hiện trong T_1, giá trị của phần tử của từ trong vector r_1 sẽ là giá trị là index của từ tương ứng trong câu T_1. Nếu không, ta cố gắng tìm từ tương đồng nhất $\widetilde{w}_i$ trong câu T_1.
- Nếu độ tương đồng giữa w_i và $\widetilde{w}_i$ lớn hơn ngưỡng cho trước, thì giá trị của phần tử của từ trong vector r_1 sẽ là index của từ $\widetilde{w}_i$ trong câu T_1.
- Nếu từ không thoả cả 2 nguyên tắc trên thì giá trị của phần tử của từ trong vector r_1 sẽ bằng 0.

Áp dụng phương pháp trên, kết hợp với phương pháp tính đã được trình bày ở mục trước vào 2 câu ví dụ, ta có:

$$r_1 = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}$$

$$r_2 = \{1\ 2\ 3\ 9\ 5\ 6\ 7\ 8\ 4\}$$

Vì vậy, một vector trật tự từ là một thông tin cấu trúc cơ bản nằm trong câu. Nhiệm vụ của việc xử lý trật tự từ là để đo mức độ giống nhau của trật tự từ trong hai câu.

Chúng tôi đề xuất một công thức để tính độ tương đồng trật tự từ của hai câu:

$$S_r = 1 - \frac{\|r_1 - r_2\|}{\|r_1 - r_2\|}$$

e. **Tính độ tương đồng hai câu**

Sự tương đồng về mặt ngữ nghĩa đại diện cho sự tương đồng từ vựng. Mặt khác, sự giống nhau về trật tự từ cung cấp thông tin về mối quan hệ giữa các từ: từ nào xuất hiện trong câu và những từ nào xuất hiện trước hoặc sau những từ khác. Cả thông tin về mặt ngữ nghĩa và cú pháp (về mặt trật tự từ) đều đóng một vai trò trong việc truyền đạt ý nghĩa của câu. Do đó, độ

tương đồng của hai câu được định nghĩa là sự kết hợp của độ tương tự ngữ nghĩa và và độ tương tự trật tự từ:

$$S(T_1, T_2) = \delta S_S + (1 - \delta) S_r$$

Trong đó, $\delta \leq 1$ quyết định sự đóng góp tương đối của thông tin ngữ nghĩa và thông tin trật tự vào việc tính toán độ tương đồng tổng thể giữa hai câu. Vì cú pháp đóng vai trò phụ thuộc trong việc xử lý ngữ nghĩa của văn bản, giá trị của δ nên lớn hơn 0.5, nói cách khác, $\delta \in (0.5, 1]$.

3. **Đánh giá thuật toán**

Về tốc độ xử lý:

```
S1 = "The weather is great today, chance of rain is unlikely"
S2 = "The hedgehog has spikes on its back."
ss = 0.0493647; T_i = (0.372; 0.346; 0.343; 0.359)
T ~= 0.355
Số word trong joint vector là 16

S1 = "Every man is a pilot in a calm sea."
S2 = "Will without reason is blind."
ss = 0.28732; T_i = (0.34; 0.308; 0.293; 0.355)
T ~= 0.324
Số word trong vector joint là 13
```

Thông qua một số kiểm thử nữa, triển khai của chúng tôi đạt được thời gian xấp xỉ *0.3s* khi độ dài của vector joint là 10~15. Tuy nhiên, đây là con số thời gian khi triển khai bắt đầu chạy, bằng việc ứng dụng kỹ thuật quy hoạch động, chúng ta không cần phải tính lại độ tương đồng giữa hai từ đã tính sẵn. Nên chắc chắn một điều rằng *0.3s* là ước lượng cho giới hạn trên của Thuật toán đối với độ dài câu trung bình.

TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

I. TRIỂN KHAI THUẬT TOÁN

Để biểu diễn thuật toán tính độ tương đồng giữa 2 câu, chúng tôi thực hiện ví dụ sau:

T_1: A glass of cider.

T_2: A full cup of apple juice.

T = {a glass of cider full cup apple juice}

Từ bảng 1, suy ra được vector ngữ nghĩa cho $T_{1:}$

S_1 = {0.0741749 0.444395 0.0574592 0.847267 0 0.320102 0.178623 0.0994593}

Tương tự, ta có:

S_2 = {0.0741749 0.320102 0.0574592 0.321152 0.367262 0.522502 0.694414 0.672591}

Làm theo cách như vậy, ta có vector trật tự từ của 2 câu:

r_1 = {1 2 3 4 0 2 2 1}

r_2 = {1 3 4 6 2 3 5 6}

Cuối cùng, độ tương đồng giữa 2 câu "A glass of cider" và "A full cup of apple juice" là 0.678, với $\delta = 0.85$.

II. ỨNG DỤNG TÌM KIẾM NGỮ NGHĨA ĐƠN GIẢN:

Cùng với tệp dữ liệu bao gồm hơn 1000 câu thành ngữ tiếng Anh, chúng tôi thực hiện nhiều phép kiểm thử là xuất ra danh sách các câu đó, với độ tương đồng giảm dần so với một câu văn chúng tôi nhập vào.

Dưới đây là hàng loạt ví dụ câu được nhập vào cùng với top 5 câu trong tệp được thuật toán đánh giá là tương đồng nhất:

S = "bird catches worm" | Runtime ~= 50.49s

Ss	**$Câu_i$**
0.913	The early bird catches the worm
0.842	To know where the shoe pinches.
0.809	Where are the crumbs there are the chickens.
0.804	The wearer best knows where the shoe pinches.
0.800	Takes a thief to catch a thief

Bảng 1: Semantic Search cho "*bird catches worm*"

S = "dogs are man's best friend" | Initial Runtime ~= 50.49s

Ss	**$Câu_i$**
0.967453	A dog is a man's best friend
0.782636	The early bird catches the worm
0.778899	As sound as a nut.
0.775355	Hold with the hare and run with the hounds
0.759824	Hold not with the hounds and run with the hare.

Bảng 2: Semantic Search cho "*dogs are man's best friend*"

S = "what is it?" | Initial Runtime ~= 10.42s

Ss	**$Câu_i$**
0.564825	Live and learn.
0.471504	A man cannot live by the air.
0.46987	Morning is the time for study.
0.439204	One half of the world does not know how the other half lives
0.431378	One half of the world knows not how the other half lives.

Bảng 3: Semantic Search cho "*what is it?*"

S = "our battle just begun" | Initial Runtime ~= 15.014s

Ss	**$Câu_i$**
0.758502	Fight the good fight
0.718596	The best lie is the worst.
0.709308	When the going gets tough, the tough get going
0.708711	Do all the good you talk of; but talk not of all the good you
0.6755	Good talk saves the food

Bảng 4: Semantic Search cho "*our battle just begun*"

III. NHẬN XÉT

- So với cách làm so từ vời từ, Thuật toán tỏ ra hiệu quả hơn khi có xác định được những ngữ cảnh của từ trong khi đo đạc độ tương đồng giữa 2 câu. Tuy nhiên, độ chính xác chưa đạt đến mức chấp nhận được khi đôi lúc thuật toán đánh giá cao hai câu theo trực giác là không liên quan nhau.
- Thời gian chạy không thấp khi Thuật toán Đo độ tương đồng giữa hai câu có factor độ phức tạp thời gian là $\mathbf{O(n^2)}$.
- Triển khai vẫn còn chưa được tối ưu hóa một cách chặt chẽ. Chưa ứng dụng các phương pháp tiền xử lý hay quy hoạch động hết sức có thể.

KẾT LUẬN CHUNG VÀ HƯỚNG PHÁT TRIỂN

Với nhu cầu thực tiễn về các ứng dụng tìm kiếm có độ chính xác cao, chúng tôi đã tập trung nghiên cứu bài toán so sánh độ tương đồng ngữ nghĩa nói chung và tìm kiếm câu tương đồng trong một bộ cơ sở dữ liệu nói riêng. Các kết quả cụ thể mà đề tài đạt được là:

- Khảo sát, nghiên cứu các phương pháp so sánh độ tương đồng ngữ nghĩa, từ đó chọn ra được phương pháp phù hợp nhất.
- Đề tài cũng đã giải quyết được bài toán tìm kiếm các câu tương đồng trong bộ cơ sở dữ liệu.
- Thử nghiệm các thuật toán đề xuất và cho được kết quả ban đầu khá khả quan.

Do hạn chế về mặt kiến thức sẵn có, đề tài chỉ dừng lại ở mức thử nghiệm với một số câu đơn giản và bộ cơ sở dữ liệu kích thước vừa phải. Với những kết quả thực nghiệm ban đầu, sẽ cần tiếp tục hoàn thiện để nâng cao hiệu suất.

Trong thời gian tới, hướng phát triển của đề tài là cải tiến thuật toán để đo được độ tương đồng giữa các tính từ, từ đó hoàn thiện hệ thống và ứng dụng vào các hệ thống tóm tắt văn bản hoặc RS, …

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

TẬP QUY TẮC CÚ PHÁP TIẾNG VIỆT - Đào Minh Thu, Đào Thị Minh Ngọc, Nguyễn Mai Vân, Lê Kim Ngân, Lê Thanh Hương, Nguyễn Phương Thái, Đỗ Bá Lâm

TẬP QUY TẮC CÚ PHÁP TIẾNG VIỆT CHO THÀNH PHẦN CÂU – Lê Thanh Hương, Đỗ Bá Lâm

Dictionary of English Proverbs and Proverbial Phrases With a Copious Index of Principal Words – Thomas Preston

How the statistical revolution changes (computational) linguistics – Mark Johnson *(Cognitive and Linguistic Sciences and Computer Science, Brown University)*

Information Retrieval Based on Semantic Similarity Using Information Content – Kishor Wagh *(Government College of Engineering, Amravati, India)*, Satish Kolhe *(North Maharashtra University, Jalgaon, India)*

Sentence Similarity Based on Semantic Nets and Corpus Statistics – Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett [*]

WordNet 3.0 Reference Manual – Princeton University "About WordNet." WordNet. Princeton University, 2010 [wordnet.princeton.edu/documentation]

Phrases.org.uk - English idioms, phrases and proverbs that we use every day, with their meanings and origins explained. [https://www.phrases.org.uk/index.html]