Nghiên cứu thuật toán đánh giá độ tương đồng văn bản, ứng dụng xây dựng hệ thống phát hiện sao chép tài liệu

Nguyễn Quốc Khánh[1], Đỗ Minh Hiếu[1*]

[1]Viện Công Nghệ Thông Tin Và Truyền Thông, Học Viện Kỹ Thuật Quân Sự

Tóm tắt nội dung

Nghiên cứu các phương pháp đánh giá độ tương đồng của văn bản và lưu ý đặc tính riêng của tiếng Việt, nghiên cứu các công cụ sẵn có và lựa chọn thử nghiệm cũng như đề xuất cho bài toán đánh giá độ tượng đồng văn bản tiếng Việt sử dụng thuật toán so khớp xấp xỉ và áp dụng thử nghiệm đánh giá độ tượng đồng các văn bản luận văn.

ĐẶT VẤN ĐỀ

Đánh giá độ tương đồng của văn bản (text similarity) là quá trình đánh giá mức độ giống nhau giữa hai hay nhiều tài liệu văn bản dựa trên nội dung của chúng. Đây là một trong những bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và truy vấn thông tin. Các phương pháp đánh giá độ tương đồng của văn bản có thể dựa trên các đặc trưng của văn bản, bao gồm cả cấu trúc, nội dung, từ vựng, cách sắp xếp các câu, hay ngữ pháp. Một số phương pháp đánh giá độ tương đồng phổ biến bao gồm:

- So sánh cosine: Phương pháp đo độ tương đồng dựa trên cosine similarity của hai vector biểu diễn cho hai văn bản.
- So sánh Jaccard: Phương pháp đo độ tương đồng dựa trên tỉ lệ số lượng từ chung của hai văn bản so với tổng số lượng từ trong hai văn bản đó.
- So sánh Levenshtein: Phương pháp đo độ tương đồng dựa trên khoảng cách Levenshtein giữa hai văn bản, khoảng cách này được tính bằng cách đếm số lượng thao tác chèn, xóa và thay đổi từ trong hai văn bản đó.
- Sử dụng mô hình học máy: Sử dụng các mô hình học máy để học cách đánh giá độ tương đồng văn bản dựa trên các đặc trưng của chúng, chẳng hạn như sử dụng mô hình word embedding để biểu diễn các từ trong văn bản.

Đánh giá độ tương đồng của văn bản có thể được áp dụng trong nhiều ứng dụng, chẳng hạn như tìm kiếm và sắp xếp kết quả truy vấn, phát hiện tin giả, nhận dạng bản sao văn bản, hay phân loại các tài liệu văn bản.

*Corresponding author: jane@smith.com
Received: October 20, 2023, Published: December 14, 2023

KIẾN THỨC TỔNG QUAN VÀ CÁC NGHIÊN CỨU LIÊN QUAN

Các đặc điểm của ngôn ngữ Tiếng Việt

Tiếng Việt được xếp vào loại hình đơn lập (isolate) hay còn gọi là loại hình phi hình thái, không biến hình, đơn tiết với những đặc điểm chính sau:

- Trong hoạt động ngôn ngữ, từ không biến đổi hình thái. Ý nghĩa ngữ pháp nằm ở ngoài từ. Ví dụ: Tôi nhìn anh ấy và anh ấy nhìn tôi
- Phương thức ngữ pháp chủ yếu là: Trật tự từ và từ hư. Ví dụ: Gạo xay và xay gạo
- Tồn tại một loại đơn vị đặc biệt, đó là "hình tiết" mà vỏ ngữ âm của chúng trùng khít với âm tiết, và đơn vị đó cũng chính là "hình vị tiếng Việt hay còn gọi là tiếng (theo tác giả Đinh Điền thì có khoảng 10.000 tiếng, nhưng theo khảo sát của hội người mù Việt Nam khi làm chương trình sách nói thì chỉ có khoảng 3000 từ)
- Ranh giới từ không xác định mặc nhiên bằng khoảng trắng như các thứ tiếng biến hình khác. Ví dụ: "học sinh học sinh học". Điều này khiến cho việc phân tích hình thái (tách từ) tiếng Việt trở nên khó khăn. Việc nhận diện ranh giới từ là quan trọng làm tiền đề cho các xử lý tiếp theo sau đó như: kiểm tra lỗi chính tả, gán nhãn từ, thống kê tần xuất từ ....
- Tồn tại loại từ đặc biệt "từ chỉ loại" (classsifier) hay còn gọi là phó danh từ chỉ loại đi kèm với danh từ như: cái bàn, cuốn sách, bức thư, ...
- Về mặt âm học, các âm tiết tiếng Việt đều mang 1 trong 6 thanh điệu (ngang, sắc, huyền, hỏi, ngã, nặng). Đây là âm vị siêu đoạn tính.
- Có hiện tượng láy trong từ tiếng Việt như: lấp lánh, lung linh ... Ngoài ra còn có hiện tượng nói lái (do mối liên kết giữa phụ âm đầu và phần vần trong âm tiết là lỏng lẻo) như: hiện đại - hại điện.

Một số khái niệm cơ bản

a) Ngôn ngữ tự nhiên (natural language) là ngôn ngữ được sử dụng bởi con người để giao tiếp với nhau. Đó là một hệ thống ký hiệu âm thanh, từ vựng và cú pháp để truyền tải ý nghĩa giữa các cá nhân hay nhóm người. Ngôn ngữ tự nhiên có thể được nói hoặc viết và được sử dụng rộng rãi trong đời sống hàng ngày, trong các lĩnh vực như kinh doanh, giáo dục, văn hóa và giải trí.

b) Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một lĩnh vực của trí tuệ nhân tạo, tập trung vào việc phân tích, xử lý và tạo ra ngôn ngữ tự nhiên, đó là ngôn ngữ mà con người sử dụng để giao tiếp với nhau. NLP sử dụng nhiều kỹ thuật và công nghệ khác nhau để phân tích và hiểu ngôn ngữ tự nhiên, bao gồm xử lý ngôn ngữ tự nhiên, phân tích cú pháp, phân tích ý định, dịch máy, học máy và nhiều hơn nữa. Các ứng dụng của NLP rất đa dạng, từ xử lý văn bản, tìm kiếm thông tin, đến chatbot và ứng dụng trợ lý ảo.

c) Corpus là một tập hợp các văn bản hoặc dữ liệu văn bản được sử dụng để nghiên cứu ngôn ngữ học hoặc xử lý ngôn ngữ tự nhiên. Corpus được sử dụng để cung cấp một nguồn dữ liệu đại diện cho một ngôn ngữ cụ thể hoặc một lĩnh vực nhất định, giúp các nhà nghiên cứu và nhà phát triển có thể phân tích và xử lý các đặc trưng của ngôn ngữ.

d) Stemming là một kỹ thuật được sử dụng để tìm ra từ gốc bằng cách loại bỏ tất cả những tiền tố, phụ tố và hậu tố. Mục đích chính của stemming là để tạo cho thuật toán khả năng tìm kiếm và trích xuất những thông tin hữu ích từ một nguồn khổng lồ như internet hoặc dữ liệu lớn.

e) Lemmatization là một quá trình xử lý ngôn ngữ tự nhiên để chuẩn hóa các từ về dạng gốc của chúng, gọi là lemma. Mục đích của việc lemmatization là để tạo ra một danh sách các từ gốc khác nhau từ các biến thể từ khác nhau của các từ trong văn bản.

f) Tokenization (còn được gọi là word segmentation hoặc text segmentation) là quá trình phân tách một đoạn văn bản thành các phần tử nhỏ hơn gọi là token. Một token có thể là một từ, một ký tự, một dấu câu, một số hoặc một biểu tượng khác. Tokenization là một quá trình quan trọng trong xử lý ngôn ngữ tự nhiên, vì nó là bước đầu tiên trong việc xử lý dữ liệu văn bản. Sau khi văn bản được phân tách thành các token, các token có thể được sử dụng để thực hiện các tác vụ xử lý ngôn ngữ tự nhiên khác.

g) N-gram là một chuỗi liên tiếp của N từ trong một văn bản. Đối với một văn bản có độ dài n, ta có thể tạo ra n - N + 1 N-gram khác nhau. Ví dụ, với văn bản "This is an example sentence", các 2-gram (hay bigram) có thể là: "This is", "is an", "an example", và "example sentence".

h) N-gram là một chuỗi liên tiếp của N từ trong một văn bản. Đối với một văn bản có độ dài n, ta có thể tạo ra n - N + 1 N-gram khác nhau. Ví dụ, với văn bản "This is an example sentence", các 2-gram (hay bigram) có thể là: "This is", "is an", "an example", và "example sentence".

i) Normalization (chuẩn hóa) là quá trình chuyển đổi một chuỗi dữ liệu văn bản (thường là các từ) thành dạng chuẩn hoá nhất, giúp tăng khả năng xử lý và so sánh dữ liệu trong các ứng dụng xử lý ngôn ngữ tự nhiên.Quá trình normalization giúp giảm số lượng các biến thể của các từ và đồng thời giúp định dạng văn bản thống nhất hơn. Nó là một bước quan trọng trong xử lý ngôn ngữ tự nhiên để cải thiện chất lượng và hiệu quả của các ứng dụng.

j) named entity recognition (NER) là quá trình xác định và phân loại các named entity trong văn bản. Quá trình này bao gồm nhận dạng các từ hoặc cụm từ trong văn bản và xác định xem chúng có liên quan đến các loại thực thể nào. Các named entity có thể được phân loại thành các loại khác nhau, ví dụ như người, địa điểm, tổ chức, sản phẩm, thời gian,..., "is an", "an example", và "example sentence".

k) Parts-of-speech (POS) tagging (tạm dịch là đánh dấu loại từ) là quá trình xác định loại từ (noun, verb, adjective, adverb,...) của mỗi từ trong một câu hoặc văn bản. Quá trình này cũng có thể bao gồm phân loại các từ đồng thời với việc đánh dấu loại từ, ví dụ như phân loại các động từ là nguyên thể hay khuyết thiếu, các danh từ là số ít hay số nhiều,...

Xây dựng mô hình vector cho Tiếng Việt

Phương pháp Words Embedding cổ điển

* Bag of Words (BoW) Đây là cách biểu diễn vector truyền thống phổ biến nhất được sử dụng. Mỗi từ hoặc n-gram từ sẽ được mô tả là một vector có số chiều bằng đúng số từ trong bộ từ vựng. Tại vị trí tương ứng với vị trí của từ đó trong túi từ, phần tử trong vector đó sẽ được đánh dấu là 1. Những vị trí còn lại sẽ được đánh dấu là 0. Biểu diễn BoW thường được sử dụng trong các phương pháp phân loại tài liệu, trong đó tần suất xuất hiện của mỗi một/hai/ba từ hỗ trợ việc đào tạo các bộ phân loại. Trong BoW, sự xuất hiện của từ được đánh giá độc lập với tần suất hoặc ngữ cảnh chúng xảy ra. Tuy nhiên, hạn chế của BoW là không mã hóa bất

kỳ thông tin nào liên quan đến ngữ nghĩa của từ. Trong phương pháp BoW, từ giống nhau sẽ được đánh trọng số như nhau. Phương pháp này không xét đến tần suất xuất hiện của từ hay nhữ cảnh từ. Và trong thực tế, để cần hiểu được nghĩa của mỗi từ, ta cần xác định từ đó trong văn cảnh hơn là xét nghĩa độc lập từ.

* TF- IDF (term frequency–inverse document frequency) – tần suất- tần suất đảo nghịch từ. Đây là một phương pháp thống kê, nhằm phản ánh độ quan trọng của mỗi từ hoặc n-gram đối với văn bản trên toàn bộ tài liệu đầu vào. TF-IDF thể hiện trọng số của mỗi từ theo ngữ cảnh văn bản. TF-IDF sẽ có giá trị tăng tỷ lệ thuận với số lần xuất hiện của từ trong văn bản và số văn bản có chứa từ đó trên toàn bộ tập tài liệu. Phương pháp này giúp cho TF-IDF có tính phân loại cao hơn so với phương pháp trước.

* Là phương pháp mà ta có thể xem xét được tổng quan trong toàn bộ ngữ cảnh. Mỗi từ sẽ được biểu diễn trên các thông tin tương hỗ (Mutual Information) với các từ khác trong tập dữ liệu. Thông tin tương hỗ có thể được biểu diễn dưới dạng tần suất xuất hiện trong ma trận đồng xuất hiện trên toàn bộ tập dữ liệu hoặc xem xét trong giới hạn tập dữ liệu lân cận hoặc xem xét trên giới hạn những từ xung quanh.

Phương pháp Words Embedding hiện đại - Neural Embedding

* Word2vec: Thay vì đếm và xây dựng ma trận đồng xuất hiện, word2vec học trực tiếp word vector có số chiều thấp trong quá trình dự đoán các từ xung quanh mỗi từ. Đặc điểm của phương pháp này là nhanh hơn và có thể dễ dàng kết hợp một câu một văn bản mới hoặc thêm vào từ vựng. Word2vec là một mạng neural 2 lớp với duy nhất 1 tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector(với số chiều khoảng vài trăm), với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian. Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian. Dự đoán chính xác cao về ý nghĩa của một từ dựa trên những lần xuất hiện trước đây.

* Continuous Bag-of-Words (CBOW) là một kiểu kiến trúc mô hình Word2Vec được sử dụng trong xử lý ngôn ngữ tự nhiên. CBOW hướng đến việc học cách dự đoán từ hiện tại dựa trên ngữ cảnh xung quanh của nó. Trong CBOW, mô hình được huấn luyện để dự đoán từ hiện tại trong một câu dựa trên các từ xung quanh nó. Cụ thể, CBOW sử dụng ngữ liệu đầu vào là một cửa sổ trượt (sliding window) có độ dài cố định quanh từ hiện tại để dự đoán từ đó. Nó sẽ sử dụng các vector biểu diễn từ (word embeddings) của các từ trong cửa sổ trượt đó để tính toán trung bình và dự đoán vector biểu diễn của từ hiện tại. Điểm mạnh của CBOW là có thể học được các từ hiếm, thậm chí là chưa được biết đến trong quá trình huấn luyện. Do đó, CBOW thường được sử dụng trong các tác vụ xử lý ngôn ngữ tự nhiên như phân loại văn bản, dịch máy, hay tóm tắt văn bản. Tuy nhiên, điểm yếu của CBOW là nó không thể mô tả được sự phức tạp của một từ thông qua các thành phần con nhưng chỉ có thể đại diện bởi một vector biểu diễn duy nhất.

* FastText, được xây dựng trên Word2Vec bằng cách học các biểu diễn vectơ cho mỗi từ và n-gram được tìm thấy trong mỗi từ. Các giá trị của các biểu diễn sau đó được tính trung bình thành một vectơ ở mỗi bước đào tạo. Trong khi điều này bổ sung rất nhiều tính toán bổ sung cho việc đào tạo, nó cho phép nhúng từ để mã hóa thông tin từ phụ. Các vectơ FastText đã được chứng minh là chính xác hơn các vectơ Word2Vec bằng một số biện pháp khác nhau

PHƯƠNG PHÁP ĐỀ XUẤT GIẢI QUYẾT

Sử dụng mô hình FastText để đánh giá độ tương đồng

Ý tưởng chung của thuật toán là: đầu tiên sử dụng mô hình FastText được huấn luyện bởi bộ dữ liệu VietNam Wikipedia để kiểm tra độ tương đồng giữa 2 câu. Sau đó đối với các câu đã có trong bộ dữ liệu, chia thành các nhóm có độ tương đồng trên 92%.Cuối cùng sử dụng một thuật toán tìm kiếm để tìm kiếm câu mới trùng với nhóm nào trong bộ dữ liệu. Các bước thực hiện như sau:

1. Tiền xử lý dữ liệu: FastText sử dụng các từ và cụm từ như là các đơn vị xử lý. Trước khi xây dựng mô hình, dữ liệu được tiền xử lý để tách các từ và cụm từ ra từ các câu.

2. Tạo ra các n-grams: FastText sử dụng các n-grams của từ để tạo ra các đặc trưng cho từ đó. N-grams là các chuỗi các ký tự liên tiếp trong từ hoặc cụm từ. Ví dụ: với từ "dog", n-grams sẽ bao gồm "d", "o", "g", "do" và "og". Các n-grams này được sử dụng để tạo ra các đặc trưng cho từ hoặc cụm từ.

3. Tạo ra các vector đại diện cho từ hoặc cụm từ: FastText sử dụng các n-grams đã tạo ra để tạo ra các vector đại diện cho từ hoặc cụm từ đó. Các vector này được tạo ra bằng cách tính trung bình các vector tương ứng với các n-grams của từ hoặc cụm từ. Ví dụ: với từ "dog", các vector tương ứng

với các n-grams "d", "o" và "g" sẽ được lấy trung bình để tạo ra vector đại diện cho từ "dog".

4. Huấn luyện mô hình: Sau khi tạo ra các vector đại diện cho từ hoặc cụm từ, FastText sử dụng chúng để huấn luyện một mô hình không gian vector. Mô hình này có thể được sử dụng để tính toán khoảng cách giữa các từ hoặc cụm từ và đánh giá độ tương đồng giữa chúng.

5. Với cách thức hoạt động này, FastText cho phép xử lý các từ đơn và cụm từ một cách hiệu quả, đồng thời tạo ra các vector đại diện cho chúng có khả năng bao quát đầy đủ các đặc trưng của từ hoặc cụm từ đó. Từ mô hình đã huấn luyện, sẽ có được độ tương đồng giữa 2 câu đầu vào.

.

Phân nhóm theo độ tương đồng

Việc sử dụng phương pháp so sánh tuần tự qua tất cả các câu trong bộ dữ liệu để tìm các câu giống nhau có một số nhược điểm như sau:

1. Tốn thời gian và tài nguyên: Với bộ dữ liệu lớn, việc so sánh tuần tự qua tất cả các câu sẽ tốn nhiều thời gian và tài nguyên tính toán.

2. Độ chính xác thấp: Việc so sánh tuần tự qua tất cả các câu có thể dẫn đến kết quả không chính xác khi các câu khác nhau nhưng có một số từ giống nhau. Điều này có thể dẫn đến việc tìm ra những câu không liên quan và bỏ qua những câu quan trọng.

3. Khó xử lý dữ liệu lớn: Việc lưu trữ tất cả các câu trong bộ dữ liệu và tính toán độ tương đồng giữa chúng sẽ tốn nhiều bộ nhớ và khó xử lý với các bộ dữ liệu lớn.

4. Không cập nhật được: Nếu bộ dữ liệu thay đổi thường xuyên, phương pháp so sánh tuần tự qua tất cả các câu sẽ không cập nhật được các thay đổi mới.

Vì vậy, chúng tôi đề xuất phương pháp chia các câu vào các nhóm có độ tương đồng trên 90% và chọn ra một câu đại diện cho mỗi nhóm để thực hiện so sánh sẽ giảm thiểu các nhược điểm này và giúp cho việc tìm kiếm các câu giống nhau trở nên hiệu quả và nhanh chóng hơn. Thuật toán được mô tả như sau:

1. Đầu tiên, ta sẽ chuyển đổi các câu trong bộ dữ liệu và câu mới sang dạng vector. Mỗi vector sẽ biểu diễn một câu bằng cách sử dụng mô hình nhúng từ (word embedding), ở đây ta sử dụng mô hình FastText, để biểu diễn từng từ trong câu bằng một vector số thực có số chiều cố định. Sau đó, ta sẽ kết hợp các vector này thành một vector duy nhất để biểu diễn toàn bộ câu.

2. Tiếp theo, ta sẽ tính toán độ tương đồng cosine giữa các vector đại diện cho các câu trong bộ dữ liệu và lấy các câu có độ tương đồng cao hơn 90% để tạo thành các nhóm.

3. Khi có một câu mới cần được so sánh với các câu trong bộ dữ liệu, ta sẽ thực hiện các bước sau đây:

    - Đầu tiên, ta sẽ chuyển đổi các câu trong bộ dữ liệu và câu mới sang dạng vector. Mỗi vector sẽ biểu diễn một câu bằng cách sử dụng mô hình nhúng từ (word embedding), ở đây ta sử dụng mô hình FastText, để biểu diễn từng từ trong câu bằng một vector số thực có số chiều cố định. Sau đó, ta sẽ kết hợp các vector này thành một vector duy nhất để biểu diễn toàn bộ câu.
    - Tiếp theo, ta sẽ tính toán độ tương đồng cosine giữa các vector đại diện cho các câu trong bộ dữ liệu và lấy các câu có độ tương đồng cao hơn 90% để tạo thành các nhóm.
    - Khi có một câu mới cần được so sánh với các câu trong bộ dữ liệu, ta sẽ thực hiện các bước sau đây:
        + Đầu tiên, ta sẽ so sánh câu mới với các câu đại diện trong các nhóm. Nếu câu mới trùng với câu đại diện trên 90%, ta sẽ coi câu mới trùng với tất cả các câu trong nhóm đó.
        + Nếu không tìm thấy bất kỳ câu đại diện nào trùng với câu mới trên 90%, ta sẽ thực hiện so sánh tuần tự với tất cả các câu trong bộ dữ liệu để tìm các câu có độ tương đồng cao nhất với câu mới.
        + Kết quả của thuật toán sẽ là tập hợp các câu trong bộ dữ liệu có độ tương đồng cao với câu mới. Từ đó, ta có thể áp dụng các phương pháp khác để phân tích hoặc sử dụng dữ liệu này cho mục đích khác.

Tổng quan về thuật toán này là nó giúp giảm thiểu thời gian và chi phí tính toán so với việc so sánh tuần tự qua tất cả các câu trong bộ dữ liệu. Bằng cách chia các câu thành các nhóm có độ tương đồng cao và chỉ so sánh với các câu đại diện, thuật toán giúp ta tìm kiếm các câu liên quan đến câu mới một cách nhanh chóng và hiệu quả hơn.

Xây dựng thuật toán so sánh với bộ dữ liệu lớn

Để giải quyết vấn đề trùng lặp câu trong bộ dữ liệu, chúng ta có thể sử dụng một phương pháp đơn giản là

so sánh tuần tự qua tất cả các câu trong bộ dữ liệu. Tuy nhiên, phương pháp này có nhược điểm là tốn thời gian và tài nguyên tính toán, đặc biệt khi bộ dữ liệu lớn. Một cách để cải thiện phương pháp trên là thay vì so sánh qua tất cả các câu, chúng ta chia các câu thành các nhóm có độ tương đồng trên 90% và chọn ra 1 câu làm câu đại diện cho mỗi nhóm. Sau đó, ta chỉ cần so sánh câu mới với các câu đại diện này để xác định xem câu mới có trùng lặp với câu nào trong bộ dữ liệu hay không. Nếu câu mới trùng với câu đại diện trên 90%, thì ta coi câu mới trùng với tất cả các câu trong nhóm tương ứng. Tuy nhiên, cách tiếp cận trên cũng có nhược điểm. Đó là nếu câu mới không trùng với bất kỳ câu đại diện nào trong các nhóm được chọn, thì phương pháp này không thể tìm ra các câu tương tự trong bộ dữ liệu. Đồng thời tốn thời gian và bộ nhớ khi thực hiện so sánh tuần tự qua từng nhóm. Để khắc phục nhược điểm này, chúng ta có thể thực hiện một số bước cải tiến thuật toán, ví dụ như sau:

1. Xây dựng ma trận độ tương đồng giữa các câu trong bộ dữ liệu. Ma trận này có thể được tính bằng cách tính độ tương đồng giữa các câu đại diện của các nhóm.

2. Chọn ra nhóm G1 là nhóm có số câu lớn nhất trong bộ dữ liệu. So sánh câu mới với các câu trong nhóm G1 và tìm câu có độ tương đồng lớn nhất với câu mới. Nếu độ tương đồng này lớn hơn 90%, thì trả về kết quả tương ứng. Nếu không, chuyển sang bước 3.

3. Xóa các nhóm có độ tương đồng lớp so với nhóm G1 với độ tương đồng lớn hơn sim + 0.1 khỏi hàng đợi so sánh, với sim là độ tương đồng lớn nhất tìm được ở bước 2. Tiếp tục chọn ra nhóm có độ tương đồng lớn nhất so với nhóm

Ưu điểm của phương pháp này là tăng hiệu suất xử lý vì chỉ cần so sánh câu mới với các nhóm có độ tương đồng cao hơn một ngưỡng xác định thay vì so sánh tuần tự qua tất cả các câu trong bộ dữ liệu. Khi số lượng câu trong bộ dữ liệu lớn, phương pháp này giúp giảm thiểu thời gian xử lý và tăng tốc độ trả lời. Ngoài ra, việc chọn ra câu đại diện cho mỗi nhóm giúp giảm độ phức tạp của việc so sánh và cải thiện độ chính xác của kết quả trả về.

THỬ NGHIỆM VÀ ĐÁNH GIÁ

This statement requires citation [1]. This statement requires multiple citations [1, 2]. This statement contains an in-text citation, for directly referring to a citation like so: Jones and Smith [2].

Subsection One

Suspendisse potenti. Vivamus suscipit dapibus metus. Proin auctor iaculis ex, id fermentum lectus dapibus tristique. Nullam maximus eros eget leo pretium dapibus. Nunc in auctor erat, id interdum risus. Suspendisse aliquet vehicula accumsan. In vestibulum efficitur dictum. Sed ultrices, libero nec fringilla feugiat, elit massa auctor ligula, vehicula tempor ligula felis in lectus. Suspendisse sem dui, pharetra ut sodales eu, suscipit sit amet felis. Donec pretium viverra ante, ac pulvinar eros. Suspendisse gravida consectetur urna. Pellentesque vitae leo porta, imperdiet eros eget, posuere sem. Praesent eget leo efficitur odio bibendum condimentum sit amet vel ex. Nunc maximus quam orci, quis pulvinar nibh eleifend ac. Quisque consequat lacus magna, eu posuere tellus iaculis ac. Sed vitae tortor tincidunt ante sagittis iaculis.

Subsection Two

Nullam mollis tellus lorem, sed congue ipsum euismod a. Donec pulvinar neque sed ligula ornare sodales. Nulla sagittis vel lectus nec laoreet. Nulla volutpat malesuada turpis at ultricies. Ut luctus velit odio, sagittis volutpat erat aliquet vel. Donec ac neque eget neque volutpat mollis. Vestibulum viverra ligula et sapien bibendum, vel vulputate ex euismod. Curabitur nec velit velit. Aliquam vulputate lorem elit, id tempus nisl finibus sit amet. Curabitur ex turpis, consequat at lectus id, imperdiet molestie augue. Curabitur eu eros molestie purus commodo hendrerit. Quisque auctor ipsum nec mauris malesuada, non fringilla nibh viverra. Quisque gravida, metus quis semper pulvinar, dolor nisl suscipit leo, vestibulum volutpat ante justo ultrices diam. Sed id facilisis turpis, et aliquet eros.

Subsubsection Example Duis venenatis eget lectus a aliquet. Integer vulputate ante suscipit felis feugiat rutrum. Aliquam eget dolor eu augue elementum ornare. Nulla fringilla interdum volutpat. Sed tincidunt, neque quis imperdiet hendrerit, turpis sapien ornare justo, ac blandit felis sem quis diam. Proin luctus urna sit amet felis tincidunt, sed congue nunc pellentesque. Ut faucibus a magna faucibus finibus. Etiam id mi euismod, auctor nisi eget, pretium metus. Proin tincidunt interdum mi non interdum. Donec semper luctus dolor at elementum. Aenean eu congue tortor, sed hendrerit magna. Quisque a dolor ante. Mauris semper id urna id gravida. Vestibulum mi tortor, finibus eu felis in, vehicula aliquam mi.

Aliquam arcu turpis, ultrices sed luctus ac, vehicula id metus. Morbi eu feugiat velit, et tempus augue. Proin ac mattis tortor. Donec tincidunt, ante rhoncus luctus semper, arcu lorem lobortis justo, nec convallis ante quam quis lectus. Aenean tincidunt sodales massa, et hendrerit tellus mattis ac. Sed non pretium nibh.

Donec cursus maximus luctus. Vivamus lobortis eros et massa porta porttitor. Nam vitae suscipit mi. Pellentesque ex tellus, iaculis vel libero at, cursus pretium sapien. Curabitur accumsan velit sit amet nulla lobortis, ut pretium ex aliquam. Proin eget volutpat orci. Morbi eu aliquet turpis. Vivamus molestie urna quis tempor tristique. Proin hendrerit sem nec tempor sollicitudin.

Tài liệu

[1] J. M. Smith and A. B. Jones. Book Title. 7th. Publisher, 2023.

[2] A. B. Jones and J. M. Smith. “Article Title”. In: Journal title 13.52 (Mar. 2024), pp. 123–456. DOI: 10.1038/s41586-021-03616-x.