

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC KINH TẾ - KỸ THUẬT
CÔNG NGHIỆP

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT
ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN
NĂM HỌC 2024 – 2025

ĐỀ TÀI: “ỨNG DỤNG MÔ HÌNH NGÔN NGỮ LỚN
ĐO LƯỜNG ĐỘ TƯƠNG TỰ CỦA VĂN BẢN
SONG NGỮ VIỆT – TRUNG”

Giảng viên hướng dẫn: TS Bùi Văn Tân

ThS Trần Thị Lan Anh

Chủ nhiệm đề tài:

Vũ Trung Hiếu

Lớp: DHTI16A1HN

Thành viên:

Lương Đức Thắng

Lớp: DHKL16A2HN

Phạm Việt Anh

Lớp: DHTI16A1HN

Nguyễn Tiến Bình

Lớp: DHTI16A1HN

HÀ NỘI, THÁNG 4 NĂM 2025

BÁO CÁO TỔNG KẾT
ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN
NĂM HỌC 2024 – 2025

ĐỀ TÀI: “ỨNG DỤNG MÔ HÌNH NGÔN NGỮ LỚN
ĐO LƯỜNG ĐỘ TƯƠNG TỰ CỦA VĂN BẢN
SONG NGỮ VIỆT – TRUNG”

Giảng viên hướng dẫn: TS Bùi Văn Tân

ThS Trần Thị Lan Anh

Chủ nhiệm đề tài: Vũ Trung Hiếu Lớp: DHTI16A1HN

Thành viên: Lương Đức Thắng Lớp: DHKL16A2HN

Phạm Việt Anh Lớp: DHTI16A1HN

Nguyễn Tiến Bình Lớp: DHTI16A1HN

HÀ NỘI, THÁNG 4 NĂM 2025

MỤC LỤC	
DANH MỤC CÁC BẢNG DỮ LIỆU TRONG ĐỀ TÀI	5
DANH MỤC CÁC HÌNH TRONG ĐỀ TÀI.....	6
DANH MỤC CÁC TỪ VIẾT TẮT	7
LỜI MỞ ĐẦU	8
THÔNG TIN NGHIÊN CỨU CỦA ĐỀ TÀI.....	9
CHƯƠNG I: KIẾN THỨC NỀN TẢNG.....	13
1.1. Mô tả bài toán “Ứng dụng mô hình ngôn ngữ lớn đo lường độ tương tự của văn bản song ngữ Việt – Trung.....	13
1.1.1. Mục tiêu bài toán.....	13
1.1.2. Quy trình thực hiện	14
1.2. Mạng Noron nhân tạo	14
1.2.1. Cấu trúc cơ bản của mạng nơ-ron.....	16
1.2.2. Cơ chế truyền tín hiệu và lan truyền ngược.....	18
1.2.3. Hàm kích hoạt (Activation Functions)	20
1.2.4. Perceptron và Multilayer Perceptron (MLP)	21
1.2.5. Mạng nơ-ron trong xử lý ngôn ngữ tự nhiên (NLP).....	23
1.2.6. Vai trò của mạng nơ-ron trong NLP hiện đại	26
1.3. Mô hình ngôn ngữ lớn	27
1.3.1. Tổng quan.....	27
1.3.2. BERT	28
1.3.3. PhoBERT	32
1.3.4. VisoBERT	33
1.3.5. SimCSE	35
CHƯƠNG 2: ĐO LƯỜNG ĐỘ TƯƠNG TỰ CỦA VĂN BẢN DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN.....	37
2.1. Một số phương pháp tính độ tương đồng văn bản.....	37
2.1.1. COSINE SIMILARITY	37
2.1.2. JACCARD SIMILARITY	38
2.1.3. LEVENSHTAIN DISTANCE.....	39
2.1.4. MANHATTAN SIMILARITY	40
2.2. MÔ HÌNH ĐỀ XUẤT	41
2.2.1. WORD EMBEDDING	41
2.2.2. SENTENCE EMBEDDING	44

2.2.3. SENTENCE TRANSFORMERS	46
2.2.4. COMBSENTSIM	48
CHƯƠNG 3: THỰC NGHIỆM VÀ PHÂN TÍCH	51
3.1. BỘ DỮ LIỆU	51
3.1.1. Quy mô và lĩnh vực nội dung	51
3.1.2. Tiền xử lý dữ liệu.....	52
3.1.3. Chọn lọc các cặp câu	53
3.1.4. Đánh giá dữ liệu.....	53
3.1.5. Cấu trúc bộ dữ liệu.....	54
3.2: THỰC NGHIỆM VÀ GIAO DIỆN	54
3.2.1. Các mô hình ngôn ngữ lớn được áp dụng.....	54
3.2.2. Công cụ và các thư viện chính:.....	55
3.2.3. Quy trình thực hiện	56
3.2.4. Chương trình giao diện trực quan.....	59
3.3. ĐÁNH GIÁ MÔ HÌNH.....	65
3.3.1. Các phương pháp đánh giá năng suất mô hình trong bài toán xử lý ngôn ngữ tự nhiên	65
3.3.2. Spearman và Kendall: Đánh giá thứ hạng và mức độ tương tự	67
3.3.3. Hệ số Kappa – Đo lường mức độ đồng thuận giữa mô hình và con người	70
3.3.4. Đánh giá.....	74
KẾT LUẬN VÀ KIẾN NGHỊ.....	76
TÀI LIỆU THAM KHẢO.....	78

DANH MỤC CÁC BẢNG DỮ LIỆU TRONG ĐỀ TÀI

Bảng 1.1: So sánh đặc trưng ngôn ngữ Tiếng Việt và Tiếng Trung

Bảng 1.2: Pre-trained models

Bảng 1.3: Thông số của PhoBERT-base và PhoBERT-large

Bảng 3.1: Diễn giải giá trị Kappa theo ngữ cảnh

DANH MỤC CÁC HÌNH TRONG ĐỀ TÀI

Hình 1.1: Cấu trúc mạng Noron nhiều lớp (Multilayer Perceptron – MLP)

Hình 1.2: Mô hình Perceptron

Hình 1.3: Mô phỏng hoạt động của MLM

Hình 1.4: Mô phỏng hoạt động của NSP

Hình 1.5: Phân loại mối quan hệ ngữ nghĩa giữa 2 câu

Hình 1.6: Công thức mất mát trong phép nhúng câu

Hình 2.1: Biểu diễn từ thành vector bằng phương pháp One - hot encoding

Hình 2.2: Biểu đồ tính similar sentence-transformers

Hình 3.1: Biểu đồ lượng cặp câu từng miền(đã điều chỉnh)

Hình 3.2: Biểu đồ tính mức độ tương đồng dựa trên thang điểm

Hình 3.3: Mô hình tổng quát

Hình 3.4 – 3.6: Giao diện

Hình 3.7 - 3.10: Chạy chương trình

Hình 3.11: Kết quả đo lường

Hình 3.12: Trực quan hóa kết quả trên biểu đồ

Hình 3.13: Đồ thị so sánh độ tương quan giữa 3 mô hình bằng độ đo Spearman

DANH MỤC CÁC TỪ VIẾT TẮT

Viết tắt	Tên đầy đủ	Ghi chú
AI	Artificial Intelligence	Trí tuệ nhân tạo
ANN	Artificial Neural Network	Mạng nơ-ron nhân tạo
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
MLP	Multilayer Perceptron	Mạng nơ-ron nhiều lớp
LLM	Large Language Model	Mô hình ngôn ngữ lớn
BERT	Bidirectional Encoder Representations from Transformers	Mô hình ngôn ngữ hai chiều
MLM	Masked Language Modeling	Mô hình ngôn ngữ bị che
NSP	Next Sentence Prediction	Dự đoán câu tiếp theo
TF-IDF	Term Frequency - Inverse Document Frequency	Tần suất từ - Tần suất nghịch đảo
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
RNN	Recurrent Neural Network	Mạng nơ-ron hồi tiếp
LSTM	Long Short-Term Memory	Bộ nhớ ngắn dài hạn
GRU	Gated Recurrent Unit	Đơn vị hồi tiếp có cổng
POS	Part-Of-Speech	Từ loại
CRF	Conditional Random Field	Trường ngẫu nhiên có điều kiện
ReLU	Rectified Linear Unit	Hàm kích hoạt ReLU
BiLSTM	Bidirectional Long Short-Term Memory	LSTM hai chiều
XLM-RoBERTa	Cross-lingual Language Model RoBERTa	Mô hình ngôn ngữ đa ngữ
GPT	Generative Pre-trained Transformer	Mô hình sinh ngôn ngữ được huấn luyện trước
RDRSegmenter	RDR-based Vietnamese Word Segmenter	Công cụ tách từ tiếng Việt
NER	Named Entity Recognition	Nhận dạng thực thể có tên

LỜI MỞ ĐẦU

Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Kinh Tế Kỹ Thuật Công Nghiệp, Khoa Công Nghệ Thông Tin đã tạo điều kiện thuận lợi để những ý tưởng nghiên cứu sinh viên của chúng em được thực hiện.

Nhóm nghiên cứu xin bày tỏ lòng biết ơn sâu sắc đến các Thầy/cô giảng viên vì sự hướng dẫn tận tình, những lời khuyên quý giá, và sự đồng hành xuyên suốt quá trình nghiên cứu. Nhờ có sự chỉ dẫn của thầy/cô, chúng em mới có thể hoàn thiện công trình nghiên cứu này một cách tốt nhất.

Bên cạnh đó, xin gửi lời cảm ơn đến các thành viên của nhóm nghiên cứu vì chúng ta đã cùng nhau đồng hành và hoàn thiện Đề tài NCKH đúng như dự kiến ban đầu. Sự đồng hành của các thành viên trong nhóm là nguồn động lực to lớn góp phần tạo ra một đề tài hoàn chỉnh, vượt qua những khó khăn và thử thách trong quá trình thực hiện nghiên cứu.

Cuối cùng, nhóm nghiên cứu chân thành cảm ơn tất cả các cá nhân và tổ chức đã cung cấp tài liệu, chia sẻ kinh nghiệm và kiến thức, cũng như hỗ trợ những điều kiện cần thiết để tôi có thể hoàn thành nghiên cứu này.

Mặc dù đã cố gắng hết sức, chúng tôi tự nhận thức rằng báo cáo nghiên cứu này khó tránh khỏi những thiếu sót. Vì vậy, chúng tôi rất mong nhận được những góp ý từ quý thầy cô, bạn bè và đồng nghiệp để có thể hoàn thiện hơn trong tương lai.

THÔNG TIN NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung

- **Tên đề tài:** “Ứng dụng mô hình ngôn ngữ đo lường độ tương tự của văn bản song ngữ Việt Trung”
- **Lý do chọn đề tài:**

Bài toán đo lường độ tương tự ngữ nghĩa giữa các văn bản song ngữ nhằm mục đích xác định mức độ tương đồng về nội dung giữa các văn bản được viết bằng các ngôn ngữ khác nhau. Đầu vào của bài toán bao gồm hai văn bản thuộc các ngôn ngữ khác nhau, trong khi đầu ra là một đánh giá số liệu hoặc vector biểu thị mức độ tương tự ngữ nghĩa giữa chúng [1, 2, 3].

Nghiên cứu này có ứng dụng rộng rãi trong các lĩnh vực như dịch máy tự động, phân tích nội dung đa ngôn ngữ, và tạo ra các hệ thống gợi ý nội dung đa ngôn ngữ [4]. Các ứng dụng cụ thể bao gồm hệ thống tìm kiếm đa ngôn ngữ, dịch thuật tự động, và phân loại văn bản đa ngôn ngữ [5].

Các hướng tiếp cận phổ biến để giải quyết bài toán này bao gồm sử dụng các phương pháp nhúng từ (word embeddings) như Word2Vec [12], GloVe; sử dụng các mô hình biểu diễn ngữ nghĩa dựa trên mạng nơ-ron học sâu như Transformer, BERT, RoBERTa, XLNet, ELECTRA, Mistral [6]; và sử dụng các phương pháp biểu diễn đồ thị tri thức (knowledge graph representations) [7].

Các tiến bộ gần đây trong lĩnh vực này bao gồm việc áp dụng mô hình BERT và các mô hình ngôn ngữ lớn cho tiếng Anh và tiếng Việt để nâng cao hiệu năng biểu diễn ngữ nghĩa và đo lường độ tương tự ngữ nghĩa giữa các văn bản. Ngoài ra, sự phát triển của các phương pháp biểu diễn ngữ nghĩa không giám sát cũng đóng góp nâng cao độ chính xác của việc đo lường độ tương tự [8].

Tuy nhiên, để đo lường chính xác độ tương tự ngữ nghĩa giữa các văn bản song ngữ vẫn là một thách thức. Các thách thức chính bao gồm: đa ngữ nghĩa và ngữ cảnh, sự khác biệt về ngữ pháp và cấu trúc câu giữa các ngôn ngữ, cũng như sự hạn chế về dữ liệu huấn luyện đa ngôn ngữ và độ phức tạp trong việc hiểu và biểu diễn ngữ nghĩa chính xác [9].

Nghiên cứu đo lường độ tương tự ngữ nghĩa giữa các văn bản song ngữ là một lĩnh vực nghiên cứu quan trọng trong xử lý ngôn ngữ tự nhiên, với tiềm năng ứng dụng rộng rãi. Mặc dù đã có nhiều tiến bộ, nhưng cần phải giải quyết nhiều thách thức để nâng cao hiệu quả và độ chính xác của các phương pháp trong việc đo lường độ tương tự ngữ nghĩa giữa các văn bản song ngữ [10, 11].

Tiêu chí	Tiếng Việt	Tiếng Trung
Loại ngôn ngữ	Ngôn ngữ đơn lập, thanh điệu	Ngôn ngữ đơn lập, thanh điệu
Bảng chữ cái	Chữ Latinh	Chữ Hán
Đơn vị từ	Phân tách từ rõ ràng bằng khoảng trắng	Không có khoảng trắng giữa các từ
Cách biểu đạt	Giàu hình ảnh, giàu nghĩa bóng	Ngắn gọn, súc tích, giàu ngữ cảnh
Từ vựng vay mượn	Nhiều từ gốc Hán, tiếng Pháp	Ít vay mượn, chủ yếu phát triển nội tại
Thách thức khi xử lý NLP	Phân tách từ, đồng âm, đa nghĩa	Tách từ, ký tự không có nghĩa độc lập

Bảng 1.1: So sánh đặc trưng ngôn ngữ Tiếng Việt và Tiếng Trung

2. Mục tiêu, đối tượng, phạm vi đề tài

- Mục tiêu:

Nghiên cứu nhằm áp dụng các mô hình ngôn ngữ lớn như BERT và các mạng nơ-ron học sâu để cải thiện độ chính xác và hiệu suất trong việc đo lường độ tương tự ngữ nghĩa giữa các văn bản song ngữ, đồng thời khảo sát và giải quyết các thách thức như đa ngữ nghĩa, sự khác biệt về ngữ pháp, và sự thiếu dữ liệu huấn luyện đa ngôn ngữ [13].

- Đối tượng, phạm vi đề tài:

- Nghiên cứu mô hình ngôn ngữ lớn.
- Nghiên cứu mạng nơ-ron học sâu.
- Nghiên cứu bài toán đo lường độ tương tự của văn bản song ngữ Việt - Trung.

- Triển khai ứng dụng trên máy tính

3. Cơ sở lý thuyết và lịch sử nghiên cứu:

- Cơ sở lý thuyết:

Đề tài nghiên cứu này dựa trên các cơ sở lý thuyết sau: Mô hình ngôn ngữ lớn (LLMs) được phát triển nhằm nắm bắt ngữ nghĩa và ngữ cảnh sâu sắc từ dữ liệu văn bản lớn, giúp cải thiện độ chính xác trong các nhiệm vụ xử lý ngôn ngữ tự nhiên. Mạng nơ-ron học sâu, đặc biệt là các kiến trúc như Transformer, được áp dụng để xử lý và so sánh các cấu trúc phức tạp của văn bản [14, 15]. Phương pháp này tận dụng khả năng học biểu diễn ngữ nghĩa phong phú, cho phép đo lường độ tương tự văn bản giữa các ngôn ngữ khác nhau bằng cách ánh xạ chúng vào cùng một không gian ngữ nghĩa [16]. Các kỹ thuật như song ngữ embeddings và các mô hình pretrained, như BERT hoặc GPT, cũng đóng vai trò quan trọng trong việc cải thiện khả năng phân tích và so sánh văn bản song ngữ.

- Lịch sử nghiên cứu:

Nghiên cứu về đo độ tương tự giữa các văn bản bắt đầu từ các phương pháp dựa trên so khớp từ khóa và tần suất từ (TF-IDF) trong những năm 1970. Đến những năm 1990, các mô hình không gian vector và cosine similarity trở nên phổ biến, cho phép biểu diễn văn bản trong không gian vector [18]. Năm 2003, mô hình LSA (Latent Semantic Analysis) được sử dụng để nắm bắt mối quan hệ ngữ nghĩa sâu hơn. Sau đó, Word2Vec (2013) [12] và GloVe (2014) mang đến cách tiếp cận dựa trên embeddings, tạo ra biểu diễn từ có ngữ nghĩa [19]. Gần đây, sự ra đời của Transformer và các mô hình ngôn ngữ lớn như BERT và GPT đã cải thiện đáng kể khả năng đo lường độ tương tự giữa văn bản, cho phép hiểu sâu hơn về ngữ nghĩa và ngữ cảnh [20].

4. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết
- Đánh giá kết quả nghiên cứu
- Xây dựng cơ sở dữ liệu
- Thực nghiệm demo sản phẩm
- Phân tích kết quả

5. Nội dung nghiên cứu

Chương 1: Kiến thức nền tảng:

- Mô hình Ngôn ngữ Lớn (LLMs)

- Khả năng học biểu diễn ngữ nghĩa sâu sắc từ dữ liệu lớn.
- Ứng dụng trong các nhiệm vụ đa ngôn ngữ và song ngữ.

- Mạng Nơ-ron Học Sâu - Kiến trúc Transformer và tác động đến NLP.

- Vai trò của các lớp attention trong việc hiểu ngữ cảnh. Đo Lường Độ Tương Tự Văn Bản
- Phương pháp dựa trên embeddings như Word2Vec, BERT.
- Đánh giá độ tương tự dựa trên khoảng cách trong không gian vector.

Chương 2: Đo lường độ tương tự văn bản dựa trên mô hình Ngôn ngữ lớn

- Áp dụng các độ đo độ tương tự văn bản như: Cosine, Jaccard, WordNet.... vào các mô hình ngôn ngữ lớn để đưa ra được độ đo tương tự văn bản trên các cặp câu.
- Thực hiện tạo bộ dữ liệu dựa trên những đánh giá thực tế của chuyên gia

Chương 3: Thực nghiệm và phân tích

- Áp dụng LLMs trong xử lý văn bản song ngữ.
- So sánh với các mô hình truyền thống về độ chính xác và hiệu quả.

- Kết quả và Ứng dụng

- Cải thiện đáng kể trong đo lường độ tương tự.
- Mô hình demo
- Ứng dụng trong dịch máy, tìm kiếm thông tin và gợi ý nội dung.

CHƯƠNG I: KIẾN THỨC NỀN TẢNG

1.1. Mô tả bài toán “Ứng dụng mô hình ngôn ngữ lớn đo lường độ tương tự của văn bản song ngữ Việt – Trung

Sống trong thời đại công nghệ phát triển vượt bậc như ngày nay, trí tuệ nhân tạo (AI) đã trở thành một phần quan trọng trong cuộc sống của con người. Không chỉ dừng lại ở việc hỗ trợ những tác vụ hàng ngày, AI còn đang mở rộng phạm vi ứng dụng vào các lĩnh vực phức tạp như y học, giáo dục, kinh doanh và nghiên cứu khoa học. Từ việc giúp tối ưu hóa hiệu suất công việc đến việc khai phá những tiềm năng chưa từng được khám phá, AI đang làm thay đổi cách chúng ta tương tác với thế giới.

Trong đó, một khía cạnh đáng chú ý là khả năng xử lý ngôn ngữ của AI, góp phần thúc đẩy sự giao tiếp và tương tác hiệu quả giữa con người với máy móc, cũng như giữa các nền văn hóa và ngôn ngữ khác nhau. Việc đo lường mức độ tương đồng ngữ nghĩa giữa các văn bản không chỉ mang lại lợi ích thực tiễn mà còn mở ra nhiều cơ hội nghiên cứu trong lĩnh vực ngôn ngữ học ứng dụng. Đó chính là lý do nghiên cứu này được thực hiện nhằm khám phá tiềm năng của AI trong việc xử lý dữ liệu ngôn ngữ song ngữ một cách hiệu quả và chính xác.

Một trong những vấn đề cốt lõi trong nghiên cứu xử lý ngôn ngữ song ngữ là làm thế nào để đo lường được mức độ tương đồng ngữ nghĩa giữa hai ngôn ngữ một cách chính xác. Kết quả đo lường không chỉ giúp kiểm tra tính nhất quán ngữ nghĩa giữa bản gốc và bản dịch trong dịch máy mà còn mang lại lợi ích to lớn trong việc xây dựng hệ thống truy vấn thông tin và gợi ý nội dung. Chẳng hạn, trong dịch máy, nếu hệ thống có thể đánh giá ngữ nghĩa giữa bản dịch và bản gốc một cách hiệu quả, độ chính xác trong dịch thuật sẽ được cải thiện rõ rệt.

1.1.1. Mục tiêu bài toán

Bài toán “Ứng dụng mô hình ngôn ngữ lớn đo lường độ tương tự của văn bản song ngữ Việt - Trung” vẫn được kế thừa và phát triển từ những bài toán đo lường ngôn ngữ trước đó, hướng tới mục tiêu xác định mức độ tương đồng ngữ nghĩa giữa 2 văn bản viết bằng tiếng Việt và tiếng Trung. Đồng thời đánh giá hiệu quả của các mô hình ngôn ngữ lớn trong việc xử lý dữ liệu song ngữ. Hướng tới ứng dụng kết

quả đo lường vào các lĩnh vực dịch thuật, phân tích ngữ nghĩa, hoặc xây dựng hệ thống hỗ trợ ngôn ngữ.

Nghiên cứu này không chỉ có ý nghĩa khoa học mà còn mang lại giá trị thực tiễn to lớn trong các lĩnh vực như dịch thuật, tìm kiếm tài liệu song ngữ, hay gợi ý nội dung thông minh. Ví dụ, hệ thống gợi ý nội dung có thể hỗ trợ người dùng khám phá những thông tin hữu ích từ cả hai ngôn ngữ, đảm bảo sự đa dạng và phù hợp với sở thích. Điều này không chỉ góp phần nâng cao hiệu quả công việc mà còn đóng góp tích cực vào việc phát triển công nghệ ngôn ngữ song ngữ trong tương lai.

1.1.2. Quy trình thực hiện

Để đạt được những mục tiêu trên, nghiên cứu được triển khai qua bốn giai đoạn cơ bản. Đầu tiên, dữ liệu văn bản tiếng Việt và tiếng Trung sẽ được xử lý, làm sạch và chuẩn hóa, đảm bảo chất lượng đầu vào cho các mô hình.

Tiếp theo, sử dụng các mô hình ngôn ngữ lớn hiện đại khác nhau đã có để mã hóa văn bản và trích xuất đặc trưng ngữ nghĩa.

Giai đoạn thứ ba là đo lường mức độ tương đồng ngữ nghĩa giữa các văn bản, áp dụng các phương pháp tính toán độ tương tự như Cosine Similarity hoặc Euclidean Distance... Độ tương đồng này có thể được đánh giá dựa trên nhiều yếu tố khác nhau như:

1. Từ vựng: Sự xuất hiện của các từ giống nhau hoặc tương tự trong hai câu cho thấy mức độ tương đồng giữa chúng.
2. Cấu trúc: Cấu trúc câu có thể cung cấp thông tin về mức độ tương đồng giữa hai câu.
3. Ngữ nghĩa: Ý nghĩa của hai câu có thể được so sánh để đánh giá độ tương đồng giữa chúng.
4. Ngữ cảnh: Các từ và câu trước và sau có thể ảnh hưởng đến mức độ tương đồng giữa hai câu.

Cuối cùng, kết quả đo lường giữa các mô hình ngôn ngữ sẽ được đánh giá, hiệu chỉnh để tối ưu hóa độ chính xác và hiệu suất của hệ thống.

1.2. Mạng Nơ-ron nhân tạo

Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) là một mô hình tính toán được phát triển dựa trên cảm hứng từ cấu trúc và cơ chế xử lý thông tin của hệ

thần kinh sinh học, đặc biệt là não bộ con người. Trong hệ thần kinh sinh học, thông tin được truyền qua các nơ-ron thông qua các xung điện và kết nối synapse. Tương tự, trong ANN, thông tin được xử lý thông qua các đơn vị tính toán cơ bản gọi là nơ-ron nhân tạo, được tổ chức thành từng lớp (layers) và kết nối với nhau bằng các trọng số (weights) có thể điều chỉnh.

Một mạng nơ-ron điển hình thường bao gồm ba loại tầng chính: tầng đầu vào (input layer), một hoặc nhiều tầng ẩn (hidden layers), và tầng đầu ra (output layer). Mỗi tầng ẩn trong mạng có vai trò học biểu diễn đặc trưng trừu tượng từ dữ liệu đầu vào, thông qua việc kết hợp tuyến tính các tín hiệu đầu vào, sau đó áp dụng một hàm kích hoạt phi tuyến. Quá trình này cho phép mạng nơ-ron có khả năng mô hình hóa các quan hệ phi tuyến phức tạp trong không gian dữ liệu, điều mà các mô hình học máy tuyến tính truyền thống thường không thể thực hiện hiệu quả.

Khả năng học của ANN được thể hiện thông qua việc điều chỉnh các trọng số kết nối giữa các nơ-ron sao cho hàm mục tiêu (thường là hàm mất mát) được tối thiểu hóa. Việc điều chỉnh này được thực hiện thông qua thuật toán lan truyền ngược (backpropagation), kết hợp với các thuật toán tối ưu như gradient descent hoặc các biến thể hiện đại như Adam, RMSprop. Nhờ đó, ANN có khả năng học từ dữ liệu để tự động khám phá và trích xuất ra các đặc trưng quan trọng phục vụ cho mục tiêu dự đoán hoặc phân loại.

Một điểm nổi bật của mạng nơ-ron nhân tạo so với nhiều mô hình học máy cổ điển là khả năng học biểu diễn tự động từ dữ liệu thô mà không cần thiết kế thủ công các đặc trưng đầu vào. Điều này đặc biệt quan trọng trong các lĩnh vực mà dữ liệu có tính chất đa chiều, phi cấu trúc như hình ảnh, âm thanh hoặc ngôn ngữ tự nhiên. Ví dụ, trong thị giác máy tính, ANN có thể nhận dạng khuôn mặt hoặc vật thể từ ảnh số; trong xử lý ngôn ngữ tự nhiên, ANN được ứng dụng trong các bài toán như dịch máy, phân loại cảm xúc, và trích xuất thông tin.

Tuy nhiên, bên cạnh những ưu điểm vượt trội, mạng nơ-ron nhân tạo cũng tồn tại một số thách thức đáng kể. Thứ nhất, mạng thường yêu cầu một lượng lớn dữ liệu để đạt hiệu suất tốt, do số lượng tham số cần học là rất lớn. Thứ hai, mô hình có xu hướng khó huấn luyện khi độ sâu tăng, đặc biệt là do các vấn đề như tiêu biến gradient

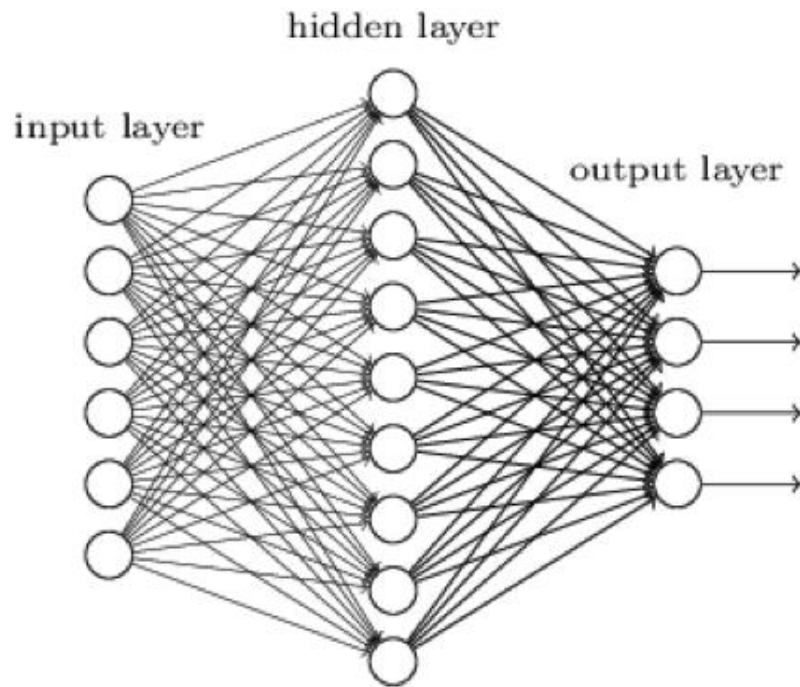
(vanishing gradient) hoặc bùng nổ gradient (exploding gradient). Thứ ba, ANN thường bị xem là mô hình "hộp đen" (black-box) vì khó diễn giải rõ ràng lý do đằng sau các dự đoán của nó, làm giảm tính minh bạch trong các ứng dụng nhạy cảm.

Bất chấp những hạn chế nêu trên, sự phát triển mạnh mẽ của ANN trong những thập kỷ gần đây – đặc biệt là với sự ra đời của các mô hình học sâu (deep learning) – đã và đang khẳng định vai trò trung tâm của kiến trúc này trong lĩnh vực trí tuệ nhân tạo hiện đại. ANN không chỉ là nền tảng lý thuyết cho các mô hình tiên tiến hơn như mạng tích chập (CNN), mạng hồi tiếp (RNN), mà còn là trụ cột trong nhiều hệ thống thông minh thực tế.

1.2.1. Cấu trúc cơ bản của mạng nơ-ron

Từ mô hình Perceptron đơn, các nhà nghiên cứu đã phát triển khái niệm nơ-ron nhân tạo (Artificial Neuron) để khắc phục các giới hạn của hàm quyết định tuyến tính, đặc biệt là việc không thể xử lý các bài toán không tuyến tính như XOR. Điểm khác biệt quan trọng giữa nơ-ron nhân tạo và Perceptron nằm ở việc thay thế hàm bước (step function) bằng các hàm kích hoạt phi tuyến như sigmoid, tanh hoặc ReLU. Điều này giúp mô hình có khả năng học được các quan hệ phức tạp trong dữ liệu thực tế.

Mạng nơ-ron nhân tạo được xây dựng từ nhiều tầng (layers) khác nhau, trong đó mỗi tầng đóng một vai trò cụ thể trong quá trình xử lý và học biểu diễn từ dữ liệu. Ba thành phần chính tạo nên kiến trúc của một mạng nơ-ron bao gồm tầng đầu vào (input layer), các tầng ẩn (hidden layers) và tầng đầu ra (output layer). Việc thiết kế và tổ chức các tầng này quyết định trực tiếp đến khả năng học và hiệu suất của mạng [21].



Hình 1.1: Cấu trúc mạng nơ-ron nhiều lớp (Multilayer Perceptron - MLP)

Tầng đầu vào là nơi tiếp nhận dữ liệu thô từ thế giới bên ngoài, chẳng hạn như ảnh, văn bản, hoặc dữ liệu dạng số. Mỗi đặc trưng (feature) của dữ liệu tương ứng với một nơ-ron ở tầng này. Tầng đầu vào không thực hiện bất kỳ phép biến đổi nào mà chỉ đóng vai trò trung gian truyền tải thông tin đến các tầng tiếp theo.

Các tầng ẩn là nơi diễn ra phần lớn quá trình học của mạng. Ở mỗi tầng ẩn, tín hiệu đầu vào được biến đổi tuyến tính thông qua trọng số và hệ số dịch (bias), sau đó được đưa qua một hàm kích hoạt phi tuyến như ReLU, sigmoid hoặc tanh. Chính các tầng ẩn là nơi mạng học được những biểu diễn trừu tượng và phức tạp hơn của dữ liệu. Ví dụ, trong một hệ thống nhận diện khuôn mặt, các tầng ẩn đầu tiên có thể học được các đường viền và cạnh, trong khi các tầng sâu hơn học được các cấu trúc phức tạp như mắt, mũi, hoặc toàn bộ khuôn mặt.

Số lượng tầng ẩn và số lượng nơ-ron trong mỗi tầng là các siêu tham số quan trọng ảnh hưởng đến năng lực học của mạng. Một mạng nơ-ron càng sâu (nhiều tầng ẩn) thì càng có khả năng mô hình hóa các quan hệ phi tuyến phức tạp, nhưng đồng thời cũng dễ rơi vào các vấn đề như quá khớp (overfitting), tiêu biến gradient hoặc

tăng thời gian huấn luyện. Do đó, cần cân nhắc kỹ lưỡng khi lựa chọn độ sâu và cấu trúc tầng ẩn cho từng bài toán cụ thể.

Tầng đầu ra là tầng cuối cùng trong mạng, có nhiệm vụ sinh ra kết quả dự đoán. Hình thức của tầng đầu ra phụ thuộc vào loại bài toán mà mạng giải quyết. Trong bài toán phân loại nhị phân, tầng đầu ra thường chỉ gồm một nơ-ron duy nhất với hàm kích hoạt sigmoid, đưa ra xác suất thuộc về một trong hai lớp. Đối với phân loại đa lớp, tầng đầu ra có kkk nơ-ron (tương ứng với kkk lớp), sử dụng hàm softmax để chuyển các giá trị đầu ra thành xác suất, đảm bảo tổng xác suất bằng 1. Còn trong các bài toán hồi quy, tầng đầu ra thường không áp dụng hàm kích hoạt, nhằm giữ nguyên giá trị thực số được tính toán.

Sự phối hợp giữa ba loại tầng này – từ việc tiếp nhận dữ liệu, trích xuất đặc trưng, cho đến sinh ra kết quả – tạo thành một quy trình xử lý khép kín, cho phép mạng học từ dữ liệu một cách toàn diện. Việc tối ưu hóa kiến trúc tầng không chỉ giúp cải thiện độ chính xác của mô hình mà còn nâng cao hiệu quả tính toán và khả năng tổng quát hóa cho dữ liệu mới.

1.2.2. Cơ chế truyền tín hiệu và lan truyền ngược

Trong quá trình học của mạng nơ-ron nhân tạo, dữ liệu được xử lý thông qua hai giai đoạn chính: lan truyền xuôi (forward propagation) và lan truyền ngược (backpropagation). Hai cơ chế này kết hợp với nhau tạo thành chu trình học liên tục, cho phép mạng tối ưu hóa trọng số để cải thiện độ chính xác của dự đoán.

Lan truyền xuôi (Forward Propagation)

Giai đoạn lan truyền xuôi bắt đầu từ tầng đầu vào, nơi dữ liệu thô được tiếp nhận và truyền qua các tầng ẩn đến tầng đầu ra. Tại mỗi tầng, tín hiệu đầu vào được biến đổi thông qua phép nhân với ma trận trọng số, cộng với hệ số dịch (bias), sau đó đưa qua hàm kích hoạt phi tuyến để tạo ra đầu ra cho tầng đó. Cụ thể, với một nơ-ron, đầu ra y được tính như sau:

$$y = \phi \left(\sum_{i=1}^n w_i x_i + p \right) = \phi(w^T x + b)$$

Trong đó $x = [x_1, x_2, \dots, x_n]$ là vector đầu vào, $w = [w_1, w_2, \dots, w_n]$ là vector trọng số, b là hệ số dịch (bias), và ϕ là hàm kích hoạt phi tuyến như sigmoid, tanh hoặc ReLU. Hàm kích hoạt đóng vai trò tạo ra tính phi tuyến trong mô hình, một yếu tố thiết yếu để mạng học được các quan hệ phức tạp mà các mô hình tuyến tính không thể nắm bắt.

Lan truyền ngược (Backpropagation)

Sau khi thu được đầu ra từ quá trình lan truyền xuôi, mạng sẽ so sánh đầu ra dự đoán với giá trị thực tế thông qua một hàm mất mát (loss function), ví dụ như hàm cross-entropy cho phân loại hay MSE cho hồi quy. Sai số này được dùng làm cơ sở để điều chỉnh các tham số của mạng. Quá trình lan truyền ngược sử dụng quy tắc chuỗi trong đạo hàm để tính toán gradient của hàm mất mát đối với từng trọng số trong mạng. Thông tin đạo hàm này sau đó được dùng trong thuật toán tối ưu như gradient descent để cập nhật trọng số:

$$w := w - \eta \frac{\partial L}{\partial w}$$

Trong đó, η là tốc độ học (learning rate), và $\frac{\partial L}{\partial w}$ là đạo hàm của hàm mất mát L theo trọng số w . Nhờ lan truyền ngược, mạng có thể học cách điều chỉnh các tham số sao cho sai số đầu ra được giảm dần sau mỗi vòng huấn luyện.

Vai trò của lan truyền xuôi và ngược

Sự kết hợp giữa lan truyền xuôi và lan truyền ngược giúp mạng không chỉ mô hình hóa dữ liệu đầu vào, mà còn tự điều chỉnh để cải thiện hiệu suất qua thời gian. Trong thực tế, quá trình huấn luyện một mạng nơ-ron hiện đại thường yêu cầu hàng nghìn hoặc hàng triệu vòng lan truyền xuôi và ngược lặp đi lặp lại, đi kèm với các kỹ thuật tối ưu nâng cao như Adam, RMSProp hay học sâu mini-batch để tăng tốc độ hội tụ và tránh mắc kẹt tại điểm tối ưu cục bộ.

Sự hiệu quả của hai cơ chế này là nền tảng cho sự phát triển mạnh mẽ của học sâu (deep learning), góp phần mở đường cho các ứng dụng thực tế như xử lý ảnh, ngôn ngữ tự nhiên, và thị giác máy tính.

1.2.3. Hàm kích hoạt (Activation Functions)

Trong mạng nơ-ron nhân tạo, hàm kích hoạt đóng vai trò then chốt trong việc đưa vào tính phi tuyến, cho phép mạng học được các quan hệ phức tạp giữa đầu vào và đầu ra. Nếu không có hàm kích hoạt, toàn bộ mạng dù có nhiều tầng đến đâu vẫn chỉ là một mô hình tuyến tính tương đương với hồi quy đa biến, không đủ khả năng biểu diễn các quan hệ phi tuyến trong dữ liệu thực tế.

Hàm kích hoạt được áp dụng tại mỗi nơ-ron sau khi thực hiện phép nhân giữa đầu vào và trọng số, cộng với hệ số dịch. Việc lựa chọn hàm kích hoạt phù hợp ảnh hưởng trực tiếp đến hiệu suất học và khả năng hội tụ của mạng. Một hàm kích hoạt tốt cần đảm bảo tính liên tục, có đạo hàm và có thể lan truyền gradient hiệu quả trong quá trình huấn luyện.

Một số hàm kích hoạt phổ biến: [22]

1. Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid biến đổi đầu vào thành giá trị nằm trong khoảng (0,1), phù hợp cho các bài toán phân loại nhị phân, đặc biệt tại tầng đầu ra. Tuy nhiên, trong các tầng ẩn, sigmoid thường gây ra hiện tượng tiêu biến gradient khi giá trị đầu vào quá lớn hoặc quá nhỏ, khiến tốc độ học chậm lại đáng kể.

2. Tanh

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Hàm Tanh có đầu ra nằm trong khoảng (-1,1), giúp dữ liệu được chuẩn hóa tốt hơn so với sigmoid. Nó khắc phục một phần nhược điểm tiêu biến gradient của sigmoid, nhưng vẫn không tránh khỏi hoàn toàn vấn đề này trong mạng sâu.

3. ReLU

$$f(x) = \max(0, x)$$

ReLU là hàm kích hoạt đơn giản nhưng rất hiệu quả, đặc biệt trong các tầng ẩn. Nó không chỉ giúp giảm thiểu tiêu biến gradient mà còn rút ngắn thời gian huấn luyện nhờ tính toán nhanh chóng. Tuy nhiên, ReLU có thể gặp vấn đề “nơ-ron chết” (dead neurons) khi giá trị đầu vào âm khiến nơ-ron ngừng học (gradient bằng 0).

4. Softmax

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

Softmax thường được dùng ở tầng đầu ra của mạng phân loại đa lớp, với vai trò chuyển vector đầu ra thành một phân phối xác suất trên các lớp. Hàm này đảm bảo tổng xác suất của tất cả các lớp bằng 1, giúp việc giải thích kết quả dễ dàng hơn.

➔ **Kết luận:** Hàm kích hoạt không chỉ ảnh hưởng đến độ chính xác mà còn đến khả năng hội tụ và ổn định trong huấn luyện mạng nơ-ron. Việc lựa chọn hàm kích hoạt cần dựa trên bài toán cụ thể và cấu trúc mạng. Trong thực tế, ReLU và các biến thể như Leaky ReLU, ELU hiện đang được ưa chuộng trong các mạng sâu nhờ hiệu năng vượt trội và tính ổn định cao.

1.2.4. Perceptron và Multilayer Perceptron (MLP)

Perceptron là một trong những mô hình mạng nơ-ron đầu tiên được đề xuất bởi Frank Rosenblatt vào năm 1958, được xem là nền móng cho sự phát triển của các mô hình học sâu ngày nay. Perceptron đơn (single-layer perceptron) mô phỏng một cách đơn giản cơ chế xử lý của nơ-ron sinh học, trong đó các tín hiệu đầu vào được nhân với trọng số, cộng với một hệ số dịch (bias), sau đó áp dụng một hàm quyết định (thường là hàm bước – step function) để đưa ra đầu ra nhị phân.

Về mặt toán học, đầu ra của một Perceptron đơn được tính theo công thức:

$$y = \begin{cases} 1 & \text{nếu } \sum_{i=1}^n w_i x_i + b > 0 \\ 0 & \text{ngược lại} \end{cases}$$

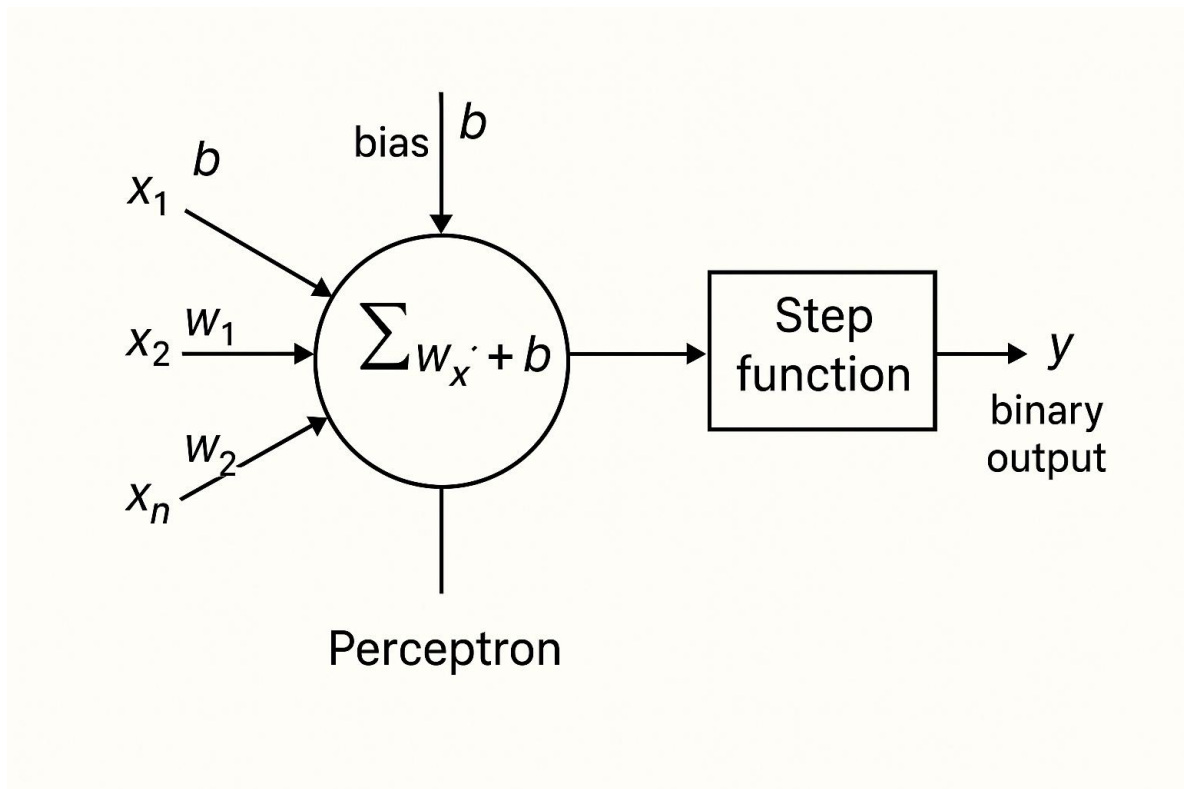
Trong đó:

$x = [x_1, x_2, \dots, x_n]$ là vector đầu vào

$w = [w_1, w_2, \dots, w_n]$ là vector trọng số,

b là hệ số dịch (bias),

y là đầu ra nhị phân.



Hình: 1.2: Mô hình Perceptron

Perceptron đơn có thể học cách phân biệt giữa hai lớp bằng cách điều chỉnh các trọng số w_i và bias b trong quá trình huấn luyện. Thuật toán học của Perceptron sử dụng một phương pháp cập nhật đơn giản dựa trên lỗi giữa đầu ra thực tế và đầu ra mong muốn. Tuy nhiên, Perceptron đơn chỉ có thể giải được các bài toán tuyến tính phân tách được (linearly separable), như bài toán AND hoặc OR, nhưng không thể giải được bài toán XOR, điều này từng dẫn đến sự hoài nghi về khả năng của mạng nơ-ron trong một thời gian dài.

Chính vì những hạn chế này, khái niệm Perceptron đã được mở rộng và phát triển thành nơ-ron nhân tạo tổng quát với các hàm kích hoạt phi tuyến, từ đó hình thành nên các kiến trúc mạng nơ-ron sâu hiện đại, mở ra một kỷ nguyên mới trong lĩnh vực học máy.

Để vượt qua giới hạn của perceptron đơn, mô hình Multilayer Perceptron (MLP) đã ra đời như một bước tiến quan trọng trong việc mở rộng khả năng học của mạng nơ-ron. MLP bao gồm nhiều tầng liên tiếp: tầng đầu vào, một hoặc nhiều tầng ẩn và tầng đầu ra. Các nơ-ron trong mỗi tầng đều được kết nối đầy đủ với các nơ-ron ở tầng liền kề, hình thành một kiến trúc mạng có khả năng học sâu hơn và linh hoạt hơn. Sự có mặt của các tầng ẩn đóng vai trò quyết định trong việc trích xuất và học các đặc trưng phi tuyến từ dữ liệu đầu vào, mở ra khả năng áp dụng mạng vào các bài toán phân loại và hồi quy phức tạp hơn nhiều so với những gì perceptron đơn có thể thực hiện.

Nhờ cấu trúc nhiều tầng và việc sử dụng các hàm kích hoạt phi tuyến, MLP có thể mô hình hóa các quan hệ phi tuyến giữa đầu vào và đầu ra, từ đó mở rộng đáng kể phạm vi ứng dụng của mạng nơ-ron nhân tạo. Mỗi tầng ẩn trong MLP không chỉ là một khối trung gian, mà còn là nơi mạng học được các biểu diễn trừu tượng hơn của dữ liệu gốc thông qua quá trình lan truyền xuôi và lan truyền ngược. Việc sử dụng các thuật toán tối ưu như gradient descent hay Adam giúp MLP điều chỉnh trọng số theo hướng giảm thiểu sai số, từ đó cải thiện dần hiệu suất dự đoán sau mỗi vòng huấn luyện.

Một cột mốc lý thuyết quan trọng củng cố sức mạnh của MLP là định lý xấp xỉ phổ quát (Universal Approximation Theorem). Theo định lý này, một mạng nơ-ron với một tầng ẩn duy nhất nhưng có đủ số lượng nơ-ron và sử dụng hàm kích hoạt phi tuyến liên tục có thể xấp xỉ bất kỳ hàm liên tục nào trên một tập hợp đầu vào đóng và bị chặn. Điều này đồng nghĩa với việc MLP có khả năng học được bất kỳ mối quan hệ nào giữa đầu vào và đầu ra, bất kể độ phức tạp của bài toán, miễn là mạng có cấu trúc đủ linh hoạt và được huấn luyện đúng cách. Đây là cơ sở lý thuyết vững chắc lý giải vì sao MLP có thể được áp dụng trong nhiều lĩnh vực như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và dự đoán chuỗi thời gian.

1.2.5. Mạng nơ-ron trong xử lý ngôn ngữ tự nhiên (NLP)

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực liên ngành giữa ngôn ngữ học, khoa học máy tính và trí tuệ nhân tạo, tập trung vào việc xây dựng các hệ thống có khả năng hiểu và sinh ngôn ngữ giống như con

người. Một trong những yếu tố quan trọng thúc đẩy sự tiến bộ của NLP chính là sự phát triển của các mô hình mạng nơ-ron nhân tạo. Từ những mô hình cơ bản như Multilayer Perceptron (MLP) cho đến những kiến trúc chuyên biệt cho dữ liệu chuỗi như Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), và sau này là Transformer, các mô hình mạng nơ-ron đã trở thành nền tảng cho nhiều ứng dụng thực tiễn trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Multilayer Perceptron (MLP) là dạng mạng nơ-ron nhân tạo nhiều tầng, trong đó các tầng được kết nối đầy đủ với nhau, chủ yếu phù hợp với dữ liệu có định dạng vector cố định. Trong NLP, MLP thường được áp dụng trong các tác vụ như phân loại văn bản, phát hiện cảm xúc hoặc phân loại chủ đề. Trước khi đưa vào MLP, văn bản được chuyển thành các biểu diễn vector thông qua các phương pháp như Bag-of-Words, TF-IDF hoặc các kỹ thuật word embedding như Word2Vec hay GloVe. Sau khi biểu diễn dưới dạng vector, dữ liệu được truyền qua các tầng ẩn của MLP để học ra các đặc trưng phân loại và đưa ra dự đoán. Tuy nhiên, do không có cơ chế xử lý chuỗi hay ghi nhớ ngữ cảnh giữa các từ, MLP thường không phù hợp với các bài toán yêu cầu hiểu ngữ cảnh dài hạn hoặc mô hình hóa thứ tự từ trong câu.

Recurrent Neural Network (RNN) được thiết kế để khắc phục điểm yếu đó của MLP bằng cách thêm khả năng ghi nhớ thông tin tuần tự trong chuỗi đầu vào. Trong RNN, tại mỗi thời điểm, đầu vào hiện tại và trạng thái ẩn từ bước trước được kết hợp để tính ra trạng thái ẩn hiện tại. Cấu trúc này cho phép RNN "nhớ" thông tin từ các bước trước và sử dụng thông tin đó cho bước hiện tại, giúp ích rất nhiều trong việc xử lý dữ liệu chuỗi như văn bản. Trong các tác vụ như phân tích cảm xúc, RNN có thể tiếp nhận từng từ trong câu theo thứ tự và xây dựng trạng thái ẩn phản ánh ngữ nghĩa toàn cục; trong gán nhãn từ loại (POS tagging), RNN học được các mẫu ngữ pháp dựa trên chuỗi từ để gán nhãn phù hợp; trong dịch máy, mô hình encoder-decoder dựa trên RNN mã hóa câu nguồn thành một vector ngữ cảnh duy nhất, sau đó giải mã để sinh ra câu đích tương ứng. Tuy nhiên, RNN truyền thống vẫn gặp khó khăn trong việc xử lý chuỗi dài do hiện tượng biến mất gradient, làm giảm khả năng học các quan hệ dài hạn.

Để khắc phục hạn chế này, LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit) được giới thiệu như các biến thể của RNN có khả năng ghi nhớ dài hạn tốt hơn. LSTM bổ sung các cơ chế cổng điều khiển dòng thông tin, bao gồm cổng vào, cổng quên và cổng đầu ra. Những cổng này cho phép mạng quyết định giữ lại hay loại bỏ thông tin tại từng thời điểm, giúp mô hình dễ dàng học được các mối quan hệ ngữ nghĩa dài hạn trong văn bản. Trong NLP, LSTM được ứng dụng hiệu quả trong nhiều tác vụ như sinh văn bản, tóm tắt văn bản và dịch máy. GRU là một phiên bản đơn giản hóa của LSTM, kết hợp một số cổng và giảm độ phức tạp tính toán trong khi vẫn giữ được hiệu quả xử lý chuỗi. GRU thường được ưa chuộng trong các ứng dụng yêu cầu tốc độ huấn luyện nhanh hoặc tài nguyên tính toán hạn chế.

Một biến thể khác là mạng nơ-ron hồi tiếp hai chiều (Bi-directional RNN – BiRNN), trong đó dữ liệu chuỗi được xử lý theo cả hai chiều: xuôi (từ đầu đến cuối) và ngược (từ cuối về đầu). Nhờ đó, mô hình có thể tận dụng ngữ cảnh từ cả hai phía để đưa ra quyết định tốt hơn tại mỗi thời điểm. BiRNN đặc biệt hữu ích trong các tác vụ cần hiểu đầy đủ bối cảnh như nhận dạng thực thể (Named Entity Recognition – NER), gán nhãn ngữ nghĩa và phân tích cú pháp. Bi-LSTM và Bi-GRU là các kiến trúc phổ biến trong nhiều hệ thống NLP hiện đại, từ chatbot, hệ thống hỏi đáp đến trích xuất thông tin.

Tùy vào từng tác vụ cụ thể trong NLP, các kiến trúc mạng nơ-ron được áp dụng linh hoạt để đạt hiệu quả tối ưu. Chẳng hạn, trong phân loại văn bản, các mô hình đơn giản như MLP có thể hoạt động tốt khi kết hợp với các biểu diễn từ thích hợp, trong khi LSTM hoặc BiLSTM lại phát huy tác dụng với văn bản dài và ngữ cảnh phức tạp. Trong gán nhãn thực thể (NER), kiến trúc BiLSTM kết hợp với CRF (Conditional Random Field) thường được sử dụng để tận dụng ngữ cảnh hai chiều và tối ưu hóa cấu trúc đầu ra. Với tác vụ gán nhãn từ loại (POS Tagging), RNN hoặc BiRNN giúp học các mối quan hệ ngữ pháp dựa trên chuỗi từ. Trong sinh văn bản, LSTM và GRU là lựa chọn phổ biến để học mô hình ngôn ngữ và sinh ra câu mới một cách tự nhiên. Với dịch máy, kiến trúc encoder-decoder dựa trên RNN/LSTM kết hợp attention được sử dụng để cải thiện đáng kể chất lượng dịch. Trong tóm tắt

văn bản và trả lời câu hỏi, các mô hình tuần tự như LSTM cũng giữ vai trò trung tâm trước khi bị thay thế bởi các mô hình hiện đại hơn như Transformer.

1.2.6. Vai trò của mạng nơ-ron trong NLP hiện đại

Trong bối cảnh công nghệ phát triển nhanh chóng, mạng nơ-ron nhân tạo đã và đang đóng vai trò trung tâm trong sự chuyển mình của lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Nhóm nghiên cứu nhận thấy rằng, từ chỗ chỉ là một công cụ hỗ trợ, các kiến trúc mạng nơ-ron ngày nay đã trở thành nền tảng cốt lõi, thúc đẩy hàng loạt tiến bộ vượt bậc trong việc xây dựng các hệ thống hiểu và sinh ngôn ngữ một cách tự động. Sự kết hợp giữa sức mạnh tính toán ngày càng tăng và khả năng học biểu diễn ngữ nghĩa từ dữ liệu lớn đã biến mạng nơ-ron thành công cụ không thể thiếu trong hầu hết các ứng dụng NLP hiện đại.

Một trong những xu hướng nổi bật là việc chuyển từ các mô hình mạng nơ-ron truyền thống (như RNN, LSTM, BiLSTM) sang các mô hình ngôn ngữ lớn (Large Language Models – LLMs) với kiến trúc Transformer làm trung tâm. Các mô hình này cho phép học biểu diễn ngữ nghĩa sâu và linh hoạt, dựa trên cơ chế attention giúp nắm bắt mối quan hệ giữa các từ trong toàn bộ câu một cách đồng thời. Nhờ đó, chúng vượt qua được các giới hạn về phụ thuộc ngữ cảnh gần hay hiện tượng biến mất gradient thường gặp ở RNN. Các mô hình ngôn ngữ hiện đại không chỉ học biểu diễn ngôn ngữ từ dữ liệu lớn mà còn có khả năng generalize tốt sang nhiều tác vụ khác nhau như phân loại, gán nhãn, sinh văn bản, dịch máy và trả lời câu hỏi.

Tại thời điểm hiện tại, nhóm thực hiện đề tài ghi nhận vai trò then chốt của các mô hình tiền huấn luyện (pre-trained language models) như BERT, PhoBERT, hay các phương pháp tối ưu hóa biểu diễn ngữ nghĩa như SimCSE trong việc nâng cao độ chính xác và khả năng thích ứng với ngôn ngữ tự nhiên ở các nhiệm vụ khác nhau. Đặc biệt, với ngôn ngữ tiếng Việt – vốn có cấu trúc cú pháp và ngữ nghĩa phức tạp – các mô hình được huấn luyện chuyên biệt như PhoBERT đã chứng minh được hiệu quả vượt trội trong nhiều bài toán, từ phân tích cảm xúc đến nhận dạng thực thể. Đồng thời, nhóm cũng ghi nhận tiềm năng của các mô hình đa phương thức như Visobert trong việc tích hợp xử lý ngôn ngữ và hình ảnh – mở ra hướng tiếp cận mới cho các ứng dụng liên quan đến ngữ cảnh đa chiều.

Sự phát triển này cho thấy mạng nơ-ron không chỉ đóng vai trò mô hình hóa ngôn ngữ, mà còn là nền tảng để xây dựng các hệ thống thông minh có khả năng học hỏi liên tục, thích ứng nhanh chóng và giải quyết hiệu quả các thách thức của NLP trong thực tiễn. Điều này càng khẳng định tầm quan trọng của việc nghiên cứu sâu hơn về các mô hình ngôn ngữ lớn, không chỉ dừng lại ở lý thuyết mà còn trong khả năng ứng dụng vào các tình huống cụ thể của tiếng Việt cũng như các ngôn ngữ khác.

Để làm rõ hơn vai trò và cách vận hành của các mô hình ngôn ngữ lớn trong NLP hiện đại, chương tiếp theo sẽ tập trung phân tích chi tiết các mô hình tiêu biểu như BERT, PhoBERT, SimCSE và Visobert. Thông qua việc trình bày kiến trúc, cơ chế hoạt động và ứng dụng thực tiễn của từng mô hình, nhóm nghiên cứu kỳ vọng có thể cung cấp cái nhìn tổng quan và có chiều sâu về cách mà mạng nơ-ron đang dẫn dắt làn sóng đổi mới trong lĩnh vực xử lý ngôn ngữ tự nhiên hiện nay.

1.3. Mô hình ngôn ngữ lớn

1.3.1. Tổng quan

Mô hình ngôn ngữ lớn (Large Language Model - LLM) là một loại trí tuệ nhân tạo sử dụng mạng nơ-ron sâu để xử lý và tạo ra ngôn ngữ tự nhiên. Các mô hình này được huấn luyện trên khối lượng dữ liệu khổng lồ nhằm hiểu, tạo và dự đoán văn bản một cách chính xác.

Kiến trúc và nguyên lý hoạt động: Hầu hết các mô hình LLM hiện nay dựa trên kiến trúc Transformer được giới thiệu vào năm 2017 [23]. LLM học hỏi từ một khối dữ liệu khổng lồ để có thể ghi nhớ các quy luật và cấu trúc ngôn ngữ. Trong quá trình huấn luyện, mô hình học các mối quan hệ thống kê giữa các từ, cụm từ và câu, cho phép nó tạo ra các đoạn văn mạch lạc và có ngữ cảnh liên quan khi được cung cấp một đoạn văn mẫu (prompt).

Một số thành phần chính của LLM:

1. Tokenization: Văn bản được chuyển thành các token nhỏ (Từ, cụm từ, ký tự).
2. Embedding Layer: Chuyển token thành vector số học có ý nghĩa.

3. Multi-Head Self-Attention: Cho phép mô hình tập chung vào các phần quan trọng của văn bản.
4. Feedforward Neural Network: Xử lý và truyền tải thông tin giữa các lớp.
5. Positional Encoding: Cung cấp thông tin về vị trí của từ trong câu.

Quá trình huấn luyện: Mô hình được huấn luyện dựa trên một lượng lớn văn bản từ nhiều nguồn như sách, bài báo, trang web. **Có hai phương pháp chính:**

1. Huấn luyện không giám sát(Unsupervised Learning): Sử dụng kỹ thuật như “Masked Language Model¹” (MLM) và “Causal Language Model²” (CLM).
2. Fine-tuning: Tinh chỉnh trên các tập dữ liệu chuyên biệt để phù hợp với các ứng dụng cụ thể.

Ứng dụng của mô hình LLM: Các mô hình ngôn ngữ lớn có nhiều ứng dụng thực tế như:

1. Chatbot và trợ lý ảo: ChatGPT, Google Gemini, Claude,...
2. Dịch thuật tự động: Google Translate, DeepL,...
3. Phân tích ngữ nghĩa và tìm kiếm thông minh.
4. Viết nội dung tự động: Tạo bài viết, tóm tắt tài liệu, sáng tác thơ, viết content, kịch bản,...
5. Lập trình và hỗ trợ mã hóa: GitHub Copilot, Code Interpreter,...

1.3.2. BERT

1.3.1.1. Giới thiệu

BERT là cái tên viết tắt của “Bidirectional Encoder Representations from Transformers”, là một mô hình ngôn ngữ tự nhiên (NLP) dựa trên mạng Transformer được phát triển bởi Google và được giới thiệu vào năm 2018. Mô hình này có khả năng hiểu được ngôn ngữ tự nhiên của con người, được sử dụng rộng rãi trong các ứng dụng NLP như phân loại văn bản, dịch máy, tóm tắt văn bản, trả lời câu hỏi, ... Một điểm đặc biệt của BERT là nó được huấn luyện trên dữ liệu lớn và đa dạng từ nhiều nguồn khác nhau, bao gồm các tài liệu trên mạng Internet và các tài liệu chuyên

¹ https://huggingface.co/docs/transformers/en/tasks/masked_language_modeling

² https://huggingface.co/docs/transformers/tasks/language_modeling

ngành. BERT được huấn luyện trên hai tác vụ chính là “Masked Language Modeling” (MLM) và “Next Sentence Prediction” (NSP). Kết hợp hai tác vụ trên, BERT có khả năng hiểu được ngữ nghĩa của các từ và câu trong ngôn ngữ tự nhiên.

BERT được ra đời nhằm giải quyết một số hạn chế của các mô hình ngôn ngữ trước đó. Trước khi BERT được giới thiệu, các mô hình ngôn ngữ thông thường như Word2Vec, GloVe chỉ có thể biểu diễn các từ một cách đơn giản, không thể hiểu được ngữ cảnh và sự phụ thuộc các từ trong một câu hoặc một đoạn văn. Để giải quyết được vấn đề này, BERT sử dụng kiến trúc Transformer để học các biểu diễn từ và câu phức tạp hơn. BERT cũng sử dụng phương pháp Pre-training (huấn luyện trước) để đào tạo mô hình với lượng dữ liệu lớn trước khi Fine-tuning (tinh chỉnh) cho các tác vụ cụ thể.

BERT đã đạt được nhiều thành công và trở thành một trong những mô hình ngôn ngữ phổ biến nhất hiện nay, được sử dụng rộng rãi trong nhiều lĩnh vực xử lý ngôn ngữ tự nhiên như dịch thuật, phân loại văn bản, tìm kiếm thông tin ... Ở thời điểm hiện tại, BERT đã được ứng dụng cho tiếng Việt. Đó chính là dự án PhoBERT được giới thiệu vào năm 2020 bởi VinAI.

1.3.1.2. Kiến trúc mô hình

BERT có hai phiên bản, BERT_{base} và BERT_{large}, bảng sau đây sẽ cung cấp số liệu so sánh giữa hai phiên bản này:

Model	Encoder Layer (L)	Attention Heads (A)	Hidden Size (H)
BERT_base	12	12	768
BERT_large	24	16	1024

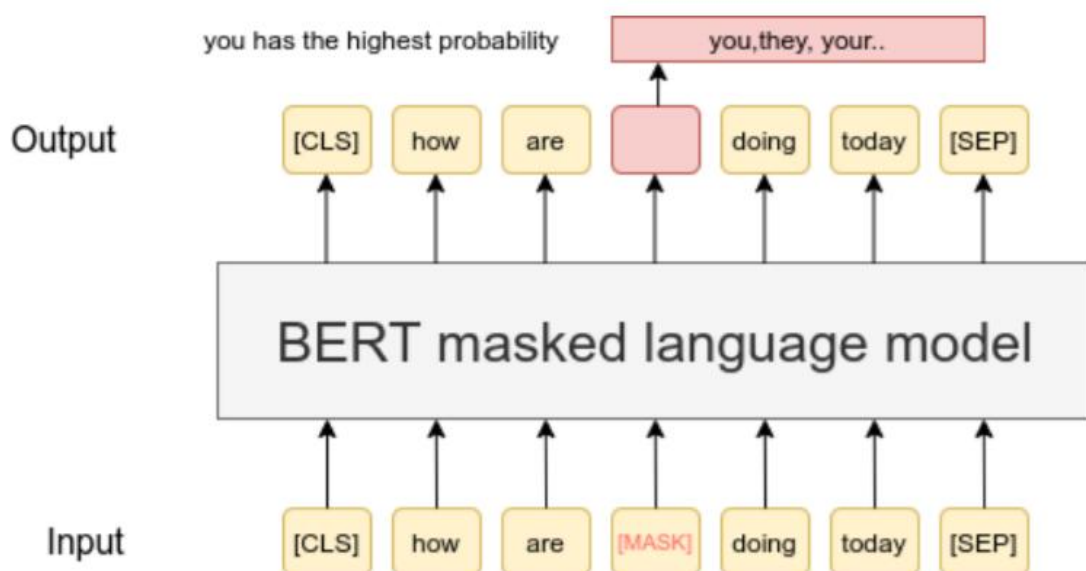
Bảng 1.2: Pre-trained models

BERT hoạt động bằng cách sử dụng một kiến trúc mã hoá hai chiều với nhiều lớp mã hoá Transformer được xếp chồng lên nhau. Mỗi lớp mã hoá Transformer có hai thành phần chính đó là *self – attention* và *feed – forward network*.

– Self – attention: Lớp này cho phép mô hình tập trung vào các phần quan trọng của đầu vào (như các từ hay cụm từ có liên quan đến nhau) để tính toán các biểu diễn vector đầu ra.

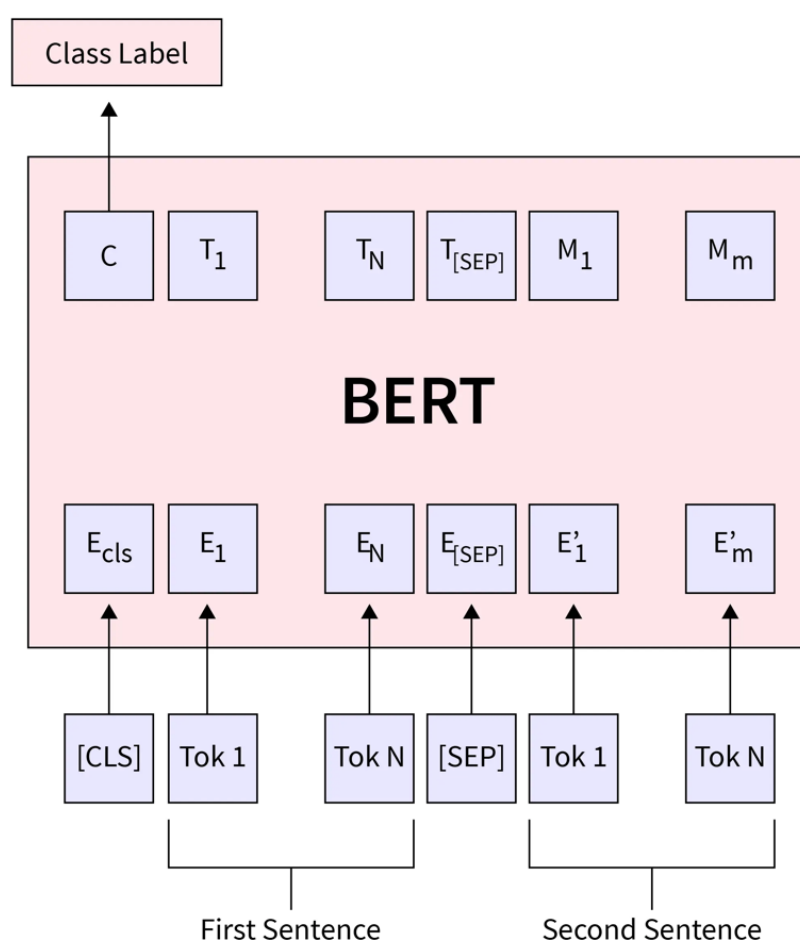
–Feed – forward network: Là một lớp mạng nơ-ron truyền thẳng (fully connected layer) trong Transformer, được sử dụng để biến đổi các biểu diễn vector đầu vào của self – attention thành các biểu diễn vector đầu ra. Mỗi lớp feed – forward network bao gồm hai lớp linear (tuyến tính) được tách bằng một hàm ReLU (Rectified Linear Unit).

Như đã nói ở phần giới thiệu, BERT sử dụng hai tác vụ chính để đào tạo mô hình đó chính là Masked Language Modeling (MLM) và Next Sentence Prediction (NSP). Trong tác vụ MLM, các từ trong câu được chọn ngẫu nhiên và thay thế bằng một ký tự đặc biệt. Mô hình phải dự đoán được từ gốc ban đầu của các từ bị thay thế. Ví dụ, ta có đầu vào (input) là: “*Hôm nay trời [MASK] quá*” thì mô hình BERT sẽ phải dự đoán từ gốc của từ được thay thế bằng [MASK]. Trong trường hợp này, từ miêu tả thời tiết có thể là ‘*nắng*’, ‘*mưa*’, ‘*đẹp*’, ‘*âm u*’, ‘*mát*’ ... tùy vào ngày hôm đó, vì vậy đầu ra dự kiến của mô hình sẽ phụ thuộc vào ngữ cảnh của câu. Tác vụ MLM giúp mô hình BERT học được cách biểu diễn các từ và từ ghép trong câu một cách toàn diện, bao gồm cả các từ đứng trước và sau từ được thay thế, giúp mô hình hiểu được ngữ nghĩa của các từ và cách chúng liên kết với nhau trong ngôn ngữ tự nhiên.



- [CLS] là ký tự đặc biệt được thêm vào đầu câu văn để giúp mô hình BERT hiểu rằng đây là một câu văn cần được phân loại.
- [SEP] là ký tự đặc biệt sử dụng để phân tách giữa hai câu trong.

Trong tác vụ NSP, mô hình được đưa cho hai câu và phải dự đoán xem liệu câu thứ hai có phải là câu tiếp theo của câu thứ nhất hay không. Ví dụ, ta có đầu vào là: “*Bàn thắng đầu tiên đã đến với đội chủ nhà. [SEP] Các cầu thủ đang ôm nhau ăn mừng*”. Trong đó, [SEP] là một ký tự đặc biệt để phân tách hai câu văn. Mô hình BERT phải dự đoán xem câu thứ hai có phải là câu văn tiếp theo của câu thứ nhất không. Với trường hợp này thì cả hai câu đều đang nói đến môn thể thao bóng đá và với ngữ cảnh như này thì câu thứ hai có thể được xem là câu tiếp theo của câu thứ nhất. Tác vụ NSP giúp BERT hiểu được cấu trúc của một đoạn văn và cách các câu liên kết với nhau trong ngôn ngữ tự nhiên. NSP giúp BERT có khả năng hiểu được ý nghĩa của câu một cách toàn diện hơn.



Hình 1.4: Mô phỏng hoạt động của NSP

Việc kết hợp hai tác vụ trên (MLM và NSP) giúp cho mô hình BERT hiểu được ngữ nghĩa của các từ và câu phức tạp hơn trong ngôn ngữ tự nhiên.

1.3.1.3. Ứng dụng thực tế của BERT

Với khả năng mà BERT đã mang lại, nó đã được áp dụng rộng rãi trong các sản phẩm và ứng dụng khác nhau.

Dưới đây là một số sản phẩm, ứng dụng đã được tích hợp BERT:

1. Google Search: Công cụ tìm kiếm trực tuyến hàng đầu do Google cung cấp được tích hợp BERT để cải thiện khả năng tìm kiếm và hiển thị kết quả tìm kiếm chính xác hơn.
2. Google Assistant: Một trợ lý ảo của Google cũng được tích hợp BERT để cải thiện khả năng hiểu các câu hỏi và trả lời câu hỏi của người dùng một cách chính xác và tự nhiên nhất có thể.
3. Microsoft Office: BERT được tích hợp vào trong Microsoft Office để cải thiện khả năng tóm tắt văn bản, trích xuất thông tin và kiểm tra chính tả.
4. Grammarly: Grammarly tích hợp BERT để cải thiện khả năng phân tích và sửa lỗi ngữ pháp và cách dùng từ trong văn bản để phù hợp với từng ngữ cảnh khác nhau.

Vẫn còn rất nhiều sản phẩm, ứng dụng thực tế khác đã tích hợp BERT bên trong, điều này cho thấy BERT là một thứ rất quan trọng đối với ngành AI hiện nay với nhiều ứng dụng mà nó mang lại.

1.3.3. PhoBERT

Với cái tên được lấy cảm hứng từ một món ăn phổ biến của người Việt Nam là ‘Phở’. PhoBERT là mô hình tiên tiến nhất được huấn luyện dành riêng cho tiếng Việt, nghiên cứu bởi VinAI được public năm 2020. PhoBERT có cách tiếp cận dựa trên RoBERTa của Facebook, một biến thể của BERT mang lại hiệu suất mạnh mẽ hơn.

PhoBERT được huấn luyện trên khoảng 20GB dữ liệu được làm sạch với khoảng 1GB lấy từ Vietnamese Wikipedia corpus và khoảng 19GB từ Vietnamese news corpus. Vietnamese news corpus là một tập hợp các bài báo và tin tức thu nhập từ các trang báo điện tử lớn tại Việt Nam, bao gồm các chuyên mục về chính trị, kinh tế, xã hội, giáo dục và văn hoá. Các tài liệu tiếng Việt sau khi thu thập từ Vietnamese news corpus được tiền xử lý bằng cách tách từ, đưa về dạng chuẩn và loại bỏ các ký tự đặc biệt sau đó dữ liệu sẽ được đưa vào để huấn luyện trên mô hình PhoBERT.

Tất cả các mô hình ngôn ngữ dựa trên BERT được công bố hỗ trợ đơn ngữ hoặc song ngữ đều không nhận ra sự khác biệt giữa các âm tiết và các đơn vị từ trong tiếng Việt. Ví dụ, một câu có năm âm tiết “*Tôi là một bác sĩ*” tạo thành 4 từ “*Tôi là một bác sĩ*” (I am a doctor). Để giải quyết vấn đề này các nhà nghiên cứu sử dụng RDRSegmenter và VnCoreNLP để thực hiện phân đoạn từ và câu trên tập dữ liệu huấn luyện trước, kết quả là ~145 triệu câu được tách từ (~ 3 tỷ từ - words token). Nếu không thực hiện bước tiền xử lý tách từ tiếng Việt, các mô hình được huấn luyện trên dữ liệu ở mức âm tiết có thể không hoạt động tốt bằng các mô hình ngôn ngữ được huấn luyện trên dữ liệu ở mức từ. Đó là lý do PhoBERT ra đời.

PhoBERT có hai phiên bản, PhoBERT_{base} và PhoBERT_{large} sử dụng kiến trúc tương tự như BERT_{base} và BERT_{large} chúng tôi đã giới thiệu và có cách tiếp cận dựa trên RoBERTa để có hiệu suất tốt hơn. RoBERTa là một biến thể của mô hình BERT, được huấn luyện trên một lượng lớn dữ liệu và các phương pháp tăng cường dữ liệu để cải thiện hiệu suất. RoBERTa loại bỏ phương pháp Masked Language Model (MLM) của BERT và thay thế bằng phương pháp đặt câu hỏi ngẫu nhiên (Next Sentence Prediction – NSP). Điều này giúp RoBERTa học được các biểu diễn từ phong phú hơn và cải thiện khả năng hiểu ngôn ngữ tự nhiên.

Model	#params	Arch.	Max length	Pre-training data
Vinai/phobert-base	135M	base	256	20GB of texts
Vinai/phobert-large	370M	large	256	20GB of texts

Bảng 1.3: Thông số của PhoBERT-base và PhoBERT-large

1.3.4. ViSoBERT

ViSoBERT là một mô hình ngôn ngữ mạnh mẽ, được phát triển bởi Trường Đại học Công nghệ Thông tin, Đại học Quốc gia TP.HCM (UIT). Họ đã nhận thấy nhu cầu cấp thiết của việc xử lý ngôn ngữ tự nhiên trên nền tảng mạng xã hội tiếng Việt, nơi dữ liệu thường không chuẩn ngữ pháp và chứa nhiều từ lóng. Tên gọi của ViSoBERT xuất phát từ "Vi" (Việt Nam) và "So" (Social - mạng xã hội), thể hiện rõ

mục tiêu tối ưu hóa xử lý ngôn ngữ tự nhiên trên các nền tảng mạng xã hội. Mô hình này dựa trên kiến trúc XLM-RoBERTa, với hai phiên bản ViSoBERTbase và ViSoBERTlarge, phù hợp cho các nhiệm vụ từ cơ bản đến nâng cao. Nó sử dụng các lớp Transformer để học biểu diễn ngữ nghĩa của văn bản.

ViSoBERT được huấn luyện trên hơn 120GB dữ liệu tiếng Việt đa dạng và chất lượng cao, bao gồm hơn 20 triệu câu từ các nền tảng như Facebook, YouTube và Twitter. Dữ liệu này đã được tiền xử lý kỹ lưỡng để loại bỏ lỗi chính tả, ký tự không cần thiết và chuẩn hóa văn bản nhằm đảm bảo độ chính xác cao. ViSoBERT tích hợp khả năng phân đoạn từ thông qua các công cụ như RDRSegmenter để xử lý ngôn ngữ tiếng Việt hiệu quả hơn.

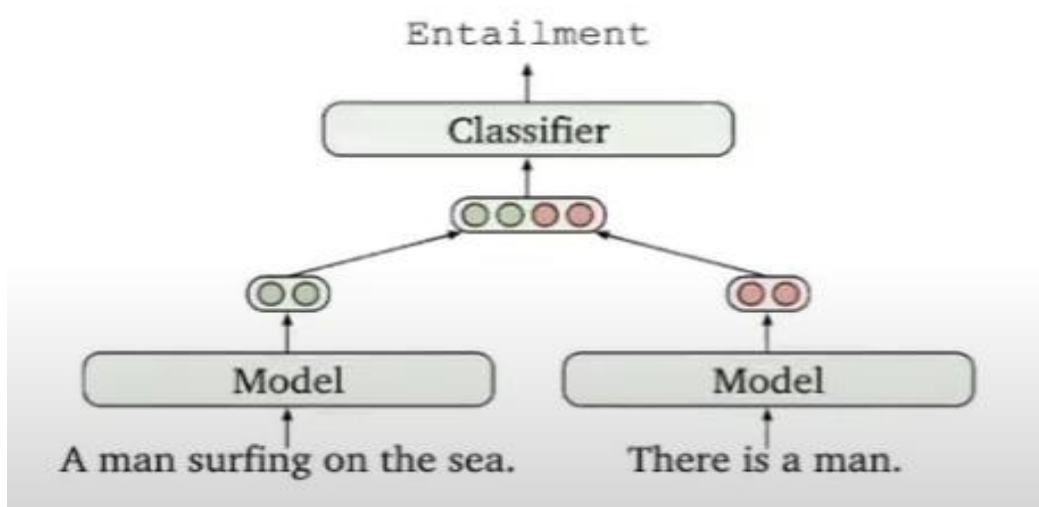
So với các mô hình trước đây như PhoBERT, ViSoBERT vượt trội trong các nhiệm vụ xử lý ngôn ngữ tự nhiên liên quan đến mạng xã hội, bao gồm:

1. **Nhận diện cảm xúc:** Phân tích cảm xúc trong các bài đăng hoặc bình luận.
2. **Phát hiện ngôn từ thù ghét:** Xác định các nội dung có tính chất xúc phạm hoặc thù địch.
3. **Phân tích cảm xúc:** Đánh giá mức độ tích cực, tiêu cực hoặc trung lập của văn bản.
4. **Phát hiện đánh giá spam:** Nhận diện các đánh giá không trung thực hoặc không liên quan.
5. **Phát hiện các đoạn văn chứa ngôn từ thù ghét:** Xác định vị trí cụ thể của các từ ngữ thù ghét trong văn bản.

Với độ chính xác trung bình vượt ngưỡng 90%, ViSoBERT đã chứng minh khả năng phân tích văn bản mạng xã hội một cách toàn diện và mạnh mẽ. Ứng dụng của ViSoBERT không chỉ giới hạn ở lĩnh vực công nghệ, mà còn góp phần thúc đẩy môi trường mạng xã hội an toàn và văn minh hơn, mở ra nhiều cơ hội nghiên cứu hành vi người dùng và phát triển các hệ thống kiểm duyệt tự động

1.3.5. SimCSE

SimCSE (Simple Contrastive Learning of Sentence Embeddings) là một phương pháp học biểu diễn câu hiệu quả, tối ưu hóa qua việc kết hợp giữa học không giám sát (unsupervised) và học có giám sát (supervised) [24]. Trước đây, các phương pháp supervised thường dựa trên việc sử dụng dữ liệu NLI (Natural Language Inference) để tạo thành các cặp câu (pairs), phân loại và dự đoán mối quan hệ giữa các câu.



Hình 1.5: Phân loại mối quan hệ ngữ nghĩa giữa hai câu

Tuy nhiên, để tối đa hóa dữ liệu đầu vào, các kỹ thuật tăng cường dữ liệu (data augmentation) như xóa ngẫu nhiên từ trong câu đã được áp dụng. Dù hiệu quả ở mức nhất định, cách này có thể dẫn đến việc tạo ra các câu vô nghĩa, ảnh hưởng tới chất lượng nhúng câu (embedding).

Ý tưởng cơ bản của SimCSE là sử dụng phương pháp học tương phản (contrastive learning) để cải thiện đáng kể chất lượng embedding. Contrastive learning hoạt động dựa trên nguyên tắc kéo gần các biểu diễn ngữ nghĩa giống nhau (neighbors) và đẩy xa các biểu diễn không tương tự (dissimilar). Điều này giúp điều chỉnh không gian nhúng, làm cho các cặp câu có liên kết về ngữ nghĩa được sắp xếp tốt hơn, đồng thời giữ các cặp không liên quan cách xa nhau.

Phương pháp này sử dụng hàm mất mát InfoNCE để tối ưu hóa:

- **Positive pairs:** Là embedding của cùng một câu nhưng được biểu diễn qua các dropout mask khác nhau.
- **Negative pairs:** Là embedding của tất cả các câu khác trong cùng một batch, được gọi là in-batch negatives.

$$l_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}$$

Hình 1.6: Công thức mất mát trong phép nhúng câu

Với Supervised SimCSE, điểm nổi bật nằm ở chỗ không chỉ so sánh embedding của các câu trong cùng một batch mà còn tận dụng các câu khác trong tập dữ liệu làm negative pairs. Kỹ thuật này kết hợp dữ liệu NLI giúp tăng cường tín hiệu giám sát, tạo ra không gian nhúng đồng nhất và có tổ chức hơn. Giả sử chúng ta có một batch gồm 5 câu. Một câu trong batch sẽ được đối chiếu với tất cả các embedding còn lại trong batch để phân biệt positive pairs và negative pairs. Điều này đảm bảo rằng mô hình học cách phát hiện các tương quan ngữ nghĩa chính xác hơn, đồng thời loại bỏ nhiễu từ các câu không liên quan.

CHƯƠNG 2: ĐO LƯỜNG ĐỘ TƯƠNG TỰ CỦA VĂN BẢN DỰA TRÊN MÔ HÌNH NGÔN NGỮ LỚN

Độ tương đồng giữa các văn bản ám chỉ tới độ tương đồng giữa nội dung của chúng. Giá trị tương đồng nằm trong khoảng 0 đến 1. Với giá trị càng gần 1 chỉ ra rằng hai văn bản có mức độ tương đồng cao và ngược lại.

2.1. Một số phương pháp tính độ tương đồng văn bản

Có nhiều cách để tính toán độ tương đồng giữa hai câu. Một số phương pháp phổ biến là:

- ✓ Cosine Similarity: Phương pháp này biểu diễn các câu dưới dạng vector và tính toán độ tương đồng giữa chúng bằng cách tính cosine của góc giữa hai vector đó.
- ✓ Jaccard Similarity: Phương pháp này tính toán độ tương đồng giữa hai câu dựa trên số lượng từ chung và tổng số từ của hai câu.
- ✓ Levenshtein Distance: Phương pháp này tính toán khoảng cách giữa hai câu bằng cách đếm số lượng các hoán vị cần thiết để chuyển đổi một câu thành câu kia.
- ✓ Manhattan Similarity: Phương pháp tính khoảng cách giữa các điểm trong không gian Euclid với hệ tọa độ Decarts. [25]

2.1.1. COSINE SIMILARITY

Văn bản được biểu diễn như mô hình cái túi của các từ (bag-of-words). Văn bản được coi như là một tập hợp của các từ mà không cần xem xét đến ngữ pháp, thứ tự các từ. Mỗi văn bản được chia thành các từ hoặc cụm từ (n-grams) sau đó được đặt vào trong túi. Kế tiếp tính tổng số lần xuất hiện của chúng rồi tạo thành một vector n chiều. Sau khi chuyển đổi hai văn bản thành 2 vector \vec{a} và \vec{b} ta sử dụng công thức sau:

$$Sim_c(\vec{a}, \vec{b}) = \frac{\vec{a} * \vec{b}}{\|\vec{a}\| \|\vec{b}\|_{(1)}}$$

- Đầu vào: Hai chuỗi A và B
- Thực hiện:
 - + Tiền xử lý: Tách các từ, tạo danh sách từ điển, ...
 - + Xây dựng bộ từ điển chung: $T = \{t_1, t_2, \dots\}$
 - + Mô hình hóa văn bản thành vector: Sử dụng T để tạo ra các vector tần số A và B là \vec{a} và \vec{b} (Dùng $TF * IDF$), tương ứng.
 - + Tính độ tương tự Cosine của hai vector tần số dùng công thức (1).
- Đầu ra: Độ tương đồng của A và B.

2.1.2. JACCARD SIMILARITY

Jaccard Similarity là một phương pháp phổ biến để tính toán độ tương đồng giữa hai tập hợp và cũng có thể được sử dụng để tính toán độ tương đồng giữa hai câu dưới dạng tập hợp các từ. Phương pháp này được đặt tên theo Paul Jaccard, một nhà toán học người Thụy Sĩ.

Độ tương tự Jaccard còn được gọi là hệ số tương tự Jaccard và Giao lộ trên Union. Jaccard ma trận được sử dụng để xác định sự giống nhau giữa hai tài liệu văn bản có nghĩa là hai tài liệu văn bản gần nhau như thế nào về ngữ cảnh của chúng, tức là có bao nhiêu từ phổ biến tồn tại trên tổng số từ.

Trong xử lý ngôn ngữ tự nhiên, chúng ta thường cần ước tính độ tương tự văn bản giữa các tài liệu văn bản. Có nhiều ma trận tương tự văn bản tồn tại như độ tương tự Cosine , Độ tương tự Jaccard và phép đo Khoảng cách Euclide . Tất cả các chỉ số tương tự văn bản này có hành vi khác nhau.

Jaccard được định nghĩa là một giao điểm của hai tài liệu được chia cho sự kết hợp của hai tài liệu đó đề cập đến số lượng từ chung trên tổng số từ.

Biểu diễn toán học độ đo tương đồng Jaccard là:

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2} \quad (2)$$

Điểm tương đồng của Jaccard nằm trong khoảng từ 0 đến 1. Nếu hai tài liệu giống hệt nhau điểm tương đồng của Jaccard là 1. Điểm tương đồng Jaccard là 0 nếu không có từ chung giữa hai tài liệu.

Để áp dụng Jaccard Similarity vào bài toán Sentence Similarity, chúng ta có thể xem mỗi câu như một tập hợp các từ và tính toán Jaccard Similarity giữa hai tập hợp đó. Tuy nhiên, phương pháp này không xử lý được các vấn đề như từ đồng nghĩa, từ viết sai hoặc thứ tự từ trong câu. Do đó, nó thường được sử dụng kết hợp với các phương pháp khác để đánh giá độ tương đồng giữa hai câu.

2.1.3. LEVENSHTein DISTANCE

Khoảng cách Levenshtein thể hiện sự khác biệt giữa hai chuỗi A và B là số bước nhỏ nhất để biến đổi chuỗi này thành chuỗi kia bằng cách sử dụng ba phép biến đổi: Phép xóa, phép chèn, phép sửa. Để tính toán đại lượng này ta sử dụng giải thuật quy hoạch động. Phép toán được thực hiện trên mảng 2 chiều có kích thước $(m + 1) * (n + 1)$, trong đó m và n là chiều dài của 2 chuỗi.

Giải thuật Levenshtein ($A[1,2,...,n], B[1,2,...,m]$):

Khởi tạo: $d[0...m, 0...n]$

For $i:=0 \rightarrow m$

$d[i,0]:=i$

For $j:=0 \rightarrow n$

$d[0,j]:=j$

For $i:=1 \rightarrow m$

For $j:=1 \rightarrow n$

If $A[i]=B[j]$ then $cost:=0$ else $cost:=1$

$d[i,j]:=min(d[i - 1,j] + 1, d[i, j - 1] + 1, d[i - 1, j - 1] + cost)$

Return $d[m,n]$

Giá trị của $d[m,n]$ chính là khoảng cách Levenshtein, s là chiều dài của chuỗi dài nhất. Độ tương tự được tính bởi công thức:

$$Sim_L(A, B) = 1 - \frac{d[m, n]}{s} \quad (3)$$

Giải thuật 1: Độ đo Levenshtein

- Đầu vào: Hai chuỗi A và B.
- Thực hiện:
 - + Tiền xử lý: Tách các từ, tạo danh sách từ điển, ...
 - + Xây dựng bộ từ điển chung: $T = \{t_1, t_2, \dots\}$
 - + Mô hình hóa văn bản thành vector: Sử dụng T để tạo ra các vector tần số A và B là \vec{a} và \vec{b} (Dùng $TF * IDF$), tương ứng.
 - + Tính khoảng cách Levenshtein.
 - + Tính độ tương tự dùng công thức (3).
- Đầu ra: Độ tương đồng của A và B.

2.1.4. MANHATTAN SIMILARITY

Đại lượng này được tính bằng cách tính tổng chiều dài các hình chiếu của các đường thẳng kết nối hai điểm:

$$D(\vec{a}, \vec{b}) = \sum_{i=1}^n |a_i - b_i| \quad (4)$$

Giá trị của $D(\vec{a}, \vec{b})$ nằm trong đoạn từ 0 đến 1. Độ tương tự được tính bởi công thức:

$$Sim_M(\vec{a}, \vec{b}) = 1 - \frac{D(\vec{a}, \vec{b})}{n} \quad (5)$$

Giải thuật 2: Độ đo Manhattan

- Đầu vào: Hai chuỗi A và B.
- Thực hiện:
 - + Tiền xử lý: Tách các từ, tạo danh sách từ điển, ...
 - + Xây dựng bộ từ điển chung: $T = \{t_1, t_2, \dots\}$
 - + Mô hình hóa văn bản thành vector: Sử dụng T để tạo ra các vector tần số A và B là \vec{a} và \vec{b} (Dùng $TF * IDF$), tương ứng.
 - + Tính hệ số tương tự Manhattan của hai vector tần số dùng công thức (5).
- Đầu ra: Độ tương đồng của A và B.

2.2. MÔ HÌNH ĐỀ XUẤT

2.2.1. WORD EMBEDDING

2.2.1.1. Giới thiệu

Word Embedding – Kỹ thuật biểu diễn từ trong xử lý ngôn ngữ tự nhiên. Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), việc biểu diễn từ dưới dạng các vector là một bước quan trọng để mô hình có thể hiểu được. Word Embedding là một kỹ thuật được sử dụng để biểu diễn từ trong văn bản dưới dạng các vector trong không gian đa chiều, nó có khả năng miêu tả được mối liên hệ, sự tương đồng về mặt ngữ nghĩa, văn cảnh (context) của dữ liệu. Không gian này bao gồm nhiều chiều và các từ trong không gian đó mà có cùng văn cảnh hoặc có cùng ngữ nghĩa sẽ có vị trí gần nhau.

Ví dụ như ‘ngôi nhà’, ‘tổ ấm’ sẽ có vị trí gần nhau trong không gian chúng ta biểu diễn do chúng có sự tương đồng với nhau về mặt ngữ nghĩa là đều đang nói đến nơi mà chúng ta trở về sau một ngày học tập và làm việc. Chúng ta hiểu rằng máy tính chỉ có thể hiểu được các con số chứ không phải từ ngữ như con người. Vì vậy để máy tính có thể xử lý được các tác vụ liên quan đến ngôn ngữ hay cụ thể hơn là để xây dựng bộ dữ liệu để huấn luyện các mô hình học máy cho các bài toán NLP việc đầu tiên cần làm là biểu diễn các từ ngữ dưới dạng con số cụ thể là các vector.

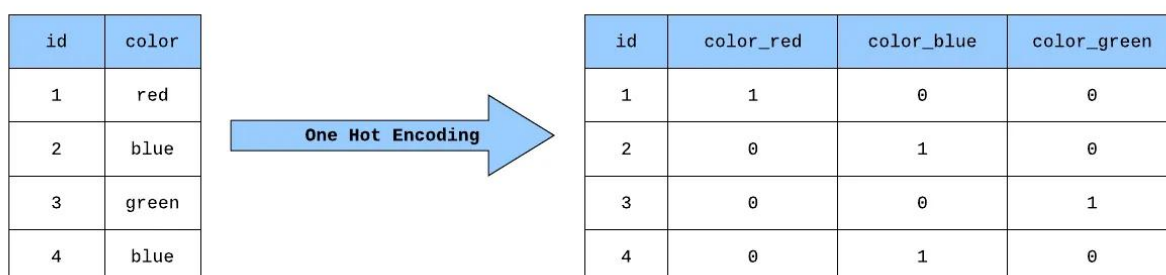
Word Embedding là một nhóm kỹ thuật đặc biệt trong xử lý ngôn ngữ tự nhiên, được sử dụng để ánh xạ một từ hoặc cụm từ trong bộ từ vựng sang một vector số thực trong không gian đa chiều. Thay vì sử dụng không gian một chiều cho mỗi từ, word embedding cho phép biểu diễn các từ dưới dạng các vector liên tục. Các vector từ được biểu diễn theo phương pháp word embedding thể hiện được ngữ nghĩa của các từ và cho phép ta nhận ra mối quan hệ giữa các từ trong không gian vector. Điều này cho phép mô hình học được các đặc trưng ngữ nghĩa của từ và sử dụng chúng để giải quyết các nhiệm vụ xử lý ngôn ngữ tự nhiên như phân loại văn bản, dịch máy và sinh văn bản tự động.

2.2.1.2. Phương pháp tạo ra Word Embedding

Có hai phương pháp chủ yếu được dùng để tính toán Word Embedding là:

1. Count – based methods.
2. Prediction – based methods.

Count – based methods: Phương pháp này dựa trên tần suất xuất hiện của các từ trong văn bản để tạo ra một ma trận thưa (sparse matrix). Ma trận này sau đó được sử dụng để tạo ra các vector biểu diễn cho các từ. Ví dụ điển hình về count – based methods là phương pháp One – hot encoding. One – hot encoding là cách biểu diễn dữ liệu dưới dạng one – hot vector, một vector toàn là giá trị 0 và chỉ có duy nhất một giá trị 1. Một ví dụ hình ảnh thể hiện rõ về phương pháp One – hot encoding:



Hình 2.1: Biểu diễn từ thành vector bằng phương pháp One - hot encoding

Chúng ta có thể thấy rằng phương pháp one – hot encoding là một phương pháp rất đơn giản, dễ thực hiện. Nhưng thông thường bộ từ điển các từ sẽ rất lớn, có thể lên tới hàng triệu từ. Như vậy, điều này dẫn đến kích thước của one – hot vector cũng phải đủ lớn gây tốn kém về không gian lưu trữ. Các bài toán về NLP thì phương pháp này không xử lý được sự liên quan giữa các từ bởi các vector biểu diễn từ được tạo ra bởi one – hot encoding là độc lập với nhau, không có sự tương quan giữa chúng.

Prediction – based methods: Phương pháp này dựa trên các mô hình dự đoán để tạo ra các vector biểu diễn cho các từ. Các mô hình này được huấn luyện trên một lượng lớn các văn bản để dự đoán từ tiếp theo trong một câu văn hoặc văn bản. Ví dụ điển hình về prediction – based methods là phương pháp Word2Vec.

Một số phương pháp word embedding phổ biến bao gồm Word2Vec, GloVe, FastText, những phương pháp này đều sử dụng các thuật toán học máy để tạo ra các biểu diễn từ. Trong đó, Word2Vec là phương pháp phổ biến nhất, sử dụng mô hình Skip-gram hoặc CBOW để tạo ra các vector biểu diễn từ. Chúng ta có thể thấy rằng các phương pháp phổ biến nhất (Word2Vec, GloVe, FastText) đều là *prediction – based methods*. Mặc dù, cả hai phương pháp count – based và prediction – based đều

được sử dụng phổ biến trong xử lý ngôn ngữ tự nhiên và học máy. Tuy nhiên, trong những năm gần đây, các phương pháp prediction – based như Word2Vec, GloVe và FastText đã trở thành phương pháp phổ biến hơn. Điều này là do các phương pháp này có thể tạo ra các biểu diễn từ tốt hơn và đáp ứng được nhu cầu của nhiều ứng dụng xử lý ngôn ngữ tự nhiên, chúng có thể tạo ra các biểu diễn từ phù hợp với nghĩa của chúng và có thể xử lý được các từ hiếm hoặc không xuất hiện trong tập dữ liệu huấn luyện. Tuy nhiên, count – based methods vẫn được sử dụng trong một số trường hợp như là khi dữ liệu đầu vào là các văn bản dài hoặc khi dữ liệu đầu vào không đủ lớn để huấn luyện các phương pháp prediction – based.

2.2.1.3. Ứng dụng của kỹ thuật Word Embedding

Word Embedding đang được sử dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là xử lý ngôn ngữ tự nhiên và học máy. Nó giúp cải thiện độ chính xác và hiệu suất của các mô hình học máy và giúp cho các ứng dụng xử lý ngôn ngữ tự nhiên hoạt động tốt hơn. Một số ứng dụng của kỹ thuật này có thể kể đến như: Phân loại văn bản, dịch thuật, tóm tắt văn bản, phân tích cảm xúc, phân tích ngữ nghĩa, tìm kiếm thông tin,...

1. Phân loại văn bản: Word Embedding được sử dụng để biểu diễn văn bản dưới dạng các vector, được sử dụng để phân loại văn bản vào các nhóm khác nhau. Ví dụ, phân loại tin tức, phân loại thư rác, phân loại sản phẩm trong các trang thương mại điện tử...
2. Dịch thuật: Word Embedding được sử dụng trong các mô hình dịch thuật máy để tạo ra các biểu diễn từ của nhiều ngôn ngữ khác nhau, giúp cải thiện độ chính xác của các mô hình dịch thuật.
3. Phân tích cảm xúc: Word Embedding được sử dụng trong các mô hình phân tích cảm xúc, giúp phân tích cảm xúc của người dùng bình luận, tin nhắn, tweet trong các bài viết trên mạng xã hội.
4. Tóm tắt văn bản: Word Embedding được sử dụng để tạo ra các biểu diễn cho các câu văn trong văn bản, giúp tóm tắt văn bản một cách tự động và hiệu quả.

Word Embedding là một kỹ thuật rất quan trọng trong lĩnh vực NLP. Việc biểu diễn các từ dưới dạng vector giúp cho các mô hình NLP có thể xử lý và phân tích thông tin trong văn bản một cách hiệu quả hơn. Các phương pháp tạo ra Word

Embedding cũng đang được nghiên cứu và phát triển liên tục để cải thiện hiệu quả của kỹ thuật này trong các ứng dụng thực tế.

2.2.2. SENTENCE EMBEDDING

Sentence embedding là tên gọi chung của một tập hợp các kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) trong đó các câu được ánh xạ tới các vector của số thực. Các phần nhúng này được thiết kế để nắm bắt ý nghĩa ngữ nghĩa của văn bản và giúp các thuật toán máy học dễ dàng xử lý và hiểu hơn và ngày càng phát triển hơn. Điều này giúp cho việc so sánh, phân loại và trích xuất thông tin từ các câu trở nên dễ dàng hơn. Các phương pháp sentence embedding thường sử dụng các mô hình học sâu như word embedding, neural network, hay transformer để biểu diễn các câu thành các vector số có tính chất phản ánh đặc trưng của câu như ý nghĩa, cú pháp, ngữ cảnh, v.v. Các phương pháp sentence embedding được sử dụng rộng rãi trong các ứng dụng xử lý ngôn ngữ tự nhiên như trích xuất thông tin, dịch máy, phân loại văn bản, v.v.

Sentence embedding đã thay đổi hoàn toàn lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) trong những năm gần đây bằng cách cho phép mã hóa các đoạn văn bản dưới dạng các vector có kích thước cố định. Một trong những bước đột phá gần đây nhất được sinh ra từ cách thể hiện dữ liệu văn bản sáng tạo này là một tập hợp các phương pháp để tạo các nhúng câu, còn được gọi là vector câu. Các phần nhúng này giúp có thể biểu thị các đoạn văn bản dài hơn bằng số dưới dạng vector mà thuật toán máy tính, chẳng hạn như mô hình máy học (Machine Learning), có thể xử lý trực tiếp.

Một câu hỏi chính trong NLP là làm thế nào để biểu diễn dữ liệu văn bản ở định dạng mà máy tính có thể hiểu và làm việc dễ dàng. Giải pháp là chuyển đổi ngôn ngữ thành dữ liệu số - các phương pháp truyền thống như TF-IDF và mã hóa một lần nóng đã được áp dụng trong lĩnh vực này trong vài thập kỷ. Tuy nhiên, những phương pháp này có một hạn chế lớn, đó là chúng không nắm bắt được thông tin ngữ nghĩa chi tiết có trong ngôn ngữ của con người. Ví dụ: phương pháp tiếp cận Bag-of-Word phổ biến chỉ tính đến việc có hay không các mục từ vựng có trong câu hoặc tài liệu, bỏ qua ngữ cảnh rộng hơn và mối quan hệ ngữ nghĩa giữa các từ.

Tuy nhiên, đối với nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên, điều quan trọng là phải có quyền truy cập vào kiến thức ngữ nghĩa vượt xa các biểu diễn dựa trên số lượng đơn giản. Các phần nhúng là các vector đa chiều, có độ dài cố định giúp có thể

trích xuất và thao tác ý nghĩa của các phân đoạn mà chúng đại diện, chẳng hạn như bằng cách so sánh mức độ giống nhau của hai câu với nhau về mặt ngữ nghĩa.

Có một số kỹ thuật và thuật toán có sẵn để tạo nhúng câu. Một số phương pháp phổ biến bao gồm như sau:

Bag of Word (BoW)

BoW là một kỹ thuật đơn giản biểu diễn một câu dưới dạng một vector tần số của từ. Nhược điểm chính của BoW là nó không xem xét thứ tự của các từ hoặc nắm bắt ý nghĩa ngữ nghĩa một cách hiệu quả.

Thuật ngữ tần số-ngịch đảo tần số tài liệu (Term Frequency-Inverse Document Frequency - TF-IDF)

TF-IDF là một phần mở rộng của BoW có tính đến tầm quan trọng của một từ trong tài liệu và toàn bộ kho văn bản. Nó gán trọng số cho các từ dựa trên tần suất của chúng trong tài liệu và nghịch đảo của tần suất của chúng trong kho văn bản.

Word Embeddings

Nhúng từ là các biểu diễn vector dày đặc của các từ riêng lẻ nắm bắt ý nghĩa ngữ nghĩa của chúng. Các kỹ thuật nhúng từ phổ biến bao gồm Word2Vec, GloVe và FastText. Các nhúng câu có thể được tạo bằng cách tính trung bình, tổng hợp hoặc nối các từ nhúng của các từ trong câu.

Doc2Vec

Doc2Vec là một phần mở rộng của Word2Vec học cách nhúng cho toàn bộ câu hoặc tài liệu thay vì từng từ riêng lẻ. Nó sử dụng cùng một kiến trúc mạng thần kinh cơ bản nhưng bổ sung thêm một vector "tài liệu" trong quá trình đ

Các mô hình ngôn ngữ được đào tạo trước

Các mô hình ngôn ngữ được đào tạo trước, chẳng hạn như BERT, GPT và RoBERTa, đã trở nên phổ biến để tạo các nhúng câu. Các mô hình này được đào tạo trên kho văn bản lớn và có thể tạo các phần nhúng theo ngữ cảnh cho mỗi mã thông báo trong một câu. Sau đó, các phần nhúng này có thể được tổng hợp (ví dụ: bằng cách tính trung bình hoặc gộp) để tạo biểu diễn vector có kích thước cố định cho toàn bộ câu.

Nhúng câu được sử dụng rộng rãi trong các tác vụ xử lý ngôn ngữ tự nhiên, chẳng hạn như:

- Phân loại văn bản
- Phân tích tình cảm
- Dịch máy
- Truy xuất thông tin
- Tóm tắt văn bản
- Trả lời câu hỏi
- Đo độ tương tự văn bản ngữ nghĩa

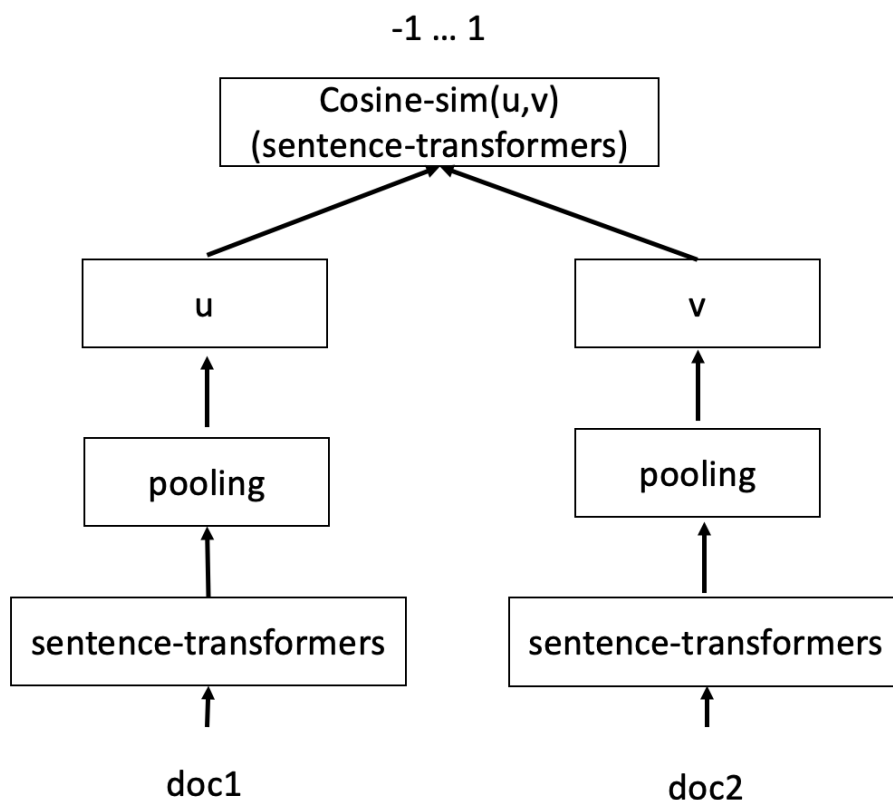
Tóm lại, nhúng câu là một công cụ thiết yếu trong xử lý ngôn ngữ tự nhiên cho phép máy xử lý, so sánh và hiểu dữ liệu văn bản. Có nhiều kỹ thuật khác nhau để tạo các nhúng câu và việc chọn đúng phương pháp phụ thuộc vào nhiệm vụ cụ thể và các yêu cầu của bài toán.

2.2.3. SENTENCE TRANSFORMERS

Sentence Transformers là một mô hình biến đổi câu: Ánh xạ các câu và đoạn văn tới một không gian vector dày đặc gồm 768 chiều. Sentence Transformers có thể được sử dụng để tính toán sự giống nhau giữa hai câu. Điều này hữu ích cho các tác vụ như truy xuất thông tin, nơi bạn cần tìm các tài liệu tương tự với một truy vấn nhất định.

Sentence Transformers câu có thể được sử dụng để nhúng các câu vào một không gian vector. Điều này rất hữu ích cho các nhiệm vụ như phân loại văn bản hoặc sự giống nhau về ngữ nghĩa khi cần so sánh các câu. Sentence Transformers câu có thể được sử dụng để thực hiện tìm kiếm ngữ nghĩa. Điều này hữu ích cho các tác vụ như trả lời câu hỏi, nơi bạn phải tìm tài liệu chứa câu trả lời cho một câu hỏi nhất định.

Một câu Transformer có thể được sử dụng để phân cụm các tài liệu. Điều này hữu ích cho các tác vụ như lập mô hình chủ đề hoặc phân loại tài liệu, khi bạn cần nhóm các tài liệu theo chủ đề hoặc danh mục.



Hình 2.2: Biểu đồ tính Similar sentence-transformers

Để đo độ tương tự giữa hai văn bản chúng ta sẽ sử dụng độ tương tự Cosine các vector của văn bản sẽ dùng sentence-transformers để tính ra các vector.

Độ tương tự Cosine: Độ tương tự Cosine đo độ tương tự giữa hai vector của một không gian tích bên trong. Nó được đo bằng cosin của góc giữa hai vector và xác định xem hai vector có chỉ theo cùng một hướng hay không. Bất kỳ tài liệu nào cũng có thể được biểu diễn bằng hàng nghìn thuộc tính, mỗi thuộc tính ghi lại tần suất của một từ cụ thể (chẳng hạn như từ khóa) hoặc cụm từ trong tài liệu. Do đó, mỗi tài liệu là một đối tượng được biểu diễn bởi cái được gọi là vector tần số thuật ngữ. Do đó hai văn bản càng giống nhau về mặt ngữ nghĩa thì độ tương tự Cosine lại gần ra 1, nếu hai văn bản không giống nhau thì lại gần ra 0.

2.2.4. COMBSENTSIM

WordNet tiếng Việt là một từ điển từ ngữ nghĩa được phát triển bởi Viện Công nghệ thông tin và Truyền thông Việt Nam (IOIT) và được công bố vào năm 2009. WordNet tiếng Việt bao gồm một tập hợp các từ được phân loại thành các nhóm đồng nghĩa (synset) và các mối quan hệ từ vựng khác nhau. Các từ trong WordNet tiếng Việt được phân loại vào các nhóm đồng nghĩa và được liên kết với các từ khác thông qua các mối quan hệ từ vựng như đồng nghĩa, trái nghĩa, tương đồng, tương phản và phụ thuộc.

WordNet có thể được sử dụng để tính toán độ tương đồng giữa các từ trong ngôn ngữ tự nhiên. Các phương pháp tính toán độ tương đồng này dựa trên số lượng synset chung giữa các từ, khoảng cách trên đồ thị từ điển WordNet, hoặc các phương pháp thống kê khác.

Một trong những phương pháp tính độ tương đồng phổ biến nhất trong WordNet là độ đo ngữ nghĩa (semantic similarity) dựa trên đồ thị WordNet. Độ đo này tính toán độ tương đồng giữa hai từ dựa trên khoảng cách trên đồ thị WordNet. Khoảng cách được tính bằng cách đếm số bước đi từ synset gốc chứa từ đầu tiên đến synset gốc chứa từ thứ hai.

WordNet cũng có thể được sử dụng để trích xuất các từ đồng nghĩa hoặc liên quan đến một từ cho trước. Điều này có thể được thực hiện bằng cách tìm các synset chứa từ đó, sau đó lấy tất cả các từ trong các synset này.

Độ tương tự giữa hai từ u và v , chúng tôi sử dụng độ đo tương tự Cosine giữa hai vector nhúng từ công thức 2.1 sau:

$$Sim_{Embed}(u, v) = \frac{\vec{v}_u \cdot \vec{v}_v}{\|\vec{v}_u\| \cdot \|\vec{v}_v\|} \quad (2.1)$$

Để đo lường chính xác hơn độ tương tự của cặp từ, chúng tôi sử dụng tập các cặp từ trái nghĩa A (Antonyms) và tập các từ đồng nghĩa S (Synonymy) được chúng tôi tổng hợp.

$$Sim_{Embed}(u, v) = \begin{cases} 0 & \text{nếu } u - v \in A \\ 1 & \text{nếu } u - v \in S \\ Sim_{Embed} & \text{nếu } u - v \notin S \cup A \end{cases} \quad (2.2)$$

Sau đó chúng tôi đo độ tương tự giữa một từ với một câu:

$$Sim_{Cross}(u, S) = \operatorname{argmax}_{v \in S} Sim_{EmbedExt}(u, v) \quad (2.3)$$

Và cuối cùng chúng ta sẽ có độ tương tự giữa hai câu:

$$Sim_{WN}(S_1, S_2) = \frac{1}{2} \times \left(\frac{1}{N} \sum_{u \in S_1} Sim_{Cross}(u, S_2) + \frac{1}{M} \sum_{v \in S_2} Sim_{Cross}(v, S_1) \right) \quad (2.4)$$

Độ đo Sim_{WN} sẽ nằm trong khoảng từ 0 đến 1.

PhoBERT hiện đang là mô hình ngôn ngữ tiếng Việt tốt nhất bởi được huấn luyện trên cơ sở bộ dữ liệu tiếng Việt rất lớn vì vậy nó có khả năng xử lý rất tốt và đạt độ chính xác cao về mức độ tương đồng ngữ nghĩa cặp câu tiếng Việt. Nên việc PhoBERT có mặt ở đây là điều dễ hiểu. Khi sử dụng PhoBERT cho tác vụ đo lường mức độ tương đồng của cặp câu chúng tôi vẫn sử dụng phương pháp “*Cosine Similarity*” để tính toán khoảng cách giữa các vector biểu diễn của các câu đó. Tuy nhiên mặc dù là mô hình dành riêng cho tiếng Việt nhưng PhoBERT vẫn không thể cho ra kết quả tốt với mọi bộ dữ liệu được, đó thực sự là một thách thức không hề nhỏ. Vì vậy với mô hình CombSentSim chúng tôi kết hợp PhoBERT, Jaccard và WordNet để giúp cho mô hình đạt được kết quả tốt hơn bằng cách sử dụng kết hợp kết quả của PhoBERT, Jaccard và WordNet với công thức 2.5 sau:

$$\alpha * PhoBERT + \beta * Jaccard + \gamma * WordNet \quad \begin{cases} \alpha, \beta, \gamma \in [0, 1] \\ \alpha + \beta + \gamma = 1 \end{cases} \quad (2.5)$$

Công thức sử dụng để tính toán trung hoà giữa hai đầu ra của PhoBERT, Jaccard và WordNet với α (α), β (β) và γ (γ) là các trọng số giữa ba giá trị đó. Với công thức này chúng tôi dự đoán trọng số α sẽ phải lớn hơn 0.5 bởi vì chúng tôi đánh giá PhoBERT cao hơn trong tác vụ này và chúng tôi mong muốn α nằm trong khoảng từ 0.60 đến 0.65 sẽ là phù hợp nhất bởi vì chúng tôi muốn giảm sự phụ thuộc vào PhoBERT trong mô hình này, điều này cho thấy Jaccard và WordNet cũng đang cho kết quả tốt.

Với Jaccard, như chúng tôi đã giới thiệu Jaccard không thể nhận biết được từ đồng nghĩa, trái nghĩa hoặc liên quan đến nhau. Vì vậy chúng tôi sử dụng thêm Word2vec để giúp tìm kiếm các từ tương tự trong câu vì thế mà làm mềm được phương pháp Jaccard để cho ra kết quả tối ưu nhất có thể.

Với tác vụ của bài toán tìm độ tương đồng ngữ nghĩa của hai cặp câu thì ta cần phân biệt được hai câu đó có phải hai câu trái nghĩa hoặc đồng nghĩa nhau hay không, ví dụ ta có hai cặp câu: “*Tôi thường_xuyên đi tập thể dục vào mỗi buổi chiều sau khi đi làm về*” và “*Tôi không_bao_giờ đi tập thể dục vào buổi chiều sau khi tan ca*”. Với hai câu ở ví dụ trên thì đối với bài toán tìm kiếm thì hai câu này có mức độ liên quan khá cao bởi chúng có cùng ngữ cảnh và sử dụng từ ngữ khá giống nhau, nhưng đối với bài toán so sánh về độ tương đồng ngữ nghĩa thì hai câu trên nên có độ tương đồng thấp bởi “*thường_xuyên*” và “*không_bao_giờ*” là hai từ trái nghĩa với nhau. Do đó, chúng tôi sử dụng WordNet để khắc phục điểm yếu này của mô hình.

CHƯƠNG 3: THỰC NGHIỆM VÀ PHÂN TÍCH

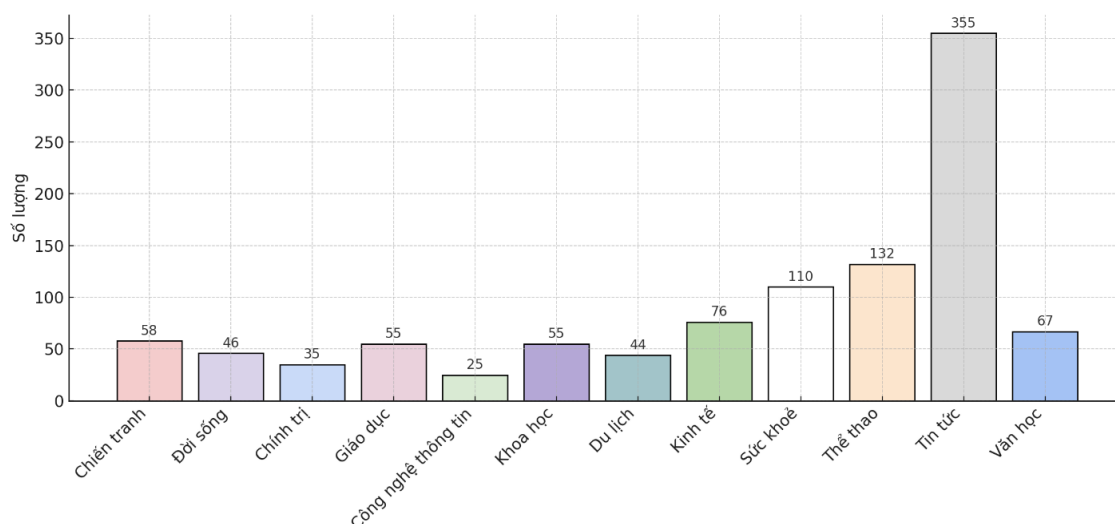
3.1. BỘ DỮ LIỆU

3.1.1. Quy mô và lĩnh vực nội dung

Bộ dữ liệu chứa khoảng hơn 1000 các cặp câu song ngữ Việt - Trung. Tuy chưa phải là một số lượng lớn nhưng ta vẫn có thể nhận được kết quả tích cực trong việc đánh giá nếu có đủ sự phong phú, đa dạng trong cú pháp, ngữ nghĩa. Trong tương lai bộ dữ liệu sẽ còn được bổ sung thêm các cặp câu để bắt kịp với xu hướng xã hội, mở rộng thêm các lĩnh vực để ngày càng hoàn thiện hơn.

Được đánh giá bởi các chuyên gia đóng vai trò như bước "kiểm định chất lượng", phát hiện và sửa lỗi trong dịch thuật, từ đó tăng độ tin cậy của dữ liệu. Bộ dữ liệu chủ yếu sử dụng văn phong chính quy, đã được chuẩn hóa nhằm đảm bảo tính chính xác, phù hợp và tạo thuận lợi cho quá trình tiền xử lý, tokenize, ... Đồng thời chỉ bao gồm các dạng ngôn ngữ chuẩn của tiếng Việt và tiếng Trung, không đề cập đến các thuật ngữ chuyên ngành, ngôn từ địa phương, từ lóng hay các từ ngữ được tạo ra theo trend, genz, ...

Bộ dữ liệu được bao gồm hơn 20 lĩnh vực nhằm đảm bảo sự phong phú, độ bao phủ rộng. Các câu trong bộ dữ liệu có mức độ phức tạp khác nhau, từ cấu trúc đơn giản đến các câu có cú pháp phức tạp hơn. Nội dung của bộ dữ liệu tập trung vào các lĩnh vực có tính thực dụng cao như báo chí, kinh tế, xã hội, chính trị, Sự phân bố đa dạng này giúp đánh giá mô hình đối với cả hai ngôn ngữ một cách phong phú, trực quan hơn



Hình 3.1: Biểu đồ lượng cặp câu từng miền

3.1.2. Tiền xử lý dữ liệu

Là bước đóng vai trò quan trọng trong việc xây dựng bộ dữ liệu Việt-Trung nhằm đảm bảo tính nhất quán, chất lượng và khả năng ứng dụng. Quá trình này đối mặt với nhiều thách thức, bao gồm việc văn bản chứa các câu dài hoặc không đúng định dạng, sự xuất hiện của ngôn ngữ khác ngoài tiếng Việt và tiếng Trung, lỗi chính tả hoặc định dạng không thống nhất, cũng như sự chênh lệch đáng kể về độ dài của hai câu có thể làm ảnh hưởng đến hiệu quả của mô hình.

Để khắc phục những vấn đề này quá trình tiền xử lý bao gồm nhiều bước. Trước tiên, cấu trúc câu được chuẩn hóa bằng cách chỉ bao gồm các cặp câu đơn lẻ thay vì đoạn văn dài, đồng thời loại bỏ hoặc tách riêng những câu chứa nhiều mệnh đề không phù hợp. Tiếp theo cần loại bỏ những từ không phải tiếng Việt hoặc tiếng Trung, cũng như các ký tự đặc biệt gây lỗi mã hóa để duy trì chất lượng dữ liệu mà lại không có nhiều ý nghĩa.

Bên cạnh đó, các chuyên gia ngôn ngữ tiến hành kiểm tra và sửa lỗi chính tả nhằm đảm bảo độ chính xác của văn bản. Để duy trì tính đồng nghĩa giữa các cặp câu song ngữ, các chuyên gia cũng thực hiện quá trình xác minh để đảm bảo rằng cả hai câu trong một cặp đều truyền tải cùng một ý nghĩa. Ngoài ra, văn bản được chuẩn hóa về dấu câu, khoảng trắng và mã hóa ký tự nhằm đảm bảo tính đồng nhất và phù hợp với các mô hình. Cuối cùng, để cân bằng giữa tính tự nhiên của câu và tính khả dụng

của bộ dữ liệu trong huấn luyện mô hình, sự chênh lệch độ dài giữa hai câu được giới hạn ở mức tối đa năm từ.

Những bước tiền xử lý này giúp nâng cao chất lượng dữ liệu, tối ưu hóa khả năng áp dụng vào mô hình, đồng thời đảm bảo độ chính xác và tính nhất quán trong khi xử lý văn bản song ngữ Việt - Trung.

3.1.3. Chọn lọc các cặp câu

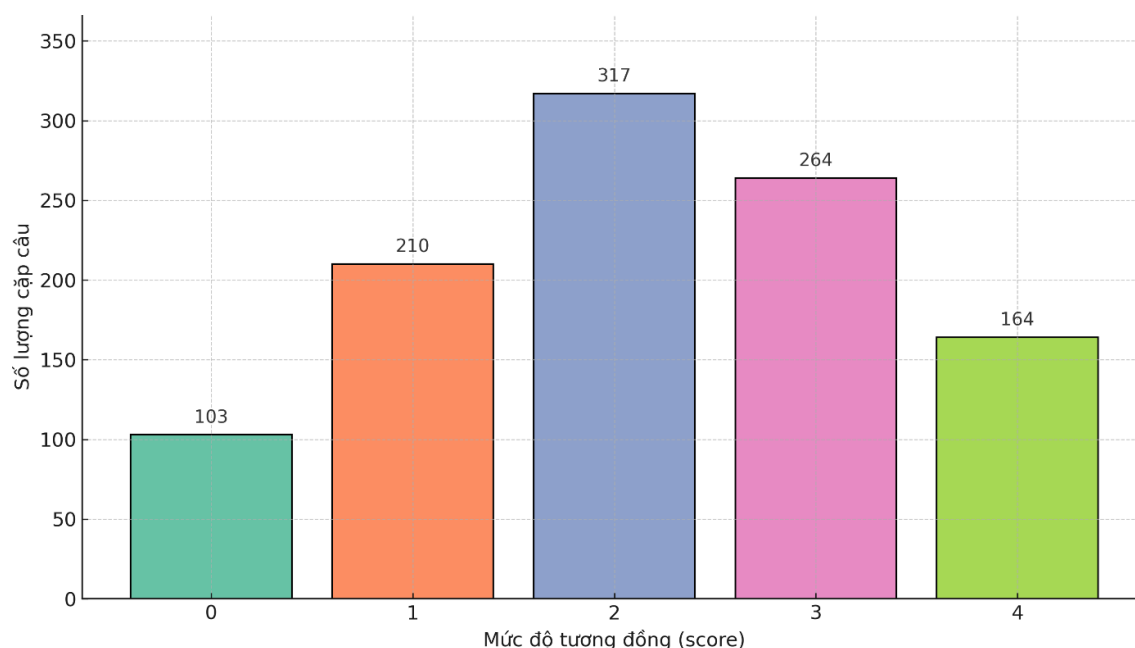
Sau khi thực hiện tiền xử lý và áp dụng các quy tắc chuẩn hóa dữ liệu, việc chọn lọc các cặp câu đóng vai trò quan trọng trong việc đảm bảo chất lượng của bộ dữ liệu. Quá trình này được thực hiện với sự cân nhắc của các chuyên gia ngôn ngữ nhằm tạo ra một tập dữ liệu có phân bố điểm tương đồng hợp lý, giúp tối ưu hóa việc đánh giá mô hình.

Các cặp câu sẽ có mức độ tương đồng về ngữ nghĩa theo thang điểm từ 0 đến 4, đảm bảo tính chính xác trong việc đánh giá mức độ đồng nghĩa. Đồng thời, để tránh tình trạng bị lệch về một nhóm điểm cụ thể. Ngoài ra, các cặp câu được chọn lọc cần đảm bảo đúng cấu trúc ngữ pháp, không có lỗi chính tả và có độ dài phù hợp để tránh mất cân bằng trong tập dữ liệu.

3.1.4. Đánh giá dữ liệu

Bộ dữ liệu sử dụng thang điểm 0 – 4 để đánh giá mức độ tương đồng giữa các cặp câu. Mỗi điểm số tương ứng với các mức độ cụ thể tương ứng:

- Không liên quan (0)
- Có chút liên quan (1)
- Liên quan nhưng không tương tự (2)
- Tương tự (3)
- Rất giống (4)



Hình 3.2: Biểu đồ tính mức độ tương đồng dựa trên thang điểm

3.1.5. Cấu trúc bộ dữ liệu

Sau khi đã có các cặp câu được chuẩn hóa và các độ tương đồng tương ứng, chúng tôi ghi dữ liệu vào file excel theo định dạng mỗi cặp câu được biểu diễn trên một dòng. Trong đó, cột thứ nhất là câu tiếng Việt, cột thứ hai là câu tiếng Trung, cột thứ ba là lĩnh vực liên quan của cặp câu và cột cuối cùng là mức độ tương đồng được các chuyên gia đánh giá. Các dòng tiếp theo gồm các cặp câu tiếp với định dạng như vậy.

Việc ghi cấu trúc dữ liệu như vậy không chỉ giúp dễ dàng đọc hiểu, rà soát, quản lý, ... mà còn tạo điều kiện thuận lợi cho việc đọc hay ghi dữ liệu sau này trong quá trình cài đặt và thực nghiệm để đánh giá sự hiệu quả của mô hình.

3.2: THỰC NGHIỆM VÀ GIAO DIỆN

3.2.1. Các mô hình ngôn ngữ lớn được áp dụng

Trong quá trình thực nghiệm, nhóm nghiên cứu đã sử dụng các mô hình:

- PhoBERT³: Được coi là một trong những mô hình ngôn ngữ hiện đại nhất dành riêng cho tiếng Việt do VinAI huấn luyện, phát triển và công bố. Là mô hình ngôn ngữ đã được huấn luyện trước dựa từ tập dữ liệu tiếng Việt rất lớn từ tập văn bản các bài báo và tập văn bản Wikipeida tiếng Việt. Tận dụng cơ chế attention để học được mối liên hệ giữa các từ trong câu, từ đó cho ra hiệu năng tốt hơn.
- VisoBERT⁴: Giống như PhoBERT là mô hình dành cho tiếng Việt. Tuy nhiên không “tổng thể” các ngữ cảnh như PhoBERT mà VisoBERT đi sâu vào mạng xã hội – Nơi mà nhưng ngôn từ được thể hệ genz sáng tạo ra không ngừng, nhưng từ ngữ này chứa đựng nhiều ý nghĩa mà các mô hình ngôn ngữ dành cho tiếng Việt gặp khó.
- SimCSE⁵: Là một framework học tương phản được dùng để sinh vector biểu diễn đại diện cho câu sao cho những câu gần thì các vectors biểu diễn gần nhau còn nhưng câu xa nhau thì các vectors cách xa nhau. SimCSE được coi là tốt hơn so với Bert vì nó huấn luyện riêng cho tương đồng câu, vector embedding rõ ràng.

3.2.2. Công cụ và các thư viện chính:

Việc triển khai các mô hình ngôn ngữ lớn đòi hỏi sự kết hợp của các công cụ lập trình, thư viện chuyên dùng và nền tảng mạnh mẽ nhằm tối ưu hiệu quả và tính khả dụng. Báo cáo này trình bày chi tiết các cài đặt, công cụ, thư viện và mô hình đã được thử nghiệm trong qua trình nghiên cứu và thực nghiệm.

Nhóm thực hiện đã chọn Python làm ngôn ngữ lập trình, do Python là ngôn ngữ hiện đại, mạnh mẽ, có một cộng đồng lớn trong việc phát triển các mô hình dịch máy nói chung hay xử lý ngôn ngữ tự nhiên nói riêng. Các công cụ và thư viện được sử dụng bao gồm:

³ [VinAIResearch/PhoBERT: PhoBERT: Pre-trained language models for Vietnamese \(EMNLP-2020 Findings\)](#)

⁴ [uitnlp/ViSoBERT: ViSoBERT: A Pre-Trained Language Model for Vietnamese Social Media Text Processing \(EMNLP'2023\)](#)

⁵ [princeton-nlp/SimCSE: \[EMNLP 2021\] SimCSE: Simple Contrastive Learning of Sentence Embeddings https://arxiv.org/abs/2104.08821](#)

- Mô hình Transformers⁶(Hugging Face): Là mô hình nền tảng cho các mô hình ngôn ngữ lớn như PhoBert, VisoBert và SimCSE đồng thời cũng là cơ sở để làm việc với các mô hình đó.
- PyTorch⁷: Thư viện học máy mạnh mẽ, hỗ trợ cho việc tính toán trên các tensor đồng thời còn cung cấp các hàm tính độ tương đồng như Cosine.
- Numpy⁸: Thư viện hỗ trợ việc tính toán trên các ma trận. Đọc ghi dữ liệu trên các loại file. Thích hợp cho việc lấy các cặp câu từ bộ dữ liệu để làm việc.

Ngoài ra chúng tôi sử dụng các IDE mạnh mẽ để hỗ trợ cho việc cài đặt mô hình bằng Python như Visual Studio Code và PyCharm.

3.2.3. Quy trình thực hiện

Theo mô hình của chúng tôi, hai câu đầu vào sẽ bao gồm một câu tiếng Việt và một câu tiếng Trung. Vì mô hình có các bước tính toán dựa vào các cặp từ đồng nghĩa, trái nghĩa được tổng hợp; sử dụng độ đo Jaccard để tính độ tương tự về mặt từ ngữ giữa hai tập hợp. Chính vì vậy cần phải dịch câu tiếng Trung về tiếng Việt để đảm bảo cho quá trình tính toán. Ở đây, chúng tôi dùng Google API từ thư viện googletrans của Python.

Dữ liệu dù đã được các chuyên gia xem xét, chọn lọc, đánh giá, sửa đổi, ... nhưng vẫn khó tránh khỏi sai sót. Bên cạnh đó việc xử lý dữ liệu đầu vào tốt còn tạo thuận lợi cho các công việc tính toán, lập trình cài đặt sau này.

Trước tiên chúng tôi thực hiện tokenize sử dụng VnCoreNLP⁹ là bộ công cụ xử lý ngôn ngữ tự nhiên tiếng Việt do nhóm nghiên cứu tại viện John von Neumann(ĐHQG TP.HCM) phát triển. Đây là một thư viện mạnh mẽ và phổ biến cho các tác vụ xử lý văn bản tiếng Việt. Đồng thời đáp ứng yêu cầu thực hiện chia từ trong câu(word-segment) của mô hình PhoBert.

⁶ <https://huggingface.co/docs/transformers/v4.17.0/en/index>

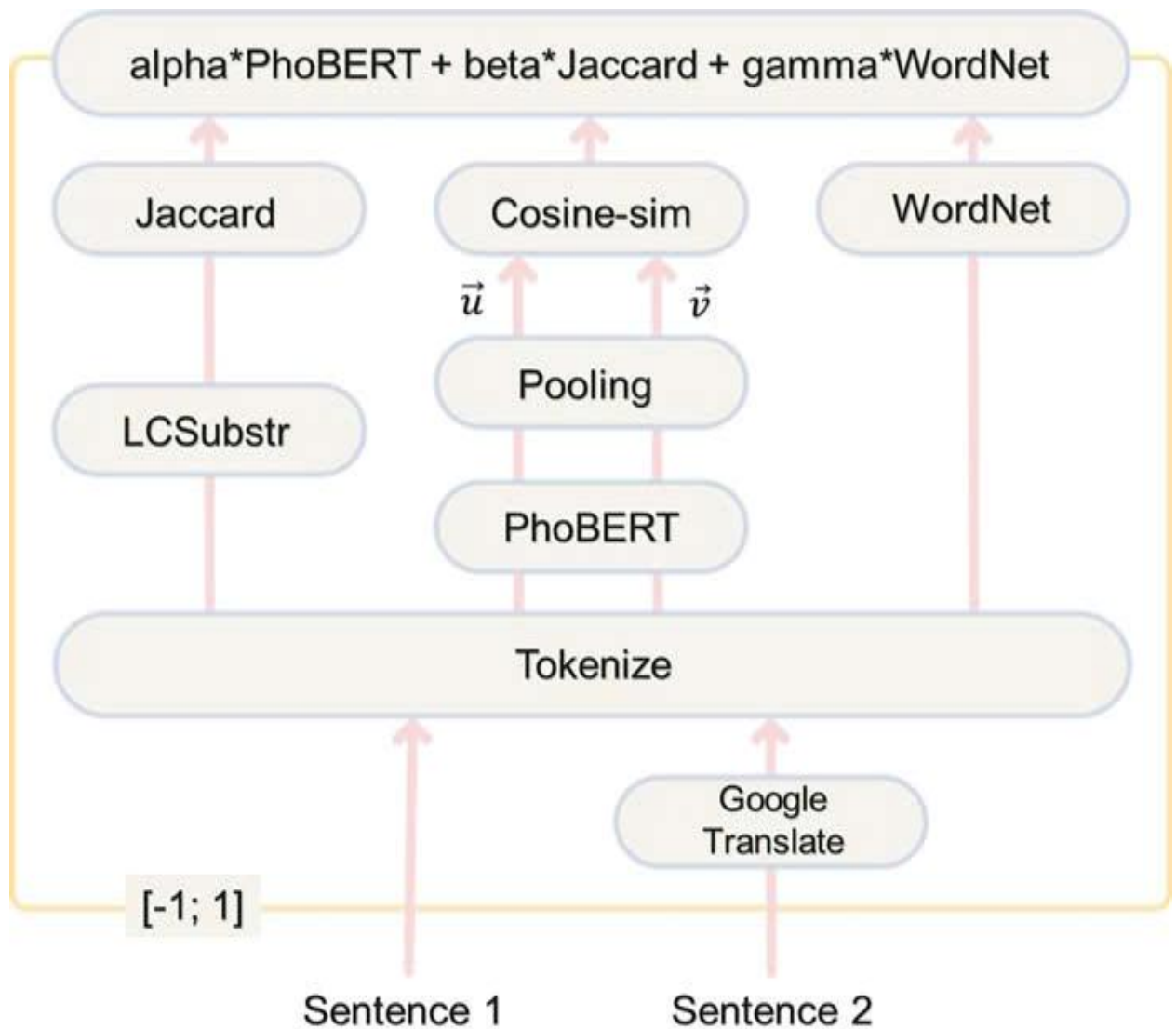
⁷ [pytorch/pytorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration](https://pytorch.org/)

⁸ [numpy/numpy: The fundamental package for scientific computing with Python.](https://numpy.org/)

⁹ [vncorenlp/VnCoreNLP: A Vietnamese natural language processing toolkit \(NAACL 2018\)](https://github.com/vncorenlp/VnCoreNLP)

Khi này câu đã được tokenize – mỗi cụm từ có nghĩa trong câu được nối với nhau bởi dấu “_” thể hiện đó là 1 cụm từ có nghĩa bao gồm cả các danh từ riêng, tên riêng hay cụm động từ. Tuy nhiên đến bước này chúng tôi chưa loại bỏ ngay các dấu câu hay ký tự đặc biệt như dấu chấm, dấu phẩy hay dấu chấm than, ... Vì các mô hình PhoBert, VisoBert hay SimCSE là mô hình hiện đại, thông minh nên nó có thể “học” được ý nghĩa của các ký tự đó với câu. Sau khi tính xong độ tương đồng sử dụng các mô hình này, chúng tôi tiếp tục chuẩn hóa một lần nữa. Lúc này câu đã được loại bỏ hoàn toàn các ký tự đặc biệt, chuyển hết về chữ thường để thực hiện bước tính toán bằng WordNet, Jaccard cuối cùng.

Khi đã có kết quả của ba độ đo dựa trên 3 phương pháp chúng tôi kết hợp ba công thức lại bằng cách gán trọng số cho từng độ đo rồi cộng tổng lại để thu được kết quả cuối cùng. Việc đánh trọng số là một công việc thực sự khó khăn, vì mỗi độ đo có những tư tưởng khác nhau, mỗi phương pháp đều cho ra những kết quả khác nhau trong những cặp câu khác nhau. Tuy nhiên qua nhiều quá trình thực nghiệm trước đó, chúng tôi tổng kết lại rằng: Mô hình PhoBert, VisoBert, Simcse tuy chưa cho ra kết quả thực sự chuẩn xác nhưng vẫn tương đối ổn định – tức sẽ không lệch quá xa khỏi độ tương đồng thực tế; còn độ đo dựa trên Jaccard và WordNet lại cho ra kết quả tốt với các cặp câu rõ ràng về ngữ nghĩa hay có nhiều từ giống nhau nhưng lại có hiệu quả kém với các cặp câu có độ phức tạp cao hơn.



Hình 3.3: Mô hình tổng quát hóa

Cuối cùng chúng tôi rút ra bộ ba trọng số tối ưu: $a = 0.6$, $b = 0.2$, $d = 0.2$ tức công thức cuối cùng là:

$$\text{ComboSim} = 0.6 * \text{PhoBert} + 0.2 * \text{Jaccard} + 0.2 * \text{WordNet}$$

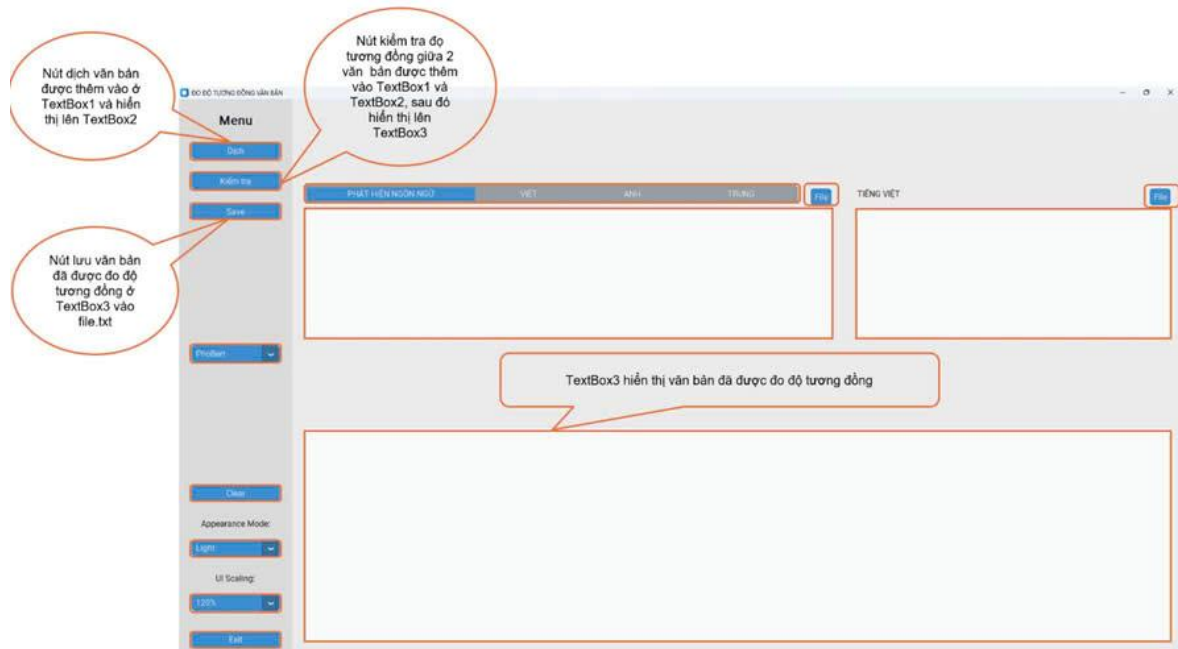
Việc sử dụng công thức này cho ra kết quả tương đối tốt khi thực nghiệm trên bộ dữ liệu hơn 1000 cặp câu song ngữ Việt-Trung của chúng tôi.

Vì cả ba mô hình ngôn ngữ mà chúng tôi đề cập trong việc kết hợp các độ đo đều dựa trên kiến trúc Transformers nên các mô hình VisoBert, SimCSE đều tương tự như mô hình trên, nên chỉ cần thay PhoBert bằng hai mô hình đó là được.

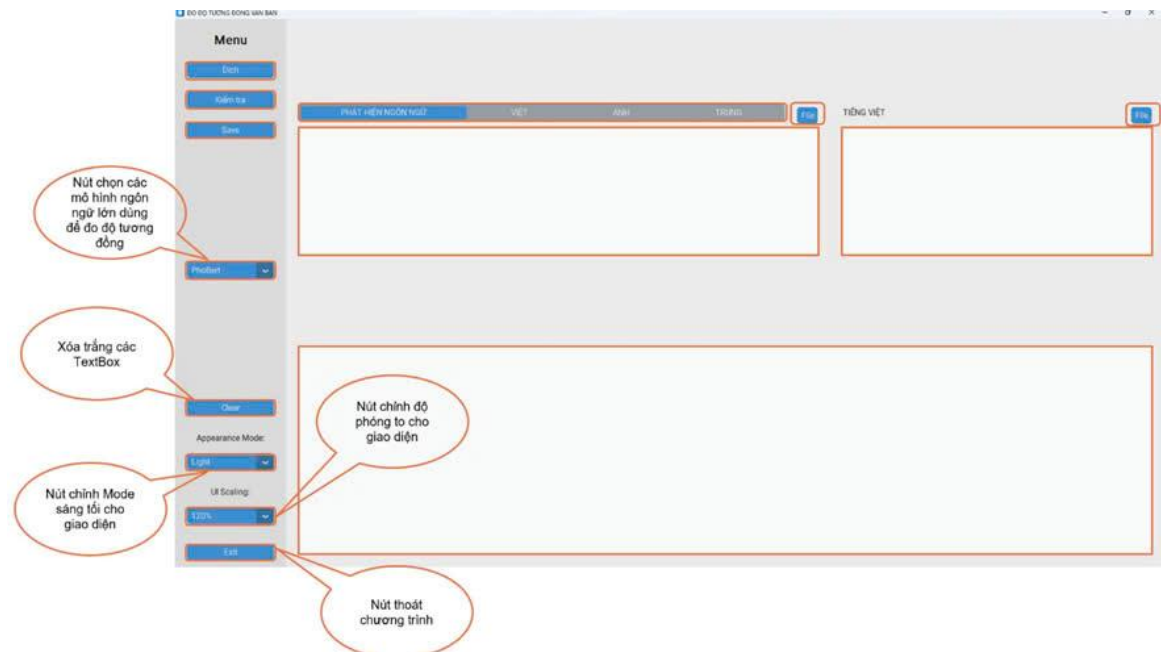
3.2.4. Chương trình giao diện trực quan

Để thuận tiện cho quá trình quan sát, theo dõi khi chạy mô hình, chúng tôi đã xây dựng một giao diện ứng dụng demo hiển thị các cặp câu, các kết quả tương ứng. Bên cạnh đó ứng dụng còn có các chức năng tùy chỉnh

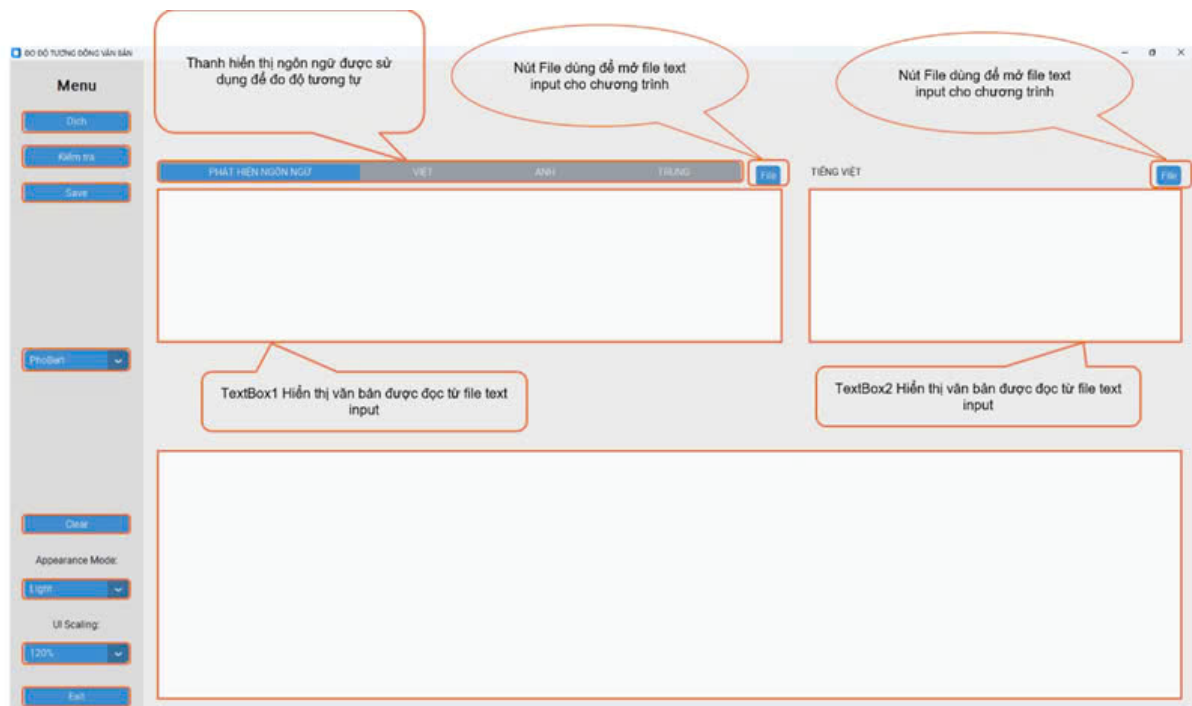
Ứng dụng demo:



Hình 3.4: Giao diện thanh Menu các nút chức năng

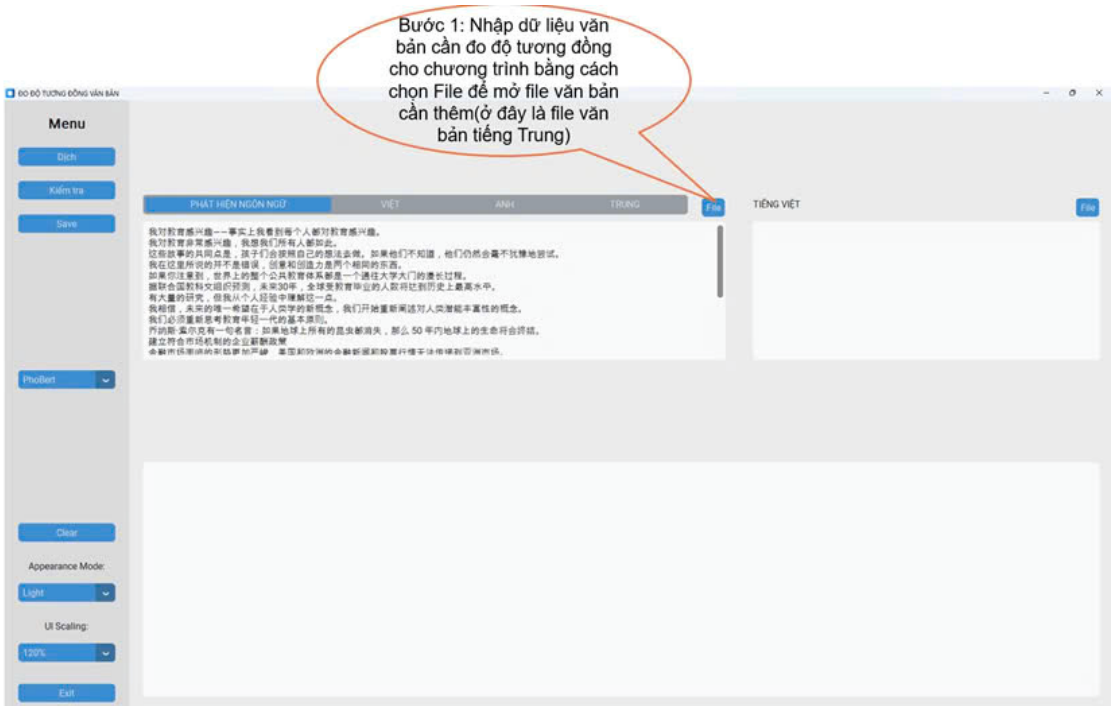


Hình 3.5: Các nút chức năng tùy chỉnh giao diện



Hình 3.6: Giao diện textbox truyền dữ liệu

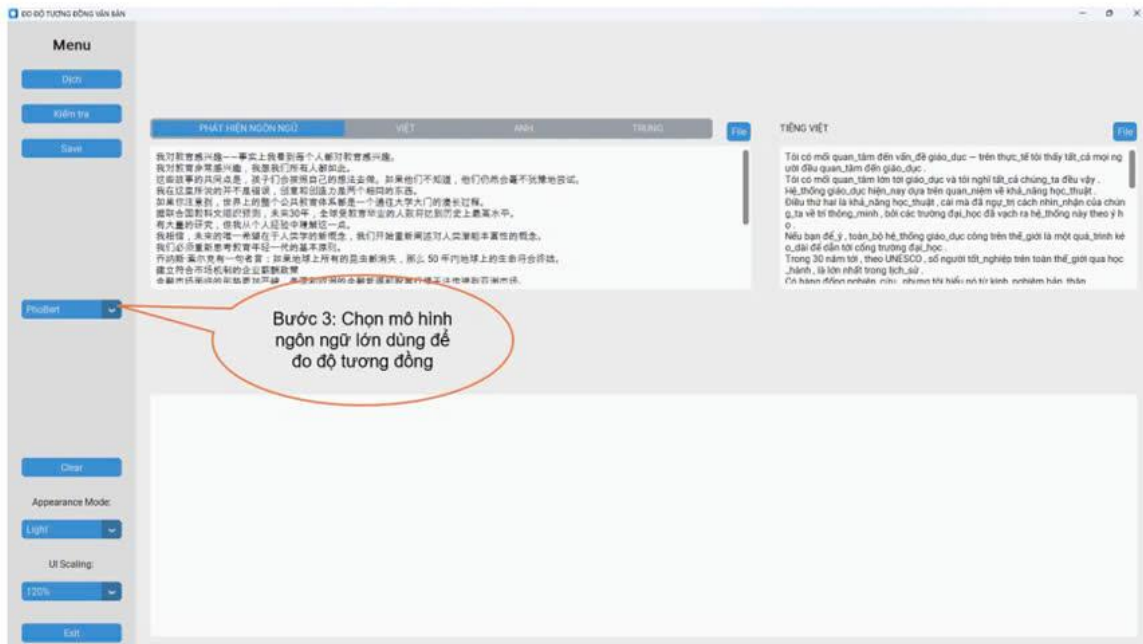
Các bước thực hiện:



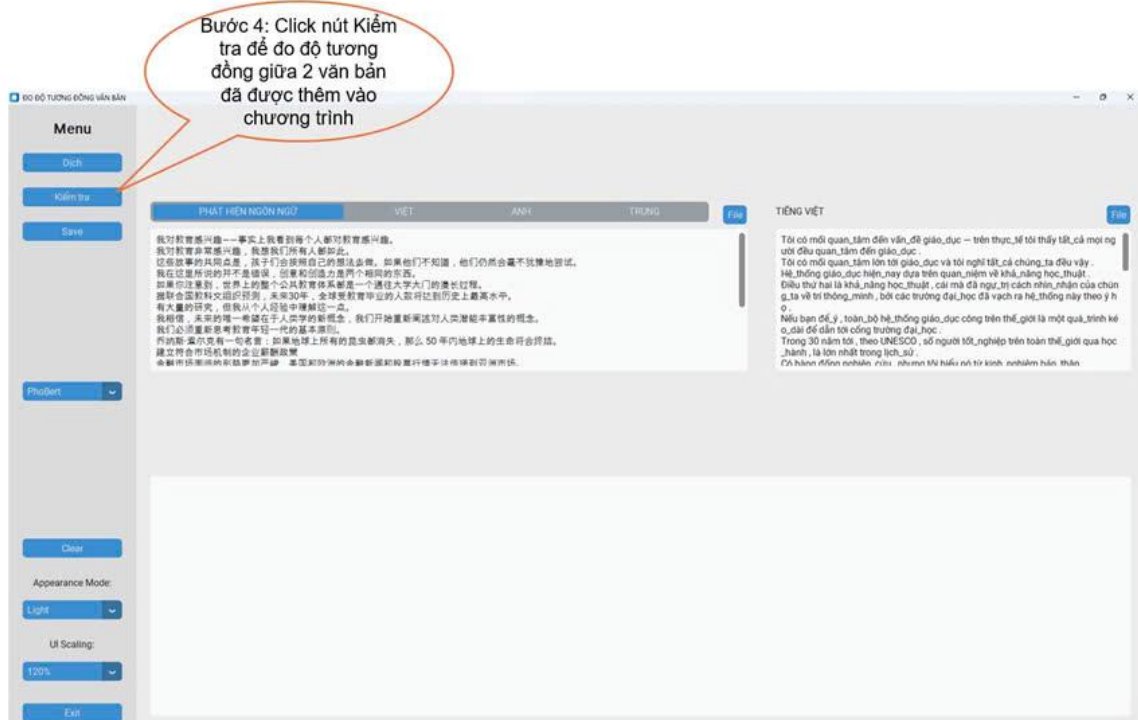
Hình 3.7: Nhập dữ liệu Tiếng Trung



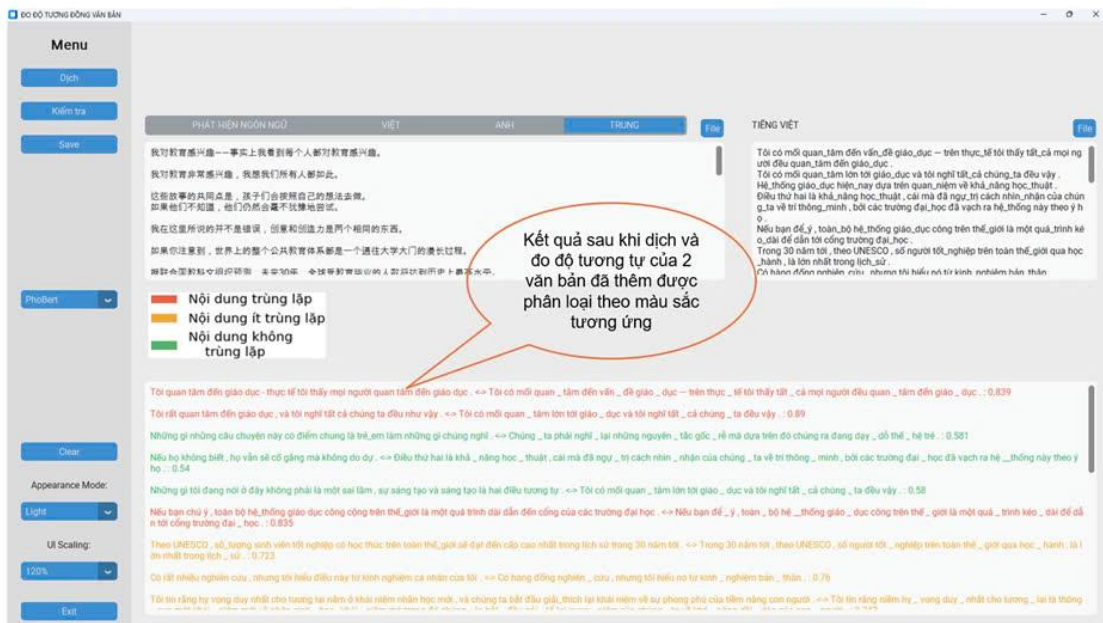
Hình 3.8: Nhập dữ liệu Tiếng Việt



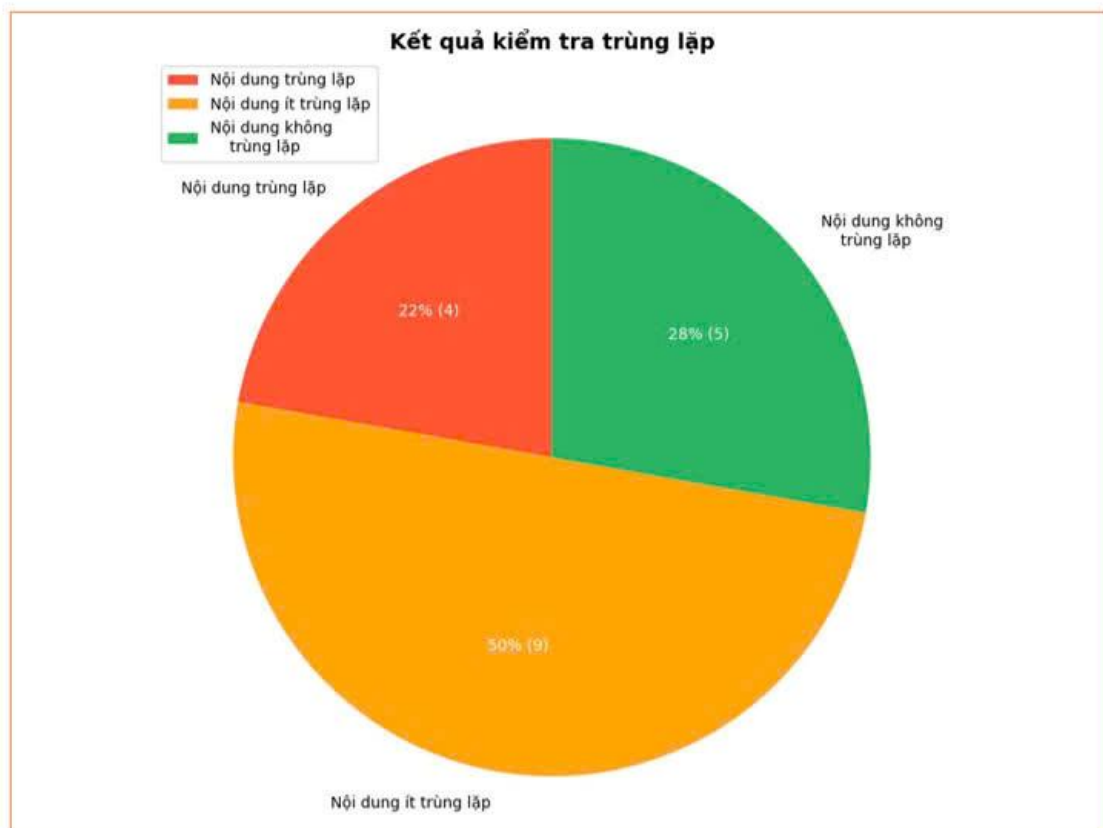
Hình 3.9: Chọn mô hình ngôn ngữ lớn để đo



Hình 3.10: Thực hiện đo lường



Hình 3.11: Kết quả đo lường



Hình 3.12: Trực quan hóa kết quả trên biểu đồ

So sánh giữa 3 mô hình PhoBERT, VisoBERT và SimCSE:



PhoBERT



VisoBERT



SimCSE

Kết quả thực nghiệm từ mô hình cho thấy rằng việc ứng dụng các mô hình ngôn ngữ lớn như PhoBERT, VisoBERT, SimCSE vào bài toán đo lường độ tương đồng ngữ nghĩa giữa văn bản song ngữ Việt – Trung mang lại kết quả tích cực.

3.3. ĐÁNH GIÁ MÔ HÌNH

3.3.1. Các phương pháp đánh giá năng suất mô hình trong bài toán xử lý ngôn ngữ tự nhiên

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), việc đánh giá năng suất của mô hình không chỉ đơn thuần dựa vào độ chính xác, mà còn cần đến các thước đo phản ánh được mức độ tương đồng, đồng thuận và tương quan giữa mô hình và dữ liệu thực tế. Đặc biệt trong các bài toán liên quan đến đo lường mức độ tương tự ngữ nghĩa, chấm điểm văn bản hoặc phân tích cảm xúc đa cấp độ, mô hình thường đưa ra kết quả dưới dạng điểm số liên tục hoặc thứ hạng tương đối. Khi đó, các chỉ số như hệ số tương quan và hệ số đồng thuận Kappa đóng vai trò trung tâm trong việc đánh giá hiệu quả của mô hình một cách toàn diện và khách quan hơn.

Một trong những nhóm chỉ số phổ biến là các hệ số tương quan, bao gồm Pearson, Spearman và Kendall. Pearson Correlation đo lường mức độ tương quan tuyến tính giữa hai biến liên tục, thường được sử dụng khi cả đầu ra của mô hình và nhãn thực đều ở dạng số thực. Tuy nhiên, trong nhiều trường hợp, dữ liệu ngôn ngữ không phân bố chuẩn và mối quan hệ giữa các biến không tuyến tính. Khi đó, hệ số tương quan bậc hai này có thể không phản ánh đúng mức độ phù hợp giữa mô hình và thực tế. Để khắc phục hạn chế trên, Spearman's Rank Correlation được sử dụng như một công cụ mạnh mẽ nhằm đo lường mức độ tương quan thứ hạng giữa hai dãy giá trị. Thay vì so sánh trực tiếp giá trị tuyệt đối, Spearman tập trung vào việc đánh giá xem mô hình có tái hiện được thứ tự sắp xếp tương đối mà con người đưa ra hay không. Điều này đặc biệt hữu ích trong các bài toán xếp hạng văn bản theo mức độ liên quan, đo độ giống nghĩa, hoặc đánh giá chất lượng tóm tắt.

Tương tự, Kendall's Tau cũng là một hệ số tương quan thứ hạng nhưng hoạt động theo nguyên lý so sánh từng cặp dữ liệu để xác định tỷ lệ cặp đồng thuận và nghịch thuận. Ưu điểm của Kendall là cung cấp cái nhìn sâu hơn về sự nhất quán cục bộ trong thứ hạng, đặc biệt hữu ích trong các tập dữ liệu nhỏ hoặc khi thứ hạng có sự trùng lặp. Mặc dù tính toán Kendall's Tau phức tạp hơn so với Spearman, nhưng đây lại là công cụ tin cậy trong đánh giá các hệ thống NLP có kết quả ở dạng thứ hạng mờ, chẳng hạn như sắp xếp kết quả tìm kiếm theo mức độ phù hợp ngữ cảnh.

Bên cạnh các hệ số tương quan, nhóm nghiên cứu cũng đặc biệt chú trọng đến các chỉ số đo lường mức độ đồng thuận, tiêu biểu là hệ số Kappa. Trong NLP, dữ liệu thường được gán nhãn bởi nhiều người với sự chủ quan không tránh khỏi. Việc đảm bảo rằng mô hình không chỉ chính xác, mà còn phản ánh đúng xu hướng đánh giá của con người, là một yêu cầu bắt buộc. Cohen's Kappa được sử dụng khi có hai đối tượng đánh giá, chẳng hạn như so sánh đầu ra của mô hình với một chuyên gia gán nhãn. Kappa điều chỉnh xác suất đồng thuận ngẫu nhiên, do đó cung cấp cái nhìn thực tế và công bằng hơn so với việc chỉ tính tỷ lệ chính xác đơn thuần. Giá trị Kappa nằm trong khoảng từ -1 đến 1, trong đó giá trị càng gần 1 thể hiện sự đồng thuận càng cao.

Tuy nhiên, để đánh giá bộ dữ liệu gán nhãn bởi nhiều người, Fleiss' Kappa được sử dụng như một phương pháp tổng quát hơn. Phương pháp này cho phép đánh giá mức độ đồng thuận giữa từ ba người trở lên, tính toán dựa trên tần suất lựa chọn từng nhãn trên mỗi mẫu. Fleiss' Kappa đặc biệt hữu ích trong giai đoạn xây dựng và kiểm định chất lượng tập dữ liệu trước khi đưa vào huấn luyện mô hình, cũng như trong các nghiên cứu đánh giá khả năng “suy nghĩ giống người” của mô hình NLP.

Việc lựa chọn chỉ số đánh giá phù hợp phụ thuộc vào đặc thù bài toán. Nếu đầu ra ở dạng số thực liên tục và có mối quan hệ tuyến tính với dữ liệu thật, Pearson là công cụ hợp lý. Ngược lại, trong các bài toán xếp hạng, Spearman và Kendall mang lại đánh giá khách quan hơn về độ tương đồng giữa mô hình và con người. Trong khi đó, các hệ số Kappa lại nhấn mạnh đến khía cạnh đồng thuận – yếu tố mang tính xã hội và chủ quan của ngôn ngữ – giúp đánh giá xem mô hình có đạt được mức độ “hiểu ngôn ngữ” tương tự như con người hay không. Việc kết hợp cả hai nhóm chỉ số sẽ mang đến cái nhìn toàn diện về hiệu quả của mô hình, không chỉ về mặt kỹ thuật mà còn về mức độ thích nghi với ngữ cảnh ngôn ngữ tự nhiên.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Trong đó x_i là giá trị dự đoán từ mô hình,

y_i là nhãn thực tế,

\bar{x} và \bar{y} lần lượt là trung bình cộng của hai chuỗi giá trị.

Giá trị của hệ số Pearson dao động từ -1 đến 1: giá trị gần 1 biểu thị mối tương quan thuận mạnh, gần -1 biểu thị mối tương quan nghịch mạnh, và giá trị gần 0 cho thấy không có mối quan hệ tuyến tính rõ ràng.

Lợi thế lớn nhất của Pearson nằm ở khả năng phản ánh mối quan hệ tuyến tính một cách nhạy bén và chính xác khi các giả định nền tảng được đảm bảo, bao gồm phân phối chuẩn của dữ liệu và tính tuyến tính giữa hai biến. Điều này khiến nó trở thành lựa chọn mặc định trong nhiều nghiên cứu, đặc biệt là khi muốn kiểm định chất lượng đầu ra của mô hình theo các tiêu chuẩn đo lường khách quan, chẳng hạn như thang điểm từ 0 đến 5 trong hệ thống chấm điểm văn bản.

Tuy nhiên, điểm yếu lớn của Pearson là tính nhạy cảm với phân phối dữ liệu và sự hiện diện của ngoại lệ (outliers). Trong xử lý ngôn ngữ tự nhiên, dữ liệu thực tế thường không phân bố chuẩn, mà bị ảnh hưởng bởi sự đa dạng ngữ nghĩa, cấu trúc ngữ pháp phong phú và cách dùng ngôn ngữ linh hoạt của con người. Ngoài ra, trong một số bài toán như phân tích cảm xúc hoặc đánh giá sự phù hợp ngữ nghĩa, mối quan hệ giữa đầu ra mô hình và nhãn thực có thể không tuân theo tính chất tuyến tính nghiêm ngặt. Khi đó, Pearson không còn là chỉ số tối ưu vì nó có thể đánh giá thấp những mô hình thực ra vẫn đang học đúng hướng nhưng theo dạng phi tuyến.

Để khắc phục những hạn chế trên, Pearson thường được kết hợp với các chỉ số tương quan thứ hạng như Spearman hoặc Kendall nhằm cung cấp góc nhìn toàn diện hơn. Dù vậy, trong những trường hợp dữ liệu đủ lớn, ít nhiễu, và đặc biệt là khi đầu ra mô hình mang bản chất tuyến tính – như khi dùng hồi quy để dự đoán điểm số – thì Pearson vẫn giữ vai trò là một chuẩn mực đánh giá cơ bản, góp phần định lượng mức độ khớp giữa mô hình và thực tế trong các ứng dụng NLP hiện đại.

3.3.2. Spearman và Kendall: Đánh giá thứ hạng và mức độ tương tự

Trong nhiều bài toán xử lý ngôn ngữ tự nhiên, đầu ra của mô hình không chỉ đơn thuần là một giá trị liên tục, mà thường phản ánh một thứ hạng, một mức độ ưu tiên hoặc độ tương đồng tương đối giữa các thực thể ngôn ngữ. Ví dụ điển hình bao gồm xếp hạng câu trả lời trong hệ thống hỏi đáp, sắp xếp đoạn văn theo mức độ liên quan đến một truy vấn, hay đánh giá mức độ tương tự ngữ nghĩa giữa hai văn bản. Trong các trường hợp này, mô hình NLP được yêu cầu tái hiện đúng thứ tự mà con

người cảm nhận được thay vì chỉ trùng khớp về mặt số học. Khi đó, các hệ số tương quan thứ hạng như Spearman's rho và Kendall's tau trở thành công cụ đánh giá phù hợp và mạnh mẽ hơn so với Pearson.

3.3.2.1. Spearman's rho

Spearman's rho ¹⁰ là một hệ số tương quan không tham số (non-parametric), được sử dụng để đánh giá mối quan hệ đơn điệu giữa hai biến bằng cách so sánh thứ hạng của chúng thay vì giá trị tuyệt đối. Điều này đặc biệt quan trọng trong NLP vì đầu ra của mô hình đôi khi không cần chính xác về mặt số lượng, mà chỉ cần đúng về mặt thứ tự. Phương pháp này đầu tiên chuyển đổi hai dãy giá trị thành các thứ hạng, sau đó tính Pearson Correlation giữa các thứ hạng đó.

Công thức tính Spearman như sau:

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Trong đó: d_i là hiệu giữa hai thứ hạng tương ứng

n là số cặp quan sát.

Giá trị p dao động từ -1 đến 1

Với giá trị gần 1 thể hiện rằng mô hình giữ được trật tự giống như con người đánh giá.

Ưu điểm nổi bật của Spearman là khả năng đánh giá chính xác độ phù hợp thứ hạng trong các trường hợp có mối quan hệ đơn điệu phi tuyến – một đặc trưng thường gặp trong bài toán NLP thực tế. Hơn nữa, vì Spearman không yêu cầu phân phối chuẩn và không bị ảnh hưởng nhiều bởi ngoại lệ, nó trở thành công cụ đáng tin cậy khi làm việc với dữ liệu ngôn ngữ có độ biến thiên lớn và giàu tính biểu cảm.

¹⁰ <https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide-2.php>

3.3.2.2 Kendall's Tau

Khác với Spearman, Kendall's tau ¹¹ đo lường mức độ đồng thuận giữa hai thứ hạng bằng cách so sánh tất cả các cặp phần tử. Hai cặp được coi là *đồng thuận* nếu thứ tự của chúng trong cả hai danh sách giống nhau, và *ngược thuận* nếu thứ tự bị đảo ngược. Hệ số Kendall được tính dựa trên tỷ lệ giữa số cặp đồng thuận và tổng số cặp so sánh được.

Công thức cơ bản của Kendall's tau:

$$\tau = \frac{(C - D)}{\frac{1}{2}n(n - 1)}$$

Trong đó: C là số cặp đồng thuận,

D là số cặp ngược thuận, và n là số quan sát.

Giá trị của τ cũng nằm trong khoảng $[-1, 1]$, tương tự Spearman.

So với Spearman, Kendall's tau thường được đánh giá là ổn định hơn, đặc biệt trong những tập dữ liệu nhỏ hoặc khi thứ hạng có nhiều giá trị trùng lặp. Nhờ vào cách tiếp cận dựa trên từng cặp quan sát, Kendall có thể cung cấp một cái nhìn chi tiết hơn về mức độ nhất quán cục bộ trong sắp xếp – điều mà Spearman có thể bỏ sót nếu chỉ nhìn vào sự khác biệt thứ hạng toàn cục.

3.3.2.3. So sánh và Ứng dụng trong NLP

Cả Spearman và Kendall đều hướng tới việc đánh giá năng suất mô hình dựa trên khả năng “hiểu thứ tự” – một khía cạnh rất quan trọng trong các ứng dụng NLP như tìm kiếm thông tin, gợi ý nội dung và phân tích cảm xúc theo mức độ. Trong các nghiên cứu học thuật, Spearman thường được sử dụng phổ biến hơn do dễ tính toán và tích hợp vào pipeline huấn luyện. Tuy nhiên, khi yêu cầu đánh giá độ ổn định hoặc đối mặt với dữ liệu có cấu trúc phức tạp, Kendall lại là lựa chọn ưu việt hơn.

Việc lựa chọn giữa Spearman và Kendall tùy thuộc vào quy mô dữ liệu, tính chất thứ hạng và mức độ nhạy cảm mong muốn trong đánh giá. Trong thực tiễn, nhiều hệ thống đánh giá mô hình hiện đại sử dụng cả hai hệ số song song để đảm bảo đánh

¹¹ <https://datatab.net/tutorial/kendalls-tau>

giá toàn diện, vừa theo góc nhìn tổng quát (Spearman), vừa theo góc nhìn cục bộ (Kendall).

3.3.3. Hệ số Kappa – Đo lường mức độ đồng thuận giữa mô hình và con người

Trong các bài toán xử lý ngôn ngữ tự nhiên, đặc biệt là những bài toán mang tính chất phân loại như phân tích cảm xúc, xác định chủ đề, hoặc chấm điểm văn bản, việc đánh giá chất lượng mô hình không chỉ dừng lại ở mức độ trùng khớp với nhãn mà còn cần phản ánh được mức độ đồng thuận với cách con người đánh giá. Ngôn ngữ tự nhiên vốn dĩ giàu tính chủ quan và bối cảnh, do đó trong nhiều trường hợp, cùng một văn bản có thể được diễn giải theo những cách khác nhau bởi các chuyên gia ngôn ngữ khác nhau. Chính vì vậy, các chỉ số như Kappa được đưa vào để đo lường mức độ nhất quán và đồng thuận – không phải giữa mô hình và một “sự thật tuyệt đối”, mà giữa mô hình và cách con người hiểu ngôn ngữ trong thực tế.

3.3.3.1 Cohen's Kappa¹²

Cohen's Kappa là chỉ số đo lường sự đồng thuận giữa hai đối tượng phân loại (thường là mô hình và con người, hoặc hai người gán nhãn) khi gán nhãn cho cùng một tập dữ liệu. Điểm đặc biệt của chỉ số này nằm ở việc điều chỉnh xác suất đồng thuận do ngẫu nhiên, từ đó mang lại một cái nhìn thực tế hơn về mức độ nhất quán thực sự. Nếu một mô hình ngẫu nhiên đoán nhãn và tình cờ trùng với nhãn thật trong một số trường hợp, thì tỷ lệ chính xác (accuracy) đơn thuần có thể đánh lừa người dùng. Cohen's Kappa loại bỏ hiệu ứng này bằng cách so sánh tỷ lệ đồng thuận quan sát được (observed agreement) với tỷ lệ đồng thuận kỳ vọng (expected agreement by chance).

Công thức của Cohen's Kappa được định nghĩa như sau:

$$k = \frac{P_o - P_e}{1 - P_e}$$

Trong đó: P_o là xác suất đồng thuận thực tế,

P_e là xác suất đồng thuận kỳ vọng.

¹² <https://datatab.net/tutorial/cohens-kappa>

Giá trị Kappa nằm trong khoảng từ -1 đến 1

Với những bài toán NLP có đầu ra là nhãn rời rạc như “tích cực – trung tính – tiêu cực” hoặc “giỏi – khá – trung bình – yếu”, Cohen’s Kappa là một chỉ số lý tưởng giúp xác định liệu mô hình có “suy nghĩ” giống như người thật hay không, vượt ra ngoài sự trùng khớp đơn giản.

3.3.3.2. Fleiss’ Kappa¹³

Trong thực tế xây dựng bộ dữ liệu NLP, việc gán nhãn thường được thực hiện bởi nhiều người chứ không chỉ hai. Khi đó, Cohen’s Kappa không còn đủ để phản ánh mức độ đồng thuận tổng thể. Fleiss’ Kappa là sự mở rộng tự nhiên của Cohen’s Kappa dành cho trường hợp nhiều người đánh giá (lớn hơn 2), và đặc biệt không yêu cầu cùng một cặp người gán nhãn cho tất cả các mẫu. Đây là điểm khác biệt quan trọng giúp Fleiss’ Kappa trở nên linh hoạt và phù hợp hơn trong các dự án chú thích ngữ liệu quy mô lớn, nơi mỗi câu chỉ được gán nhãn bởi một tập con người gán.

Fleiss’ Kappa đo lường mức độ đồng thuận tổng thể giữa các annotator trên toàn bộ dữ liệu, bằng cách so sánh tỷ lệ đồng thuận quan sát được và tỷ lệ đồng thuận kỳ vọng xảy ra do ngẫu nhiên. Kết quả đánh giá được chuẩn hóa về thang đo $[-1, 1]$, với ý nghĩa tương tự như Cohen’s Kappa: giá trị càng gần 1 biểu thị mức độ đồng thuận cao, gần 0 biểu thị mức đồng thuận tương đương ngẫu nhiên, và giá trị âm cho thấy sự bất đồng vượt ngẫu nhiên.

Về mặt toán học, giả sử có N đối tượng (câu, văn bản), mỗi đối tượng được gán nhãn bởi n annotator, và có tổng cộng k nhãn có thể chọn. Gọi n_{ij} là số người gán nhãn đối tượng thứ i với nhãn thứ j . Khi đó, Fleiss’ Kappa được tính qua các bước sau:

Trước tiên, tính mức độ đồng thuận cho từng đối tượng:

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^k n_{ij}(n_{ij} - 1)$$

Tiếp theo, tính tỷ lệ đồng thuận trung bình toàn bộ tập:

¹³ <https://datatab.net/tutorial/fleiss-kappa>

$$\bar{P} = \frac{1}{N} \sum_{i=1}^N P_i$$

Tính tỷ lệ chọn từng nhãn trên toàn bộ dữ liệu:

$$P_j = \frac{1}{Nn} \sum_{i=1}^N n_{ij}$$

Từ đó, xác suất đồng thuận do ngẫu nhiên:

$$\bar{P}_e = \sum_{j=1}^k P_j^2$$

Và cuối cùng, Fleiss' Kappa được tính bằng:

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

Một điểm mạnh đáng chú ý của Fleiss' Kappa là khả năng phát hiện sự thiếu nhất quán trong quy trình gán nhãn dữ liệu – một yếu tố ảnh hưởng trực tiếp đến chất lượng đầu vào của mô hình NLP. Nếu giá trị Kappa thấp, điều đó có thể phản ánh sự mơ hồ của dữ liệu, sự thiếu thống nhất trong tiêu chí đánh giá, hoặc sự khác biệt về chuyên môn giữa các annotator. Trong các nghiên cứu hiện đại, Fleiss' Kappa còn được mở rộng để đánh giá mức độ "đồng thuận giống con người" của mô hình ngôn ngữ bằng cách so sánh đầu ra của mô hình với phân phối lựa chọn trung bình của con người, từ đó kiểm tra mức độ mô hình có tái hiện được cách con người hiểu và phân loại ngôn ngữ hay không.

3.3.3.3. Ý nghĩa trong đánh giá mô hình NLP

Hệ số Kappa đóng vai trò như một cầu nối giữa độ chính xác kỹ thuật và tính hợp lý xã hội trong các hệ thống xử lý ngôn ngữ tự nhiên. Khác với giả định truyền thống cho rằng nhãn đúng là một chuẩn mực tuyệt đối, Kappa cho phép nhóm nghiên

cứu tiếp cận ngôn ngữ như một hiện tượng xã hội – nơi mà cách hiểu và đánh giá thường mang tính chủ quan và phụ thuộc vào bối cảnh. Do đó, thay vì chỉ đo mức độ khớp đơn thuần giữa đầu ra của mô hình và nhãn gán, Kappa giúp đánh giá mức độ tương thích về mặt nhận thức giữa mô hình và con người.

Điều này đặc biệt quan trọng trong các bài toán như mô hình sinh ngôn ngữ, phân tích cảm xúc đa cấp độ hoặc chấm điểm tự động – nơi không có một đáp án tuyệt đối duy nhất. Trong các tình huống này, việc mô hình đạt được đồng thuận cao với đánh giá của con người không chỉ phản ánh hiệu suất kỹ thuật, mà còn thể hiện mức độ "thấu cảm ngôn ngữ" – tức năng lực nắm bắt ý nghĩa và cảm xúc trong ngữ cảnh giống như con người.

Một cách tiếp cận đánh giá hiện đại là kết hợp Kappa với các hệ số tương quan thứ hạng như Spearman hoặc Kendall. Sự kết hợp này tạo ra một hệ thống đánh giá hai chiều: vừa đo lường tính logic và nhất quán trong sắp xếp (dựa trên tương quan thứ hạng), vừa kiểm tra khả năng đồng cảm và phù hợp với chuẩn mực xã hội (dựa trên mức độ đồng thuận). Đây là một xu hướng quan trọng trong việc phát triển các mô hình NLP có tính nhân văn và phù hợp với thực tiễn giao tiếp ngôn ngữ đa dạng.

Diễn giải giá trị Kappa theo ngữ cảnh

Giá trị κ	Mức độ đồng thuận
< 0.00	Không có đồng thuận
$0.00-0.20$	Rất thấp
$0.21-0.40$	Thấp
$0.41-0.60$	Trung bình
$0.61-0.80$	Cao
$0.81-1.00$	Rất cao

Bảng 3.1: Diễn giải giá trị Kappa theo ngữ cảnh

Tuy nhiên, các mức diễn giải này không nên áp dụng một cách cứng nhắc, mà cần được điều chỉnh tùy theo bài toán cụ thể. Ví dụ, trong bài toán phân tích cảm xúc đa cấp độ (negative/neutral/positive), giá trị $\kappa=0.6$ đã có thể được xem là mức đồng thuận cao, trong khi với một bài toán nhị phân đơn giản hơn, cùng giá trị này chỉ thể hiện mức đồng thuận trung bình. Do đó, nhóm nghiên cứu cần linh

hoạt trong việc đọc hiểu và áp dụng chỉ số Kappa, tùy vào ngữ cảnh và đặc điểm ngôn ngữ của từng tác vụ.

3.3.3.4. Vai trò của Kappa trong đánh giá năng suất mô hình NLP

Hệ số Kappa không chỉ đóng vai trò như một chỉ số đánh giá năng suất mô hình từ góc nhìn kỹ thuật, mà còn góp phần phản ánh tính ổn định và độ tin cậy xã hội của mô hình trong các tác vụ có yếu tố chủ quan cao. Trong những ứng dụng thực tế như hệ thống chấm điểm tự động, chatbot tư vấn, hay trợ lý ngôn ngữ cá nhân hóa, mô hình không chỉ cần đúng mà còn cần phản ánh được cách con người hiểu và xử lý ngôn ngữ. Khi đó, Kappa trở thành một chỉ báo đạo đức và kỹ thuật quan trọng, giúp xác định liệu mô hình có thể được triển khai trong môi trường thực tế mà không gây ra những phản ứng tiêu cực về sự thiếu "nhân tính".

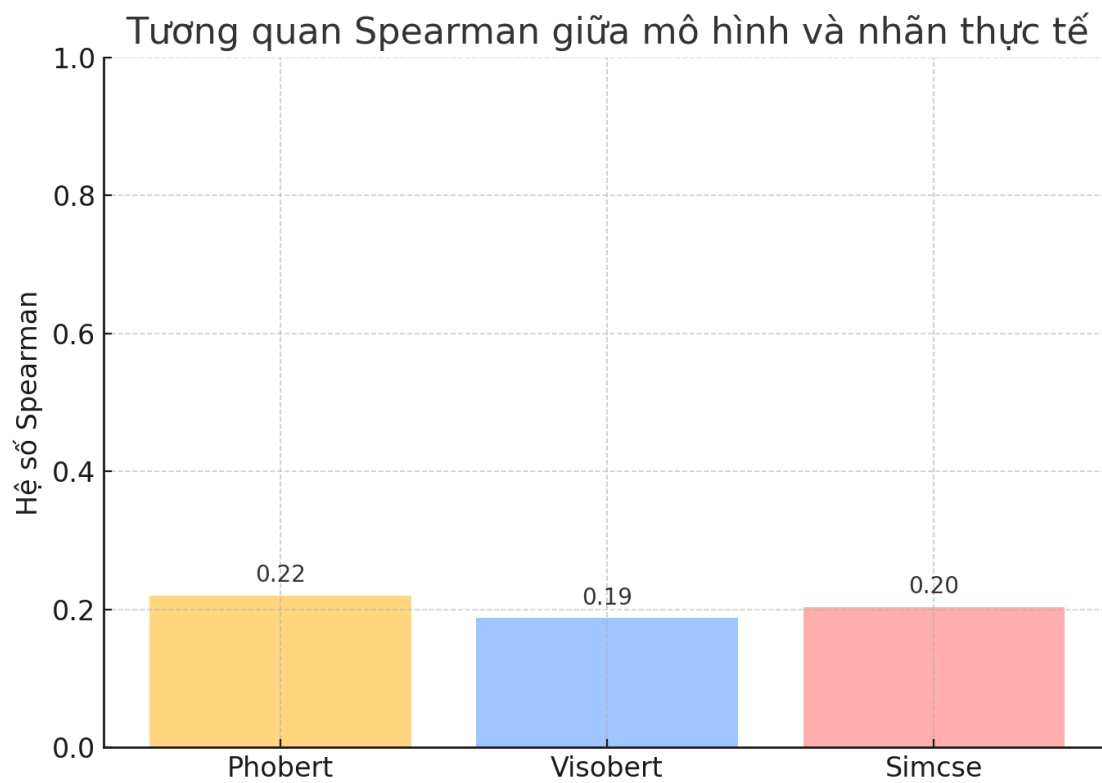
Bên cạnh đó, Kappa còn là một công cụ hữu hiệu trong giai đoạn kiểm định bộ dữ liệu huấn luyện, đặc biệt khi có nhiều annotator tham gia gán nhãn. Một giá trị Kappa thấp có thể phản ánh sự không thống nhất trong cách hiểu và định nghĩa về các nhãn – từ đó đặt ra nhu cầu phải tái cấu trúc tiêu chí gán nhãn hoặc đào tạo lại annotator. Khi bộ dữ liệu huấn luyện có độ đồng thuận cao, mô hình được huấn luyện trên đó cũng có khả năng tổng quát hóa tốt hơn và ổn định hơn khi gặp dữ liệu mới.

Cuối cùng, việc sử dụng Kappa song song với các hệ số tương quan như Pearson, Spearman hay Kendall không chỉ cho thấy khả năng tái tạo thứ hạng hay độ tương tự giữa các biểu diễn ngữ nghĩa, mà còn giúp kiểm tra xem mô hình có đang bắt chước cách con người đưa ra quyết định hay không. Chính vì vậy, trong hệ sinh thái đánh giá mô hình NLP hiện đại, hệ số Kappa giữ một vị trí đặc biệt – là điểm giao nhau giữa độ chính xác, tính nhân văn và độ tin cậy xã hội.

3.3.4. Đánh giá

Để so sánh chúng tôi sử dụng độ đo tương quan Spearman giữa độ tương đồng thực tế và các mô hình:

$$r = 1 - \frac{6 \sum D^2}{n(n^2 - 1)}$$



Hình 3.13: Đồ thị so sánh độ tương quan giữa 3 mô hình bằng độ đo Spearman

KẾT LUẬN VÀ KIẾN NGHỊ

Trong thời đại hội nhập và phát triển mạnh mẽ của công nghệ trí tuệ nhân tạo, việc nghiên cứu và ứng dụng các mô hình ngôn ngữ lớn vào xử lý văn bản song ngữ đang trở thành một xu thế tất yếu, đặc biệt trong bối cảnh nhu cầu tương tác và trao đổi thông tin giữa các nền ngôn ngữ ngày càng gia tăng. Đề tài “*Ứng dụng mô hình ngôn ngữ lớn đo lường độ tương tự của văn bản song ngữ Việt – Trung*” đã góp phần cụ thể hóa hướng tiếp cận này thông qua việc nghiên cứu, xây dựng và đánh giá một hệ thống có khả năng đo lường chính xác mức độ tương đồng ngữ nghĩa giữa các văn bản tiếng Việt và tiếng Trung.

Trong quá trình thực hiện, nhóm nghiên cứu đã:

- Tiếp cận và tổng hợp cơ sở lý thuyết về mạng nơ-ron học sâu và các mô hình ngôn ngữ lớn như BERT, PhoBERT, VisoBERT, SimCSE;
- Khảo sát và áp dụng các phương pháp đo lường độ tương đồng văn bản dựa trên biểu diễn vector ngữ nghĩa;
- Triển khai thực nghiệm với nhiều cặp văn bản song ngữ thực tế, đánh giá hiệu quả của mô hình qua các chỉ số định lượng;
- So sánh với các phương pháp truyền thống, qua đó rút ra ưu điểm nổi bật của hướng tiếp cận sử dụng LLMs.

Kết quả nghiên cứu cho thấy mô hình ngôn ngữ lớn, đặc biệt là các mô hình được tiền huấn luyện cho tiếng Việt và ứng dụng kỹ thuật sentence embedding, có khả năng đo lường ngữ nghĩa chính xác và ổn định hơn đáng kể so với các phương pháp truyền thống. Từ đó, đề tài mở ra tiềm năng ứng dụng mạnh mẽ trong các lĩnh vực như dịch máy, tìm kiếm tài liệu đa ngôn ngữ, xây dựng hệ thống hỗ trợ học tập và nghiên cứu song ngữ.

Trong tương lai, nhóm nghiên cứu mong muốn:

- Mở rộng quy mô và độ đa dạng của bộ dữ liệu song ngữ;
- Ứng dụng các mô hình ngôn ngữ đa ngữ như XLM-R, mBERT để tăng khả năng khái quát hóa;

- Xây dựng hệ thống đánh giá chất lượng bản dịch theo ngữ cảnh ngữ nghĩa thay vì chỉ dựa vào đối sánh từ vựng;
- Phát triển một sản phẩm demo hoàn chỉnh có thể áp dụng trong thực tiễn như hỗ trợ người dùng tra cứu hoặc kiểm định bản dịch tự động.

Thông qua nghiên cứu này, nhóm đề tài hy vọng có thể đóng góp một phần nhỏ vào tiến trình phát triển các hệ thống xử lý ngôn ngữ tự nhiên thông minh, hỗ trợ tốt hơn việc giao tiếp và trao đổi tri thức giữa các ngôn ngữ khác nhau.

TÀI LIỆU THAM KHẢO

- [1]. Aliguliyev, R.M.: A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Syst. Appl.* 36(4), (2009).
- [2]. Burke, R., Hammond, K., Kulyukin, V., Tomuro, S.: Question Answering from Frequently Asked Question Files. *AI Magazine* 18(2), 57-66 (1997).
- [3]. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41(6), 391-407 (1990)
- [4]. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. pp. 4171-4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019).
5. Farouk, M., Ishizuka, M., Bollegala, D.: Graph matching based semantic search engine. In: Garoufallou, E., Sartori, F., Siatri, R., Zervas, M. (eds.) *MTSR. Communications in Computer and Information Science*, vol. 846, pp. 89-100. Springer (2018).
6. Ferreira, R., Lins, R.D., Simske, S.J., Freitas, F., Riss, M.: Assessing sentence similarity through lexical, syntactic and semantic analysis. *Comput. Speech Lang.* 39, 1-28 (2016).
7. Heo, T.S., Kim, J.D., Park, C.Y., Kim, Y.S.: Global and local information adjustment for semantic similarity evaluation. *Applied Sciences* 11(5) (2021).

8. Lee, M.C., Chang, J.W., Hsieh, T.C.: A grammar-based semantic similarity algorithm for natural language sentences. *The Scientific World Journal* 2014, 17 (2014).
9. Lee, M.C., Zhang, J.W., Lee, W.X., Ye, H.Y.: Sentence similarity computation based on pos and semantic nets. In: Kim, J., Delen, D., Park, J., Ko, F., Rui, C., Lee, J.H., Wang, J., Kou, G. (eds.) *NCM*. pp. 907-912. IEEE Computer Society (2009).
10. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development* 1, 309-317 (1957).
11. Manning, C.D., MacCartney, B.: *Natural language inference* (2009).
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013).
13. Morris, A.C., Maier, V., Green, P.D.: From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In: *INTERSPEECH. ISCA* (2004).
14. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Schuurmans, D., Wellman, M.P. (eds.) *AAAI*. pp. 2786-2792. AAAI Press (2016).
15. Nguyen, D.Q., Nguyen, A.T.: Phobert: Pre-trained language models for vietnamese. In: Cohn, T., He, Y., Liu, Y. (eds.) *EMNLP (Findings)*. pp.1037-1042. Association for Computational Linguistics (2020).
16. Nguyen, P.T., Pham, V.L., Nguyen, H.A., Vu, H.H., Tran, N.A., Truong, T.T.H.: A two-phase approach for building vietnamese wordnet. *the 8th Global Wordnet Conference* pp. 259-264 (2015).
17. Nguyen, T.M.H., Romary, L., Rossignol, M., Vu, X.L.: A lexicon for vietnamese language processing. *Language Resources and Evaluation* 40(3-4), 291-309 (2006).
18. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *EMNLP*. vol. 14, pp. 1532-1543 (2014)

19. Wang, Z., Mi, H., Ittycheriah, A.: Sentence similarity learning by lexical decomposition and composition. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) COLING. pp. 1340-1349. ACL (2016).
20. Yang, M., Wang, R., Chen, K., Utiyama, M., Sumita, E., Zhang, M., Zhao, T.: Sentence-level agreement for neural machine translation. In: Korhonen, A., Traum, D.R., M_arquez, L. (eds.) ACL (1). pp. 3076-3082. Association for Computational Linguistics (2019).
21. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
22. M. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.
23. A. Vaswani *et al.*, “Attention is All You Need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
24. T. Gao, X. Yao, and D. Chen, “SimCSE: Simple Contrastive Learning of Sentence Embeddings,” *Proceedings of EMNLP*, 2021.
25. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
26. T. Wolf *et al.*, “Transformers: State-of-the-Art Natural Language Processing,” *EMNLP*, 2020.