**BÁO CÁO PHÂN TÍCH CHUYÊN SÂU: CÁC MÔ HÌNH BIỂU DIỄN TỪ NGỮ CẢNH HÓA NÂNG CAO (ADVANCED CONTEXTUALIZED WORD EMBEDDINGS)**

**I. GIỚI THIỆU: KHỦNG HOẢNG TÍNH ĐA NGHĨA VÀ SỰ CHUYỂN ĐỔI MÔ HÌNH**

**1.1. Bối Cảnh và Hạn Chế của Biểu Diễn Từ Tĩnh**

Lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) đã chứng kiến một sự thay đổi căn bản trong cách các từ được biểu diễn dưới dạng số. Các mô hình nền tảng như Word2Vec và GloVe đóng vai trò quan trọng trong việc thiết lập tiêu chuẩn cho các biểu diễn từ, hay còn gọi là *word embeddings*. Các mô hình này hoạt động dựa trên nguyên tắc phân phối (Distributional Hypothesis), cho rằng các từ xuất hiện trong ngữ cảnh tương tự có xu hướng mang ý nghĩa tương tự.

Tuy nhiên, Word2Vec và GloVe tạo ra các biểu diễn từ *cố định* (static).1 Điều này có nghĩa là mỗi từ trong bộ từ vựng chỉ được mã hóa thành một vector duy nhất, bất kể ngữ cảnh cụ thể mà từ đó xuất hiện trong văn bản.2 Vector cố định này cố gắng kết hợp tất cả các ý nghĩa và sắc thái khác nhau của từ đó vào một biểu diễn số duy nhất.2

Hạn chế cốt lõi của phương pháp tĩnh này nằm ở việc nó không thể nắm bắt được sự phức tạp và tính đa sắc thái ngữ nghĩa của ngôn ngữ tự nhiên. Việc gán một vector cố định cho một từ thất bại trong việc giải quyết vấn đề từ đa nghĩa (polysemy) và sự mơ hồ ngữ nghĩa (semantic ambiguity), từ đó giới hạn hiệu suất của các hệ thống NLP trong các tác vụ phức tạp.

**1.2. Tính Đa Nghĩa (Polysemy) và Nhu cầu về Ngữ Cảnh Hóa**

Tính đa nghĩa (Polysemy) là hiện tượng phổ biến, trong đó một từ có nhiều nghĩa khác nhau tùy thuộc vào ngữ cảnh xuất hiện.3 Ví dụ kinh điển là từ "bank," có thể mang nghĩa là "bờ sông" (*river bank*) hoặc "ngân hàng tín dụng" (*credit bank*).4 Trong các mô hình tĩnh như Word2Vec, cả hai lần xuất hiện của từ "bank" trong các ngữ cảnh khác nhau này đều được mã hóa thành cùng một vector. Điều này buộc mô hình downstream phải tự giải quyết sự mơ hồ đó, dẫn đến sai sót trong việc phân tích ngữ nghĩa.

Mục tiêu chính của các mô hình Nhúng Ngữ cảnh Hóa (Contextualized Embeddings) là khắc phục hạn chế này bằng cách tạo ra các vector nhúng *động*. Các mô hình này có khả năng sinh ra các vector khác nhau cho cùng một từ tùy thuộc vào vị trí và ngữ cảnh của nó trong câu hoặc văn bản.1 Ví dụ, từ "apple" sẽ có hai vector khác nhau khi xuất hiện trong câu "I ate an apple" (trái cây) và câu "Apple releases iPhone" (công ty công nghệ).4 Sự linh hoạt này giúp các mô hình NLP hiểu rõ hơn ý nghĩa chính xác của từ theo từng ngữ cảnh cụ thể, từ đó giảm đáng kể sự mơ hồ ngữ nghĩa và nâng cao độ chính xác trong các tác vụ như phân loại văn bản, trích xuất thông tin, hoặc hỏi đáp.4 Sự chuyển đổi từ biểu diễn tĩnh sang biểu diễn động này đánh dấu bước tiến quan trọng nhất trong NLP hiện đại.

**II. NỀN TẢNG KỸ THUẬT: CÔNG CỤ XỬ LÝ VỐN TỪ (TOKENIZATION)**

**2.1. Vấn đề Từ Vựng Ngoài Từ Điển (OOV)**

Trước khi các mô hình từ tĩnh có thể được huấn luyện, văn bản cần được xử lý thành các đơn vị cơ bản, hay còn gọi là *token*. Tokenization truyền thống, đặc biệt là tokenization cấp độ từ (word-based), hoạt động bằng cách chia văn bản dựa trên khoảng trắng và so sánh các từ với một từ điển cố định (vocabulary).

Vấn đề nghiêm trọng nhất của phương pháp này là Từ Vựng Ngoài Từ Điển (Out-Of-Vocabulary - OOV).6 Khi mô hình gặp một từ mới (ví dụ: một từ ghép, một tên riêng chưa từng thấy, hoặc một biến thể chính tả mới), nó không thể gán một vector nhúng cụ thể cho từ đó. Thay vào đó, nó buộc phải sử dụng một token [UNK] (Unknown) chung chung, dẫn đến việc mất mát thông tin ngữ nghĩa và cú pháp quan trọng. Các mô hình nền tảng như Word2Vec, vốn phụ thuộc vào tokenization cấp độ từ, dễ gặp phải vấn đề OOV này.6

**2.2. Sự Đột Phá của Tokenization Cấp Độ Subword**

Để giải quyết hiệu quả vấn đề OOV và đồng thời kiểm soát kích thước bộ từ vựng, các mô hình ngữ cảnh hóa hiện đại đã chuyển sang sử dụng Subword Tokenization (token hóa cấp độ subword). Phương pháp này đạt được sự cân bằng tối ưu: đơn vị token lớn hơn một ký tự nhưng nhỏ hơn một từ hoàn chỉnh.6 Bằng cách này, ngay cả những từ phức tạp hoặc chưa từng thấy cũng có thể được phân tách thành các subword có ý nghĩa đã có trong từ điển.

Các thuật toán subword phổ biến bao gồm Byte-Pair Encoding (BPE) và WordPiece. BPE hoạt động bằng cách thống kê và hợp nhất các cặp ký tự hoặc subword xuất hiện thường xuyên nhất trong ngữ liệu.7 WordPiece Tokenization, được sử dụng rộng rãi trong các mô hình Transformer như BERT, là một phiên bản cải tiến của BPE. WordPiece dựa trên xác suất và điểm số để quyết định cặp subword nào nên được hợp nhất, thay vì chỉ dựa vào tần suất đơn thuần.8 Quá trình này tiếp diễn cho đến khi đạt được kích thước từ vựng mong muốn, giúp mô hình tối ưu hóa khả năng biểu diễn ngữ nghĩa đồng thời giữ vững khả năng xử lý các từ OOV.

Việc áp dụng Subword Tokenization là nền tảng kỹ thuật thiết yếu cho sự thành công của các mô hình ngữ cảnh hóa quy mô lớn, vì nó cho phép các mô hình này mở rộng bộ từ vựng hiệu quả và xử lý các từ hiếm mà không làm phình to ma trận nhúng.

**2.3. Sự Khác Biệt Chiến Lược trong Xử Lý OOV**

Trong sự phát triển của Contextualized Embeddings, đã có một sự khác biệt đáng chú ý trong cách các mô hình tiên phong xử lý OOV. Mô hình ELMo đã áp dụng đầu vào cấp độ ký tự (character-level inputs) thông qua phép tích chập ký tự (character convolutions).2 Cách tiếp cận này mang lại tính linh hoạt cực kỳ cao đối với OOV, vì bất kỳ từ nào, dù là từ mới hay từ hiếm, đều có thể được xây dựng từ các ký tự cơ bản.

Ngược lại, các mô hình dựa trên Transformer như BERT đã áp dụng Subword Tokenization (WordPiece).2 Sự khác biệt này không chỉ là một lựa chọn kỹ thuật ngẫu nhiên mà còn phản ánh sự đánh đổi kiến trúc và tính toán. Hệ thống dựa trên ký tự (ELMo/Bi-LSTM) chậm hơn về mặt tính toán nhưng cực kỳ linh hoạt. Trong khi đó, hệ thống dựa trên subword (BERT/Transformer) nhanh hơn và cung cấp các token có ý nghĩa ngữ pháp/ngữ nghĩa hơn cho cơ chế Self-Attention hoạt động hiệu quả. Việc các kiến trúc Transformer chọn Subword Tokenization đã xác lập nó là tiêu chuẩn vàng cho các mô hình ngôn ngữ quy mô lớn sau này.

**III. ELMo: NGƯỜI TIÊN PHONG BIỂU DIỄN TỪ NGỮ CẢNH SÂU**

**3.1. Kiến Trúc Bi-LSTM Sâu**

ELMo (Embeddings from Language Models), được các nhà nghiên cứu tại Allen Institute for Artificial Intelligence (AI2) phát triển và công bố vào năm 2018, là mô hình đầu tiên chứng minh được hiệu quả vượt trội của biểu diễn từ ngữ cảnh sâu.4 ELMo đánh dấu sự chuyển dịch quan trọng khỏi các mô hình tĩnh.

Về mặt kiến trúc, ELMo sử dụng một mạng LSTM hai chiều (Bi-LSTM) sâu.9 Mô hình được huấn luyện dựa trên một ngữ liệu khổng lồ (khoảng 30 triệu câu và 1 tỷ từ).9 ELMo đạt được sự hiểu biết ngữ cảnh bằng cách tận dụng một Mô hình Ngôn ngữ Hai Chiều (Bi-LM). Bi-LM này không chỉ là một mô hình đơn thuần; nó bao gồm hai mô hình ngôn ngữ độc lập được huấn luyện đồng thời: một chạy xuôi (forward), dự đoán từ tiếp theo dựa trên các từ trước đó; và một chạy ngược (backward), dự đoán từ trước đó dựa trên các từ tiếp theo.11 Sự kết hợp này cho phép mô hình có "cảm nhận" về cả ngữ cảnh bên trái và bên phải của từ đang được xử lý.11

Một đặc điểm độc đáo khác của ELMo là nó lấy đầu vào là các vector nhúng cấp độ ký tự (character-level inputs).2 Như đã đề cập ở trên, cách tiếp cận dựa trên ký tự này giúp mô hình xử lý hiệu quả các từ OOV, mang lại sự ổn định và khả năng khái quát hóa cao cho vốn từ vựng mở.

**3.2. Cơ chế Tổng hợp Ngữ cảnh (Weighted Summation)**

Do sử dụng mạng Bi-LSTM sâu, ELMo không chỉ tạo ra một vector duy nhất cho mỗi từ, mà nó sinh ra một *tập hợp* các vector ẩn (hidden states) tương ứng với các lớp khác nhau của mạng.1 Các nghiên cứu cho thấy rằng các lớp khác nhau này nắm bắt các loại thông tin ngôn ngữ khác nhau: các lớp thấp hơn có xu hướng nắm bắt các tính năng cú pháp (syntactic features), trong khi các lớp cao hơn tập trung vào các tính năng ngữ nghĩa (semantic features).9

Để tạo ra Contextualized Embedding cuối cùng cho một từ, ELMo áp dụng cơ chế tổng hợp có trọng số (weighted summation) của các hidden state từ tất cả các lớp của Bi-LSTM (bao gồm cả embedding ban đầu).11 Các trọng số này không cố định mà được học trong quá trình huấn luyện tinh chỉnh (fine-tuning) cho từng tác vụ downstream cụ thể. Bằng cách điều chỉnh trọng số, mô hình downstream có thể quyết định mức độ quan trọng của thông tin cú pháp so với thông tin ngữ nghĩa trong tác vụ đó.

ELMo được coi là một mô hình trích xuất tính năng (feature-based).5 Vector ngữ cảnh được tạo ra được ghép (concatenation) hoặc cộng vào đầu vào của mô hình chuyên biệt cho tác vụ cuối cùng (ví dụ: mô hình phân loại, NER). ELMo đã chứng minh rằng Biểu diễn Từ Ngữ cảnh Sâu là cực kỳ hữu ích cho nhiều tác vụ NLP, bao gồm giải quyết đồng tham chiếu (coreference resolution) và giải quyết tính đa nghĩa.9

**3.3. Hạn Chế Kiến Trúc của Bi-LSTM**

Mặc dù ELMo là mô hình tiên phong, nó có những hạn chế cố hữu do kiến trúc Bi-LSTM. Mô hình này được tạo thành từ hai mô hình ngôn ngữ đơn hướng (Left-to-Right và Right-to-Left) được huấn luyện độc lập và chỉ được kết hợp (concatenation hoặc summation) ở tầng cuối cùng.11 Điều này tạo ra một vấn đề là sự tương tác *thực sự* hai chiều (nơi thông tin từ cả hai hướng ảnh hưởng đến việc tính toán ẩn ở giữa) không diễn ra trong mỗi lớp mạng như trong cơ chế Self-Attention.

Thành công của ELMo đã chứng minh khái niệm Contextualized Embeddings là khả thi.12 Tuy nhiên, những hạn chế về tốc độ xử lý tuần tự của RNN/LSTM và việc thiếu sự tương tác hai chiều thực sự trong quá trình tính toán ngữ cảnh đã mở đường cho kiến trúc Transformer, vốn có khả năng song song hóa cao hơn và nắm bắt các phụ thuộc xa hiệu quả hơn.

**IV. BERT: ĐỘT PHÁ CỦA BIỂU DIỄN MÃ HÓA HAI CHIỀU BẰNG TRANSFORMER**

**4.1. Kiến Trúc Nền Tảng: Transformer Encoder**

Sự ra đời của BERT (Bidirectional Encoder Representations from Transformers) vào năm 2018 bởi Google đã tạo ra một bước nhảy vọt khác trong lĩnh vực NLP. BERT được xây dựng dựa trên kiến trúc **Transformer Encoder** 13, loại bỏ các mạng tuần tự như RNN/LSTM.

Cơ chế cốt lõi của Transformer là **Self-Attention**.11 Self-Attention cho phép mô hình tính toán mối quan hệ giữa một từ với *tất cả* các từ khác trong câu đầu vào trong một lần tính toán, độc lập với khoảng cách giữa chúng. Điều này cho phép mô hình xem xét toàn bộ ngữ cảnh đầu vào (full input text context) 13 một cách đồng thời, tạo ra một biểu diễn ngữ cảnh hóa mạnh mẽ hơn nhiều so với Bi-LSTM của ELMo, vốn bị giới hạn bởi tính tuần tự.

**4.2. Mục tiêu Huấn luyện Chính: Masked Language Model (MLM)**

Yếu tố quyết định giúp BERT đạt được tính hai chiều *thực sự* (bidirectional) là mục tiêu huấn luyện tự giám sát (self-supervised learning) mang tên Masked Language Model (MLM).13

Trong MLM, thay vì cố gắng dự đoán từ tiếp theo như các mô hình ngôn ngữ tự hồi quy (autoregressive) truyền thống, BERT che (mask) ngẫu nhiên 15% các token đầu vào và sau đó cố gắng dự đoán chính xác các token bị che đó.13 Cơ chế này buộc mô hình phải tích hợp thông tin ngữ cảnh từ cả phía trước và phía sau của từ bị che để có thể điền vào chỗ trống một cách chính xác.14 Khả năng xử lý thông tin hai chiều đồng thời này là lý do BERT đạt được sự hiểu biết sâu sắc về cấu trúc và ngữ nghĩa của ngôn ngữ con người.

**4.3. Next Sentence Prediction (NSP)**

Bên cạnh MLM, trong giai đoạn tiền huấn luyện (pre-training) ban đầu, BERT cũng sử dụng một tác vụ nhị phân thứ cấp gọi là Next Sentence Prediction (NSP).13 Tác vụ này yêu cầu mô hình dự đoán liệu một câu thứ hai (Sentence B) có phải là câu tiếp theo tự nhiên của câu thứ nhất (Sentence A) trong tài liệu gốc hay không.13

Mục đích của NSP là giúp mô hình phát triển sự hiểu biết về mối quan hệ giữa các câu, một kỹ năng cần thiết cho các tác vụ downstream đòi hỏi sự hiểu biết cấp độ văn bản (document-level tasks) như Question Answering (Hỏi Đáp) hoặc Inference (Suy luận).13

**4.4. Xử Lý Đầu Vào và Biểu Diễn Ngữ Cảnh**

Đầu vào của BERT được xử lý bằng Subword Tokenization (WordPiece) 2, tạo ra các Token Embedding. Vector nhúng cuối cùng của một token trong BERT là một tổ hợp (summation) của ba loại nhúng:

1. **Token Embedding:** Biểu diễn từ/subword (WordPiece).
2. **Segment Embedding:** Phân biệt nếu token thuộc về Câu A hay Câu B (quan trọng cho NSP).
3. **Positional Encoding:** Mã hóa vị trí của token trong chuỗi, vì kiến trúc Transformer (dựa trên Attention) không tự nhiên xử lý thứ tự.13

Vector nhúng ngữ cảnh được tạo ra ở đầu ra của mô hình Encoder (thường có kích thước 768 trong phiên bản BERT Base).11 Vector này chứa đựng thông tin ngữ cảnh đã được mã hóa sâu sắc và sau đó được sử dụng trực tiếp cho các tác vụ NLP downstream thông qua phương pháp *fine-tuning* (tinh chỉnh).13

**4.5. Phân Tích Độ Hạn Chế của MLM**

Mặc dù MLM là cơ chế cốt lõi mang lại tính hai chiều cho BERT, nó cũng chứa đựng một hạn chế lý thuyết quan trọng. Việc đưa token nhân tạo vào dữ liệu đầu vào trong giai đoạn pre-training tạo ra một *sự không khớp* (discrepancy hoặc mismatch) giữa giai đoạn huấn luyện ban đầu (với các token) và giai đoạn fine-tuning (với đầu vào không bị che).13 Hạn chế kiến trúc nội tại này là một động lực thúc đẩy các mô hình kế nhiệm tìm kiếm các mục tiêu huấn luyện tinh vi hơn, loại bỏ sự phụ thuộc vào các token nhân tạo.

Đồng thời, sự chuyển đổi từ ELMo sang BERT cũng là sự chuyển đổi từ các kiến trúc tuần tự (Bi-LSTM) sang các kiến trúc dựa trên Attention. Việc sử dụng Self-Attention 11 là nguyên nhân trực tiếp cho khả năng nắm bắt ngữ cảnh hiệu quả và tốc độ xử lý nhanh hơn của BERT. Attention cho phép các phụ thuộc giữa các từ được tính toán ngay lập tức, vượt qua vấn đề quên thông tin (vanishing gradient) và phụ thuộc vào khoảng cách mà các kiến trúc RNN/LSTM thường gặp phải.

**V. TỐI ƯU HÓA TRANSFORMER VÀ CÁC MÔ HÌNH NGỮ CẢNH NÂNG CAO**

Sự ra đời của BERT đã định hình lại tiêu chuẩn NLP, nhưng nó cũng nhanh chóng được cải tiến thông qua việc tối ưu hóa quy trình huấn luyện và mục tiêu mô hình hóa. Các mô hình như RoBERTa và XLNet là những ví dụ điển hình cho sự tinh chỉnh này.

**5.1. RoBERTa: Tối Ưu Hóa Robust (Robustly Optimized BERT Approach)**

RoBERTa (Robustly Optimized BERT Approach), được phát triển bởi Facebook AI, giữ nguyên kiến trúc Transformer Encoder của BERT nhưng tập trung vào việc tối ưu hóa triệt để quy trình huấn luyện, chứng minh rằng các thay đổi chiến lược trong huấn luyện có thể mang lại hiệu suất vượt trội.13

Các điểm cải tiến chính của RoBERTa bao gồm:

1. **Loại bỏ NSP:** RoBERTa loại bỏ tác vụ Next Sentence Prediction (NSP).13 Các nghiên cứu chỉ ra rằng NSP không đóng góp đáng kể vào hiệu suất cuối cùng trên hầu hết các tác vụ downstream, và việc loại bỏ nó giúp tập trung tài nguyên huấn luyện vào MLM.
2. **Dynamic Masking:** Thay vì cố định các token bị che trong suốt quá trình huấn luyện (như BERT gốc), RoBERTa sử dụng dynamic masking.13 Các token bị che được thay đổi ngẫu nhiên ở mỗi epoch (hoặc mỗi lần đi qua dữ liệu), tạo ra các mẫu huấn luyện đa dạng hơn và buộc mô hình phải học các biểu diễn ngữ nghĩa và cú pháp sâu hơn, thay vì chỉ ghi nhớ các vị trí bị che cố định.
3. **Tăng Quy Mô Huấn Luyện:** RoBERTa được huấn luyện trên lượng dữ liệu lớn hơn đáng kể (lên đến 160GB, so với 16GB của BERT gốc), sử dụng kích thước batch lớn hơn nhiều, và huấn luyện trong thời gian dài hơn.13

Sự tổng hợp của những tối ưu hóa này đã giúp RoBERTa vượt trội hơn BERT trên hầu hết các bộ dữ liệu benchmark.13 Kết quả này củng cố một xu hướng quan trọng: trong NLP hiện đại, *quy mô* và *chiến lược* huấn luyện (tối ưu hóa mục tiêu, dữ liệu lớn) đóng vai trò then chốt trong việc thúc đẩy hiệu suất, thậm chí còn quan trọng hơn các thay đổi lớn về kiến trúc cơ bản.

**5.2. XLNet: Mô Hình Ngôn Ngữ Hoán Vị (Permutation Language Modeling)**

XLNet, được đề xuất bởi Đại học Carnegie Mellon và Google, tập trung giải quyết hạn chế lý thuyết cốt lõi của BERT là sự không khớp giữa pre-training và fine-tuning do việc sử dụng token.

XLNet giới thiệu **Permutation Language Modeling (PLM)**.14 Thay vì sử dụng cơ chế che token giả tạo, PLM huấn luyện mô hình dự đoán các token trong một thứ tự hoán vị ngẫu nhiên của chuỗi đầu vào. Điều quan trọng cần lưu ý là XLNet vẫn giữ nguyên thứ tự vật lý của chuỗi đầu vào (để giữ Positional Encoding), nhưng sử dụng mặt nạ chú ý (attention mask) trong Transformer để điều khiển *thứ tự tính toán* (factorization order) theo hoán vị.15

Cơ chế này mang lại hai lợi ích lớn:

1. **Tính hai chiều không che:** Cho phép mô hình nhìn thấy ngữ cảnh từ tất cả các token khác (trái và phải) mà không cần dựa vào token nhân tạo, giải quyết được sự không khớp của MLM.14
2. **Tính tự hồi quy (Autoregressive):** Bằng cách mô hình hóa xác suất có điều kiện của một token dựa trên các token đã được dự đoán trước đó trong chuỗi hoán vị, XLNet duy trì được các ưu điểm của mô hình ngôn ngữ tự hồi quy, vốn mạnh mẽ hơn trong việc tạo ra chuỗi (sequence generation).15

Hơn nữa, XLNet tích hợp kiến trúc **Transformer-XL** làm nền tảng.16 Transformer-XL bao gồm cơ chế đệ quy phân đoạn (segment recurrence) và nhúng vị trí tương đối (relative positional encoding).16 Các tính năng này cho phép XLNet xử lý các phụ thuộc dài hạn tốt hơn nhiều so với BERT hoặc RoBERTa, vốn bị giới hạn nghiêm ngặt bởi độ dài đầu vào tối đa cố định của Transformer gốc.14

Sự phát triển của XLNet đại diện cho một bước tiến phức tạp hơn về mặt lý thuyết, tìm cách đạt được sự kết hợp giữa khả năng mã hóa hai chiều mạnh mẽ và tính tự hồi quy tự nhiên, điều mà các mô hình dựa trên MLM không thể làm được một cách hoàn hảo.

**VI. VAI TRÒ CHUYÊN SÂU CỦA NHÚNG NGỮ CẢNH: GIẢI QUYẾT TÍNH ĐA NGHĨA VÀ NÂNG CAO HIỆU SUẤT NLP**

Các mô hình Contextualized Embeddings đã không chỉ cải thiện hiệu suất mà còn thay đổi căn bản cách các nhà khoa học dữ liệu tiếp cận các vấn đề ngôn ngữ. Vai trò chuyên sâu nhất của chúng là giải quyết tính đa nghĩa, vốn là rào cản lớn nhất đối với các hệ thống NLP truyền thống.

**6.1. Xử Lý Hiệu Quả Tính Đa Nghĩa (Polysemy Resolution)**

Khả năng tạo ra các vector nhúng động cho cùng một từ tùy thuộc vào ngữ cảnh 4 là thành tựu cốt lõi. Như đã minh họa, từ "apple" được nhúng khác biệt đáng kể khi nó là chủ thể của hành động "ăn" (I ate an apple - trái cây) so với khi nó là chủ thể của hành động "phát hành" (Apple releases iPhone - công ty công nghệ).4

Việc phân biệt ngữ nghĩa tinh tế này được thực hiện thông qua cơ chế học tập sâu của mô hình (Bi-LSTM trong ELMo hoặc Self-Attention trong BERT). Các vector nhúng ngữ cảnh hóa mạnh mẽ đến mức chúng có thể mã hóa không chỉ ý nghĩa hiện tại mà còn cả vai trò cú pháp và ngữ pháp của từ trong câu. Điều này trực tiếp loại bỏ sự mơ hồ ngữ nghĩa, giúp các mô hình downstream diễn giải chính xác văn bản.

**6.2. Cải Thiện Tác Vụ Downstream**

Việc sử dụng Contextualized Embeddings đã dẫn đến những cải thiện đáng kể về độ chính xác trên nhiều tác vụ NLP, bao gồm:

**Nhận dạng Thực thể Có tên (Named Entity Recognition - NER)**

Các mô hình như ELMo và BERT đặc biệt hiệu quả trong NER.9 Khả năng hiểu ngữ cảnh giúp mô hình phân biệt một từ là tên người, địa điểm, hay tổ chức trong một câu. Trong một ví dụ thực tiễn, việc ứng dụng ELMo embeddings trong hệ thống NER pháp lý đã giúp tăng độ chính xác nhận dạng thực thể lên tới 24%.4 Mức tăng hiệu suất rõ rệt này là động lực chính thúc đẩy việc áp dụng các kỹ thuật nhúng ngữ cảnh trong các lĩnh vực yêu cầu độ chính xác cao như pháp lý, y tế và tài chính.

**Giải quyết Đồng Tham Chiếu và Phân loại Ý định**

Contextualized Embeddings, đặc biệt là các mô hình dựa trên Transformer với khả năng nắm bắt phụ thuộc xa, cải thiện mạnh mẽ khả năng giải quyết đồng tham chiếu (Coreference Resolution), tức là xác định đại từ (ví dụ: "nó") đang tham chiếu đến thực thể nào trong văn bản. Ngoài ra, trong các ứng dụng chatbot hoặc hệ thống hỏi đáp, ELMo và các mô hình Transformer giúp phân biệt ý định người dùng (Intent Classification) dựa trên sắc thái ngữ cảnh của câu nói.4

Sự ra đời của các mô hình này đã làm thay đổi hoàn toàn phương pháp luận NLP. Thay vì dành thời gian cho việc thiết kế tính năng (feature engineering) thủ công, các nhà nghiên cứu hiện có thể dựa vào các vector nhúng ngữ cảnh đã được "tiền mã hóa" (pre-encoded) ngữ nghĩa và cú pháp sâu sắc, cho phép họ tập trung vào việc tinh chỉnh và thiết kế kiến trúc downstream hiệu quả hơn.

**VII. PHÂN TÍCH SO SÁNH CHUYÊN SÂU VÀ THÁCH THỨC TRIỂN KHAI**

Để minh họa cho sự phát triển của các mô hình nhúng từ, cần phân tích sự khác biệt về mặt kiến trúc và cơ chế học tập giữa các thế hệ.

**7.1. So Sánh Kiến Trúc và Cơ Chế Học**

Các mô hình Contextualized Embeddings được phân biệt rõ ràng bởi kiến trúc cốt lõi và phương pháp huấn luyện của chúng.

So Sánh Cơ Chế Hoạt Động Của Các Mô Hình Nhúng Từ Chính

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô Hình** | **Kiến Trúc Cốt Lõi** | **Phương Pháp Huấn Luyện Chính** | **Đầu Vào Tokenization** | **Sự Phụ Thuộc Ngữ Cảnh** |
| Word2Vec/GloVe | Feedforward/Ma trận | CBOW/Skip-gram | Cấp từ | Tĩnh (Context-Independent) 2 |
| ELMo | Bi-LSTM Sâu | Bi-directional LM | Cấp ký tự 9 | Ngữ cảnh hóa (Feature-Based) 4 |
| BERT | Transformer Encoder | MLM + NSP 13 | Subword (WordPiece) 2 | Ngữ cảnh hóa (Fine-Tuning) |
| XLNet | Transformer-XL | Permutation LM (PLM) 15 | Subword (SentencePiece) | Ngữ cảnh hóa (Fine-Tuning) |

Phân Tích Chi Tiết Sự Khác Biệt Giữa Các Mô Hình Dựa Trên Transformer

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc Điểm Huấn Luyện** | **BERT** | **RoBERTa** | **XLNet** |
| Kiến Trúc Nền Tảng | Transformer Encoder | Transformer Encoder | Transformer-XL 16 |
| Mục Tiêu Chính (Pre-training) | MLM + NSP 13 | MLM (Dynamic Masking) 13 | Permutation LM (PLM) 14 |
| Xử Lý Ngữ Cảnh Hai Chiều | Thông qua Che (Masking) | Thông qua Che (Dynamic) | Thông qua Hoán vị Thứ tự Tính toán 15 |
| Giải Quyết Chuỗi Dài | Kém (Giới hạn độ dài cố định) | Kém (Giới hạn độ dài cố định) | Tốt (Sử dụng Recurrence Mechanism) 16 |
| Dữ Liệu Huấn Luyện | Vừa (16GB) | Lớn (160GB+) 13 | Lớn (Tương đương RoBERTa) 14 |

**7.2. Phân Tích Thực Tiễn: Thách Thức và Chi Phí Triển Khai**

Mặc dù các mô hình nhúng ngữ cảnh nâng cao mang lại hiệu suất vượt trội, chúng đi kèm với yêu cầu về tài nguyên tính toán và thách thức về tính mở rộng (scalability) đáng kể.4

**Yêu cầu Tính Toán và Rào cản Tài nguyên**

Việc huấn luyện (pre-training) và tinh chỉnh (fine-tuning) các mô hình như ELMo (dựa trên Bi-LSTM sâu) và đặc biệt là các mô hình Transformer (BERT, RoBERTa) đòi hỏi tài nguyên tính toán mạnh mẽ, bao gồm GPU hoặc TPU chuyên dụng.4 Sự thành công của các mô hình tối ưu hóa như RoBERTa phụ thuộc trực tiếp vào quy mô huấn luyện (dữ liệu lớn hơn, kích thước batch lớn hơn).13

Yêu cầu này ngụ ý rằng cuộc đua hiệu suất trong NLP hiện đại đã trở thành cuộc đua về khả năng tiếp cận và đầu tư vào tài nguyên tính toán. Điều này tạo ra rào cản gia nhập cao hơn cho các nhóm nghiên cứu hoặc doanh nghiệp nhỏ không có đủ nguồn lực để sao chép các quy trình pre-training quy mô lớn này.

**Thách thức về Tính mở rộng và Lưu trữ**

Các vector nhúng ngữ cảnh hóa có chiều cao lớn (ví dụ: 768 hoặc 1024, thậm chí 1536 đối với các mô hình lớn).11 Kích thước lớn này tạo ra thách thức đáng kể về lưu trữ và tính toán khi các ứng dụng thực tế cần xử lý hàng triệu hoặc hàng tỷ vector.

Đặc biệt, trong các hệ thống yêu cầu tìm kiếm độ tương đồng thời gian thực (real-time similarity searches), việc xử lý hàng triệu vector chiều cao đòi hỏi cơ sở hạ tầng mạnh mẽ.17 Để duy trì hiệu suất, các nhà triển khai buộc phải sử dụng các kỹ thuật tối ưu hóa tiên tiến, chẳng hạn như Approximate Nearest Neighbor (ANN), thay vì tìm kiếm khoảng cách Euclidean đầy đủ.17 Độ sâu và chiều của Contextualized Embeddings, trong khi giàu ngữ nghĩa, lại tăng độ phức tạp kỹ thuật và chi phí vận hành cho việc triển khai.

**7.3. Chiến Lược Ứng Dụng (Feature-based vs. Fine-tuning)**

Các mô hình nhúng ngữ cảnh có thể được áp dụng theo hai chiến lược chính:

**Phương pháp Trích xuất Tính năng (Feature-based - Đặc trưng của ELMo)**

Trong chiến lược này, các vector ngữ cảnh hóa được tính toán và sau đó được sử dụng như các tính năng bổ sung, được ghép nối (concatenation) với đầu vào của một mô hình downstream truyền thống.5 Phương pháp này phù hợp khi tài nguyên tính toán hạn chế hoặc khi nhà phát triển muốn tích hợp các tính năng ngữ cảnh vào một kiến trúc mô hình đã tồn tại. ELMo thường được sử dụng theo cách này.

**Phương pháp Tinh chỉnh Toàn bộ (Fine-tuning - Đặc trưng của BERT/RoBERTa/XLNet)**

Chiến lược này chiếm ưu thế hiện nay. Toàn bộ mô hình tiền huấn luyện (ví dụ: tất cả các lớp Transformer) được tinh chỉnh trên một tập dữ liệu nhãn nhỏ hơn cho tác vụ downstream cụ thể (ví dụ: phân loại văn bản, NER). Phương pháp này thường mang lại hiệu suất cao nhất 13 nhưng yêu cầu phải có đủ dữ liệu nhãn cho tác vụ cuối cùng.

**VIII. KẾT LUẬN VÀ XU HƯỚNG TƯƠNG LAI**

**8.1. Tóm Tắt Các Đột Phá Chính**

Sự chuyển đổi từ biểu diễn từ tĩnh (Word2Vec, GloVe) sang các biểu diễn từ ngữ cảnh hóa (Contextualized Embeddings) là một cột mốc cách mạng trong NLP. Nhúng ngữ cảnh hóa đã giải quyết thành công vấn đề tính đa nghĩa bằng cách tạo ra các vector động, thay đổi theo ngữ cảnh, mang lại sự hiểu biết sâu sắc hơn về ngôn ngữ.1

Sự phát triển từ ELMo (người tiên phong dựa trên Bi-LSTM và đầu vào cấp ký tự) đến BERT (sử dụng Transformer và MLM để đạt được tính hai chiều thực sự), và sau đó là các phiên bản tối ưu hóa về huấn luyện (RoBERTa) và lý thuyết mô hình hóa (XLNet với PLM) đã định hình lại toàn bộ chiến lược phát triển mô hình ngôn ngữ. Các mô hình dựa trên Transformer đã trở thành kiến trúc chuẩn mực nhờ khả năng song song hóa và cơ chế Self-Attention vượt trội trong việc nắm bắt phụ thuộc ngữ cảnh xa.

**8.2. Hướng Nghiên Cứu và Phát Triển Tiếp Theo**

Lĩnh vực Contextualized Embeddings vẫn đang phát triển nhanh chóng, với các hướng nghiên cứu chính tập trung vào:

**Tối Ưu Hóa Hiệu Quả và Nén Mô Hình**

Do các mô hình ngữ cảnh hóa có kích thước khổng lồ và yêu cầu tài nguyên lớn, xu hướng hiện tại là tập trung vào các kỹ thuật nén mô hình (model compression) như rút gọn kiến trúc (Distillation - ví dụ: DistilBERT), chia sẻ tham số (ALBERT), và lượng tử hóa (quantization). Mục tiêu là giảm thiểu chi phí tính toán và yêu cầu lưu trữ cho việc triển khai thời gian thực, đồng thời duy trì hiệu suất cao.17

**Xử lý Chuỗi Dài và Kiến Trúc Tự Hồi Quy**

Kỹ thuật được giới thiệu trong XLNet (Transformer-XL, sử dụng cơ chế đệ quy phân đoạn) đã chứng minh tầm quan trọng của việc xử lý các phụ thuộc dài hơn độ dài đầu vào tối đa cố định của BERT/RoBERTa.16 Các Mô hình Ngôn ngữ Lớn (LLMs) gần đây, vốn dựa trên kiến trúc tự hồi quy (Autoregressive) như GPT, tiếp tục kế thừa và mở rộng khả năng nắm bắt ngữ cảnh trên các chuỗi văn bản rất dài.

**Đa Ngôn Ngữ và Thích Ứng Miền**

Nghiên cứu đang chuyển dịch sang việc phát triển các mô hình đa ngôn ngữ (Multilingual Models - ví dụ: XLM-R) có thể chuyển giao kiến thức từ các ngôn ngữ có tài nguyên phong phú sang các ngôn ngữ có tài nguyên khan hiếm. Ngoài ra, việc tinh chỉnh các mô hình ngữ cảnh hóa trên các bộ dữ liệu chuyên ngành (domain-specific data) để đạt được độ chính xác cao hơn trong các lĩnh vực như y tế, pháp lý, và tài chính cũng là một lĩnh vực trọng tâm.

**Nguồn trích dẫn**

1. [D] What are the main differences between the word embeddings of ELMo, BERT, Word2vec, and GloVe? : r/MachineLearning - Reddit, truy cập vào tháng 10 29, 2025, <https://www.reddit.com/r/MachineLearning/comments/aptwxm/d_what_are_the_main_differences_between_the_word/>
2. What are the main differences between the word embeddings of ELMo, BERT, Word2vec, and GloVe? - Quora, truy cập vào tháng 10 29, 2025, <https://www.quora.com/What-are-the-main-differences-between-the-word-embeddings-of-ELMo-BERT-Word2vec-and-GloVe>
3. Hiện tượng từ đa nghĩa "Polysemy" trong tiếng Anh và cách giải quyết - ZIM.vn, truy cập vào tháng 10 29, 2025, <https://zim.vn/tu-da-nghia-polysemy-trong-tieng-anh>
4. ELMo Embeddings là gì - Viện FMIT, truy cập vào tháng 10 29, 2025, <https://fmit.vn/tu-dien-quan-ly/elmo-embeddings-la-gi>
5. Sự khác nhau giữa word embedding, contextualized embedding và sentence representation là gì vậy? : r/LanguageTechnology - Reddit, truy cập vào tháng 10 29, 2025, <https://www.reddit.com/r/LanguageTechnology/comments/p6nvqm/what_is_the_difference_between_a_word_embedding_a/?tl=vi>
6. Tokenization vs Embeddings - GeeksforGeeks, truy cập vào tháng 10 29, 2025, <https://www.geeksforgeeks.org/nlp/tokenization-vs-embeddings/>
7. Chi tiết thuật toán tách token Byte-Pair Encoding (BPE) - ProtonX, truy cập vào tháng 10 29, 2025, <https://protonx.io/news/chi-tiet-thuat-toan-tach-token-byte-pair-encoding-bpe-1702295274898>
8. WordPiece tokenization - Hugging Face LLM Course, truy cập vào tháng 10 29, 2025, <https://huggingface.co/learn/llm-course/vi/chapter6/6>
9. ELMo - Wikipedia, truy cập vào tháng 10 29, 2025, <https://en.wikipedia.org/wiki/ELMo>
10. Udrasht/ELMO-Embeddings-from-Language-Models - GitHub, truy cập vào tháng 10 29, 2025, <https://github.com/Udrasht/ELMO-Embeddings-from-Language-Models/>
11. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) - Jay Alammar, truy cập vào tháng 10 29, 2025, <https://jalammar.github.io/illustrated-bert/>
12. Deep Contexualized Representation | PDF - Slideshare, truy cập vào tháng 10 29, 2025, <https://www.slideshare.net/slideshow/deep-contexualized-representation/119201444>
13. RoBERTa vs BERT: A Comprehensive Comparison - DhiWise, truy cập vào tháng 10 29, 2025, <https://www.dhiwise.com/post/roberta-vs-bert-a-comprehensive-comparison>
14. What are the key differences between BERT, RoBERTa, and XLNet in terms of architecture and performance? - Massed Compute, truy cập vào tháng 10 29, 2025, [https://massedcompute.com/faq-answers/?question=What+are+the+key+differences+between+BERT%2C+RoBERTa%2C+and+XLNet+in+terms+of+architecture+and+performance%3F](https://massedcompute.com/faq-answers/?question=What+are+the+key+differences+between+BERT,+RoBERTa,+and+XLNet+in+terms+of+architecture+and+performance?)
15. XLNet: Generalized Autoregressive Pretraining for Language Understanding - arXiv, truy cập vào tháng 10 29, 2025, <https://arxiv.org/pdf/1906.08237>
16. Understand how the XLNet outperforms BERT in Language Modelling | by Karan Purohit | Saarthi.ai | Medium, truy cập vào tháng 10 29, 2025, <https://medium.com/saarthi-ai/xlnet-the-permutation-language-model-b30f5b4e3c1e>
17. Tokenization vs Embedding - How are they Different? - Airbyte, truy cập vào tháng 10 29, 2025, <https://airbyte.com/data-engineering-resources/tokenization-vs-embeddings>