The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Week 8 - Personal Note
By: Quynh Nhi Tran

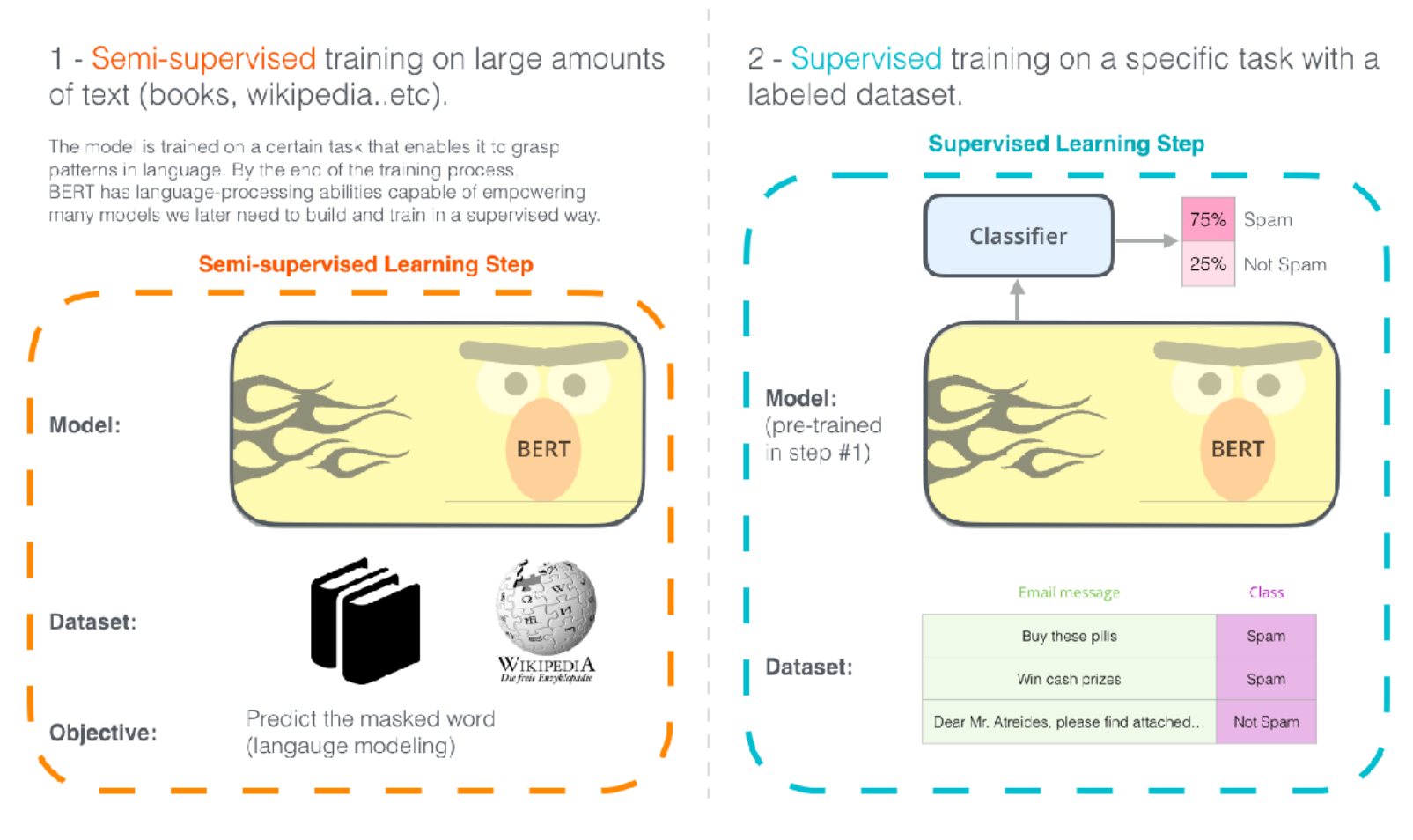
Điểm bùng phát trong NLP năm 2018 và vai trò của Transfer Learning

- Năm 2018 là thời điểm quan trọng đối với NLP. Đây là lúc mà cộng đồng NLP đã có một bước tiến lớn, đặc biệt trong việc tìm ra cách biểu diễn từ ngữ và câu sao cho nắm bắt được ý nghĩa và mối quan hệ ngữ nghĩa.
- Khái niệm **Transfer Learning** đã bùng nổ trong NLP, tạo ra một tình huống được ví như "NLP's ImageNet moment" (tương tự như sự phát triển của Machine Learning trong Computer Vision nhờ bộ dữ liệu ImageNet).
- **Transfer Learning** giúp các mô hình có thể học từ các mô hình lớn đã được huấn luyện trên tập dữ liệu khổng lồ và sau đó tùy chỉnh (fine-tuning) để ứng dụng vào các bài toán cụ thể mà không phải huấn luyện từ đầu.

Tác động của Học chuyển giao

BERT: Một cột mốc trong NLP

- **BERT (Bidirectional Encoder Representations from Transformers)**
- Google đã cung cấp mã nguồn cùng với các phiên bản model đã được huấn luyện
- Với BERT, ai cũng có thể tải xuống và sử dụng mô hình đã huấn luyện trên các tập dữ liệu khổng lồ, tiết kiệm được nhiều thời gian và tài nguyên so với việc phải tự huấn luyện mô hình từ đầu.

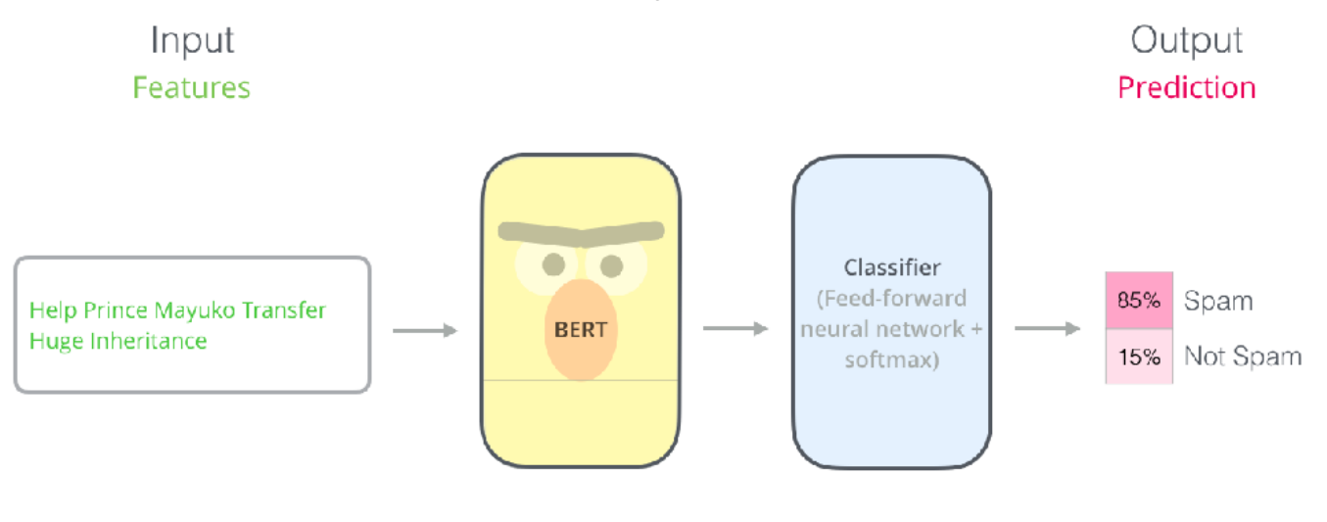


Các nền tảng mà BERT xây dựng lên

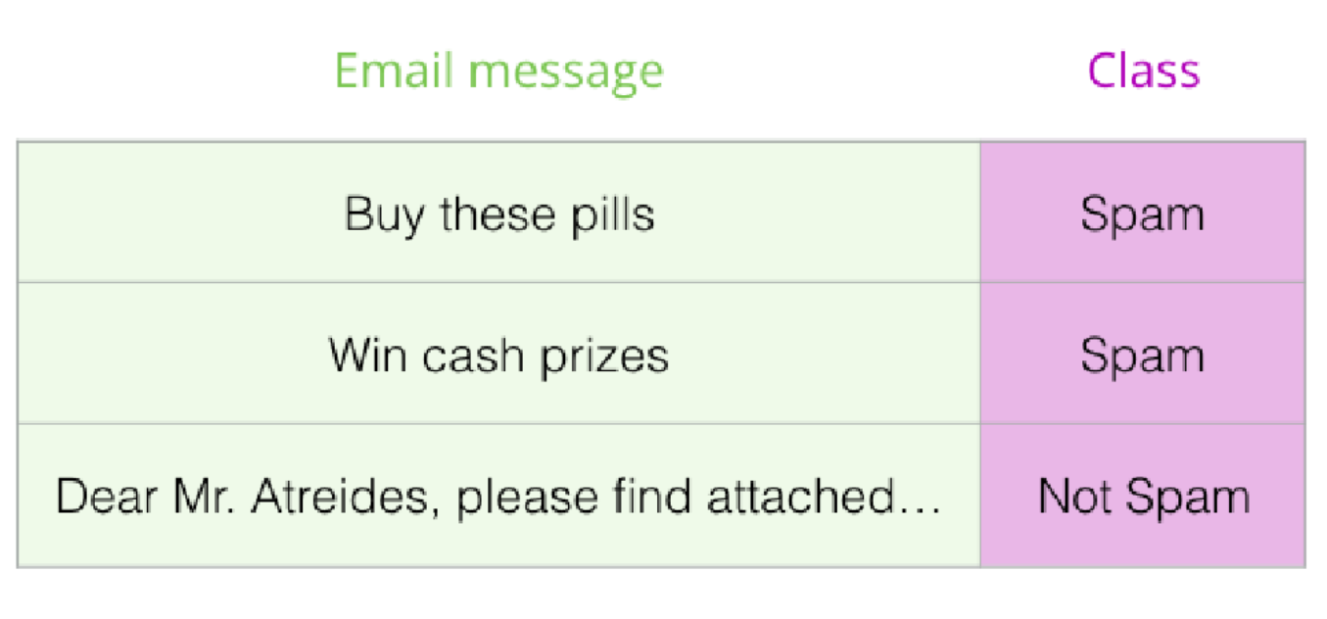
- BERT dựa trên nhiều ý tưởng đã được giới thiệu trước đó trong cộng đồng NLP như:
    - **Semi-supervised Sequence Learning**
    - **ELMo (Embeddings from Language Models)**
    - **ULMFiT (Universal Language Model Fine-tuning)**
    - **OpenAI Transformer**
    - **Transformer** - được sử dụng làm cơ sở kiến trúc chính cho BERT.

Ứng dụng của BERT: Phân loại câu

- Một trong những ứng dụng đơn giản nhất của BERT là **phân loại** một đoạn văn bản . Ví dụ, mô hình có thể được huấn luyện để phân loại email là "spam" hay "not



- Quá trình huấn luyện mô hình này chủ yếu tập trung vào phần **classifier** (phân loại) ở phần đuôi, chỉ tinh chỉnh một chút với BERT (Fine-Tuning).
- Đây là một phương pháp thuộc **supervised learning**, đòi hỏi một **labeled dataset** (tập dữ liệu có gắn nhãn) để mô hình học hỏi và đưa ra dự đoán.

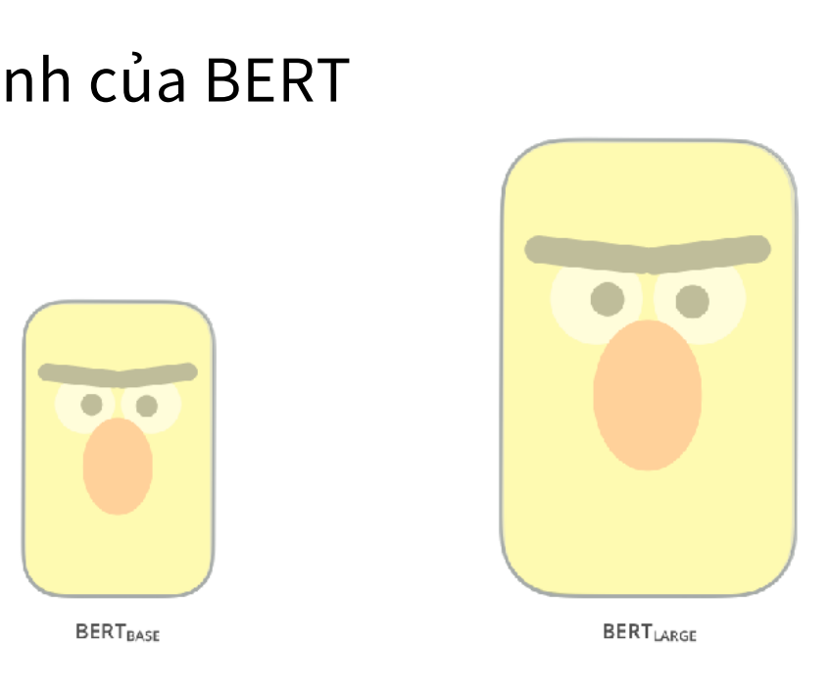


Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

Một số ví dụ khác về ứng dụng BERT trong NLP

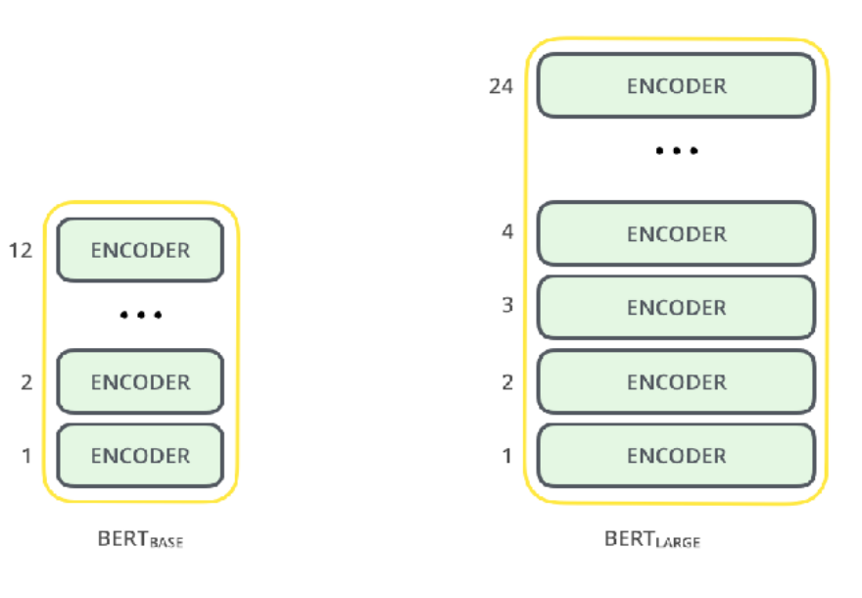
- **Sentiment Analysis**: Phân tích đánh giá (tích cực hoặc tiêu cực).
    - Example: https://nlp.stanford.edu/sentiment/
- **Fact-checking**: Phân loại một câu là "Claim" (khẳng định) hoặc "Not Claim" (không phải khẳng định), và có thể sau đó kiểm tra tính xác thực của nó.
    - Video: https://www.youtube.com/watch?v=ddf0lgPCoSo

Kiến trúc mô hình của BERT



BERT có 2 kích thước mô hình:

- **BERT BASE**: tương đương về kích thước với mô hình Transformer của OpenAI.
- **BERT LARGE**: lớn hơn rất nhiều và đạt kết quả tốt nhất vào thời điểm đó.



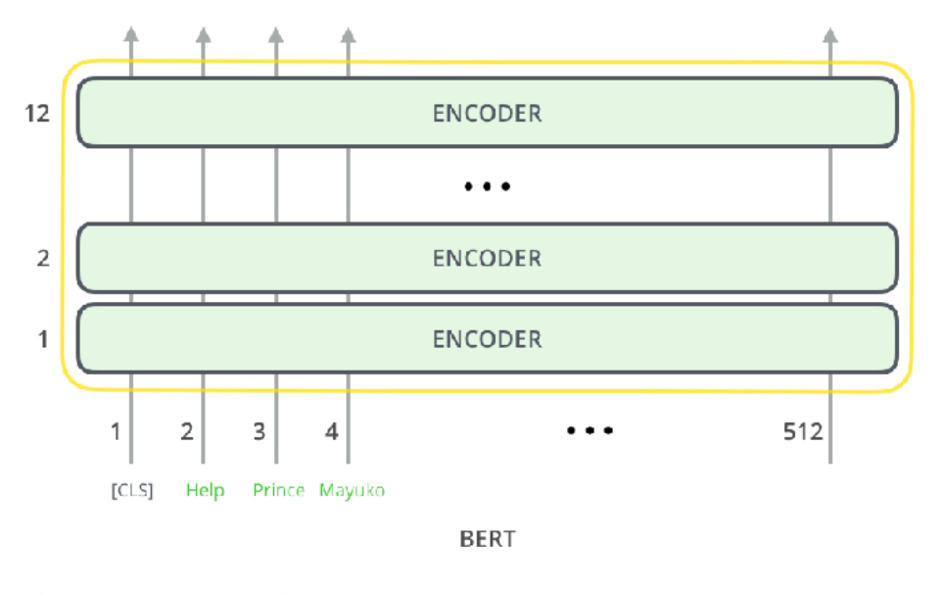
- BERT là một **Transformer Encoder Stack** với 12 lớp **Encoder** trong phiên bản Base và 24 lớp trong phiên bản Large, mỗi lớp có nhiều đơn vị mạng nơron và **attention heads** hơn so với Transformer ban đầu.

Kiến trúc mô hình
→ BERT BASE → 12 lớp mã hóa
→ BERT LARGE → 24 lớp mã hóa

- Ref: Transformer model (foundational concept for BERT)
  https://jalammar.github.io/illustrated-transformer/

Input và Output của BERT

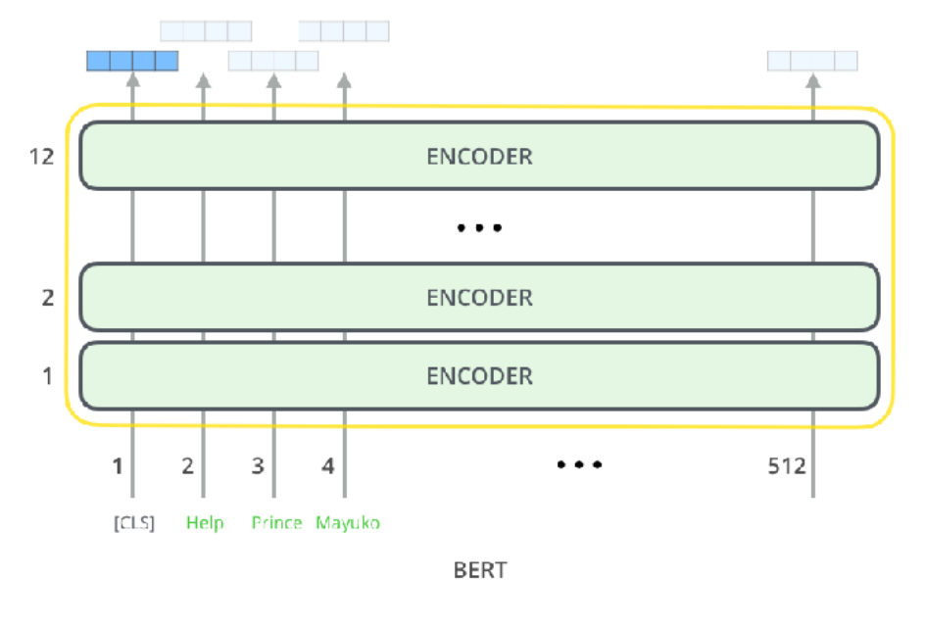
- Input của BERT: token đặc biệt [CLS] cho mục đích phân loại. ([CLS]: Classification)
- Cấu trúc hoạt động tương tự như **Transformer Encoder**: mô hình nhận sequence of words làm input, và qua mỗi lớp oder, nó áp dụng **self-attention** và đưa qua **feed-forward network**

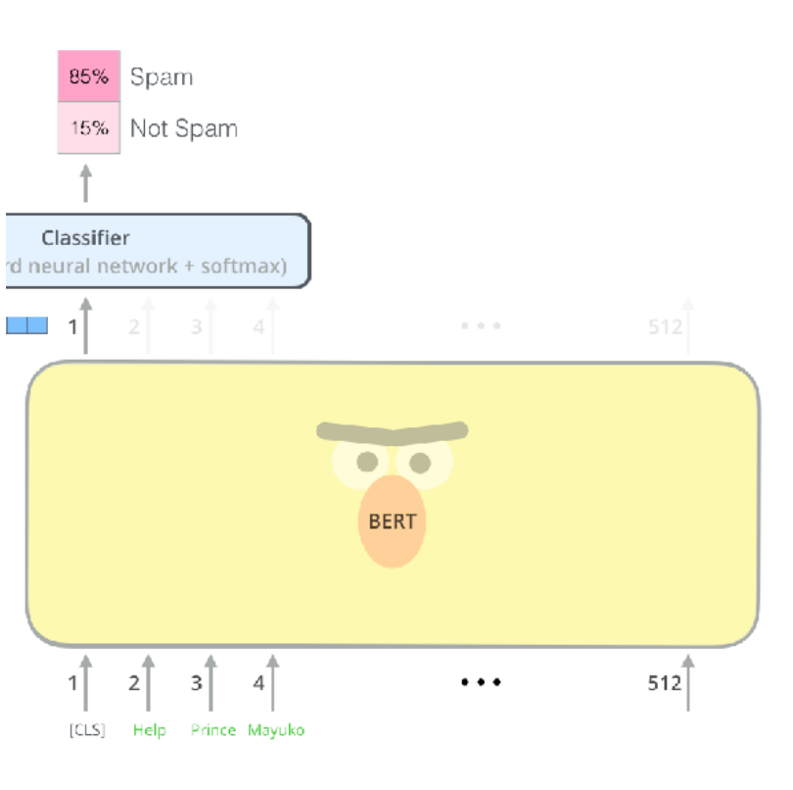


- Mỗi vị trí trong chuỗi đầu vào trả về một **vector** có kích thước 768 (với BERT Base) ở output.

Đầu vào và Đầu ra
→ Đầu vào: [CLS] Token
→ Đầu ra: Kích thước vector 768

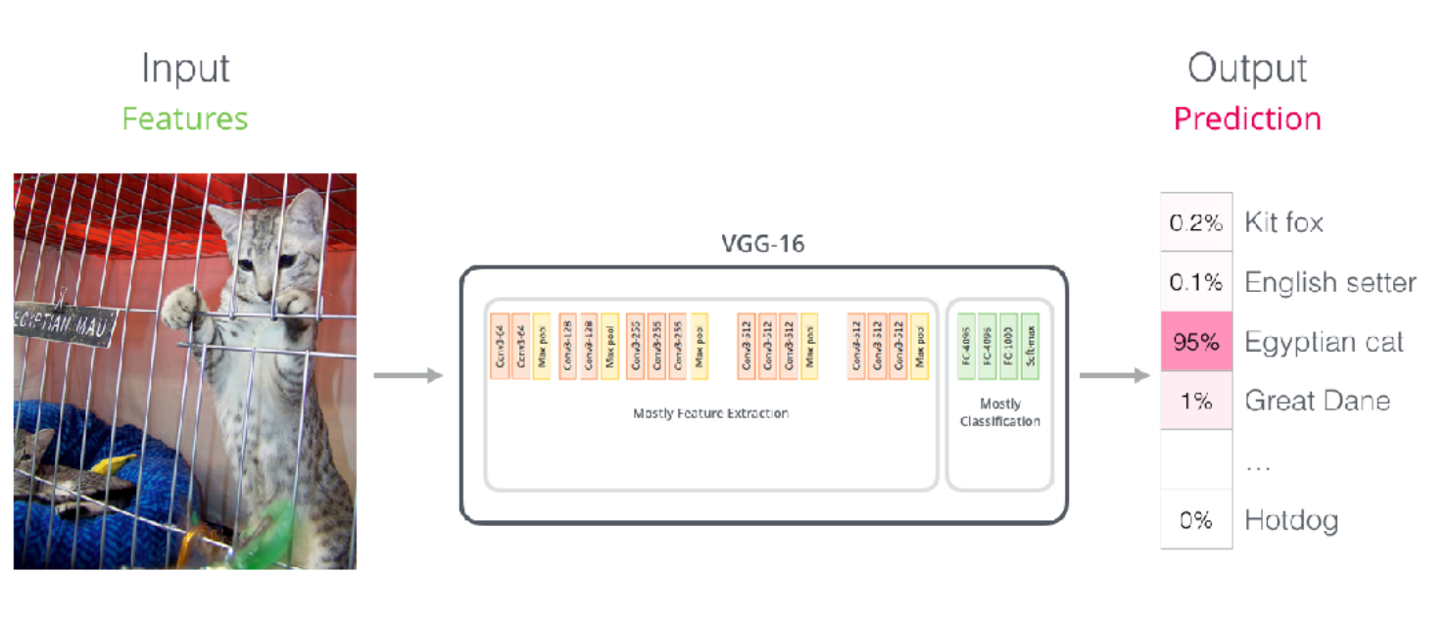
- Đối với bài toán phân loại câu, vector tại vị trí đầu tiên (tương ứng với token [CLS]) được sử dụng làm input cho một **classifier.**

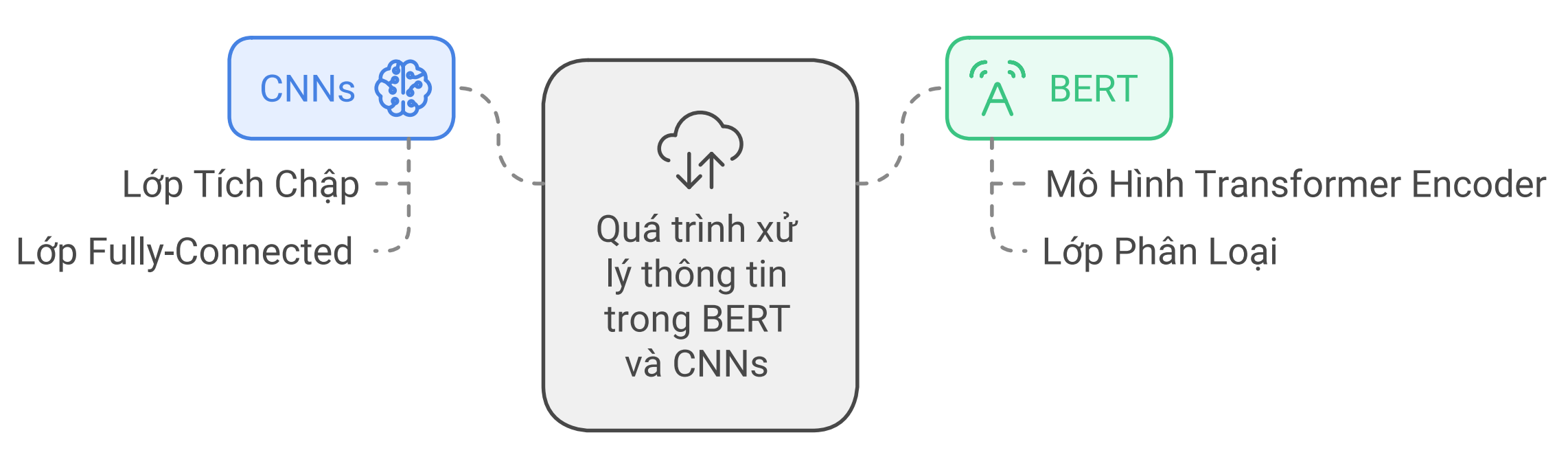




Parallels with Convolutional Nets

Khi làm việc với BERT, cấu trúc truyền tải thông tin giữa các thành phần của mạng giống với cách mà mạng nơ-ron tích chập (CNN) hoạt động trong Computer Vision. Tương tự như cách CNN truyền tải dữ liệu từ phần tích chập (convolution) sang phần fully-connected để phân loại, BERT chuyển một vector đầu ra của mô hình Transformer Encoder tới một lớp classifier để thực hiện các tác vụ xử lý ngôn ngữ.





Kỷ Nguyên Mới của Word Embedding

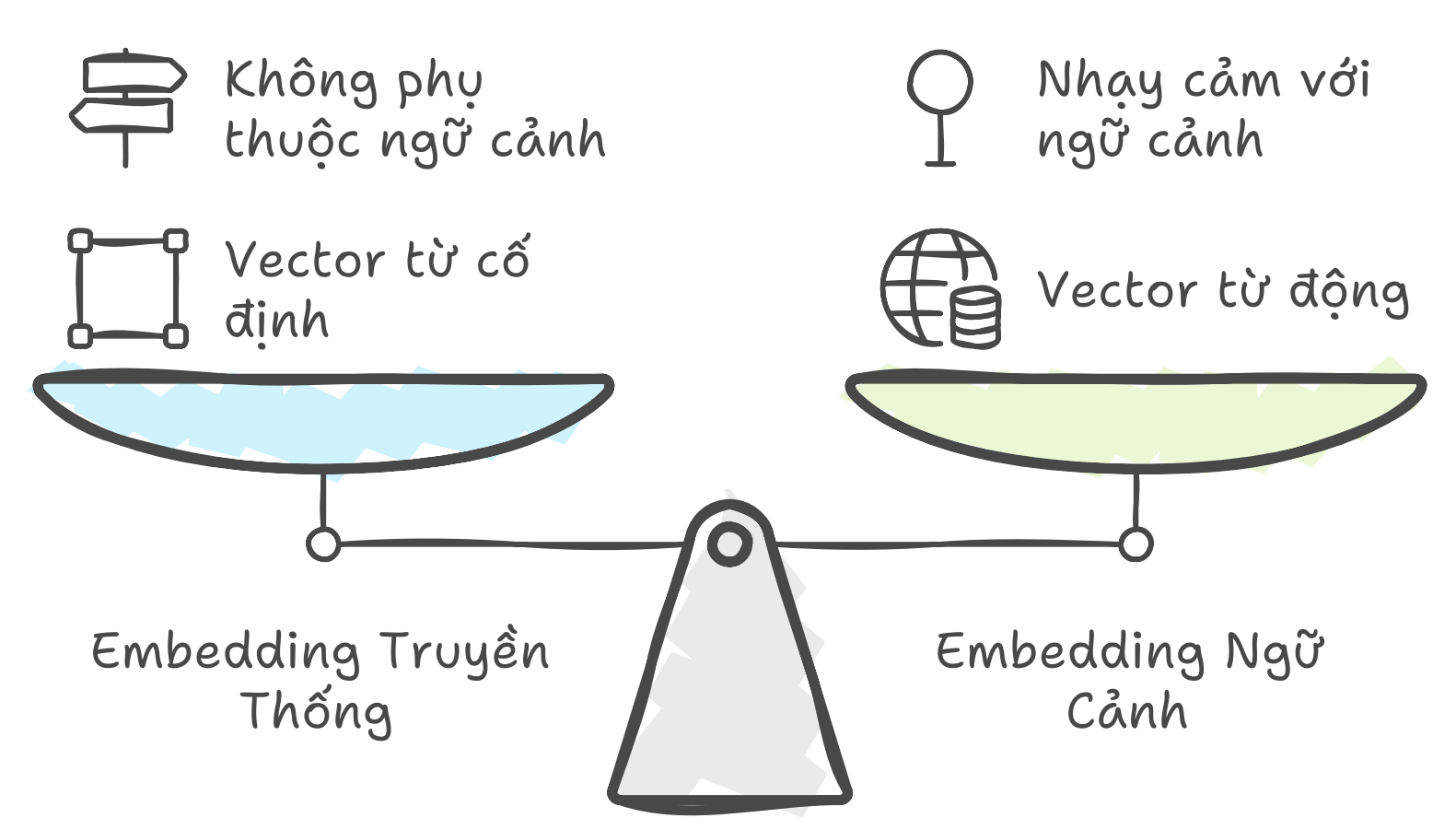
Sự phát triển mới trong NLP đã thay đổi cách mã hóa từ vựng. Trước đây, các phương pháp như Word2Vec và GloVe đã thống trị việc xử lý từ ngữ bằng cách tạo ra các vector để biểu diễn từ ngữ theo các mối quan hệ ngữ nghĩa và ngữ pháp. Những mô hình này cho phép sử dụng các embedding đã được tiền huấn luyện trên một lượng lớn dữ liệu văn bản, giúp giảm công sức và thời gian huấn luyện từ đầu.

Word Embedding Recap

Ví dụ, GloVe embedding của từ "stick" là một vector bao gồm 200 giá trị, giúp biểu diễn ý nghĩa từ vựng của "stick" nhưng vẫn cố định trong mọi ngữ cảnh. Tuy nhiên, vấn đề ở đây là khi "stick" được sử dụng trong các ngữ cảnh khác nhau, ý nghĩa của nó có thể thay đổi, và cách biểu diễn này không thể nắm bắt sự đa nghĩa của từ.

-0.34	-0.84	0.20	-0.26	-0.12	0.23	1.04	-0.16	0.31	0.06	0.30	0.33	-1.17	-0.30	0.03	0.09	0.35	-0.28

The GloVe word embedding of the word "stick" - a vector of 200 floats (rounded to two decimals). It goes on for two hundred values.

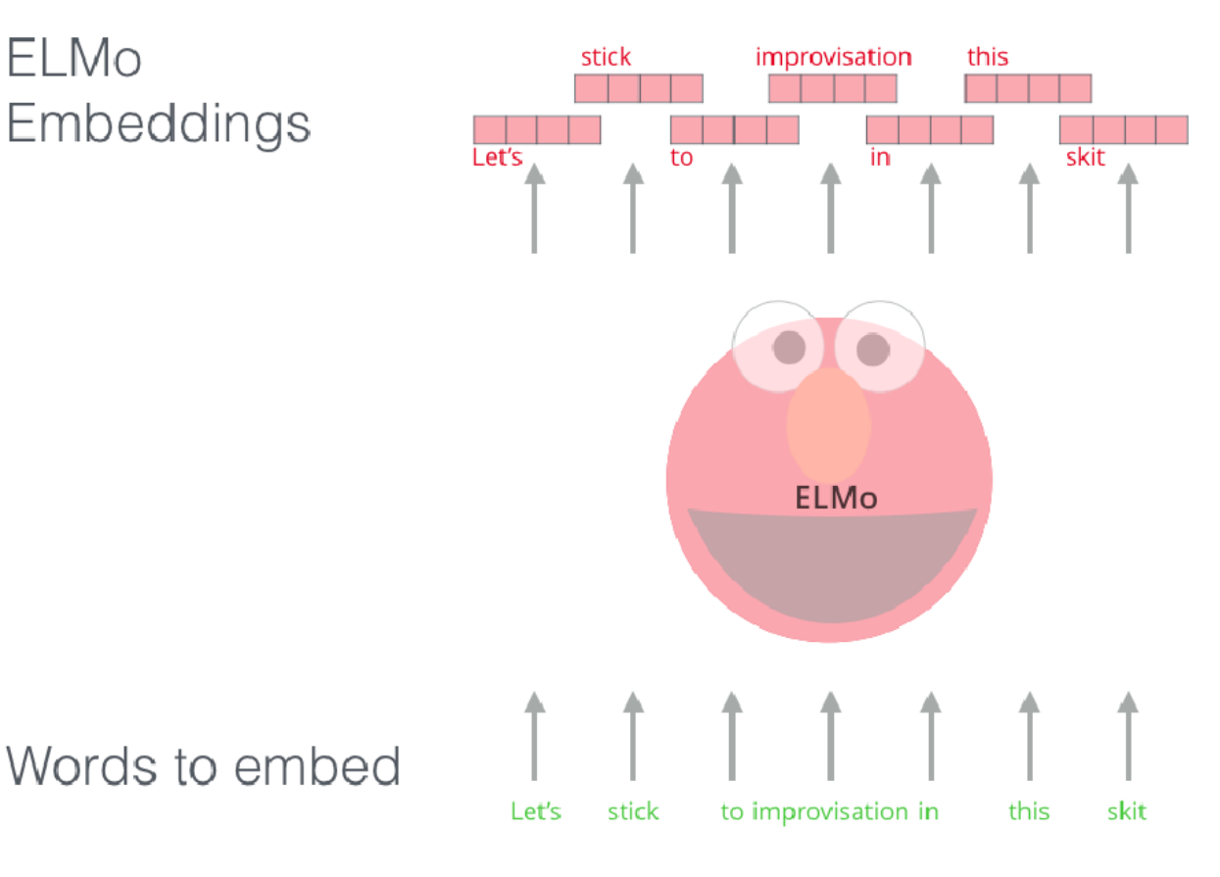


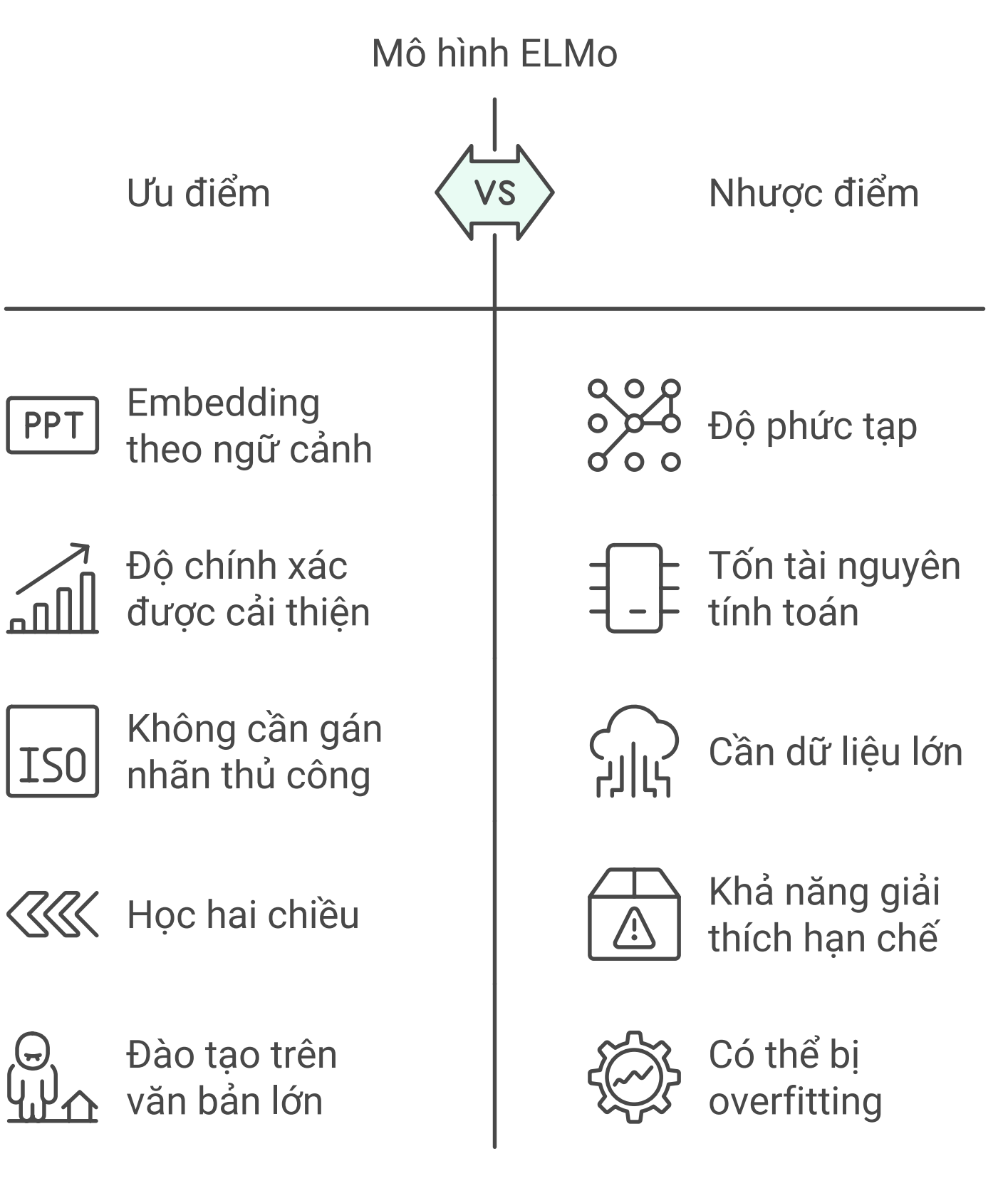
So sánh các embedding từ truyền thống và ngữ cảnh.

ELMo: Ngữ Cảnh Là Quan Trọng

Để giải quyết vấn đề trên, **ELMo (Embeddings from Language Models)** được giới thiệu nhằm tạo embedding dựa trên ngữ cảnh, từ đó nắm bắt ý nghĩa của từ ngữ trong từng câu cụ thể. Mô hình ELMo sử dụng mạng LSTM hai chiều (**bi-directional LSTM**) để gán embedding cho từng từ dựa vào toàn bộ câu chứa nó. Điều này giúp ELMo hiểu được ngữ cảnh tổng quát của từ, đồng thời cải thiện độ chính xác trong các tác vụ xử lý ngôn ngữ.

Điểm đặc biệt của ELMo là được huấn luyện thông qua nhiệm vụ Language Modeling, tức là dự đoán từ tiếp theo trong một chuỗi từ. Việc này cho phép ELMo học từ các mẫu văn bản lớn mà không cần gắn nhãn thủ công, đồng thời mô hình có thể dự đoán từ tiếp theo dựa trên từ trước đó và ngược lại nhờ cấu trúc hai chiều của nó.





ULMFiT: Transfer Learning cho NLP

Trước đây, các mô hình NLP gặp khó khăn khi muốn tận dụng transfer learning giống như trong lĩnh vực thị giác máy tính (Computer Vision). ULMFiT đã cải thiện vấn đề này bằng cách cung cấp một mô hình ngôn ngữ (language model) và một quy trình tinh chỉnh (**fine-tuning)** để tận dụng tối đa những gì mô hình học được từ giai đoạn tiền huấn luyện.

Transformer: Vượt Ra Khỏi LSTM

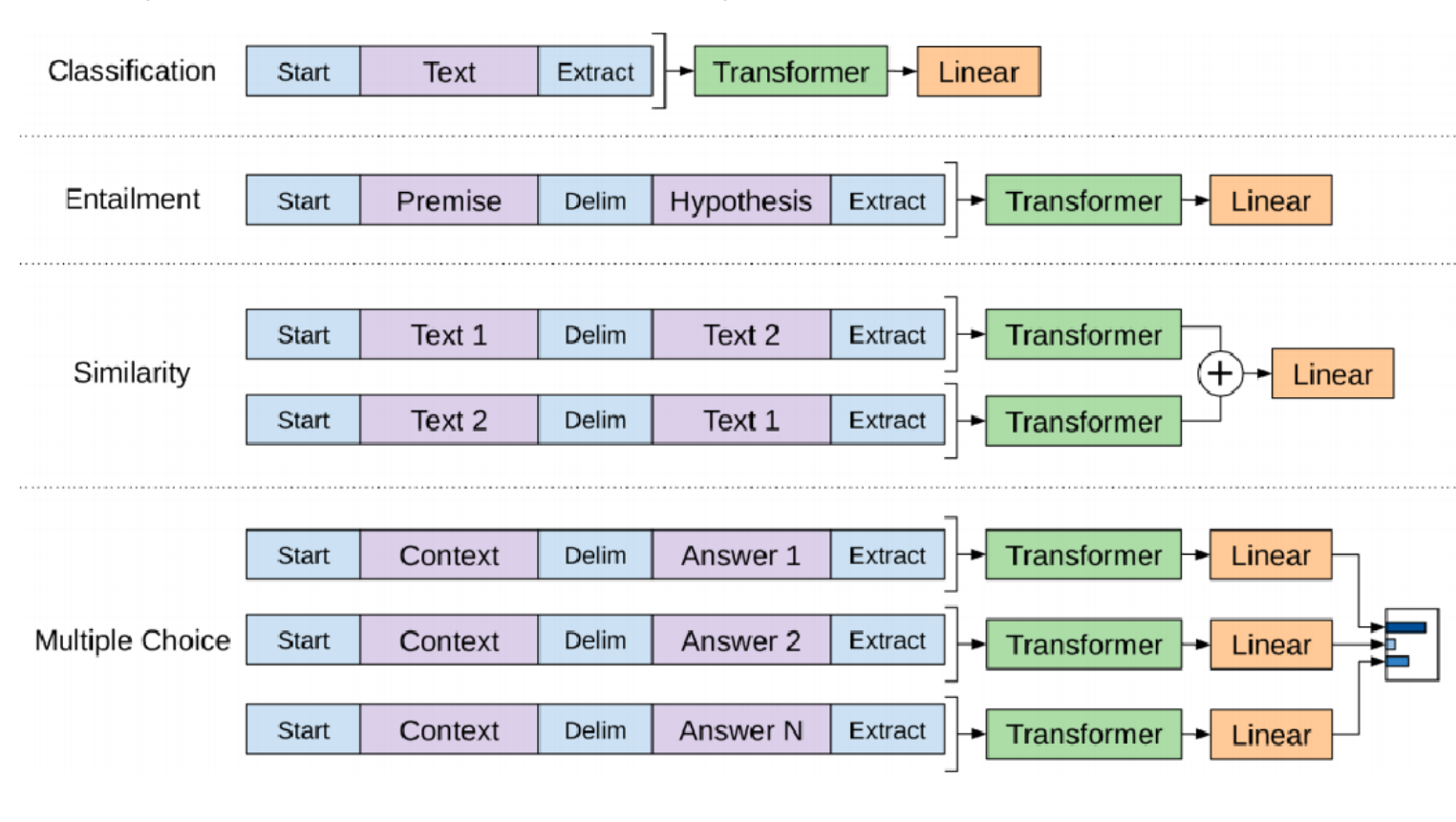
Sự ra đời của mô hình Transformer, với cấu trúc Encoder-Decoder, đã tạo ra bước ngoặt trong NLP, đặc biệt trong các tác vụ yêu cầu phụ thuộc dài hạn như dịch máy. Transformer giúp mô hình hóa các phụ thuộc dài hạn tốt hơn LSTM và phù hợp cho các tác vụ như dịch máy. Tuy nhiên, để sử dụng nó trong phân loại câu hoặc các tác vụ khác, việc tinh chỉnh thêm là cần thiết.

OpenAI Transformer: Huấn luyện sơ bộ với Decoder Layers

Trong lĩnh vực NLP, để thực hiện transfer learning, chúng ta không cần toàn bộ Transformer mà chỉ cần phần decoder. Decoder là lựa chọn tự nhiên cho **language modeling** vì nó có khả năng dự đoán từ tiếp theo bằng cách mask các từ ở tương lai. Trong thiết kế của OpenAI, mô hình được xếp chồng 12 decoder layers, nhưng không bao gồm **encoder-decoder attention sublayer** như ở Transformer nguyên bản. Chỉ có **self-attention layer** để mô hình không "nhìn" trước các từ ở tương lai.

OpenAI Transformer được huấn luyện với dữ liệu lớn, ví dụ như văn bản từ 7.000 cuốn sách. Dữ liệu dạng này cho phép mô hình học cách liên kết thông tin liên quan ngay cả khi chúng cách xa nhau, hữu ích cho các ngữ cảnh văn bản dài, không giống như tweets

Khi đã huấn luyện xong, mô hình này có thể sử dụng để thực hiện các tác vụ như phân loại câu (ví dụ: phân loại email thành "spam" hay "not spam"). OpenAI cũng đã nghiên cứu một số kỹ thuật biến đổi input để xử lý nhiều loại tác vụ khác nhau.



BERT: Từ Decoder sang Encoder với Bidirectional Context

Mặc dù OpenAI Transformer đã thành công với mô hình unidirectional (chỉ nhìn về phía trước), nhưng vẫn còn một khoảng trống. BERT đã giải quyết vấn đề này bằng cách sử dụng **bidirectional conditioning** để mô hình nhìn được cả ngữ cảnh phía trước và phía sau của một từ. Thay vì dùng decoder, BERT dùng các lớp **encoder** được xếp chồng và giới thiệu phương pháp **Masked Language Model (MLM)**. Với MLM, BERT chọn một số từ trong câu và mask chúng để mô hình dự đoán từ đó, nhờ vậy mô hình tận dụng ngữ cảnh hai chiều mà không tự "nhìn thấy" mình ở các lớp khác.

BERT cũng bổ sung thêm một **two-sentence classification task** trong quá trình huấn luyện sơ bộ. Nhiệm vụ này giúp mô hình học cách xác định liệu một câu có khả năng là câu tiếp theo của một câu khác không, giúp BERT xử lý tốt hơn các tác vụ yêu cầu nhận biết mối quan hệ giữa các câu, như kiểm tra tính tương đồng và trả lời câu hỏi.

Mô hình Chuyên Biệt và Tính năng Feature Extraction với BERT

Ngoài fine-tuning, BERT còn có thể dùng để tạo **contextualized embeddings** cho các từ trong văn bản. Bằng cách tận dụng embeddings từ BERT đã được huấn luyện sẵn, các mô hình không cần fine-tuning cho từng tác vụ vẫn có thể cải thiện đáng kể hiệu quả, đạt gần đến hiệu quả của các mô hình fine-tuning đầy đủ. BERT cung cấp nhiều lựa chọn embeddings cho các tác vụ khác nhau trong NLP.

Việc sử dụng BERT cho feature extraction, cùng với fine-tuning, là cách tiếp cận mạnh mẽ cho nhiều ứng dụng NLP, từ nhận diện tên thực thể đến đo lường tính tương đồng

Thử nghiệm với BERT

Để thử nghiệm BERT, bạn có thể bắt đầu với notebook **BERT FineTuning** với Cloud TPUs trên Google Colab, hỗ trợ TPUs, CPUs và GPUs. Các tài nguyên chính bao gồm:

- **modeling.py** (class BertModel) chứa cấu trúc chính của BERT.
- **run_classifier.py** là ví dụ về quá trình fine-tuning và tạo lớp phân loại.
- **tokenization.py** là tokenizer của BERT, chia nhỏ từ thành **WordPieces** thay vì dùng từ nguyên bản.

Hiện có nhiều mô hình BERT được huấn luyện sẵn, hỗ trợ nhiều ngôn ngữ khác nhau từ tiếng Anh đến tiếng Trung, cũng như một mô hình đa ngôn ngữ được huấn luyện trên Wikipedia với 102 ngôn ngữ.

Source: https://jalammar.github.io/illustrated-bert/?authuser=0