TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Ngô Phù Hữu Đại Sơn

Sử dụng GNN biểu diễn các Embeddings cho mô hình Transformer

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 07/2022

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Ngô Phù Hữu Đại Sơn- 18120078

Sử dụng GNN biểu diễn các Embeddings cho mô hình Transformer

ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

NGƯỜI HƯỚNG DẪN
Nguyễn Ngọc Thảo

Tp. Hồ Chí Minh, tháng 07/2022

Lời cảm ơn

Trải qua thời gian dài học tập trong trường, đã đến lúc những kiến thức của em được vận dụng vào thực tiễn công việc. Em lựa chọn làm khóa luận tốt nghiệp để tổng hợp lại kiến thức của mình. Đề tài của em là: "Sử dụng GNN biểu diễn các Embeddings cho mô hình Transformer". Trong suốt quá trình làm khóa luận, em đã nhận được sự hướng dẫn, giúp đỡ quý báu của các thầy cô, các anh chị và các bạn. Em xin được bày tỏ lời cảm ơn chân thành tới:

Th.S Nguyễn Ngọc Thảođã hướng dẫn và truyền đạt những kinh nghiệm quý báu cho em trong suốt thời gian làm khóa luận tốt nghiệp của mình.

Em cũng cảm ơn gia đình và bạn bè đã giúp đỡ em hoàn thành tốt khóa luận.

Khóa luận của em còn những hạn chế về năng lực và những thiếu sót trong quá trình nghiên cứu. Em xin lắng nghe và tiếp thu những ý kiến của giáo viên phản biện để hoàn thiện, bổ sung kiến thức.

Em xin chân thành cảm ơn!

Mục lục

Danh sách hình

Danh sách bảng

Chương 1

Giới thiệu

1.1 Đặt vấn đề

Ngôn ngữ là một loại phương tiện giúp con người có thể giao tiếp và truyền đạt suy nghĩ, ý kiến của mình cho những người xung quanh. Theo trang Ethnologue.com, tính đến năm 2022, trên thế giới có 7151 ngôn ngữ. Vì vậy, một người không thể nào học và hiểu hết mọi ngôn ngữ trên thế giới. Từ đó, có thế độ cần thiết của việc dịch từ một ngôn ngữ sang một ngôn ngữ khác. Ngành khoa học máy tính, cụ thể hơn là xử lý ngôn ngữ tự nhiên, chúng ta cũng tâm đến bài toán trên.

1.2 Bài toán dịch máy (Machine Translation)

Bài toán dịch máy là một lĩnh vực trong ngành khoa học máy tính. Đầu vào được nhập vào máy tính là một đoạn văn bản từ một ngôn ngữ (ngôn ngữ nguồn). Và qua quá trình xử lý, đưa ra được đoạn văn bản tương ứng ở một ngôn ngữ khác (ngôn ngữ đích). Khác với các mô hình xử lý ngôn ngữ khác, khi chúng chỉ cần một bộ ngữ liệu của một ngôn ngữ. Các mô hình dịch máy cần ích nhất hai bộ ngữ liệu của ngôn ngữ nguồn và ngôn ngữ đích. Bộ ngữ liệu bao gồm tập các đoạn văn bản. Với mỗi

đoạn văn bản từ ngôn ngữ nguồn sẽ được ánh xạ đến một đoạn văn bản có ý nghĩa tương ứng ở ngôn ngữ đích. Các bộ ngữ liệu này có thể tổng hợp từ các nguồn khác nhau: từ các bản dịch (subtitle) của các bộ phim, bản dịch sách và cả các bộ dữ liệu được thiết kế riêng cho bàn toàn dịch máy (các bản dịch từ các chuyên gia).

Để có thể hiểu hơn sâu hơn bài toán và tìm ra hướng giải quyết, ta cần phải hiểu được cách tự nhiên mà con người dịch một đoạn văn bản ngôn ngữ nguồn sang ngôn ngữ đích như thế nào. Quá trình này có thể chia làm hai bước:

- Trích xuất ngữ nghĩa (context) của ngôn ngữ nguồn thành thông tin.
- Chuyển hóa thông tin thu thập được thành ngôn ngữ đích.

1.3 Các cách tiếp cận trước

Với tính chất trên của bài toán, ta có thể thấy bài toán này là một bài toán sequence-to-sequence(S2S) và có thể giải quyết bằng kiến trúc encoder-decoder. Các mô hình sử dụng các mạng nơ ron hồi quy (Recursive Neural Network) sử dụng cơ chế long-short-term memory và cả cơ chế attention.

Các mô hình hồi quy (Recurrent model) cho các các kết quả tốt trong bài toán dịch máy. Tuy nhiên, các mô hình này lại sử dụng cơ chế hồi quy. Ở mỗi bước tính toán, mô hình sử dụng thông tin ẩn (hiden state) được tổng hợp từ đầu văn bản đến hiện tại h_t để để làm đầu vào tính toán. Quá trình này lặp lại cho mỗi bước. Từ đó, ta có thể thấy mô hình toán toán bước tiếp theo phải phụ thuộc vào bước trước đó, dẫn đến không thể song song quá trình tính toán này được. Điều này khiến cho việc tối ưu thời gian huấn luyện lẫn hiệu quả tính toán của mô hình trước nên khó khăn.

1.4 Transformer

Trong khi đó, Transformer - một phương pháp dựa hoàn toàn trên cơ chế attention, bỏ qua các cấu trúc của mạng nơ ron hồi quy và mạng nơ ron tích chập phức tạp, giúp đơn giản hóa mô hình những vẫn thể hiện được độ hiệu quả của mô hình. Theo **Paper attention is all your need**, mô hình cho kết quả 41.8 BLEU khi huấn luyện trên tập WMT 2014, cao hơn tất cả các mô hình dịch máy trước đó.

Nhờ không sử dụng kiến trúc RNN, Transformer tránh được nhược điểm chí mạng của các mô hình loại này. Dựa hoàn toàn vào cơ chế attention, cụ thể hơn là giới thiệu kiến trúc multihead-attention giúp việc song song tính toán mô hình. Từ đó mà tăng độ hiệu quả huấn luyện cũng như hiệu quả tính toán.

1.5 Tokenization & Word Embeddings

Trong bài toán dịch máy, các đoạn văn bản thô cần phải được tiền xử lý, chuyển đổi thành các dạng dữ liệu mà máy tính của thể hiểu được. Quá trình đó được thực hiện như sau:

- Tokenization là quá trình tách đoạn văn bản ra thành các từ thành phần (token). Các token này đã được quy định sẵn trong một tập các từ đã biết trước (vocabulary). Với các từ lạ, không thuộc trong vocabulary sẽ được đánh dấu là UNK (unknown).

- Các token sau khi được tách ra vẫn chưa thể đưa vào mô hình do chúng vẫn ở dạng dữ liệu mà các mô hình chưa thể hiểu. Do đó, với mỗi token, ta cần chuyển chúng thành các vector N chiều (N chọn trước) mang tính chất và đại diện cho từ đó. Với mỗi chiều của vector được biểu diễn bằng một số thực. Các vector này được gọi là các Word Embeddings. Các Word Embeddings sẽ là đầu vào của mô hình. Một số phương pháp để tính toán các Word Embeddings trước

đây gồm: ...

1.6 Graph Convolution Network (GCN)

Khai thác quan hệ giữa các điểm dữ liệu với nhau là một đề tài được nhiều sự quan tâm trong học máy. Trong các mô hình học sâu trước đây, ta chỉ có thể trích xuâts được các mối quan hệ thông thường từ các bộ dữ liệu Euclidian.

Trong thực tế, không phải mọi dữ liệu đều biểu diễn ở dạng Euclidian. Do đó, để khai thác được thông tin từ các bộ dữ liệu đồ thị (non-Euclidian), ta cần sử dụng các mô hình Graph Neural Network (GNN). Trong vài năm qua, nhiều biến thể của đã được phát triển. Trong đó, Graph Convolution Network (GCN) là một biến thể quan trọng được xem như là biến thể cơ bản nhất của GNN.

GCN ứng dụng phép tích chập được giới thiệu trong convolution layer của CNN:

- Đối với phép tích chập trên CNN, một đoạn dữ liệu của lớp hiện tại sẽ được nhân vô hướng (tích chập) với một bộ lọc (*filter* hoặc *kernel*). Bộ lọc sẽ trượt trên bộ dữ liệu và các đầu ra của phép tích chập sẽ được truyền vào lớp tiếp theo của mạng.

- Với phép tích chập trên GCN. Ta xét từng nút(node) của đồ thị. Với mỗi nút, ta nhóm nút này với các nút kề của chính nó lại và thực hiền phép tích chập tương tự với một màn lọc. Do số lượng nút kề của mỗi nút là khác nhau, nên kích thước của filter sẽ không cố định. Phép tích chập sẽ được thực hiện trên tất cả các nút của đồ thị.

1.7 Word GCN

Ngôn ngữ là một dạng dữ liệu non-Euclidian. Do đó, trích xuất thông tin của chúng bằng các mô hình GCN được kì vọng là có kết quả tốt hơn

các phương pháp trước đó. Các loại thông tin có thể khai thác bao gồm:

- Thông tin về cú pháp (syntactic) của ngôn ngữ. Các từ trong một câu sẽ sẽ có các quy định về các mối quan hệ của chúng trong câu. Biểu diễn các mối quan hệ này là các cạnh còn các từ trong câu là các nút của đồ thị. Từ đó mà ta có thể khai thác được các đặc trưng cú pháp của các từ. Mô hình SynGCN sẽ giúp ta huấn luyện được các embeddings mang các thông tin trên

- Thông tin về ngữ nghĩa (semantic) của ngôn ngữ. Ý nghĩa của các từ sẽ có các quan hệ với nhau như: đồng nghĩa, trái nghĩa, kế thừa,... Nhờ các mối quan hệ đó, ta có thể biểu diễn được đồ thị ngữ nghĩa của các từ trong bộ từ vựng. Từ đó mà ta có thể khai thác được các đặc trưng ngữ nghĩa của các từ. Mô hình SemGCN sẽ giúp ta huấn luyện được các embeddings mang thông tin ngữ nghĩa.

1.8 Mô hình đề xuất

Ở mô hình đề xuất, tôi muốn tích hợp các embeddings được huấn luyện thông qua WordGCN để đưa vào mô hình các mô hình dịch máy. Mô hình dịch máy để tích hợp vào là transformer do tính hiệu quả cao của mô hình transformer và khả năng tính toán song song của nó.

Chương 2

Tổng quan lý thuyết

2.1 Mô hình Transformer dịch máy

2.1.1 Tổng quan mô hình

Transformer là một mô hình có kiến trúc encoder-decoder. Mô hình bao gồm 2 thành phần chính là Encoder(bộ mã hóa) và Decoder(bộ giải mã). Khi đưa một đoạn văn bản nguồn vào, mô hình sẽ xử lý vào đưa ra đoạn văn bản có ngữ nghĩa tương ứng ở ngôn ngữ đích.

(Hình minh họa encoder-decoder)

Cụ thể hơn, Bộ mã hóa sẽ bao gồm nhiều lớp mã hóa xếp chồng lên nhau. Theo paper transformer, họ sử dụng 6 lớp mã hóa để tạo thành một bộ mã hóa. Bộ giải mã, tương tự cũng được xếp chồng bởi các lớp giải mã. Số lượng lớp của 2 thành phần phải bằng nhau.

(hình transformer high level)

Encoder

Các lớp của bộ mã hóa là độc lập nhau và có cấu trúc tương tự nhau. Đầu vào của lớp đầu tiên sẽ là các word embeddings đã được xử lý qua lớp positional encoding, các lớp phía trên sẽ có đầu vào là các vector output từ lớp ngay dưới. Các vector đầu vào của các lớp này được gọi là context

vector.

Với mỗi lớp mã hóa sẽ bao gồm 2 lớp con:

- Self-attetion: Với mỗi từ trong đoạn văn bản, xem xét độ liên quan của nó với các từ khác trong câu đầu vào. Đưa ra phân bố xác xuất đối với từng từ một.
- Feed forwarding: Mã hóa phân bố xác xuất tính toán được thành các context vector để đưa vào các lớp mã hóa phía trên.

(hình minh họa cụ thể encoder)

Decoder

Các lớp của bộ giải mã cũng có 2 lớp con là Self-attention và Feed-forwarding giống với bộ mã hóa. Tuy nhiên giữa 2 lớp con này có một lớp trung gian là Encoder-Decoder attention. Lớp này có cơ chế giống với cơ chế attention trong mô hình Seq2Seq với thông tin của các hidden layer chính là đầu ra của bộ mã hóa.

(hình minh họa cụ thể decoder)

2.1.2 Cơ chế Self-attention

Cơ sở

Self-attention là một cơ chế mới được giới thiệu trong "attention is all need". Cơ chế này khác với cơ chế attention trong các mạng Seq2Seq trước đó. Self-attention cho phép ta biểu diễn lại mối quan hệ giữa các từ trong một đoạn văn bản.

(Hình minh họa self-attention)

Hình trên cho ta một minh họa về cơ chế self-attention. Xét từ "two", theo tư duy của con người, ta sẽ phân tích xem từ "two" trong câu đang có quan hệ gì với những từ khác. Ngoài ra, ta còn xem xét tầm quan trọng

của các từ khác tác động lên ý nghĩa của câu. Từ đó mà ta có thể có được một cái nhìn rõ ràng hơn về vai trò của từ này trong câu.

Cụ thể hơn, khi mô hình xử lý từ "two", self-attention cho ta thấy được mối liên kết của nó với từ "specialists". Thể hiện vai trò của nó là dùng để chỉ số lượng của các chuyên gia là hai.

Chi tiết

Việc tính toán ở mỗi lớp self attention được tính toán dựa trên công thức sau:

$$Attention = Softmax(\frac{QK^T}{\sqrt{d}})V$$

Trong đó, Q, K, V lần lượt là các ma trận Query, Key, Value. Hai ma trận query và key được dùng để tính toán mối quan hệ giữa các từ trong câu. Còn mỗi dòng thứ của ma trận V đại diện cho từ thứ i trong câu. Từ phân bố softmax, ta tính được context vector, tổng hợp được các thông tin của từ hiện tại và các từ liên quan đến nó.

Trong công thức, ta thấy được tích vô hướng của ma trận Q và K được chuẩn hóa bởi hệ số $\sqrt{d}$ (d là số chiều của vector embedding). Lý do cho việc chuẩn hóa này là do khi d có giá trị lớn, tích vô hướng của Q và K sẽ có giá trị lớn theo do đó, nếu không chuẩn hóa về giá trị nhỏ hơn thì hàm softmax sẽ có độ hội tụ khá chậm.

(minh họa đồ thị đạo hàm softmax)

Từ công thức tính của self-attention, ta có thể thấy rõ sự khác biệt giữa bộ mã hóa của transformer và bộ mã hóa của các mô hình Seq2Seq sử dụng RNN. Đối với RNN, dữ liệu đầu vào phải được mã hóa một cách tuần tự. Trong khi đối với self-attention, các từ trong văn bản có thể được mã hóa một cái song song và không bị phụ thuộc vào nhau. Nhờ đó mà có thể tăng được hiệu quả tính toán.

2.1.3 Cơ chế Multihead-attention

Đầu ra của self-attention cho ta biết được mối quan hệ của các từ trong câu dựa trên một góc nhìn nào đó. Bằng các xếp chồng nhiều lớp self-attention lại với nhau. Ta có thể biết được sự liên quan của các từ ngữ trong câu với nhiều gốc nhìn khác nhau. Từ đó mà có được thông tin đầy đủ hơn về câu cần dịch.

Cơ chế xếp chồng nhiều lớp self-attention lại với nhau được gọi là Multihead-attention.

Ngoài ra, sử dụng cơ chế self-attention còn giúp ta tránh được trường hợp một từ phụ thuộc hoàn toàn vào chính nó. Ta mong muốn một phân bố xác suất quan hệ giữa một từ với các từ có ảnh hưởng đến nó.

Với việc sử dụng N lớp self-attention chạy song song với các bộ trọng số khác nhau, ta có được N ma trận context khác nhau. Lúc này, ta cần có một phương phát khác để tổng hợp thông tin từ các lớp self-attention này lại để đưa vào lớp feed-forward.

(Hình minh họa multi context vector)

Để làm được việc đó, transformer ghép theo chiều ngang các ma trận context lại rồi nhân với một bộ trọng số để đưa ra một kết quả duy nhất là một ma trận với các dùng là các context vector cũng đồng thời là đầu vào cho bộ giải mã ở phía trên.

(Hình minh họa feed-forward)

Hình dưới minh họa cho việc sử dụng nhiều lớp self-attention trên cùng một câu với mỗi một màu tương ứng với kết quả của một lớp self-attention khác nhau.

(Hình minh họa multi head attention)

2.1.4 Positional encoding

Cách tính toán song song của cơ chế self-attention dẫn đến một vấn đề đối với các từ trong đầu vào. Embeedings biểu diễn các từ này chưa biểu diễn được thông tin về thứ tự của các từ trong câu. Trong khi thông tin

này là một thông tin quan trọng vì thay đổi vị trí của các từ có thể dẫn đến một câu hoàn toàn khác ý nghĩa.

(Hình minh họa về thứ tự từ hoặc câu minh họa)

Transformer sử dụng cơ chế positional encoding. Cơ chế này giúp đưa thông tin về vị trí của các từ vào trong các embeddings. Cụ thể hơn, trước khi embeddings được đưa vào trong mô hình, nó được cộng với một vector tương ứng với vị trí trong câu. Những vector này tuân theo một một quy định nhất định. Chúng phải thể hiện được sự khác biệt về vị trí của các từ trong câu và cả khoảng cách của các từ trong câu.

Công thức tính toán các vector này như sau:

$$PE_{(pos,2i)} = \sin \frac{pos}{10000^{\frac{2i}{d_model}}}$$

$$PE_{(pos,2i+1)} = \cos \frac{pos}{10000^{\frac{2i}{d_model}}}$$

(Lý giải về công thức)

2.1.5 residuals

Để tránh các trường hợp vanishing graient, Transformer kết hợp các đường residual và mạng encoder. Từ đó mà thông tin từ các lớp trước có thể được sử dụng lại trong khi huấn luyện các lớp sau.

(Hình minh họa Residual)

2.1.6 Tổng kết mô hình

2.2 Mô hình WordGCN huấn luyện embedding cho bộ ngữ liệu

2.2.1 Lý thuyết đồ thị cơ bản

2.2.2 Mô hình Graph Convolution Network

2.2.3 Mô hình WordGCN

Mô hình SynGCN

Mô hình SemGCN

Chương 3

Phương pháp

3.0.1 Tạo và chuyển dữ liệu cho chủ dữ liệu

3.0.2 Chia sẻ dữ liệu ngang hàng

Chương 4

Ứng dụng và cài đặt

4.0.1 Web Application (cho nhà cung cấp nội dung)

4.0.2 Mobile Application (cho người dùng)

Tài liệu tham khảo