# Ứng dụng của mô hình ngôn ngữ trong thêm dấu tiếng việt

Nguyễn Phú Trường, Lê Trần Hải Tùng, Lê Đức Tùng, Nguyễn Xuân Trường

Ngày 5 tháng 6 năm 2021

**Tóm tắt nội dung**

Việc áp dụng mô hình ngôn ngữ vào các công việc cụ thể đang ngày càng thành công do sự phát triển của các mô hình deep learning. Trong báo cáo này, chúng tôi tập chung vào việc ứng dụng các mô hình xử lý ngôn ngữ để giải quyết bài toán phục hồi dấu câu cho tiếng việt. Chúng tôi tập chung vào ba loại mô hình và so sánh chúng trên các tập dữ liệu khác nhau. Cuối cùng chúng tôi xây dựng một trang web để dễ dàng tương tác với mô hình thông qua giao diện.

## Mục lục

# 1 Giới thiệu

Bài toán thêm dấu câu là một bài toán con trong bài toán chỉnh sửa lỗi chính tả. Đầu vào của của bải toán con này bao gồm một câu bị lược hết dấu câu, công việc của chúng ta là khôi phục lại dấu của câu đó. Việc khôi phục dấu câu được coi là một công việc khó do trong tiếng việt, một câu không có dấu có thể dịch ra rất nhiều nghĩa khác nhau. Dấu câu của một từ không chỉ phụ thuộc vào từ đó mà còn phụ thuộc vào vai trò của nó trong câu. Ví dụ câu: *"tram nam trong coi nguoi ta"* có thể dịch ra thành *"trăm năm trong cõi người ta"* hoặc *"trăm năm trông coi người ta"*, cả hai câu đều đúng ngữ pháp. Vì vậy việc xây dựng một giải pháp tối ưu hoàn toàn là điều không thể, nhưng ta có thể xây dựng các phương pháp có độ chính xác cao để hỗ trợ trong việc sửa lỗi sai.

# 2 Dữ liệu

Việc lựa chọn dữ liệu phù hợp với công việc là một phần tối quan trọng. Việc phục hồi dấu câu thường được thực hiện trên những câu ngắn, hiếm khi có một đoạn văn dài không có dấu câu trong thực tế. Trong bài toán này chúng tôi đánh giá mô hình dựa trên 3 bộ dữ liệu.

## 2.1 Bộ dữ liệu 1

Bộ dữ liệu 1 bao gồm 101.000 câu, là các đầu mục của các bài báo trên internet. Lý do chúng tôi sử dụng bộ dữ liệu này là do các đề mục thường là những câu ngắn gọn, súc tích và các đầu mục này đến từ rất nhiều các thể loại khác nhau. Chiều dài các câu thay đổi từ 1 đến 56 từ. Chúng tôi chia ra 100.000 câu để huấn luyện và đánh giá trên 1000 câu còn lại.

## 2.2 Bộ dữ liệu 2

Bộ dữ liệu 2 bao gồm 202.273 câu là bộ dữ liệu được giảng viên cung cấp. Dữ liệu huấn luyện gồm có 201.775 câu và dữ liệu đánh giá gồm 499 câu. Về đánh giá tổng quan, bộ dữ liệu có chiều dài các câu khá khác nhau, chiều dài các câu thay đổi từ 2 đến 768 từ.

## 2.3 Bộ dữ liệu 3

Bộ dữ liệu 3 bao gồm 2.001.000 câu, bao gồm cả những câu trong bộ dữ liệu 1 (1000 câu để đánh giá giống nhau). Các câu này cũng là đề mục của các bài báo trên internet. Độ dài của các câu trong bộ dữ liệu này thay đổi từ 1 đến 62 từ. Lý do chúng tôi sử dụng bộ dữ liệu này là muốn thử tính scale của các mô hình đối với một lượng lớn dữ liệu.

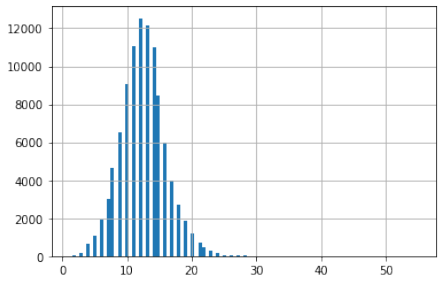
## 2.4 Tiền xử lý dữ liệu

Chúng tôi thực hiện tiền xử lý dữ liệu thông qua các bước:

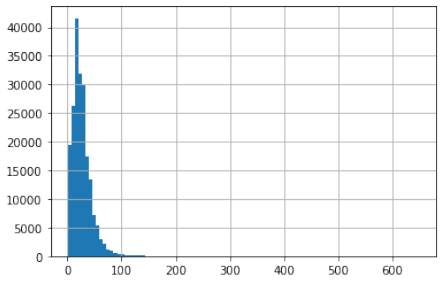
- Lọc bỏ tất cả các kí tự không thuộc trong tiếng việt.
- Bỏ dấu của tất cả các kí tự có dấu để tạo thành câu nguồn.

# 3 Phương pháp

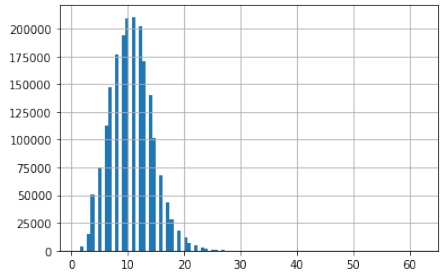
Bài toán này có thể đưa về dạng bài toán dịch máy với ngôn ngữ gốc là gồm các từ không dấu, ngôn ngữ đích gồm các từ có dấu. Trong bài toán dịch máy, kiến trúc deep learning được sử dụng rộng rãi nhất là kiến trúc encoder-decoder. Đặc biệt trong bài toán này ta có thể sử dụng mô hình n-gram bằng cách tạo ra tất cả trường hợp dấu của từ đó vì nó hữu hạn, sau đó có thể tính xác suất của từ đó bằng mô hình n-gram.




(a) Phân bố độ dài trong bộ dữ liệu 1.




(b) Phân bố độ dài trong bộ dữ liệu 2.

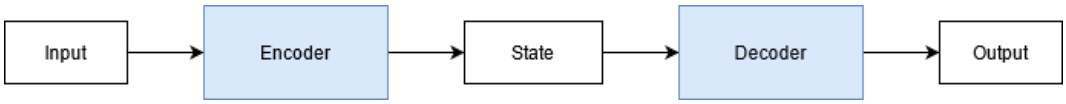



(c) Phân bố độ dài trong bộ dữ liệu 3.

Hình 1: Phân bố độ dài trên các tập dữ liệu.

## 3.1 Cấu trúc Encoder-Decoder

Cấu trúc encoder-decoder được sử dụng rộng rãi trong bài toán dịch máy. Về tổng quan, cấu trúc gồm 2 phần chính, encoder và decoder. Encoder có nghiệm vụ nhận vào một câu có độ dài bất kì và biến đổi nó thành 1 trạng thái có độ dài cố định. Từ trạng thái đó, decoder sẽ sinh ra câu đích cho đến khi gặp dấu hiệu dừng câu. Các cấu trúc thường được sử dụng cho Encoder và Decoder là các mô hình mạng hồi quy (recurrent neural network) như Long Short Term Memory (LSTM) [3], Gated Recurrent Unit (GRU) [2]. Gần đây các cấu trúc như Transformers [5] cũng đạt kết quả rất cao và có thời gian training nhanh hơn do không có cấu trúc hồi quy.




Hình 2: Cấu trúc encoder decoder

## 3.2 Cơ chế Attention

Attention là một công cụ thường được tích hợp với các mạng hồi quy. Thay vì tập chung vào tất cả mọi từ đầu vào, việc dịch một từ có thể chỉ tập chung vào một vài từ quan trọng trong câu. Cơ chế attention giúp cho việc học các mối quan hệ trở nên dễ dàng hơn. Trong báo cáo này, chúng tôi phát hiện ra rằng cơ chế attention là yếu tố tối quan trọng để các mô hình deep learning có thể hội tụ nhanh chóng. Gần đây, cơ chế self-attention dùng trong [5] cũng đang được ứng dụng rộng rãi do khả năng tính toán song song của nó, giúp cho thời gian huấn luyện giảm đáng kể mà không ảnh hưởng đến độ chính xác.

## 3.3 Mô hình ngôn ngữ N-gram

Mô hình n-gram dựa trên mô hình Markov bậc n để tính toán xác suất của 1 từ thông qua ngữ cảnh của nó. Đối với những từ không biết, ta có thể nội suy ra nó bằng các phương pháp như KneserNey. Đặc biệt đối với bài toán này, ta có thể kiểm tra từng từ trong câu đầu vào, sinh ra

tất cả các trường hợp có dấu của từ đó và tính xác suất câu hiện tại. Phương pháp này hoạt động khá hiệu quả với lượng dữ liệu vừa phải, nhưng việc tìm kiếm rất lâu.

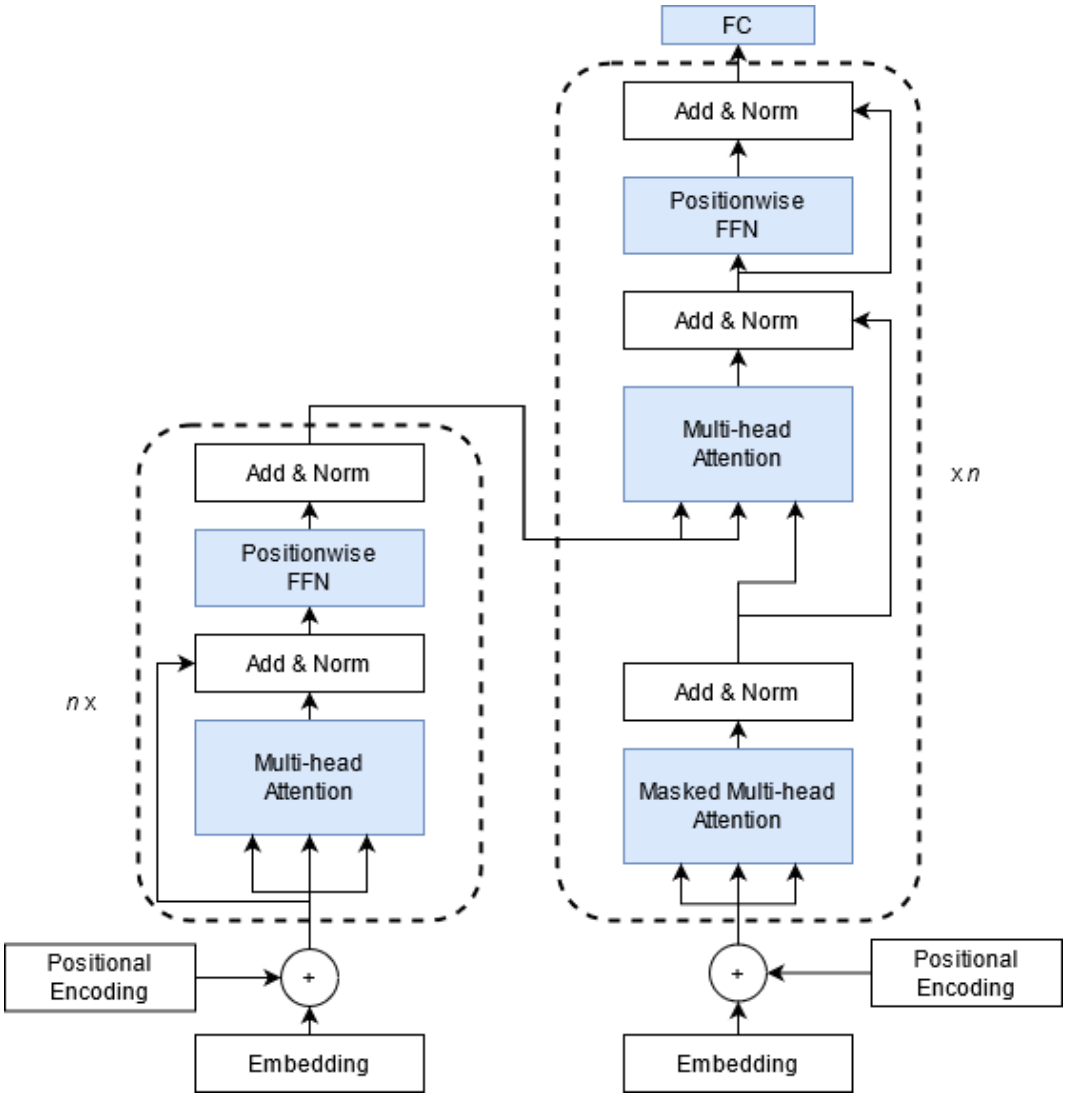
### 3.4 Phương pháp đánh giá mô hình

Đối với bài toán cụ thể này, chúng tôi sử dụng cách tính điểm dựa trên việc từ hiện tại có được dự đoán chính xác hay không. Tổng điểm sẽ được tính bằng trung bình độ chính xác trên tất cả các câu. Ví dụ: *da bao nhieu nam -> da bao nhiêu năm* (câu chính xác: *đã bao nhiêu năm*) sẽ có độ chính xác 3/4 = 0.75

## 4 Thí nghiệm

Chúng tôi thử nghiệm 3 mô hình khác nhau gồm có Transformers, GRU Encoder-Decoder, N-gram.

### 4.1 Transformers




Hình 3: Mô hình transformers

Transformer là một cấu trúc phức tạp nhưng hiệu quả được đưa ra tại (here). Các tham số quan trọng để huấn luyện mô hình là:

- Hidden size: Độ lớn của embedding
- Số heads: Số lượng head trong Multi-head Attention [5]
- Số layers: Số lượng layer trong Encoder và Decoder, ở đây ta cho số lượng layer ở cả 2 phần bằng nhau.
- Learning rate: Tốc độ học ảnh hưởng rất lớn đến khả năng hội tụ của mô hình

Mô hình transformers được chia ra làm 2 mô hình SMALL và BASE. Tham số của 2 mô hình được thể hiện trong bảng.

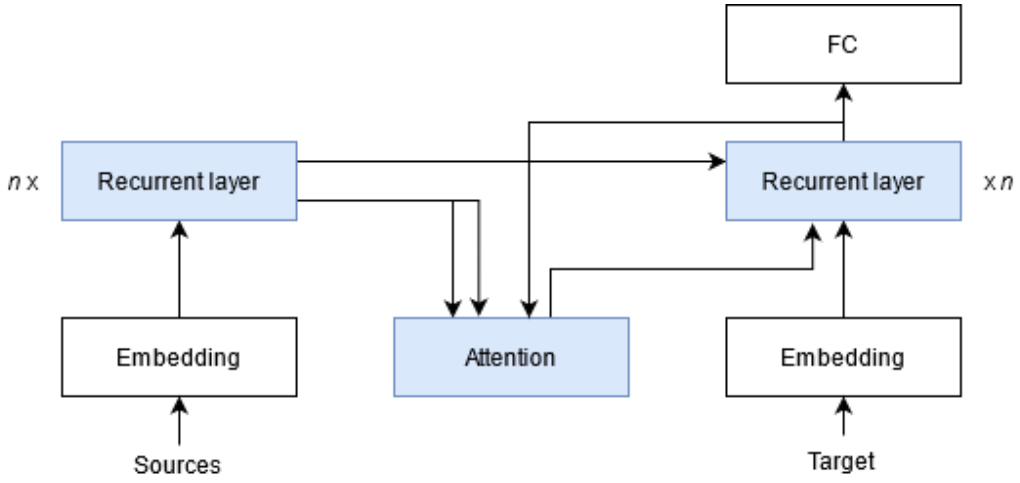
| **Mô hình** | **Hidden size** | **Số heads** | **Số layers** | **lr** | **Epochs** |
|---|---|---|---|---|---|
| SMALL | 128 | 4 | 2 | 0.001 | 20 |
| BASE | 512 | 8 | 6 | 0.0001 | 30 |

Bảng 1: Bảng tham số của mô hình

Mô hình transformers có thời gian train ngắn hơn các mô hình khác do không có cơ chế recurrent.

## 4.2 GRU Encoder Decoder

Mô hình GRU Encoder Decoder áp dụng GRU vào encoder và decoder với sự hỗ trợ của cơ chế attention trong decoder. Cụ thể hàm tính attention ở đây là additive attention [1].




Hình 4: Mô hình GRU Encoder Decoder

Các tham số quan trọng của mô hình trên là:

- Hidden size: Độ lớn của vector đầu vào
- Số layers: Số layer của GRU
- Bidirections: Sử dụng GRU 2 chiều hay không
- Learning rate: Tốc độ học

| Mô hình | Hidden size | Số layers | Bidirections | lr | Epochs |
|---|---|---|---|---|---|
| GRU | 128 | 1 | Không | 0.001 | 20 |

Bảng 2: Bảng tham số của mô hình

### 4.3 N-gram

Phương pháp n-gram là baseline được tham khảo ở [4]. Chúng tôi sử dụng mô hình n-gram với n = 3, phương pháp làm mịn Kneser–Ney. Với mỗi từ đang xét hiện tại, chúng tôi sinh ra tất cả các trường hợp có dấu của từ đó, ví dụ từ *tinh* có thể sinh ra các trường hợp {*tinh*, *tình*, *tính*, *tĩnh*, *tỉnh*, *tịnh*}. Sau đó ta chọn trường hợp có xác suất cao nhất thông qua mô hình n-gram.

### 4.4 Beam search

Trong tất cả các phương pháp, để tạo ra kết quả cuối cùng, chúng tôi đều sử dụng beam search với độ lớn 4. Việc thực hiện beam search trong các mô hình deep learning có lợi thế hơn do có thể tận dụng khả năng tính toán song song.

## 5 Đánh giá

Đánh giá của các mô hình được thể hiện trong các bảng dưới dựa trên 3 tiêu chí.

### 5.1 Độ chính xác

| Mô hình | Dữ liệu 1 (%) | Dữ liệu 2 (%) |
|---|---|---|
| Transformers (SMALL) | **0.742** | 0.766 |
| GRU Encoder Decoder | 0.712 | 0.712 |
| N-gram | 0.722 | **0.813** |

Bảng 3: Độ chính xác của các mô hình

### 5.2 Thời gian train

| Mô hình | Thời gian train DL1 (h) | Thời gian train DL2 (h) |
|---|---|---|
| Transformers (SMALL) | 0.19 | 1.0 |
| GRU Encoder Decoder | 0.71 | 2.46 |
| N-gram | 0.01 | 0.03 |

Bảng 4: Thời gian huấn luyện của các mô hình

### 5.3 Thời gian đánh giá

Nhận thấy mô hình Transformers [5] có khả năng scale tốt, chúng tôi tiếp tục huấn luyện mô hình Transformers (BASE) với số lượng tham số lớn hơn và thực hiện đánh giá mô hình này.

### 5.4 Đánh giá tổng quan

Các mô hình deep learning thường có thời gian huấn luyện lâu hơn, cần nhiều dữ liệu hơn nhưng có khả năng suy luận nhanh hơn các mô hình xác suất. Trên một lượng dữ liệu đủ lớn, mô hình như Transformers có khả năng scale thời gian huấn luyện tốt và cho kết quả khá tốt.

| Mô hình | Thời gian đánh giá DL1 (h) | Thời gian đánh giá DL2 (h) |
|---|---|---|
| Transformers (SMALL) | 0.01 | 0.005 |
| GRU Encoder Decoder | 0.007 | 0.004 |
| N-gram | 2.27 | 2.13 |

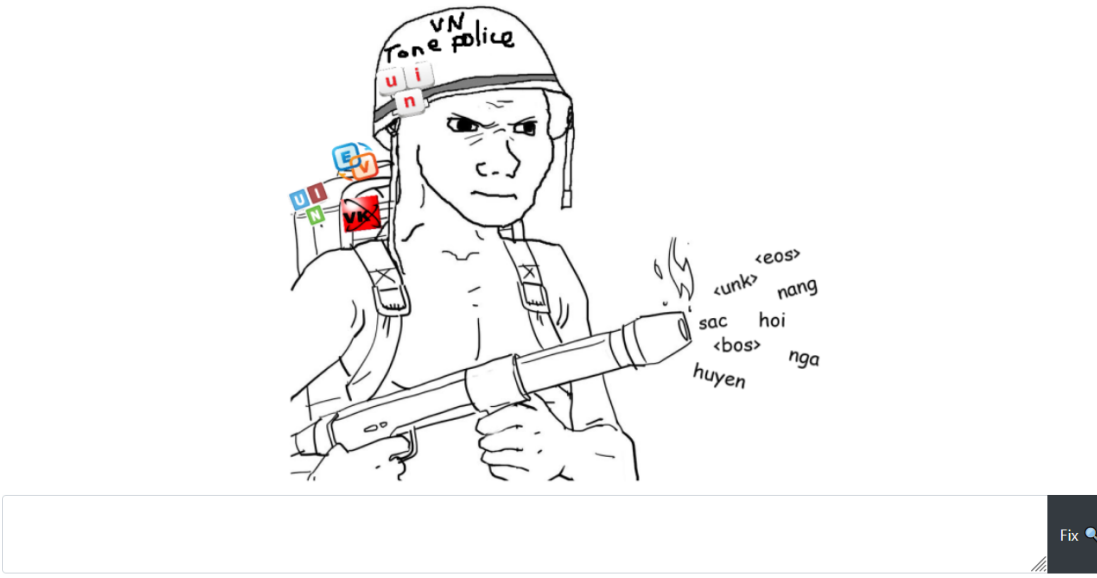
Bảng 5: Thời gian train của các mô hình

| Mô hình | Độ chính xác DL1 (%) | Độ chính xác DL2(%) |
|---|---|---|
| Transformers (BASE) | 0.937 | 0.818 (One shot) |

Bảng 6: Độ chính xác của mô hình Transformers (BASE)

# 6 Giao diện web

Để tiện hơn cho việc đánh giá cũng như tương tác trực tiếp với mô hình, chúng tôi xây dựng 1 trang web với Flask backend và Bootstrap frontend. Sau khi xử lý, người dùng còn có thể xem giải thích về kết quả của mô hình như self-attention với Transformers, attention với GRU Encoder Decoder và điểm log xác suất của mô hình n-gram. Giao diện được thể hiện trên Hình 5, Hình 6 Hình 7, Hình 8

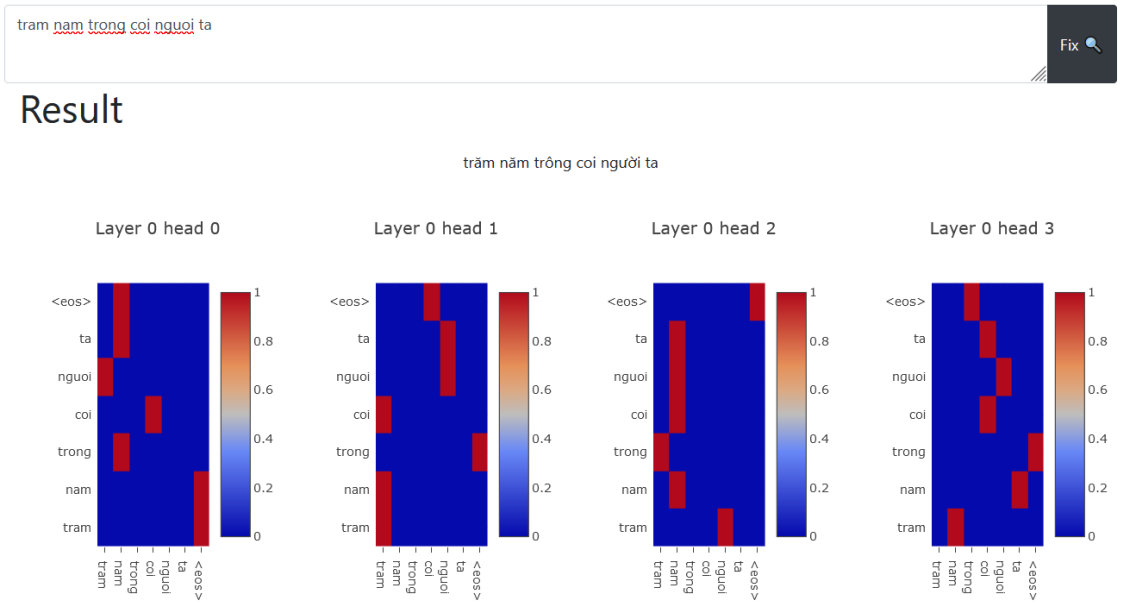



Hình 5: Giao diện để nhập dữ liệu

# 7 Kết luận

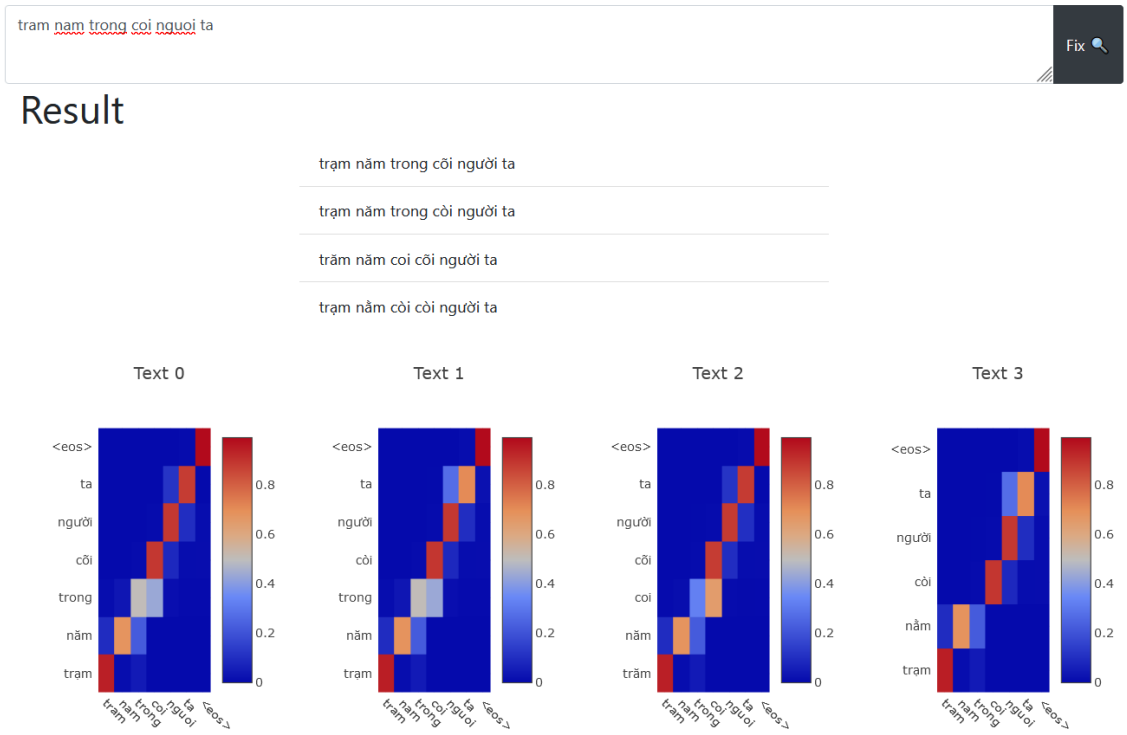
Trong thí nghiệm này, chúng tôi đã thử nghiệm 3 mô hình được sử dụng rộng rãi nhất trong bài toán dịch máy, bao gồm mô hình Transformers, GRU Encoder Decoder, N-gram. Các mô hình deep learning có cấu trúc phức tạp, thời gian huấn luyện lâu nhưng có thời gian suy luận nhanh hơn mô hình N-gram. Với lượng data lớn, mô hình Transformers có độ chính xác khá cao với thời gian huấn luyện trung bình. Mặc dù không thể đánh giá mô hình n-gram trên lượng dữ liệu lớn, chúng tôi tin rằng mô hình này cũng có thể đạt được độ chính xác cao khi câu đầu vào ngắn vừa phải. Trong các câu dài, các mô hình deep learning có thể có lợi thế hơn do có cơ chế attention.

Cũng trong thí nghiệm này, chúng tôi chưa thể thử nghiệm các phương pháp với dữ liệu ở mức kí tự. Đầu ra hiện tại mới chỉ là các từ có dấu từ input, bỏ qua các kí tự không phải là từ.

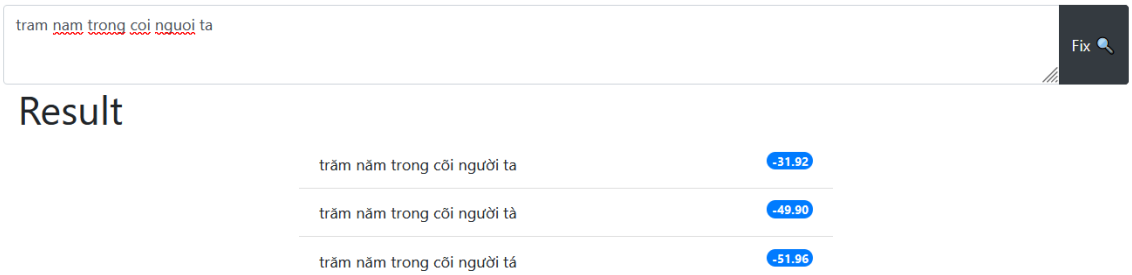



Hình 6: Kết quả của Transformers gồm câu đầu ra và self-attention

Điều này có thể cải thiện bằng cách không loại bỏ các kí tự đó và vẫn cho vào từ điển, hoặc ta có thể hậu xử lý kết quả. Đây cũng là những điều mà chúng tôi muốn cải thiện trong tương lai.




Hình 7: Kết quả của GRU Encoder Decoder là câu đầu ra và attention




Hình 8: Kết quả của N-gram là câu đầu ra và điểm log xác suất

# Tài liệu


[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] B.T. Tung. Language Modeling - Mô hình ngôn ngữ và bài toán thêm dấu câu trong Tiếng Việt.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.