CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI

4.1 GIỚI THIỆU VỀ PYTHON VÀ THƯ VIỆN TENSORFLOW

4.1.1 Python

Mã nguồn xây dựng mô hình huấn luyện của đề tài được phát triển dựa trên Python. Python là một ngôn ngữ lập trình thông dịch được sử dụng rất phổ biến trong lĩnh vực khoa học máy tính nhờ những ưu điểm sau:

❖ Đa nền tảng

Python có thể chạy trên nhiều hệ điều hành như Windows, MacOS, Linux/Unix và một số hệ điều hành khác trên máy tính. Ngoài ra, Python còn có cả những phiên bản chạy được trên .NET, máy ảo Java. Tất cả chỉ với cùng một mã nguồn cho một công việc.

❖ Đơn giản

Python có cú pháp rất đơn giản, rõ ràng. Cú pháp của Python dễ viết và dễ đọc hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như Java, C/C++, C#, JavaScript, ... Điều này cũng giúp cho nhà phát triển tập trung vào việc phát triển giải pháp thay vì cú pháp.

❖ Mã nguồn mở

Python là một dự án mã nguồn mở nên nhà phát triển có thể thoải mái sử dụng cho các mục đích cá nhân và vì vậy nên cộng đồng phát triển Python thường xuyên đưa ra những bản cập nhật mới nhằm tăng trải nghiệm cũng như tối ưu hoá Python.

❖ Nhiều thư viện hỗ trợ

Python có một khối lượng lớn các thư viện tiêu chuẩn giúp cho công việc của nhà phát triển trở nên dễ dàng hơn rất nhiều, đặc biệt là các thư viện xử lý toán học của Python cực kỳ đa dạng và mạnh mẽ.

4.1.2 Tensorflow

Thư viện Tensorflow được sử dụng trong việc tính toán các biểu đồ và các dữ liệu dưới dạng số hoá trong sản phẩm khoá luận. Là một thư viện mã nguồn mở hỗ trợ mạnh mẽ các phép toán học để tính toán trong máy học. Để xây dựng một mô hình huấn luyện cho đề tài, nhóm sử dụng các giao diện lập trình cấp thấp (low level APIs) mà Tensorflow cung cấp:

❖ Tensor

Đây là một sự khái quát hóa các vectơ và ma trận cho các kích thước có khả năng cao hơn. Là cấu trúc dữ liệu đại diện cho tất cả các loại dữ liệu trong Tensorflow. Một tensor sẽ có 3 thuộc tính cơ bản nhất bao gồm:

- Số bậc (rank): giúp phân loại dữ liệu của tensor. (Scalar, Vector, Matrix, N- Tensor)

- Số chiều (shape): giúp xác định mức độ tương hợp giữa các tensor khi thực hiện tính toán.
- Kiểu dữ liệu (type): kiểu dữ liệu cho toàn bộ các thành phần (elements) trong tensor.

❖ Graph

Đây là một loại đồ thị với các đỉnh (node) là đại diện cho biến đầu vào hoặc một phép tính toán và các cạnh (edge) là đại diện cho dữ liệu truyền bên trong đồ thị tức dữ liệu đầu vào và đầu ra của các phép tính tại một đỉnh. Và trong tensorflow, tất cả thành phần bên trong một đồ thị đều ở dạng tensor. Cách xử lý tính toán theo hướng đồ thị này có thể giúp tensorflow tận dụng được khả năng tính toán song song bằng việc chia tách các phép toán độc lập và khả năng phân tán khi chia nhỏ công việc xử lý cho nhiều CPU, GPU khác nhau.

❖ Session

Đây là một phiên xử lý được định nghĩa trong thư viện tensorflow. Một đối tượng phiên (session) cung cấp quyền truy cập vào các thiết bị trong máy cục bộ và các thiết bị từ xa bằng cách sử dụng thời gian chạy phân tán. Nó cũng lưu trữ thông tin về đồ thị (graph) để có thể chạy cùng một tính toán hiệu quả nhiều lần. Nếu không có phiên (session), mọi tính toán trong đồ thị (graph) sẽ gần như không được triển khai.

4.2 DỮ LIỆU HUẤN LUYỆN MÔ HÌNH

Để huấn luyện một hệ thống dịch máy từ tiếng Anh sang tiếng Việt dựa trên mô hình nhóm sinh viên phát triển thì bộ dữ liệu nhóm sử dụng là "IWSLT'15 English-Vietnamese data" bao gồm khoảng 133.000 câu do đại học Stanford phát triển (nhóm sinh viên chỉ lấy những câu có độ dài bé 100 nên chỉ có khoảng 100.000 câu) có hai thành phần chính bao gồm:

- ❖ Văn bản tiếng Anh

  Để huấn luyện hệ thống dịch máy từ tiếng Anh sang tiếng Việt đủ tốt thì lượng dữ liệu văn bản dùng để huấn luyện cũng phải đủ nhiều và đủ tốt. Nhóm sinh viên đã thu thập được khoảng 100.000 câu song ngữ để tiến hành huấn luyện. Ngoài ra dữ liệu khi huấn luyện cũng cần điều chỉnh sao cho độ dài bé hơn 100 từ để bảo đảm mô hình huấn luyện không bị thiếu tài nguyên.

- ❖ Văn bản tiếng Việt

  Là bản dịch tương ứng với nội dung của câu tiếng Anh. Dữ liệu tiếng Việt với khoảng 100.000 câu và để bảo đảm mô hình huấn luyện không thiếu tài nguyên ta cũng nên hạn chế độ dài lớn hơn 100 từ vì nhóm sinh viên giới hạn độ dài câu. Nếu câu dài hơn sẽ bị cắt bỏ và mất đi các nội dung quan trọng, mô hình sẽ huấn luyện lâu hơn, sai sót.

Trong quá trình thu thập dữ liệu, nhóm sinh viên đã gặp rất nhiều vấn đề về chất lượng dữ liệu như: các tập dữ liệu song ngữ English – Vietnamese có khá nhiều với các dự án như là …. Tuy nhiên các dự án này lại không

công khai dữ liệu nên nhóm sinh viên phải thu thập khắp nơi. Đối với những mẫu có mức độ sai lệch nhỏ nhóm cố gắng tinh chỉnh sao cho phù hợp nhất. Những mẫu bị sai lệch nhiều hoặc chất lượng quá thấp buộc nhóm sinh viên phải bỏ. Việc này một phần sẽ giảm bớt tình trạng gây nhiễu cho mô hình trong quá trình huấn luyện. Điều này dẫn đến thời gian huấn luyện mô hình còn khoảng (?323332) giờ cho khoảng 100.000 câu song ngữ. Trong đó, dữ liệu được chia nhỏ thành 3 bộ train, dev, test với kích thước như sau:

❖ Bộ train: chứa khoảng 100.000 câu song ngữ English-Vietnamese.

❖ Bộ dev: chứa khoảng 1500 câu song ngữ English-Vietnamese.

❖ Bộ test: chứa khoảng 1200 câu song ngữ English-Vietnamese.

4.3 CÀI ĐẶT

4.3.1 Giới thiệu

Mã nguồn được nhóm sinh viên phát triển dựa trên tham khảo các bài báo như: *Sequence to Sequence Learning with Neural Networks* do nhóm tác giả đến từ google được ông bố vào năm 2014 tại Silicon Valley AI Lab đã trình bày ý tưởng cụ thể để xây dựng một mô hình mạng nơ-ron hồi quy tối ưu với hướng đi mới so với các hệ thống dịch máy truyền thống kết hợp cùng với cơ chế chú ý (Attention mechanism) từ bài báo *Effective Approaches to Attention-based Neural Machine* được thực hiện bởi nhóm tác giả đến từ đại học Stanford vào năm 2015.

Từ đó nhóm sinh viên tự phát triển mô hình dịch máy cho tác vụ dịch tiếng Anh sang tiếng Việt để phục vụ cho luận văn. Mô hình được pháp triển trên Python3 và thư viện Tensorflow là chính.

4.3.2 Cài đặt

Đầu tiên ta cần tải về mã nguồn mô hình dịch máy của nhóm sinh viên phát triển từ github (https://github.com/nmtri1912/Model).

Phần hướng dẫn cài đặt của nhóm sinh viên yêu cầu bắt buộc về những thư viện cũng như công cụ mà nhóm sinh viên có khuyến nghị trên liên kết github phía trên để có thể chạy mô hình của nhóm:

- ❖ Python 3.6
- ❖ Tensorflow 1.x
- ❖ Hệ điều hành MacOS hoặc Linux, Ubuntu

Để huấn luyện mô hình dịch máy cho tác vụ dịch ta cần cài đặt theo hướng dẫn của nhóm sinh viên để tránh gặp lỗi không đáng có.

Để huấn luyện mô hình với bộ dữ liệu câu Anh-Việt, ta cần chuyển thành các số để mô hình có thể huấn luyện và phương pháp đó là sử dụng từ nhúng (word embedding).

Nhóm sinh viên sử dụng pre-trained model Word2vec tại http://vectors.nlpl.eu/repository/ cho tiếng Anh và tiếng Việt có thông số như sau:

❖ English CoNLL17 corpus (ID = 40): được xây dựng dựa trên thuật toán “Word2Vec continuons Skipgram” với 100 chiều và hơn 4.000.000 từ.

❖ Vietnamese CoNLL17 corpus(ID = 74): được xây dựng dựa trên thuật toán “Word2Vec continouns Skipgram” với 100 chiều và khoảng 3.800.000 từ.

4.4 HUẤN LUYỆN MÔ HÌNH

Để mô hình có thể sử dụng mô hình để đưa ra các dự đoán dịch chính xác, ta cần tìm ra các tham số phù hợp cho mô hình, nhóm sinh viên tham khảo và tiến hành công việc huấn luyện mô hình (điều chỉnh các siêu tham số - hyperparameters) để tìm ra những giá trị tối ưu để mô hình dự đoán chính xác nhất.

Quá trình huấn luyện được nhóm triển khai trên Google Colab có cấu hình GPU 12GB. Sau một thời gian huấn luyện nhóm sinh viên thống kê được các tham số như sau:

- Epoch = 70: một epoch là một lần duyệt qua hết tất cả số lượng các mẫu trong tập huấn luyện.
- batch_size = 64: số lượng mẫu được sử dụng cho mỗi lần cập nhật trọng số.
- num_layers = 2: số lớp của mô hình.
- num_hiddens: độ rộng của một lớp được dùng khi khởi tạo các lớp trong mô hình.
- learning_rate = 0.001: tỉ lệ học tập của mô hình.

- keep_prob = 0.85: tỉ lệ giữ lại cho các quá trình chuyển tiếp giữa các lớp trong mô hình.
- beam-width = 10: độ rộng của thuật toán tìm kiếm chùm tia (Beam Search).

Để huấn luyện mô hình chúng ta cần chạy câu lệnh:

```
python3 train.py
  --language_src data/train-en-vi/train.en
  --language_targ data/train-en-vi/train.vi
  --vocab_src vocab_english/
  --vocab_targ vocab_vietnamese/
  --word_emb_src word_embedding/model_en.bin
  --word_emb_targ word_embedding/model_vn.bin
  --num_layer 1
  --num_hiddens 512
  --learning_rate 0.001
  --keep_prob 0.85
  --beam_width 10
  --batch_size 64
  --checkpoint NMT.ckpt
```

Sau khi huấn luyện với các tham số như trên nhóm sinh viên có bảng đánh giá như sau:

Mô hình	Bộ dữ liệu	BLEU score

4.5 ĐÓNG GÓI MÔ HÌNH

Nhóm sinh viên lưu các tham số mô hình học được sau mỗi lần chạy xong 1 epoch (một lần duyệt qua toàn tập huấn luyện) với định dạng **NMT.ckpt.** Tập tin này có thể hiểu là các tham số được chọn lọc trong quá trình huấn luyện.

Để sử dụng tập tin này để thực hiện tác vụ dịch máy, chúng ta cần định nghĩa lại một số thư cần thiết như: từ điển word2int, int2word, từ nhúng (word embedding), xử lí đầu vào và mô hình.

4.6 XÂY DỰNG MÁY CHỦ (SERVER)

Flask Framework và AWS EC2 (hoặc AWS Elastic Beanstalk) là hai nền tảng được nhóm sinh viên chọn để xây dựng hệ thống máy chủ nhằm đóng vai trò làm cầu nối giữa ứng dụng và mô hình nhận dạng âm thanh. Với các yếu tô như tốc độ triển khai nhanh gọn, sự tiện ích và tính thông dụng nên việc chọn hai nền tảng này để xây dựng máy chủ là quyết định phù hợp với nhu cầu đặt ra của nhóm sinh viên.

Hệ thống máy chủ trong giới hạn luận văn này sẽ cung cấp ra bên ngoài duy nhất một giao diện lập trình ứng dụng (Application Programming Interface-API) để chuyển đổi văn bản tiếng Anh (dạng text) nhận được và trả về dữ liệu văn bản tiếng Việt tương ứng.

Trong khi đó Web ứng dụng để sử dụng API mà hệ thống cung cấp, nhóm sinh viên sử dụng Reactjs Framework để xây dựng giao diện và được triển khai trực tiếp lên Heroku.

4.7 MỘT SỐ VẤN ĐỀ PHÁT SINH VÀ GIẢI PHÁP

- Thiếu nguồn dữ liệu ->

- Thiếu tài nguyên để huấn luyện mô hình (GPU) -> Sử dụng dịch vụ google colab

- Khi deploy lên server gặp phải trường hợp front-end gọi API được 2 đến 3 lần thì server bị tắt -> cài nginx.

- Khi huấn luyện trên google colab thì gặp phải vấn đề giới hạn của google colab: train tối đa khoảng 10-12 tiếng sẽ bị mất kết nối, hoặc có sự cố phát sinh thì sẽ mất kết quả huấn luyện trước đó -> đặt checkpoint để lưu lại kết quả sau mỗi epoch. Khi mất kết nối thì ta chỉ cần tải lại checkpoint để huấn luyện tiếp mà không phải huấn luyện lại từ đầu.

4.8 TỔNG KẾT

Trong chương 4, nhóm sinh viên đã trình bày về cách thức cài đặt và triển khai cho các thành phần bao gồm hệ thống dịch máy, trang web chạy thử API của hệ thống. Nội dung chi tiết cho một số phần cài đặt được nhóm sinh viên trình bày chi tiết ở phần phụ lục, chương 5 sẽ là các tổng kết về quá trình thược hiện luận văn của nhóm sinh viên.