Báo cáo

* Báo cáo em làm, em chỉ liệt kê những thay đổi so với lần cuối cùng mà em gặp thầy, để giải thích rõ hơn thì em muốn hẹn gặp offline thầy ạ.

1. Dataset:

* Trong lúc tìm kiếm, research những bài báo khác thì em kiếm được thêm data từ 2000 -> ~5000 problems (trung bình 3code/problem).

1. Preprocessing:

* Sử dụng cpp-clang để xoá những hàm không sử dụng.

1. Splitting dataset

* Chia dataset theo tỉ lệ 8/2 với train/test set
* Sử dụng method iterative stratification để chia problems, cụ thể là multilabelstratifiedshufflesplit.
* Nếu chia 1 cách random (train\_test\_split), sau khi chia problems, các target(tags) sẽ không cân bằng, nghĩa là có các tag xuất hiện rất ít hoặc gần như không có trong test set
* Method này tạo ra train/test set có tỉ lệ target cân bằng (balance).

1. Word Embedding

* Có 2 method word embedding định dùng là Word2VEC và BERT.
* Ban đầu em tiếp cận word2vec thì thấy việc train word2vec quá lâu nên em đã tìm và sử dụng pretrained model, nhưng word2vec có điểm yếu là Out of Vocabulary (OOV). Việc OOV dẫn đến những từ quan trọng (sort, compare,…) không nằm trong embedding
* Perfomance của model tăng, nhưng em thấy không tốt bằng BERT.
* Sử dụng BERT, cụ thể là dùng pretrained model bert-base-uncased.

1. Predict label

* Em đã sử dụng KNN để predict multi label thay vì phải chia 1 label, 2 label như trước, nhưng độ chính xác không được tốt (~45%).

1. Tại sao em hỏi thầy có được thay đổi tên đề tài trong khi nghiên cứu hay không ?

* Trong lúc research thì em mới thấy 1 paper từ năm 2020.

<https://www.researchgate.net/publication/346814822_AlgoLabel_A_Large_Dataset_for_Multi-Label_Classification_of_Algorithmic_Challenges>

* Tóm tắt lại về paper này thì mục đích của nó là multi label classification các competitive programming problem, trong đó code classification của họ theo ý tưởng của paper code2vec (AST) nên trùng lặp với ý tưởng của mình.

Target của paper chỉ gồm 4 classes là implementation, dp, graph và math.

* Vì vậy trong lúc tìm research tìm idea thì em có thấy những paper NLP xử lí code gần đây, họ sử dụng code property graph (CPG) làm input.

<https://ieeexplore.ieee.org/document/6956589>

<https://arxiv.org/pdf/2106.10918.pdf>

* Code property graph gồm những các graph biểu diễn source code, ở đây gồm 3 loại chính họ dùng là abstract syntax tree (AST), control flow graph (CFG) và program dependence graph (PDG),…
* **Em đang generate ra các path của những graph trên và apply vào model, sau khi xong em sẽ báo cáo tiếp cho thầy.**