

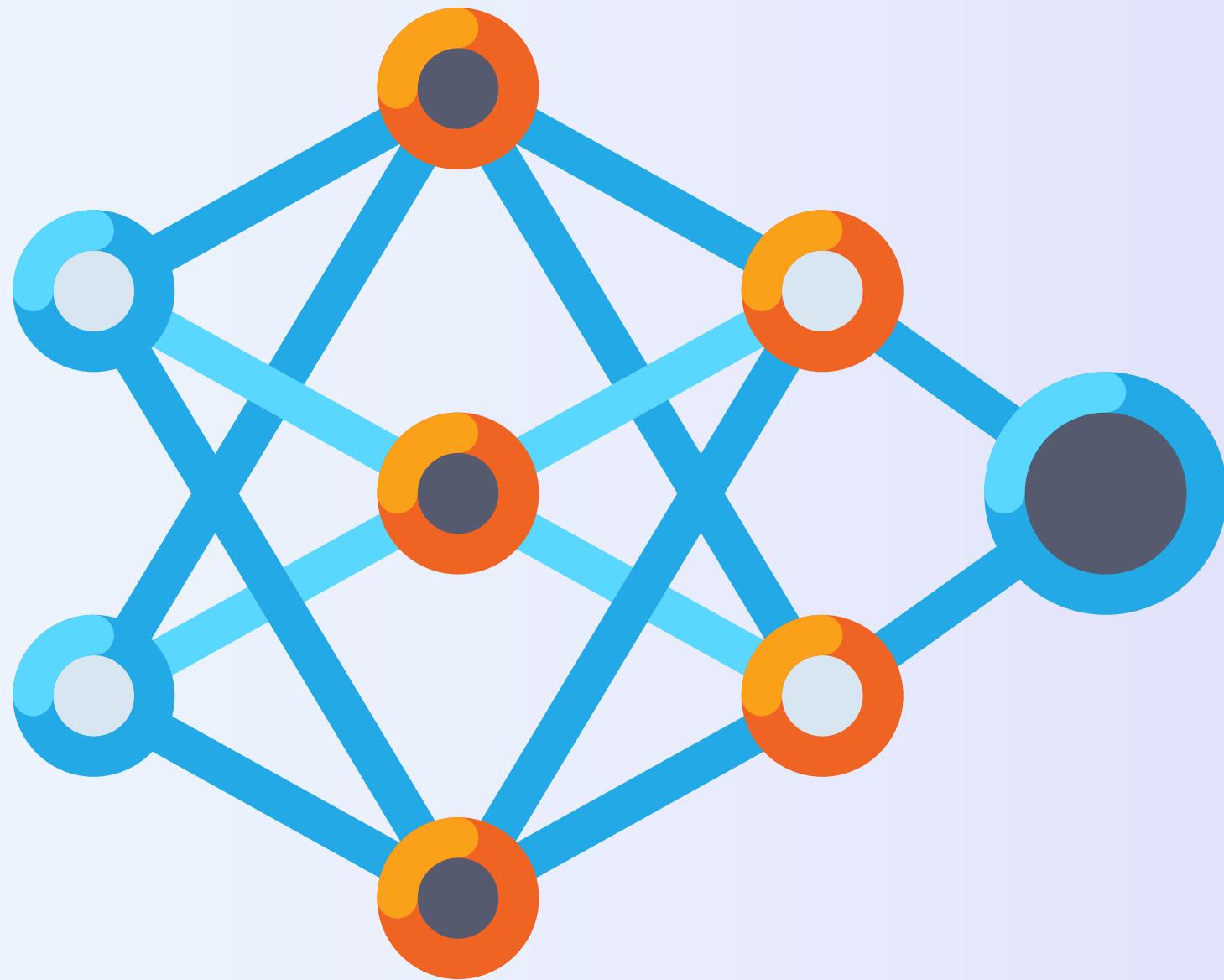
NỘI DUNG

Nội dung sẽ phủ từ lý thuyết mô hình đến kết quả thực nghiệm của các bài toán Q2, Q3 trong đồ án.

1 MÔ HÌNH & THỰC NGHIỆM Q2

2 MÔ HÌNH & THỰC NGHIỆM Q3

3 THẢO LUẬN & SO SÁNH

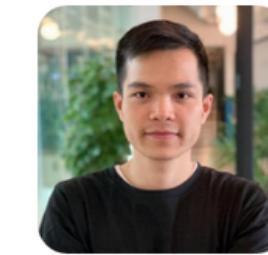


1
**MÔ HÌNH
THỰC NGHIỆM
Q2**

1 **EMBEDDING**

Sử dụng Word2vec với các mô hình pretrained cụ thể là PhoW2V & FastText để nhúng các từ trước khi đưa vào mô hình mạng.

**datquocnguyen/
PhoW2V**



Pre-trained Word2Vec syllable- and word-level embeddings for Vietnamese



1
Contributor



0
Issues



43
Stars



3
Forks



**facebookresearch/
fastText**



Library for fast text representation and classification.



55
Contributors



6k
Used by



25k
Stars



5k
Forks



Tên mô hình	Âm tiết/từ	Chiều nhúng
PhoW2V_syllables_100dims	Syllable-level	100
PhoW2V_syllables_300dims	Syllable-level	300
PhoW2V_words_100dims	Word-level	100
PhoW2V_words_300dims	Word-level	300

PHOW2V

Là model Word2Vec được huấn luyện
trên tập ngữ liệu tiếng Việt

fastText

Là model Word2Vec được huấn luyện bởi Facebook AI Research. FastText hỗ trợ 157 ngôn ngữ, trong đó có tiếng Việt.

- ✓ Số ngôn ngữ hỗ trợ: 157
- ✓ Đồ án sử dụng FastText CBOW, position weight, kích thước chiều là 300

2

CLASSIFIER

Sử dụng CNN để nhúng các câu và lớp fully connected để thực hiện việc phân loại.

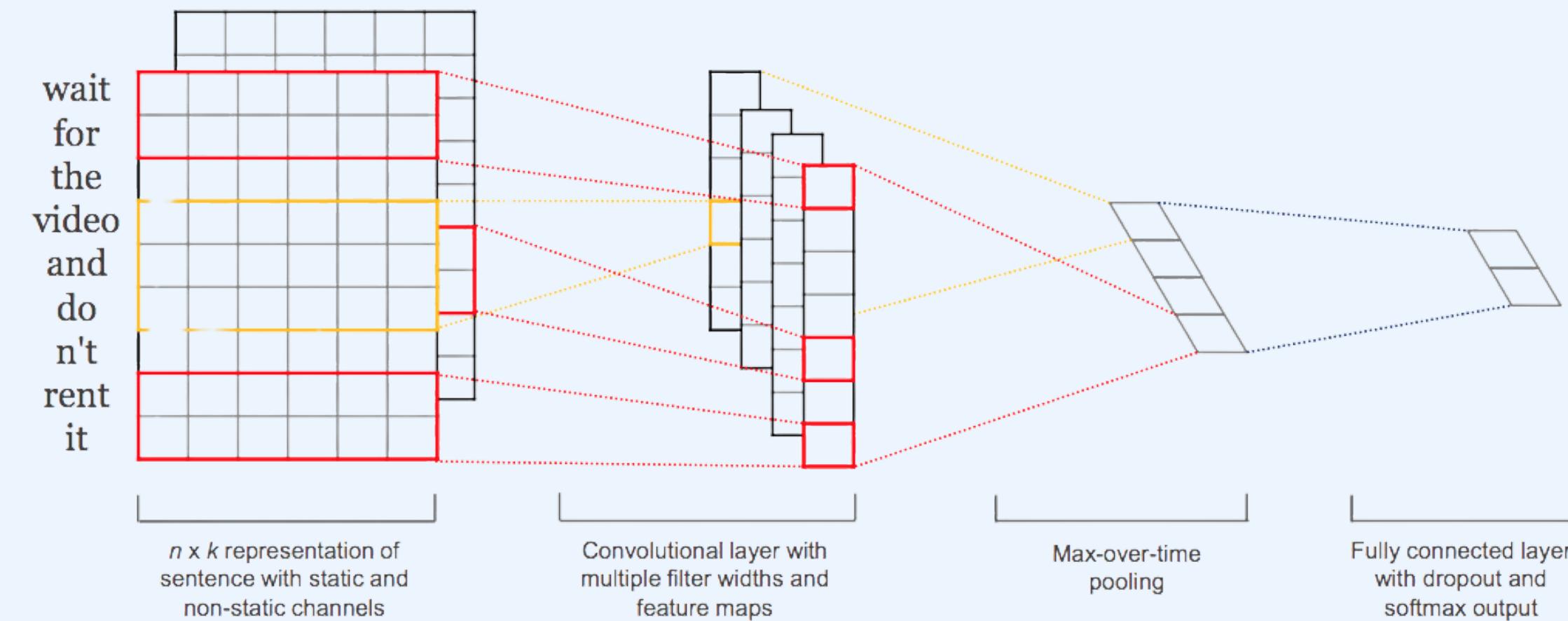
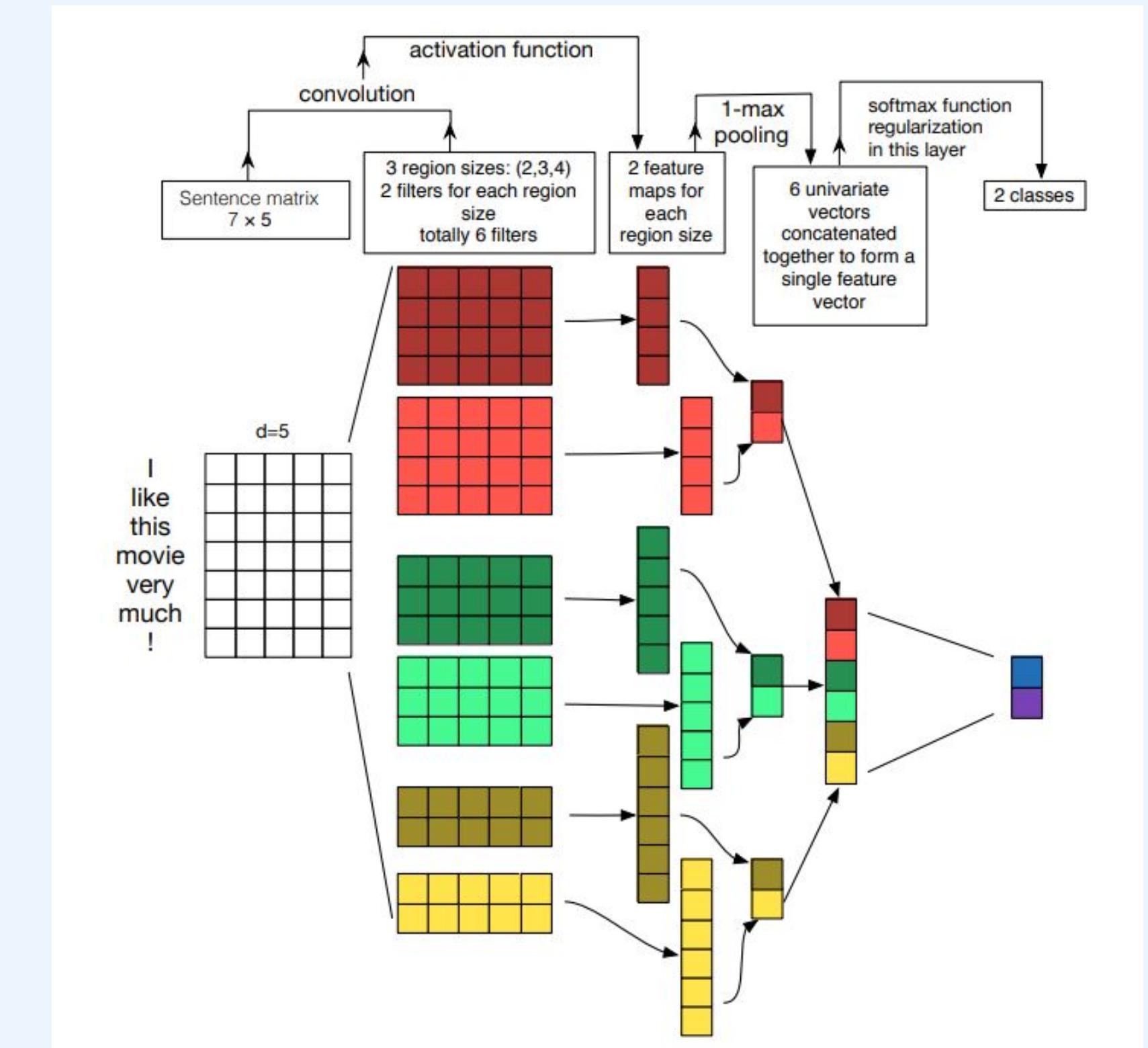


Figure 1: Model architecture with two channels for an example sentence.

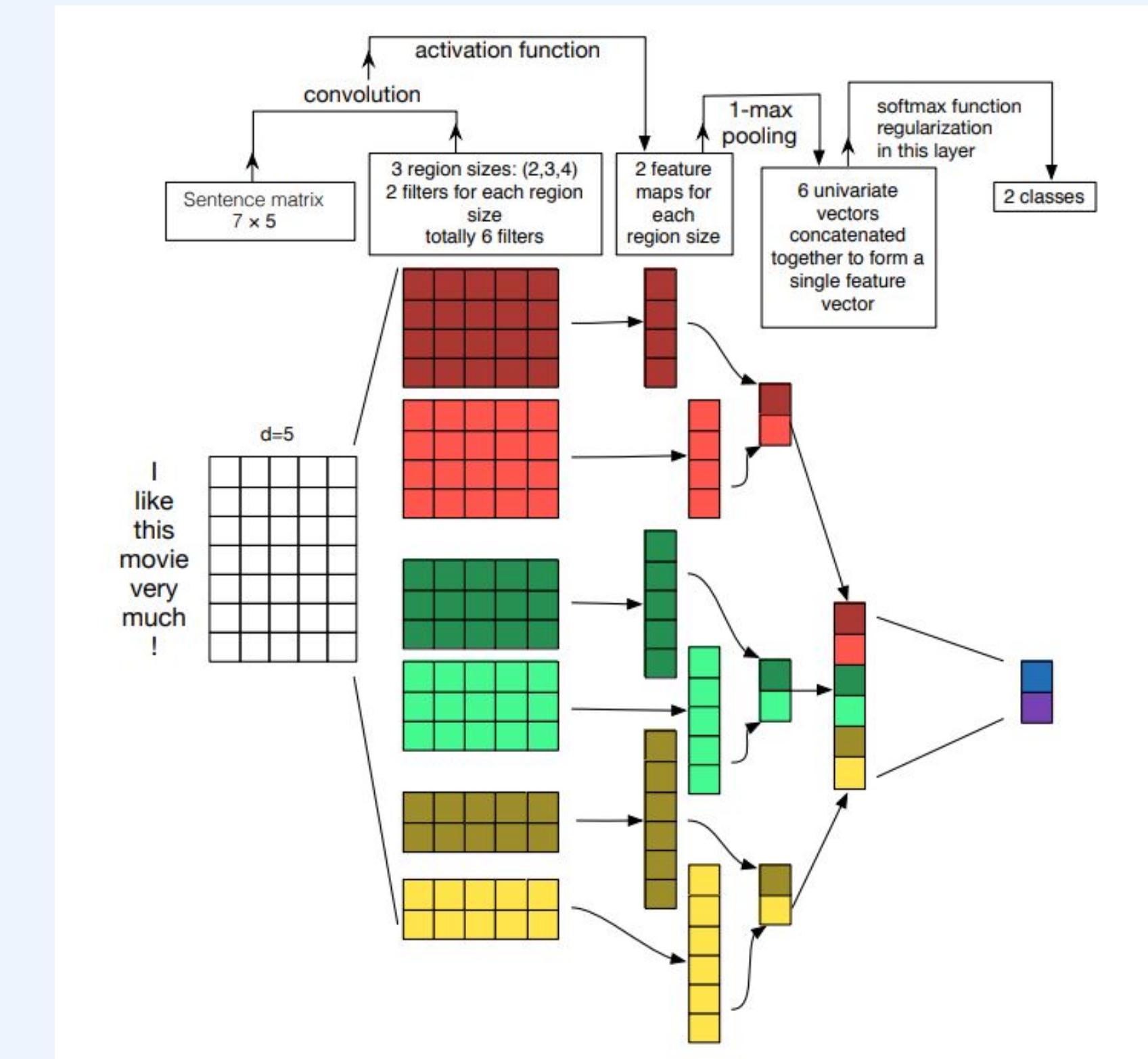
SENTENCE EMBEDDING

- ✓ Là một mạng CNN - 1D dùng để trích xuất đặc trưng quan hệ giữa các từ trong 1 câu.
- ✓ Mạng CNN này sẽ dùng các bộ lọc [filter hay kernel] với các kích thước k khác nhau để trích xuất đặc trưng quan hệ giữa k từ liên tiếp nhau trong 1 câu
- ✓ Áp dụng max-pooling cho các feature map
- ✓ Ghép các giá trị đã pooling thành 1 vector hoàn chỉnh, đây chính là embedding của toàn bộ 1 câu.



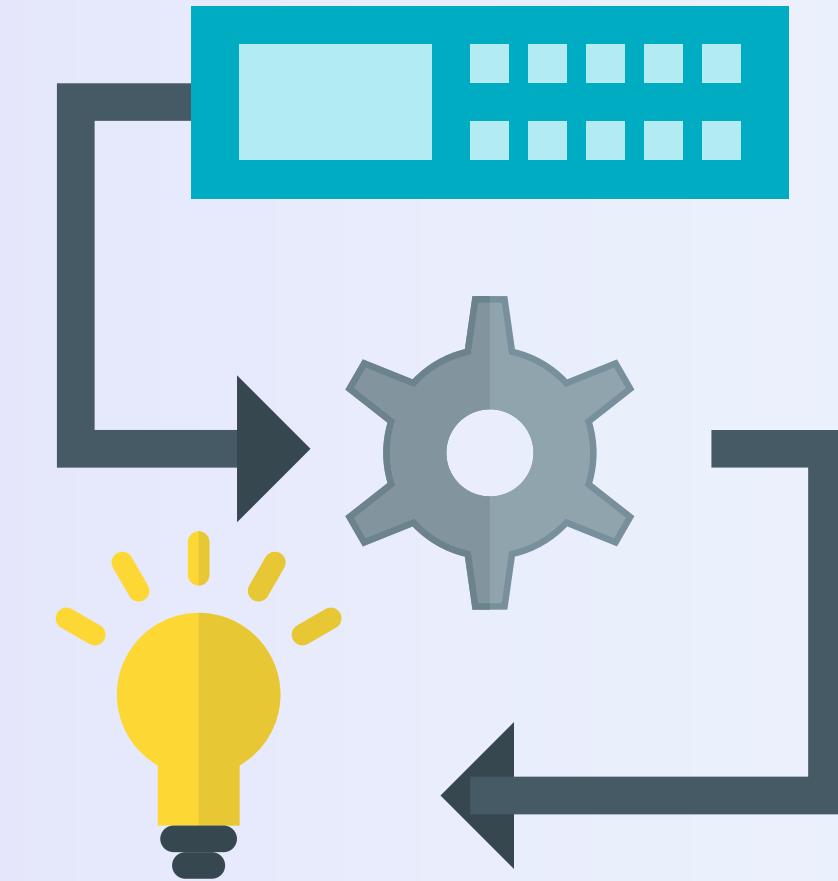
CLASSIFIER

- ✓ Là 1 lớp kết nối đầy đủ với hàm kích hoạt là softmax hoặc sigmoid
- ✓ Đẩy vector embedding của câu thu được từ mạng CNN trước đó là input



3 **EXPERIMENTS**

Những thực nghiệm và các kết luận của nhóm với phần Q2. Phần này tập trung mô tả các thông số và kết quả cuối cùng của mô hình.



MODELS

Có 5 models được huấn luyện, chia thành 2 loại chính

FREEZEE EMBEDDING

1

- Freeze Word2Vec Embedding + Classifier
- Freeze FastText Embedding + Classifier

TRAINABLE EMBEDDING

2

- Fine-tune Word2Vec Embedding + Classifier
- Fine-tune FastText Embedding + Classifier
- Build-in Embedding + Classifier

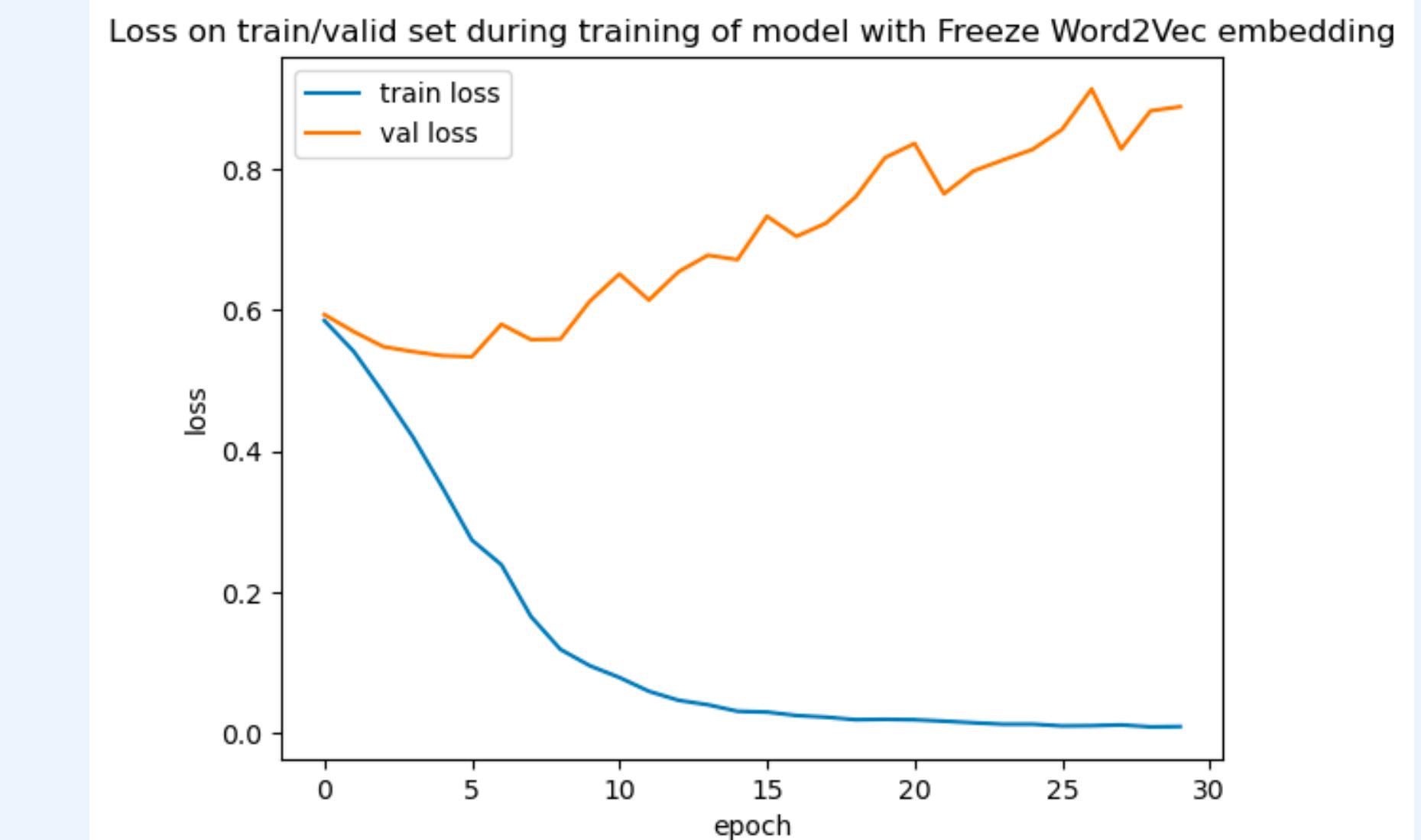
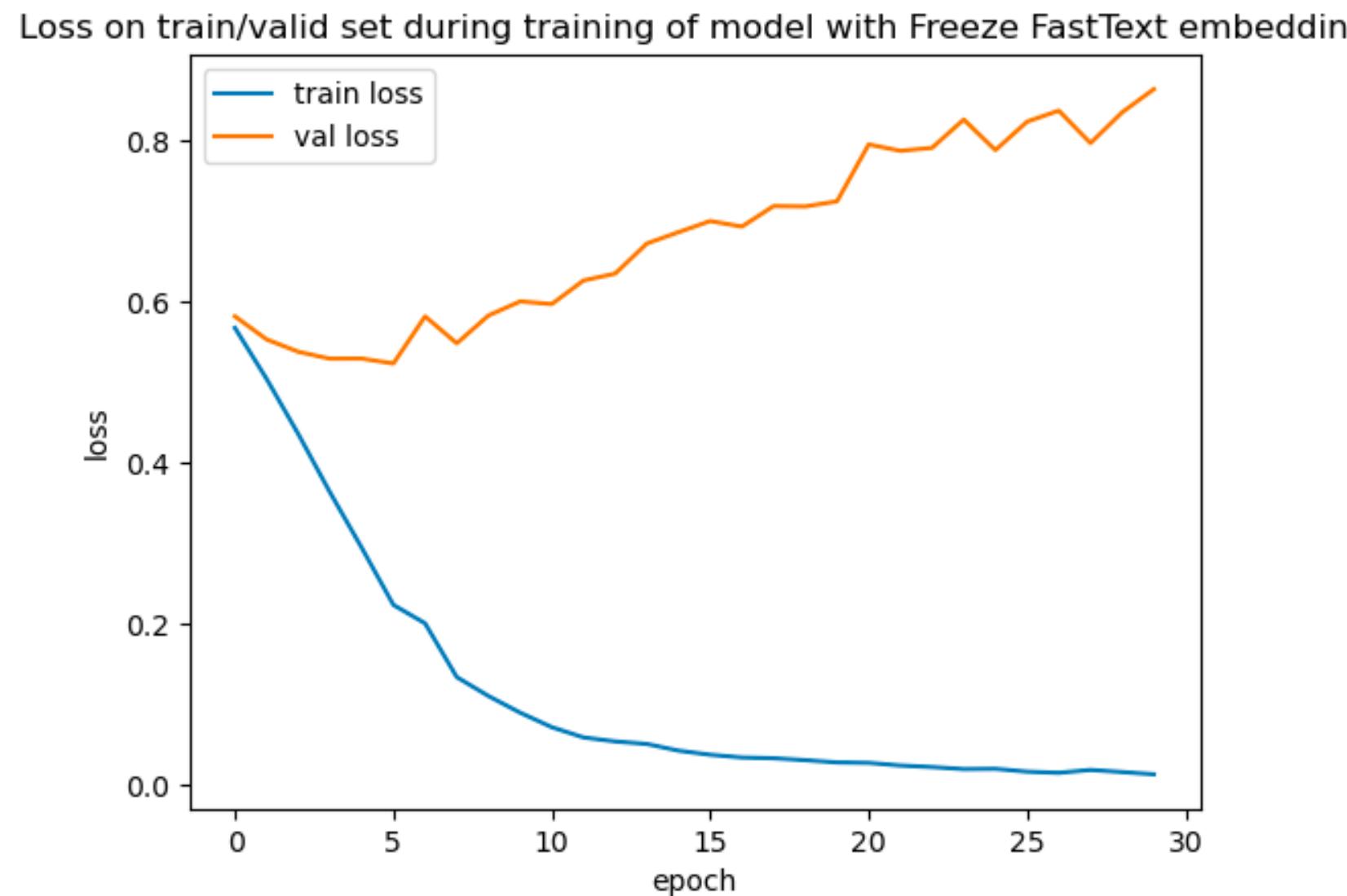
PARAMETERS

Các tham số mô hình mạng để thực hiện huấn luyện mô hình. Những tham số này được tham khảo từ bài báo Convolutional Neural Networks for Sentence Classification của Yoon Kim

MÔ TẢ THAM SỐ	GIÁTRI
Word vector đầu vào	FastText/PhoW2
Embedding size	300
Filter size	(3, 4, 5)
Số các filters	(100, 100, 100)
Hàm kích hoạt	ReLU
Pooling	1-max pooling
Dropout rate	0.5

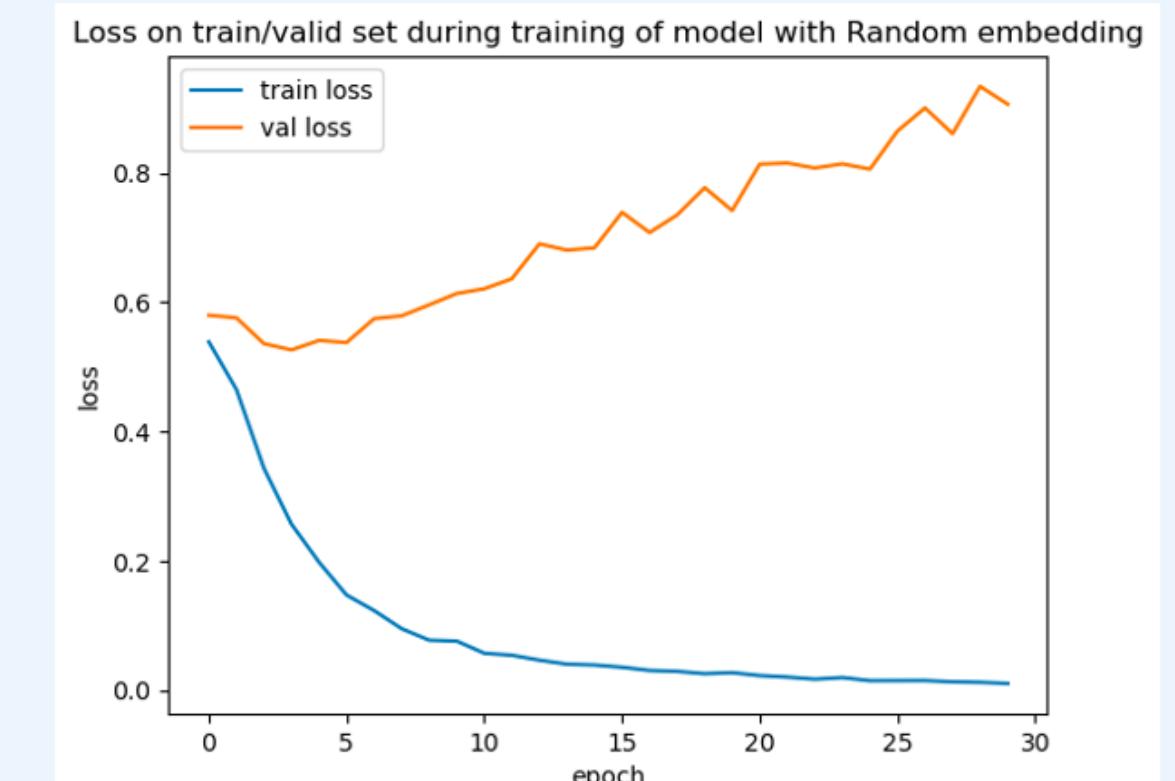
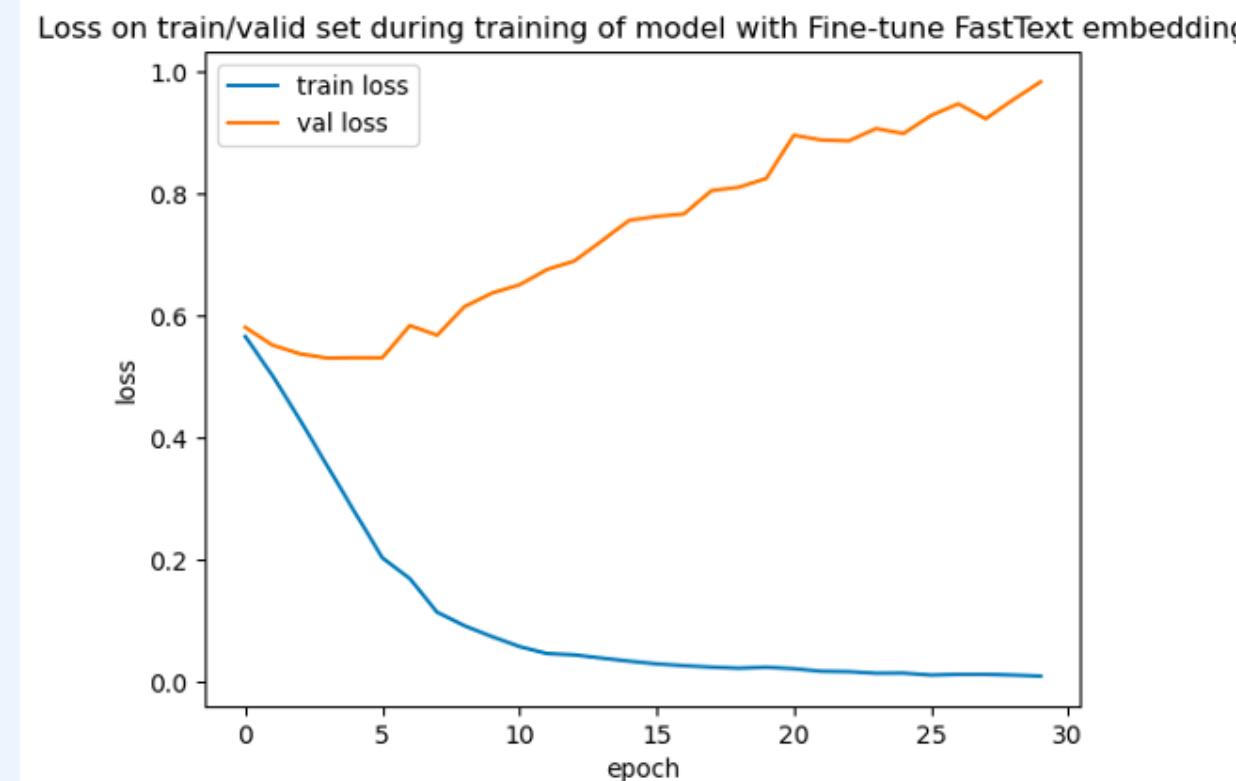
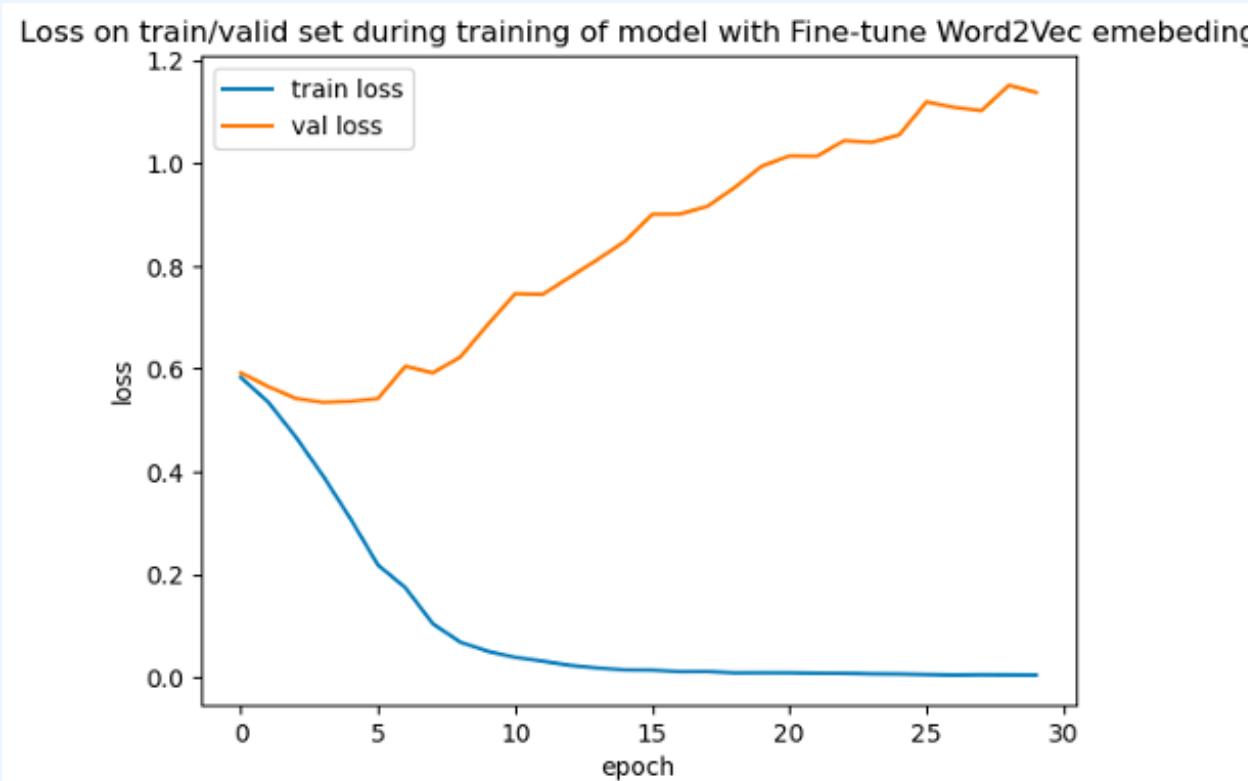
RESULT - LOSS

Freeze embedding models



RESULT - LOSS

Trainable embedding models



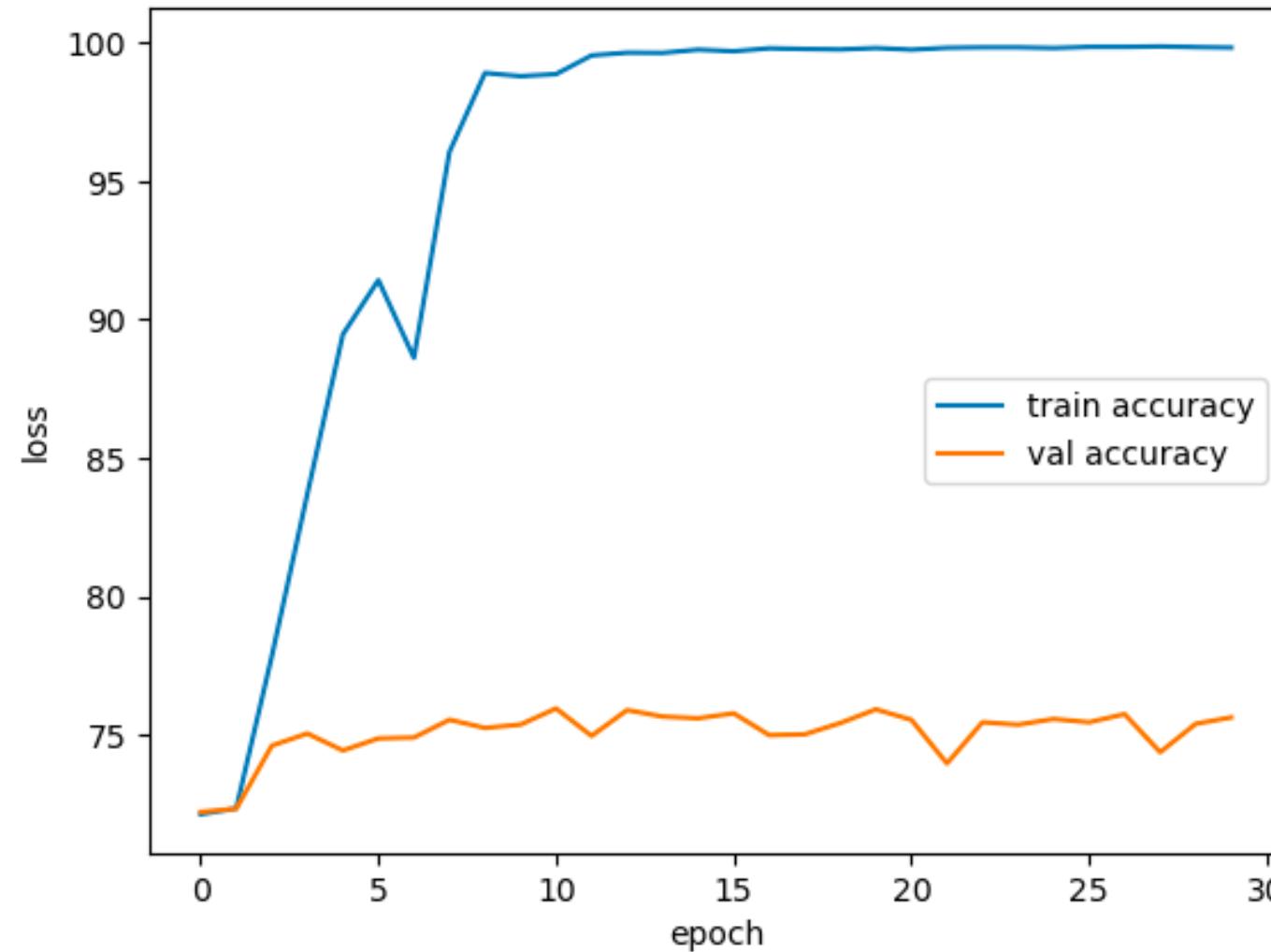
RESULT - ACCURACY

TÊN MÔ HÌNH THỬ NGHIỆM	GIÁ TRỊ ACCURACY TỐT NHẤT
Freeze Word2Vec embedding	75.95
Fine-tune Word2Vec embedding	75.92
Freeze FastText embedding	76.92
Fine-tune FastText embedding	76.07
Random embedding	76.91

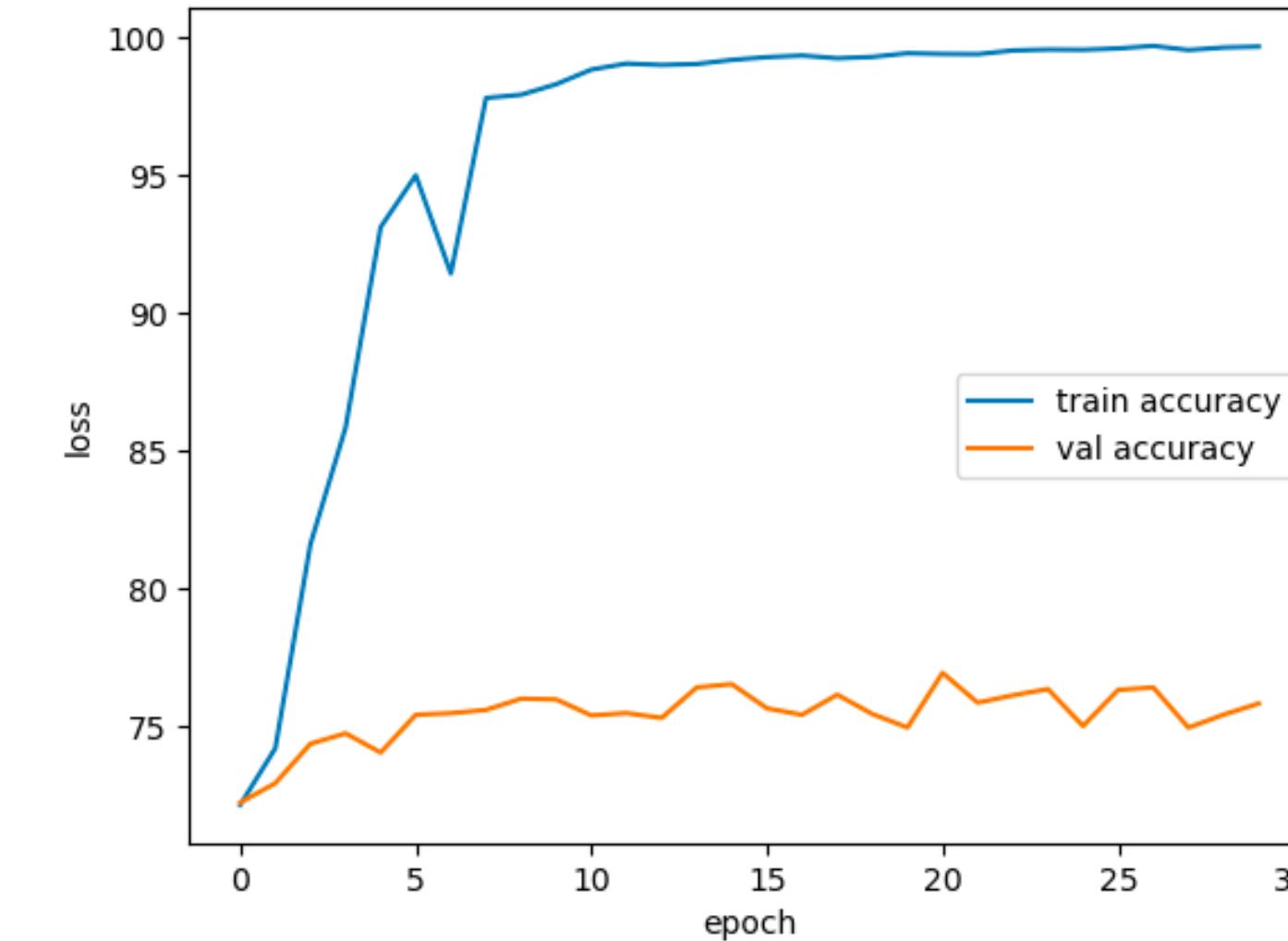
RESULT - ACCURACY

Freeze embedding models

Accuracy on train/valid set during training of model with Freeze Word2Vec embedding

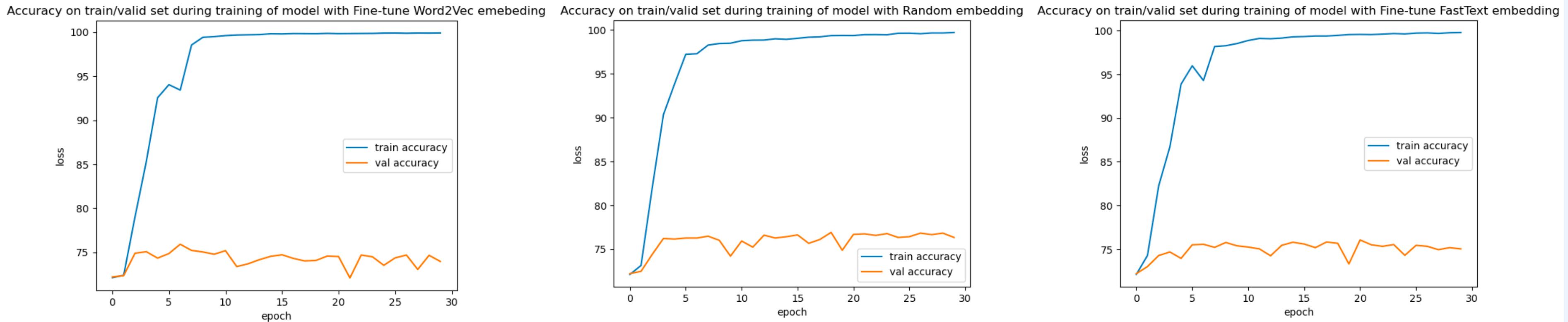


Accuracy on train/valid set during training of model with Freeze FastText embedding



RESULT - ACCURACY

Trainable embedding models



COMMENT #1

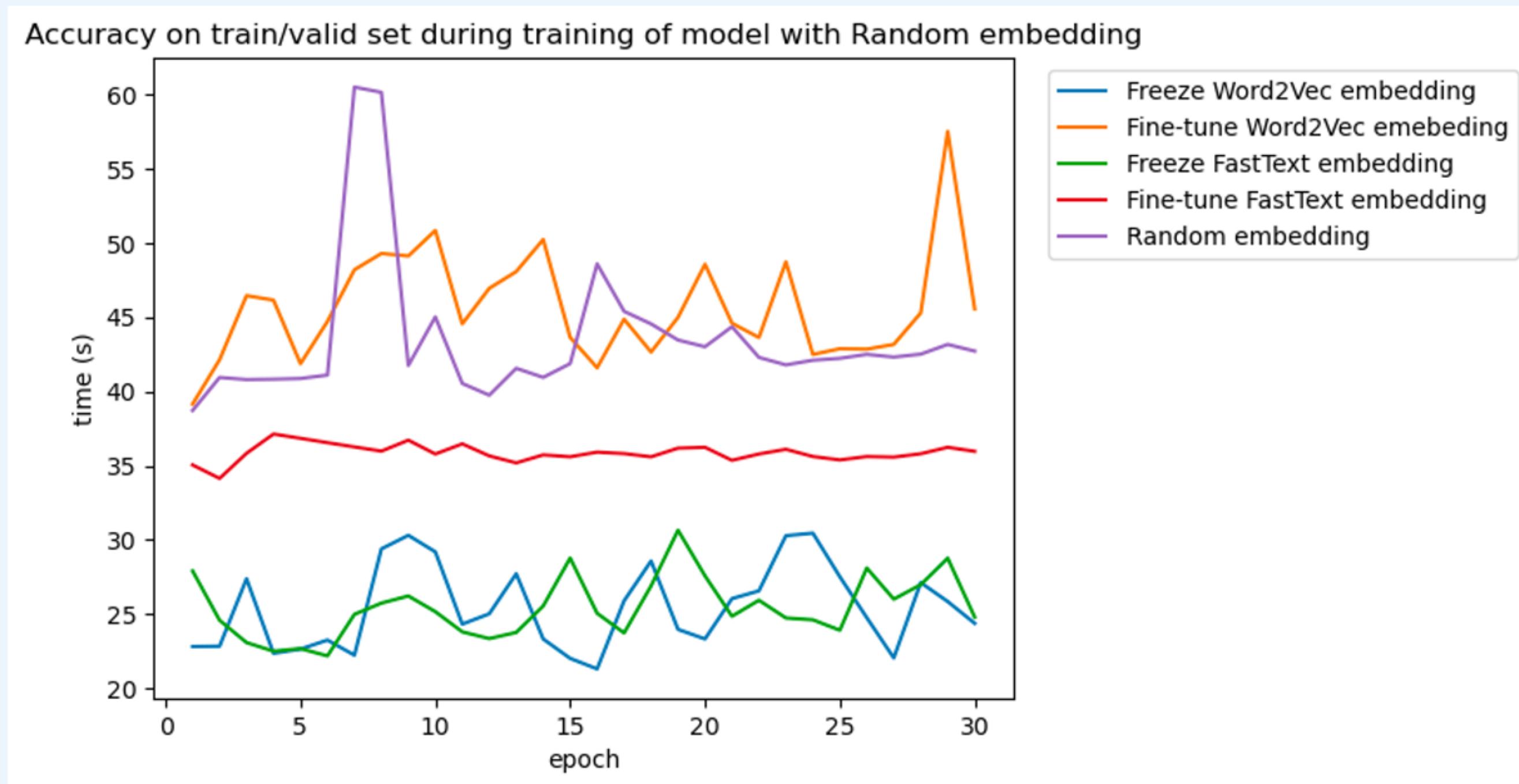
Tất cả các models đều giảm độ lỗi
và tăng độ chính xác trên tập train
trong suốt quá trình huấn luyện

COMMENT #2

Tuy nhiên, độ lỗi chỉ giảm (và độ chính
xác chỉ tăng) ở vài epoch đầu sau đó
độ lỗi tăng và độ chính xác chững lại

→ **OVERFITTING**

RESULT - TRAINING TIME



COMMENT #1

2 models giữ nguyên trọng số lớp embedding có thời gian huấn luyện ngắn, trong khoảng 20-30s/epoch.

COMMENT #2

3 models phải train trọng số cho lớp embedding có thời gian dài hơn

COMMENT #3

Các model không cần huấn luyện lớp embedding có thời gian huấn luyện ngắn hơn đáng kể các model phải huấn luyện

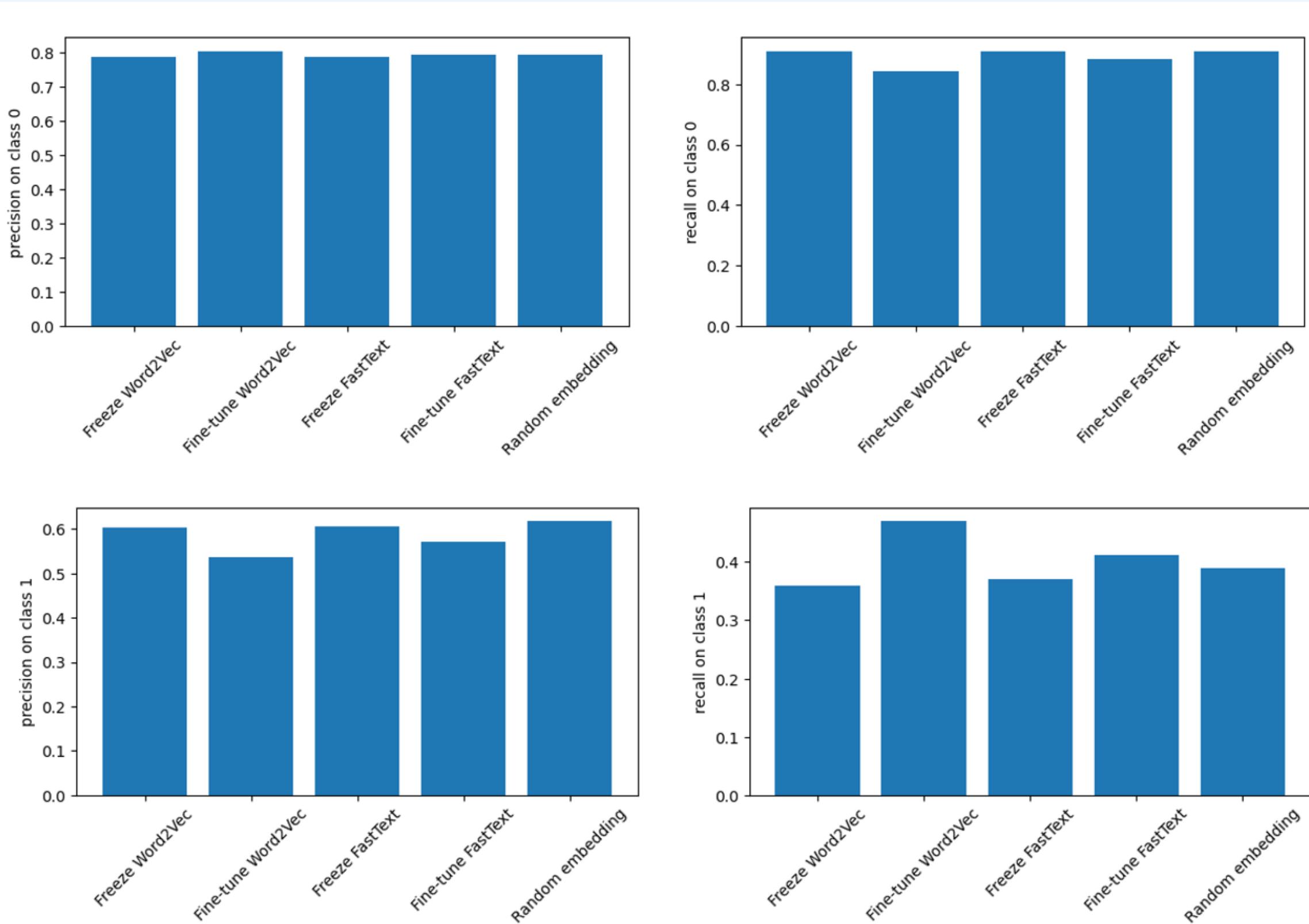
RESULT - METRICS

Đánh giá trên tập valid

	Freeze Word2Vec embedding	Fine-tune Word2Vec embedding	Freeze FastText embedding	Fine-tune FastText embedding	Random embedding
Accuracy	0.756283	0.739626	0.758036	0.750438	0.763296
Precision on class 0	0.786489	0.805255	0.789362	0.795396	0.794191
Recall on class 0	0.909348	0.843383	0.906920	0.881020	0.907325
Precision on class 1	0.603540	0.535971	0.605489	0.570803	0.617696
Recall on class 1	0.358570	0.470032	0.371188	0.411146	0.389064

RESULT - METRICS

Đánh giá trên tập valid



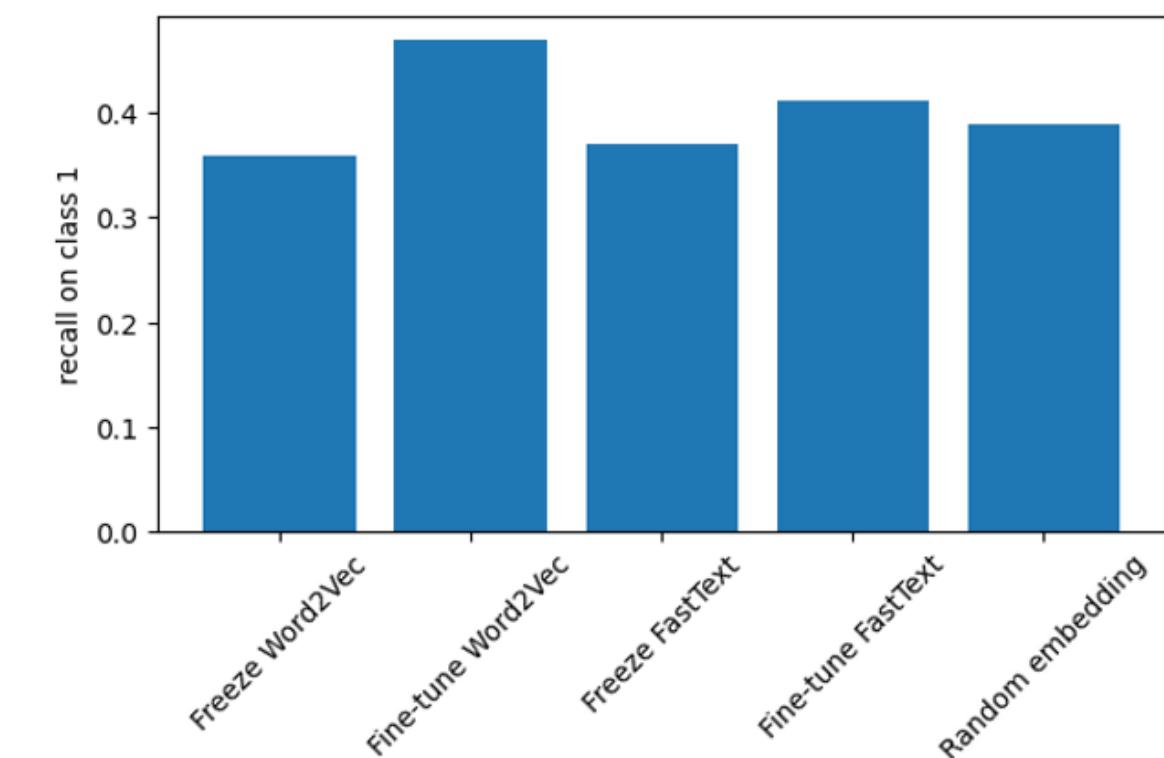
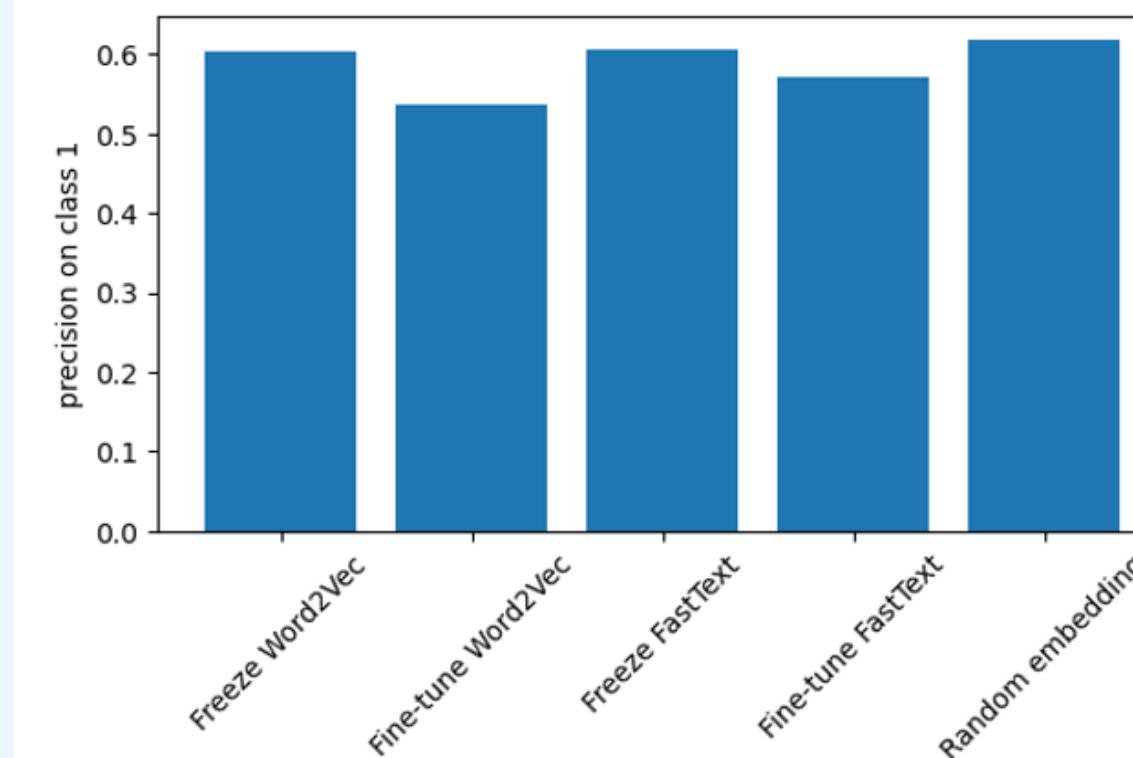
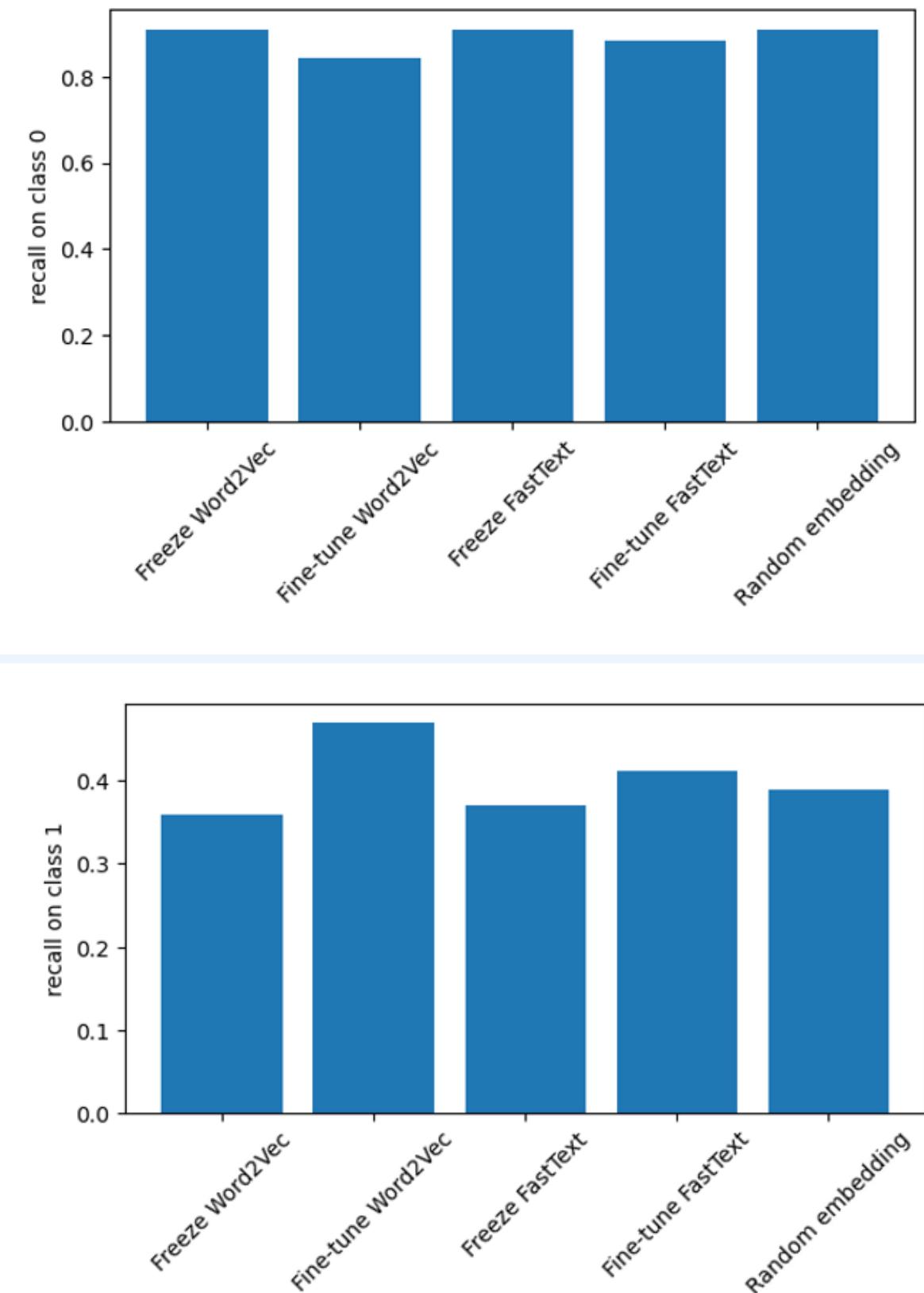
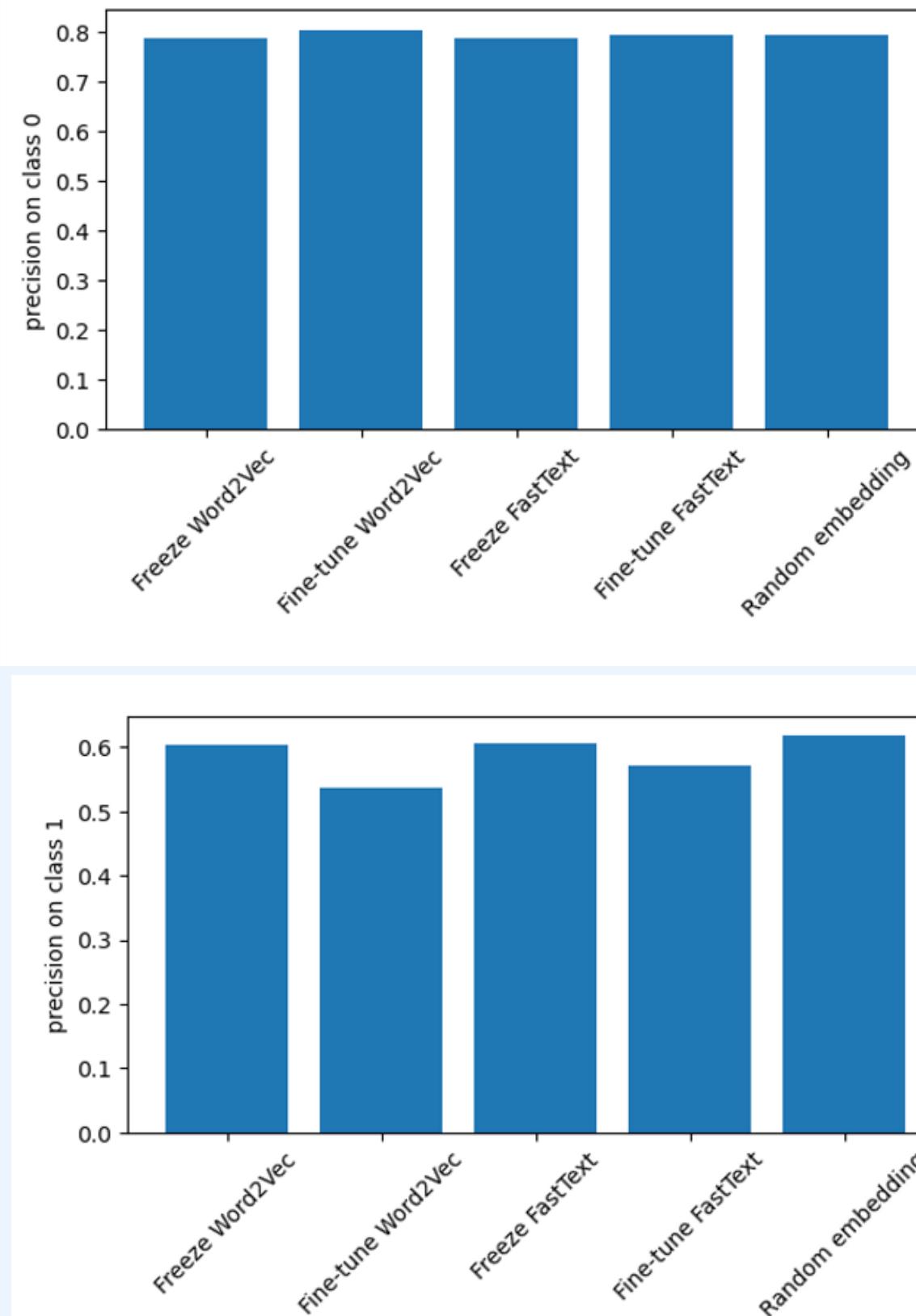
RESULT - METRICS

Đánh giá trên tập test

	Freeze Word2Vec embedding	Fine-tune Word2Vec embedding	Freeze FastText embedding	Fine-tune FastText embedding	Random embedding
Accuracy	0.754673	0.728972	0.762850	0.761682	0.769276
Precision on class 0	0.779381	0.785026	0.784584	0.794853	0.788966
Recall on class 0	0.919708	0.858881	0.924574	0.901865	0.927818
Precision on class 1	0.614786	0.520661	0.640927	0.613419	0.660305
Recall on class 1	0.329854	0.394572	0.346555	0.400835	0.361169

RESULT - METRICS

Đánh giá trên tập test



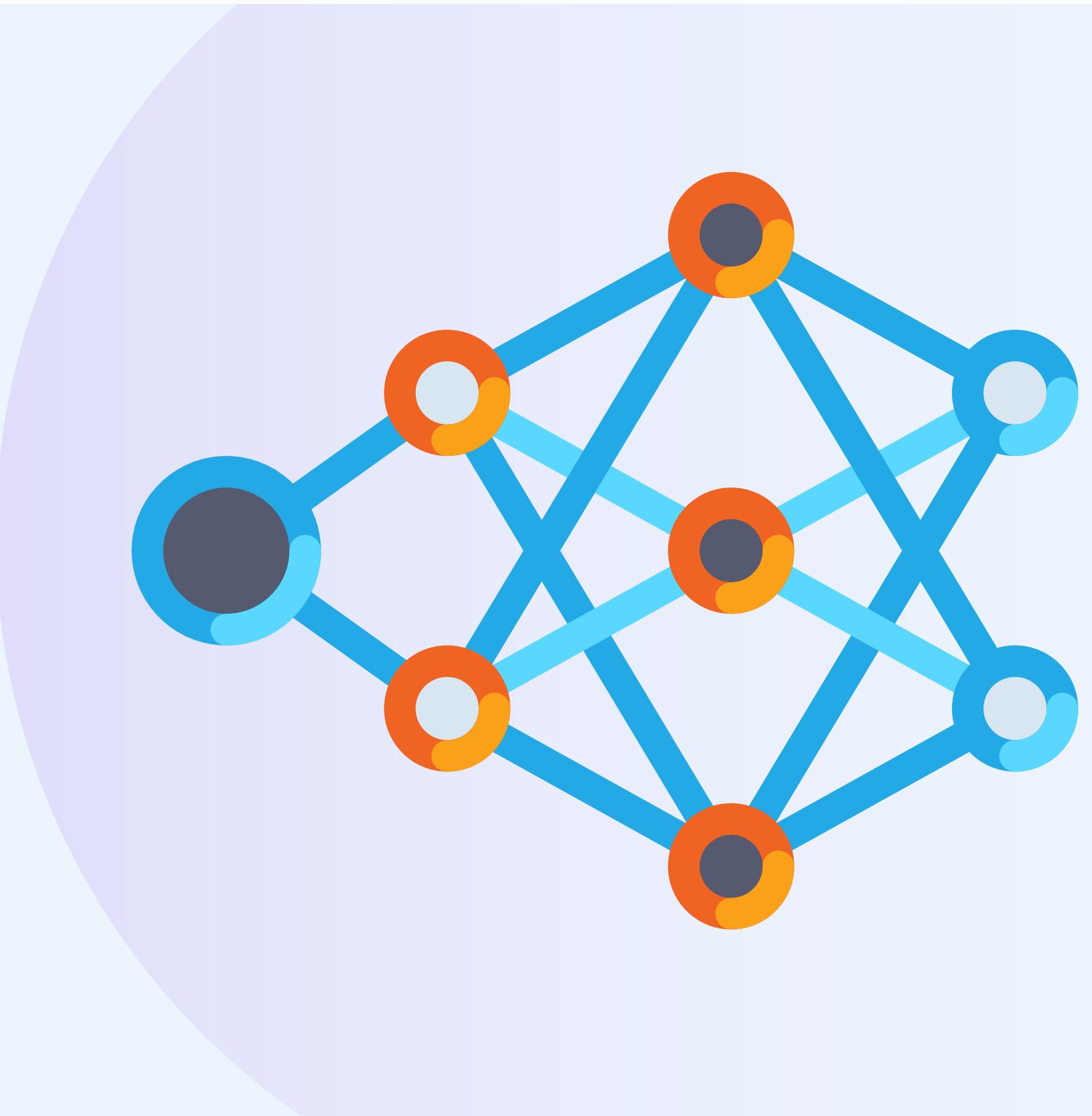
COMMENT

- ✓ Các models chỉ có accuracy từ 72-76% trên cả tập valid lẫn test
- ✓ Các models đạt được các giá trị recall trên class False khá tốt, từ 85-92%
- ✓ Nhưng độ precision trên cả tập valid lẫn test đều chưa tốt khi chỉ đạt từ 78-80%. Models dự đoán nhiều class False nên recall trên class 0 tốt.
- ✓ Với class True, tất cả các models đều khá tệ khi precision chỉ đạt từ 50 tới 60% và recall đạt 30 đến gần 50%
- ✓ Các models đang bị ảnh hưởng bởi vấn đề bộ dữ liệu mất cân bằng (imbalanced data) khi các đánh giá trên class False tốt hơn class True bởi vì các mẫu False chiếm đa số

2

MÔ HÌNH THỰC NGHIỆM

Q3



1 Giới Thiệu SimeCSE_Vietnamese

- ✓ SimeCSE_Vietnamese (viết tắt của Simple Contrastive Learning of Sentence Embeddings with Vietnamese) là một mô hình nhúng dành cho Tiếng Việt.
- ✓ Encode dữ liệu đầu vào sử dụng mô hình pre-trained là PhoBert.
- ✓ Kỹ thuật huấn luyện dựa vào mô hình pre-trained SimCSE.
- ✓ Mô hình được công bố vào năm 2021 và đủ nhẹ khi sử dụng với Google Colab.

Sentence Similarity

Source Sentence

Mỗi hiệp bóng đá kéo dài bao lâu

Sentences to compare to

Một trận thi đấu bóng đá thông thường diễn ra trong hai hiệp chính thức liên tiếp , mỗi

Một trận đấu bóng đá thông thường có hai hiệp , mỗi hiệp 45 phút với khoảng thời gian

Sau chức vô địch U-21 quốc gia 2013 , Nguyễn Quang Hải mới 16 tuổi lập tức được HLV

Cũng trong thập niên 1850 , các đội bóng nghiệp dư bắt đầu được thành lập và thườn

Việc mỗi đội bóng có luật chơi khác nhau khiến việc điều hành mỗi trận đấu giữa họ c

Compute

INPUT

OUTPUT

0.723
Một trận thi đấu bóng đá thông thường diễn ra trong hai hiệp chính thức liên tiếp , mỗi hiệp gồm 45 phút ngắn cách bằng 15 phút nghỉ giữa giờ . Sau khi hiệp 1 , hai đội bóng sẽ phải đổi sân cho nhau để có sự công bằng trong vòng 1 phút .

0.708
Một trận đấu bóng đá thông thường có hai hiệp , mỗi hiệp 45 phút với khoảng thời gian 15 phút nghỉ giữa hai hiệp .

0.090
Sau chức vô địch U-21 quốc gia 2013 , Nguyễn Quang Hải mới 16 tuổi lập tức được HLV Phan Thanh Hùng điền vào danh sách của đội bóng thủ đô tham dự V-League 2014 .

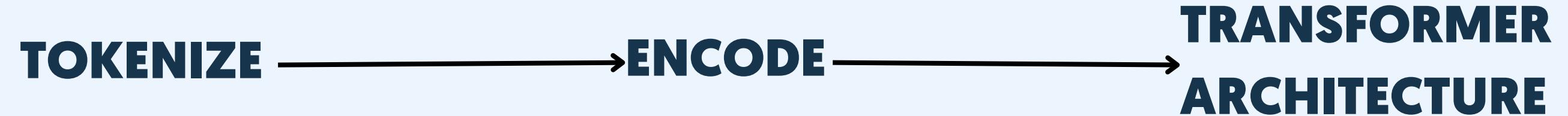
0.392
Cũng trong thập niên 1850 , các đội bóng nghiệp dư bắt đầu được thành lập và thường mỗi đội xây dựng cho riêng họ những luật chơi mới của môn bóng đá , trong đó đáng chú ý có câu lạc bộ Sheffield F.C .. Việc mỗi đội bóng có luật chơi khác nhau khiến việc điều hành mỗi trận đấu giữa họ diễn ra rất khó khăn .

0.455
Việc mỗi đội bóng có luật chơi khác nhau khiến việc điều hành mỗi trận đấu giữa họ diễn ra rất khó khăn . Nỗ lực đáng kể nhất trong việc chuẩn hóa luật chơi môn bóng đá là việc thành lập Hiệp hội bóng đá Anh (The Football Association , thường viết tắt là FA) vào ngày 26 tháng 10 năm 1863 tại Great Queen Street , Luân Đôn .

2

CÁCH HOẠT ĐỘNG

SlmeCSE_Vietnamese



2

CÁCH HOẠT ĐỘNG

Tokenization

Một số kiểu tokenize	Hạn chế
Tokenize theo word level	Sai ngữ nghĩa với các từ đa âm tiết
Tokenize theo multi-word level	Phụ thuộc vào từ điển, gia tăng chi phí tính toán
Tokenize character level	Các token không có ý nghĩa khi đứng độc lập

Phương pháp mới BETA (SOTA)

2 **CÁCH HOẠT ĐỘNG**

Phương pháp BETA (SOTA)

- Bước 1: Khởi tạo từ điển
- Bước 2: Biểu diễn mỗi từ trong bộ văn bản bằng kết hợp của các ký tự với token $\langle \backslash w \rangle$ ở cuối cùng đánh dấu kết thúc một từ
- Bước 3: Thống kê tần suất xuất hiện theo cặp của toàn bộ token trong từ điển.
- Bước 4: Gộp các cặp có tần suất xuất hiện lớn nhất để tạo thành một n-gram theo level character mới cho từ điển.
- Bước 5: Lặp lại bước 3 và bước 4 cho tới khi số bước triển khai merge đạt đỉnh hoặc kích thước kỳ vọng của từ điển đạt được.

2

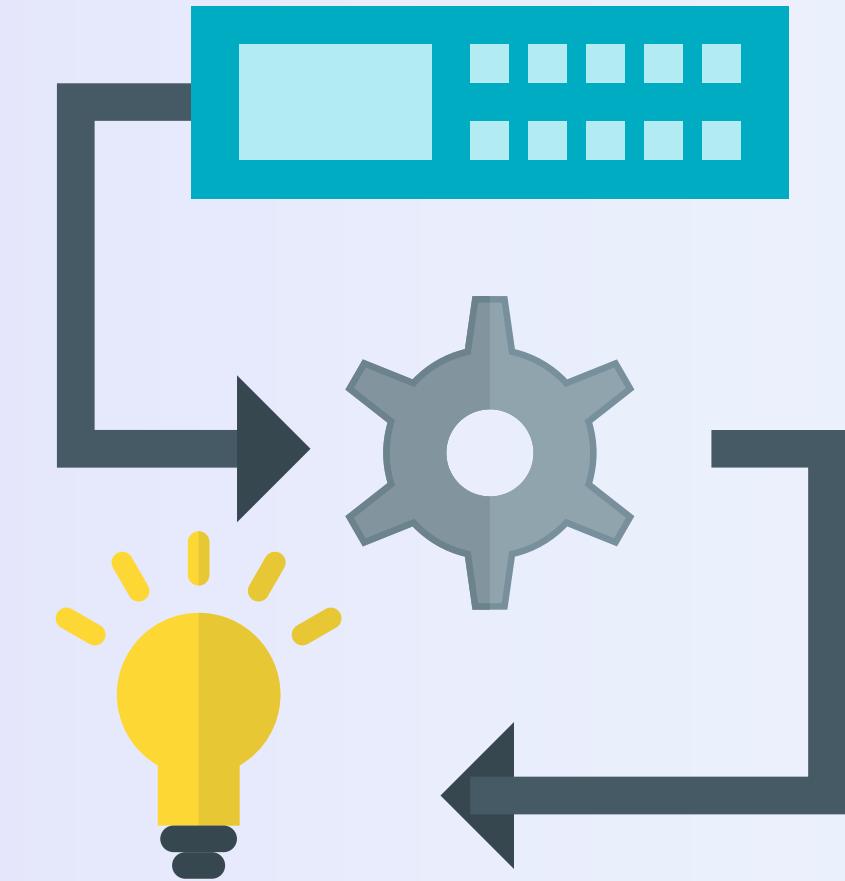
CÁCH HOẠT ĐỘNG

Phương pháp BETA (SOTA)

low, lower, lowest → **fastBPE** → low, er, est

3 **EXPERIMENTS**

Những thực nghiệm và các kết luận của nhóm với phần Q3. Phần này tập trung mô tả các thông số và kết quả cuối cùng của mô hình.



3

CÁCH TINH CHỈNH

SlmeCSE_Vietnamese

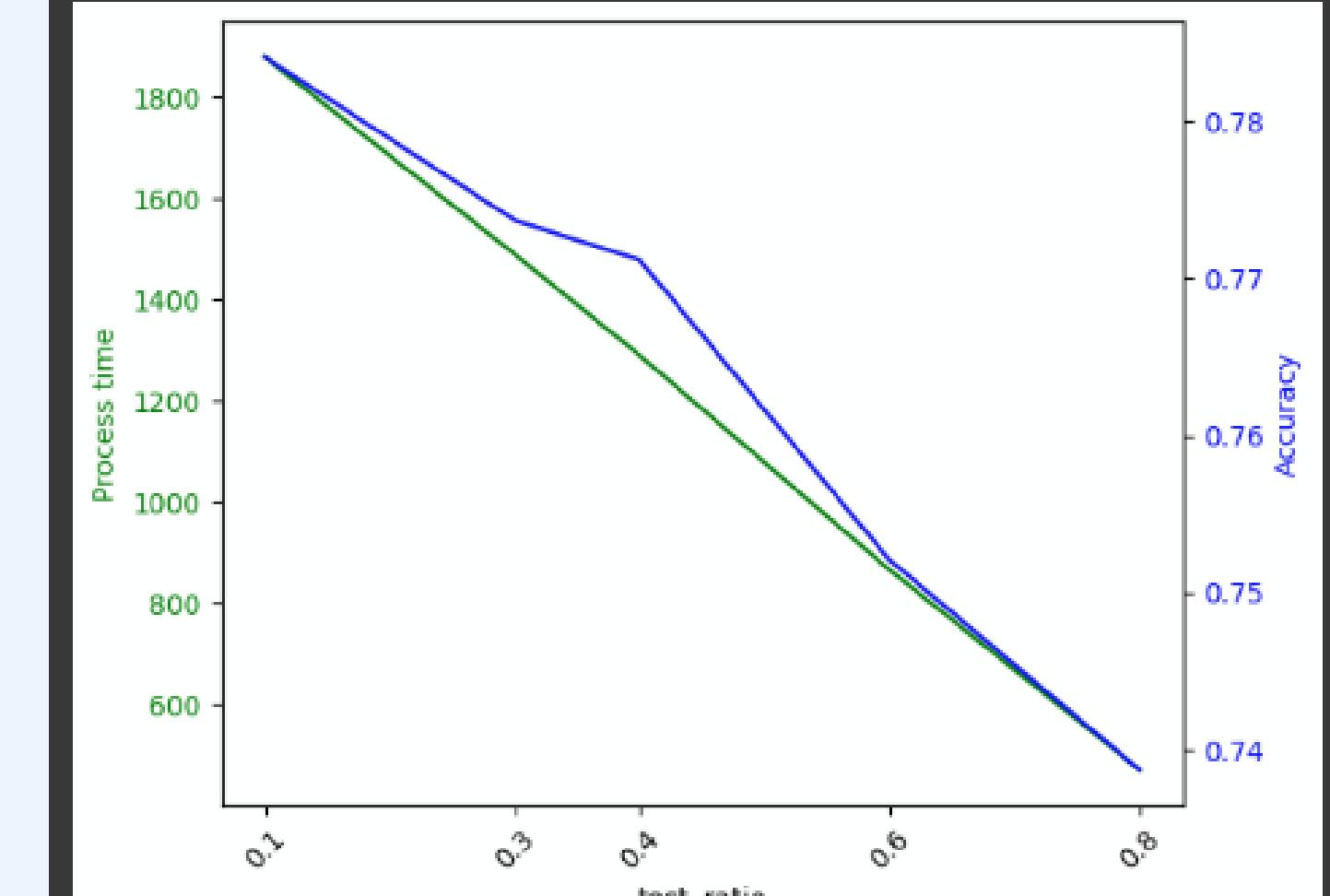
1. Đọc dữ liệu và tiền xử lý:

- Tập tin dữ liệu ở Github được tải về Google Colab với wget.
 - Tiền xử lý dữ liệu (chuẩn hóa unicode, loại bỏ stopword...).
- Dữ liệu thu được là một pandas dataframe với 18056 dòng (mẫu dữ liệu) và 3 cột (question, text, label).

2. Fine-tune và khảo sát mô hình:

- Mô hình được fine-tune sử dụng API sentence_transformers.
- Mô hình được khảo sát với các tham số test_ratio, batch_size, epoch.
- Tập dữ liệu chỉ gồm hai phần là huấn luyện và kiểm thử. Cấu hình khảo sát mặc định: [test_ratio=0.4, shuffle=True, batch_size=32, epoch=5].

	test_ratio	Process time	Accuracy	Recall	Precision	F1-score
0	0.1	1879.588830	0.784053	0.517241	0.731707	0.606061
1	0.3	1490.133618	0.773675	0.510133	0.698652	0.589692
2	0.4	1290.562403	0.771148	0.504161	0.688397	0.582048
3	0.6	866.678664	0.752077	0.454334	0.658516	0.537694
4	0.8	471.967292	0.738733	0.377096	0.654820	0.478585

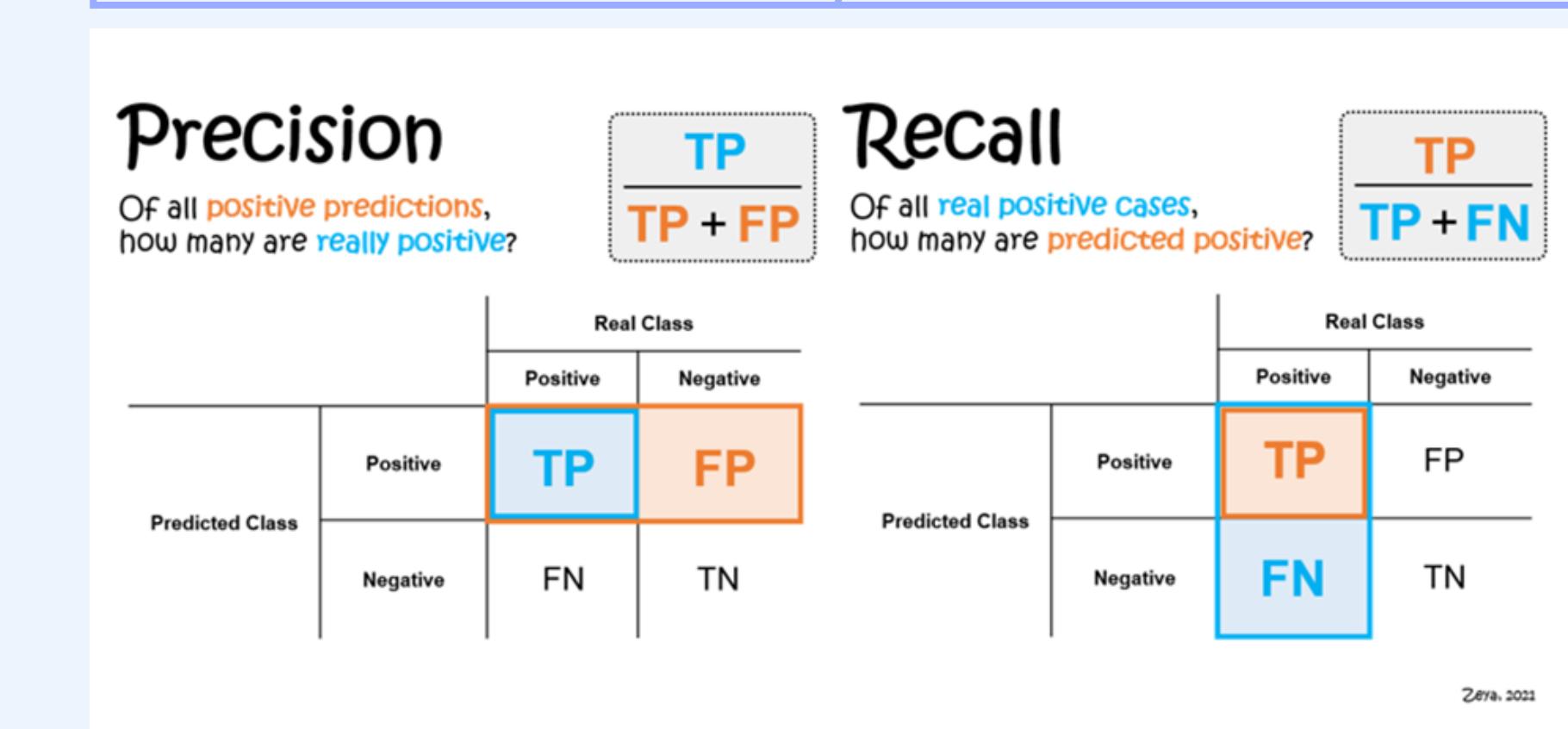


PARAMETERS

Huấn luyện trên GPU Tesla T4 của Google Colab.

Ở bài toán hiện tại, độ đo precision sẽ quan trọng hơn.

MÔ TẢ THAM SỐ	GIÁ TRỊ
Test ratio	[0.1, 0.3, 0.4, 0.6, 0.8]
Batch size	[8, 16, 32]
Số epoch	[1, 3, 10]

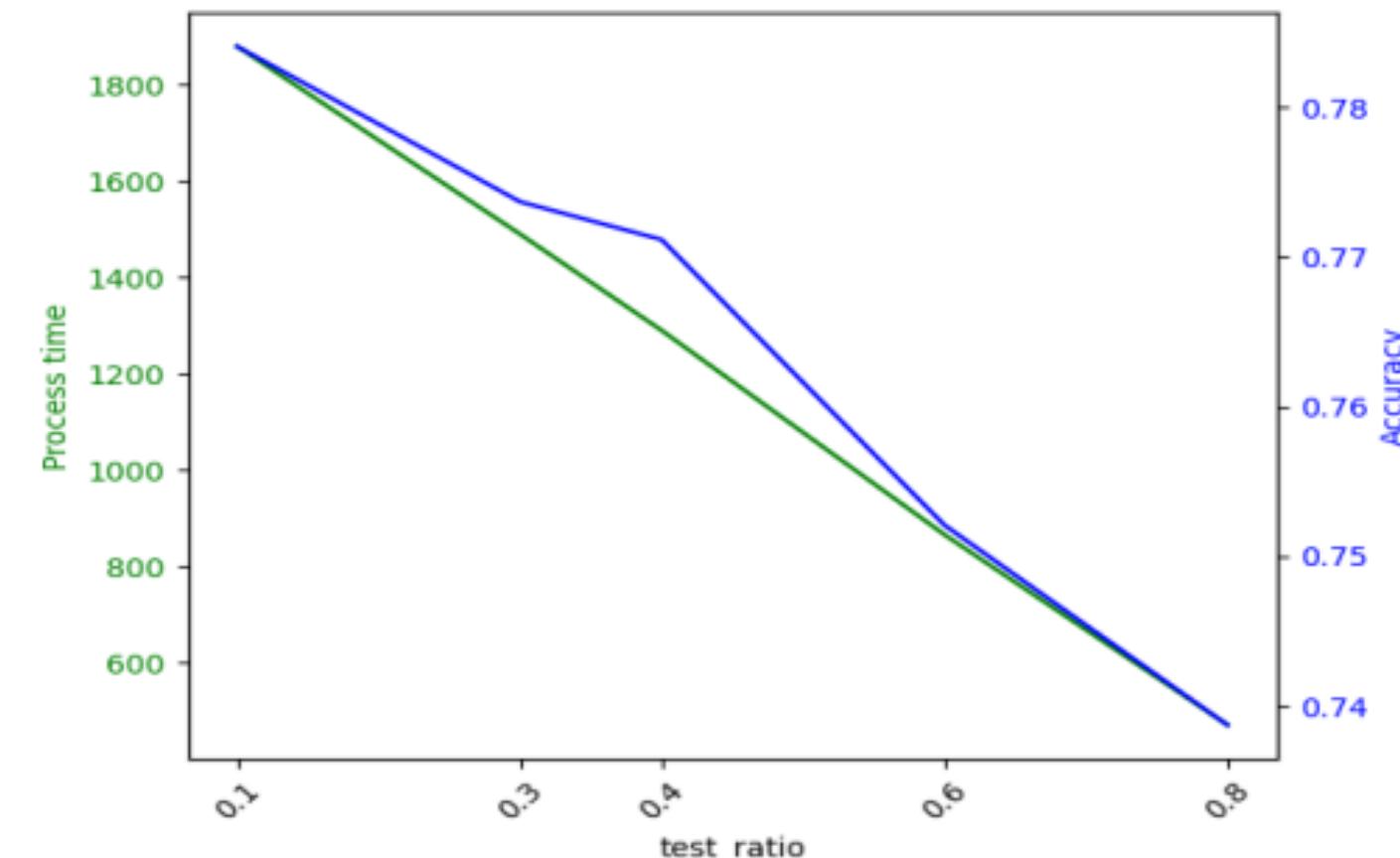


RESULTS

TEST-RATIO

- Với giá trị test_ratio 0.1, mô hình có độ chính xác cao nhất (78%) và thời gian xử lý lâu nhất (1879.59 giây).
- Sự tăng dần của giá trị test_ratio là sự giảm dần của thời gian xử lý và độ chính xác → hiện tượng overfitting.

Test Ratio	Process time	Accuracy	Recall	Precision	F1-Score
0.1	1879.588830	0.784053	0.517241	0.731707	0.606061
0.3	1490.133618	0.773675	0.510133	0.698652	0.589692
0.4	1290.562403	0.771148	0.504161	0.688397	0.582048
0.6	866.678664	0.752077	0.454334	0.658516	0.537694
0.8	471.967292	0.738733	0.377096	0.654820	0.478585

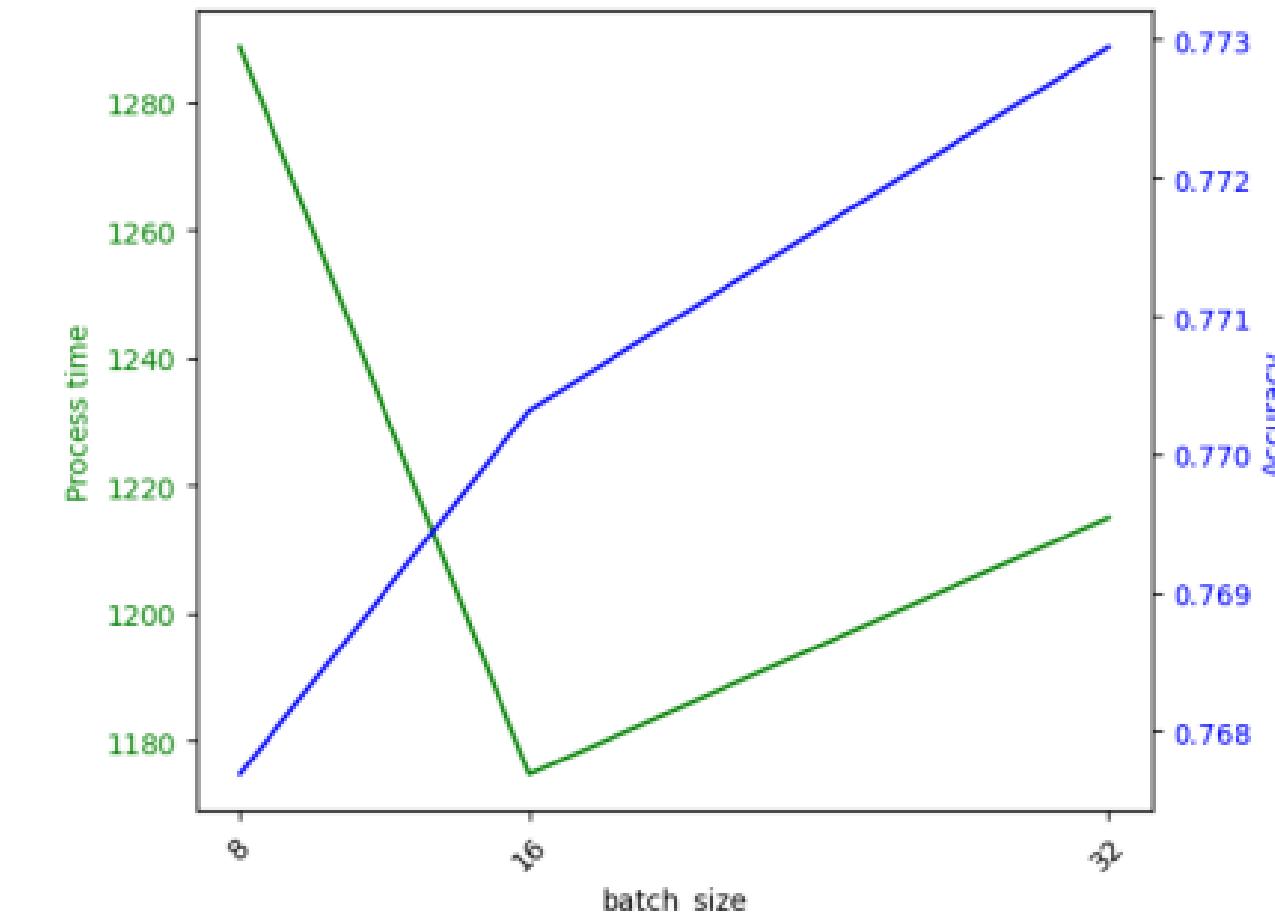


RESULTS

BATCH SIZE

- Với batch_size là 8 thì mô hình có độ chính xác thấp nhất (76.76%).
- Đồ thị thể hiện sự thay đổi gần như là tuyến tính của độ chính xác.

Batch Size	Process time	Accuracy	Recall	Precision	F1-Score
8	1288.733133	0.767687	0.528252	0.667405	0.589731
16	1174.743665	0.770317	0.502409	0.686826	0.580319
32	1214.957929	0.772948	0.504161	0.693791	0.583968

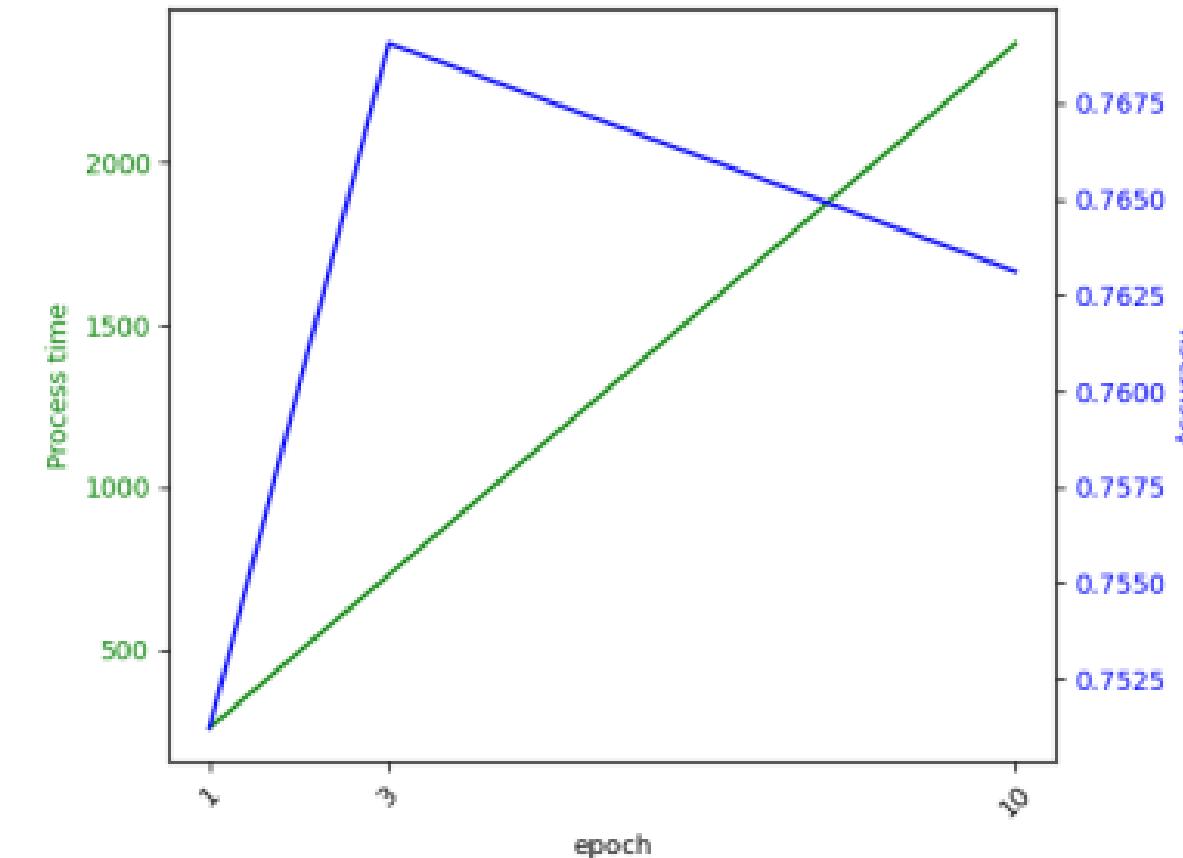


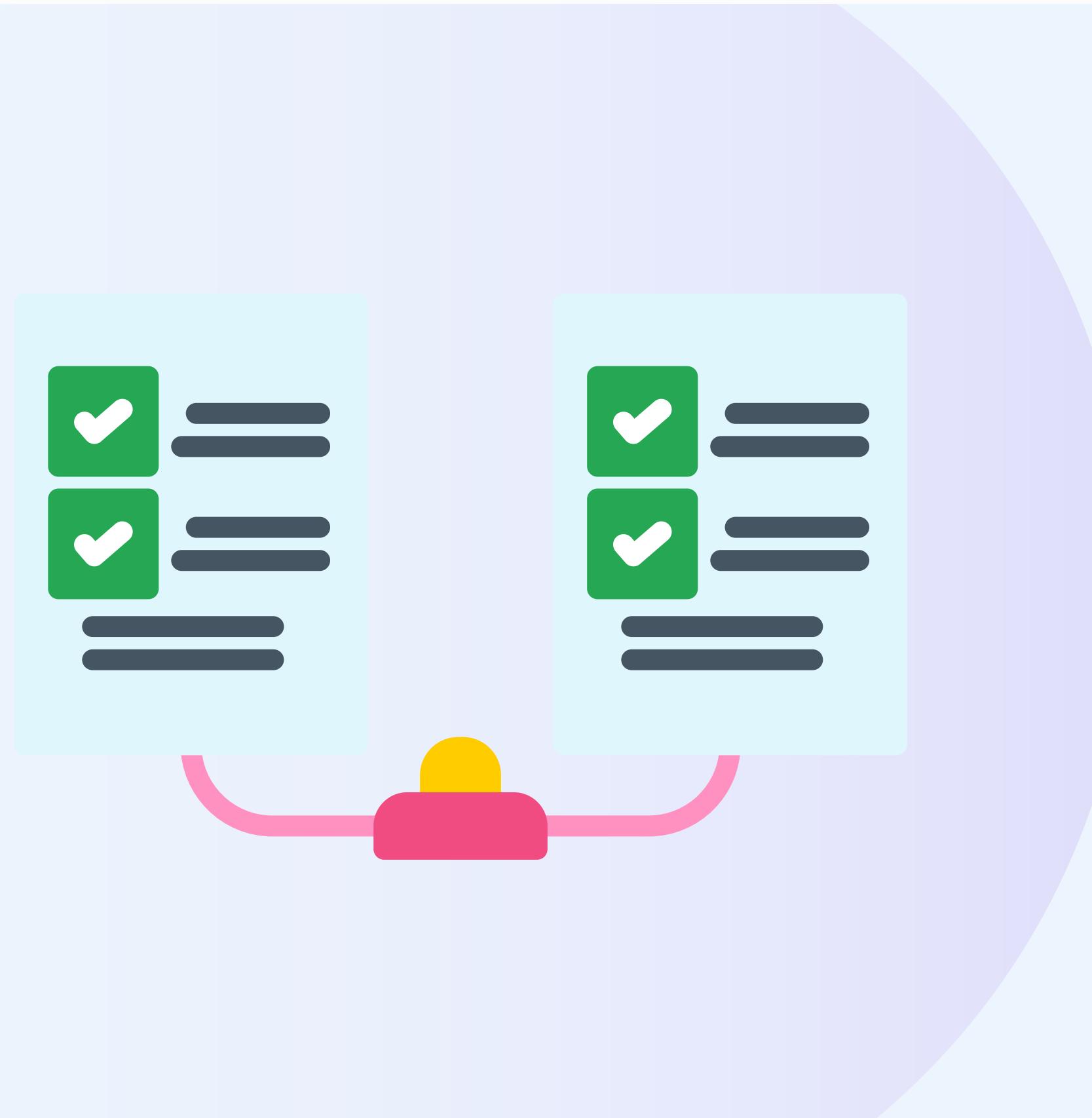
RESULTS

EPOCH

- Mô hình có độ chính xác thấp nhất với epoch là 1 (75.12%).
- Sự thay đổi của thời gian xử lý khi giá trị epoch tăng dần là tuyến tính.
- Giá trị epoch quá nhỏ (1 epoch) khiến việc huấn luyện mô hình không quá hiệu quả. Với 3 epoch huấn luyện, độ chính xác của mô hình được cải thiện nhưng xảy ra hiện tượng overfitting.

Epoch	Process time	Accuracy	Recall	Precision	F1-Score
1	256.586392	0.751211	0.416995	0.671368	0.514456
3	728.933487	0.769071	0.468682	0.701639	0.561975
10	2369.249527	0.763118	0.5265	0.656114	0.584204





3

THẢO LUẬN SO SÁNH

Q2

	Freeze Word2Vec embedding	Fine-tune Word2Vec embedding	Freeze FastText embedding	Fine-tune FastText embedding	Random embedding
Accuracy	0.754673	0.728972	0.762850	0.761682	0.769276
Precision on class 0	0.779381	0.785026	0.784584	0.794853	0.788966
Recall on class 0	0.919708	0.858881	0.924574	0.901865	0.927818
Precision on class 1	0.614786	0.520661	0.640927	0.613419	0.660305
Recall on class 1	0.329854	0.394572	0.346555	0.400835	0.361169

- Mô hình ở Q2 có độ chính xác cao nhất là 76.9% (random embedding).
- Độ chính xác của các phiên bản dao động từ ≈75-76%.

Q3

Test Ratio	Process time	Accuracy	Recall	Precision	F1-Score
0.1	1879.588830	0.784053	0.517241	0.731707	0.606061
0.3	1490.133618	0.773675	0.510133	0.698652	0.589692
0.4	1290.562403	0.771148	0.504161	0.688397	0.582048
	Process time	Accuracy	Recall	Precision	F1-score
0	71.875227	0.505538	0.775952	0.367367	0.498652

- Mô hình ở Q3 có độ chính xác cao nhất là 78.4% khi khảo sát trên tham số test_ratio.
- Trước khi fine-tune, độ chính xác của mô hình ≈50.5% và dao động từ ≈77-78% sau khi fune-tune.

Mô hình có sẵn ổn định hơn so với mô hình tự xây dựng. Cả hai đều bị ảnh hưởng bởi tình trạng mất cân bằng nhãn của tập dữ liệu.



fit@hcmus

VNUHCM - UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

COURSE: TEXT MINING
THEORY LECTURER: PhD. LE THANH TUNG
PRACTICE LECTURER: Ms. NGUYEN TRAN DUY MINH

**THANK YOU
FOR LISTENING**

GROUP: 1
MEMBERS: 20127258 - 20127655 - 20127625 - 20127597
PHƯỚC NGUYỄN - QUỐC TRUNG - HOÀNG THÁI - TẤN PHƯƠNG