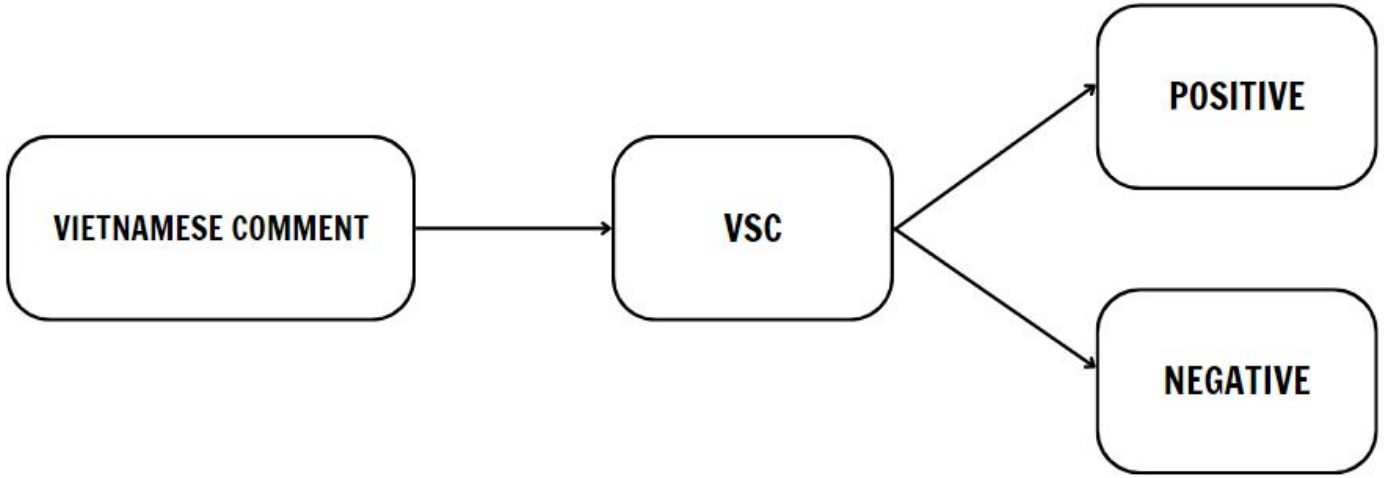
Vietnamese Sentiment Classification

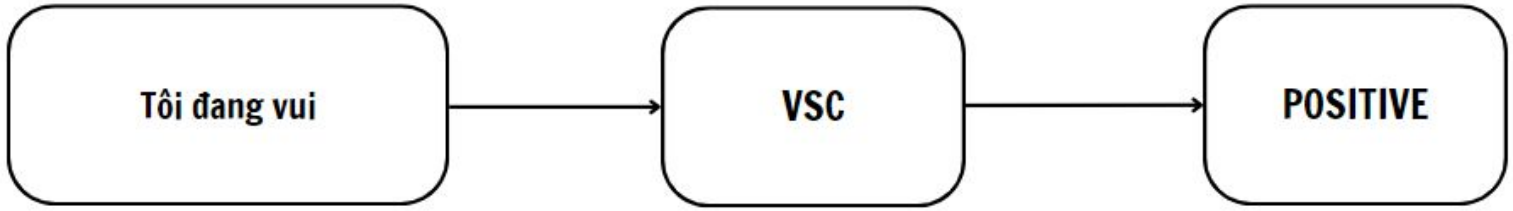
Mục tiêu : Phân loại cảm xúc của các bình luận trên mạng xã hội như Tiktok, Facebook,...

Duong Minh Duc
Dang Van Khai
Nguyen Viet Bac

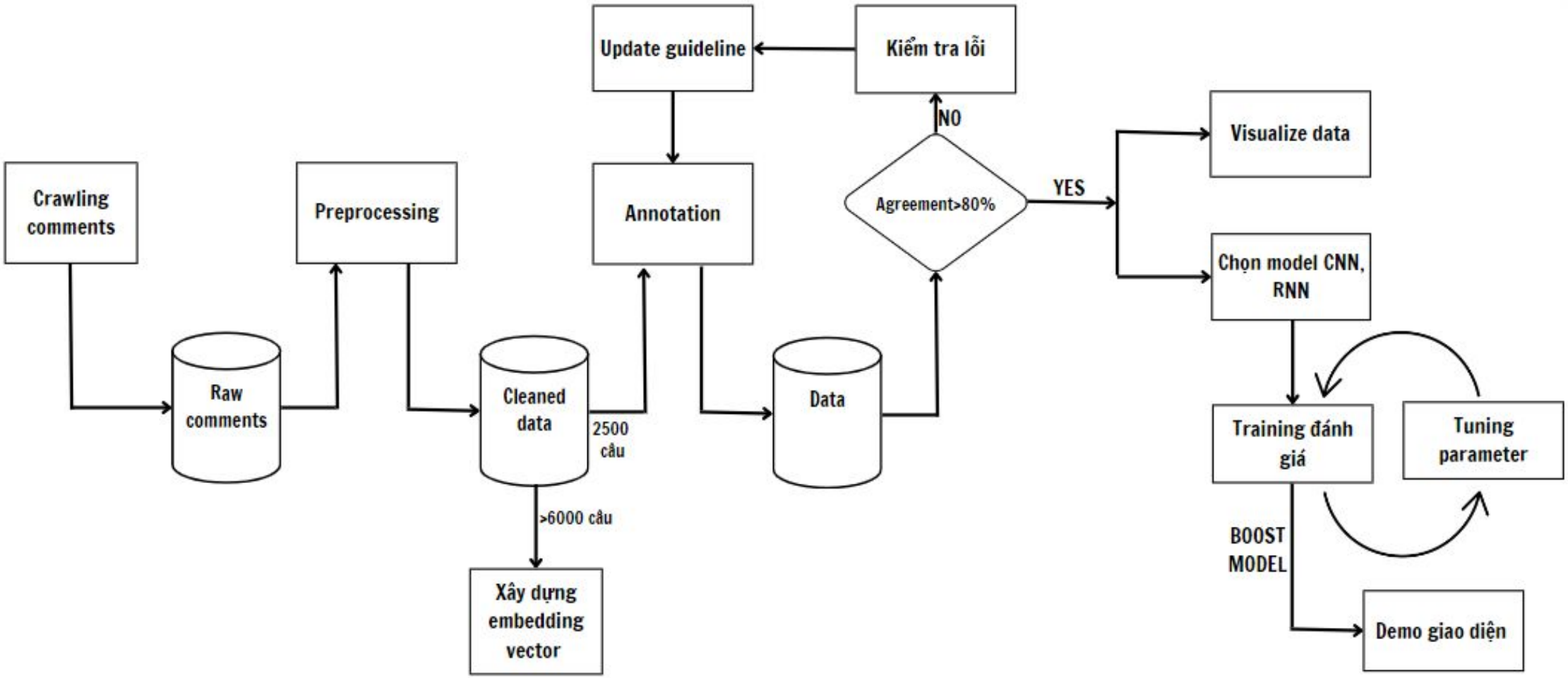
Task Description



Example

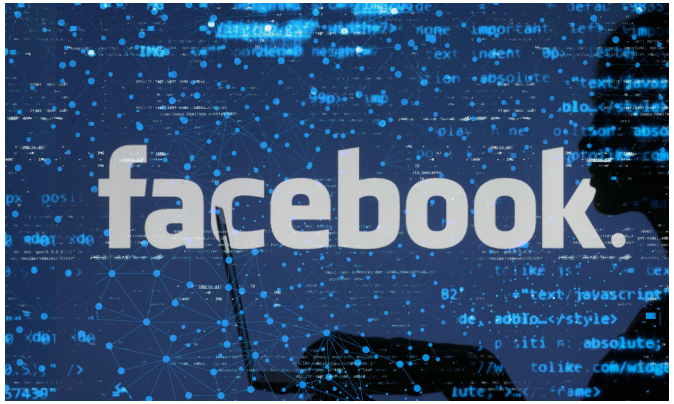


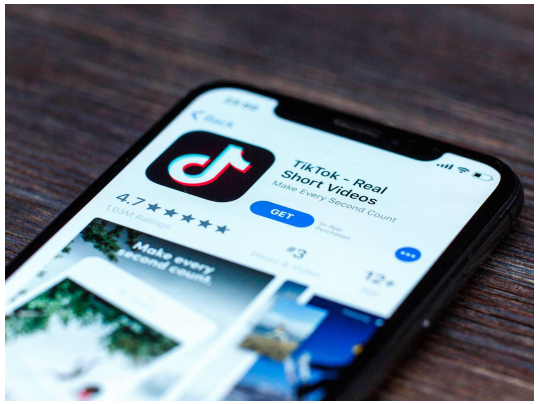
Modules


Crawling comments
Raw comments
Preprocessing
Cleaned data
2500 câu
>6000 câu
Xây dựng embedding vector
Annotation
Update guideline
Kiểm tra lỗi
NO
Agreement>80%
YES
Data
Visualize data
Chọn model CNN, RNN
Training đánh giá
Tuning parameter
BOOST MODEL
Demo giao diện

Data

- Crawl data từ MXH, chủ yếu là Facebook, Tiktok, Youtube,...
- Sử dụng API, Selenium, tools, …
  => Kết quả thu được: >10000 câu





Annotation

· Giai đoạn 1 : Cả 3 thành viên sẽ cùng gán nhãn cho cùng 500 comments, nếu độ đồng thuận dưới 80% thì sẽ lấy ra những comment bị gán nhãn khác nhau và đưa đến quyết định label cuối cùng cho comment đó

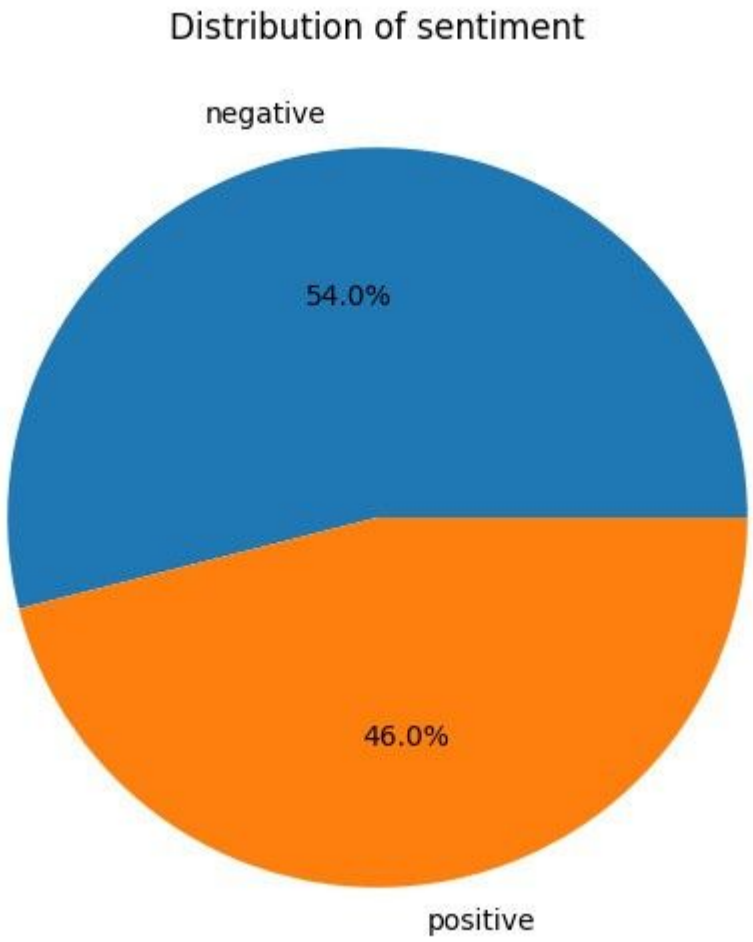
· Giai đoạn 2 : Tiến hành gán nhãn cho 2000 comments, 2 thành viên (Đức, Bắc) mỗi người sẽ gán nhãn độc lập với nhau, mỗi người 1000 comments, thành viên còn lại (Khải) sẽ tiến hành kiểm tra nhãn của toàn bộ 2000 comments và cũng yêu cầu độ đồng thuận giữa 2 người là trên 80%

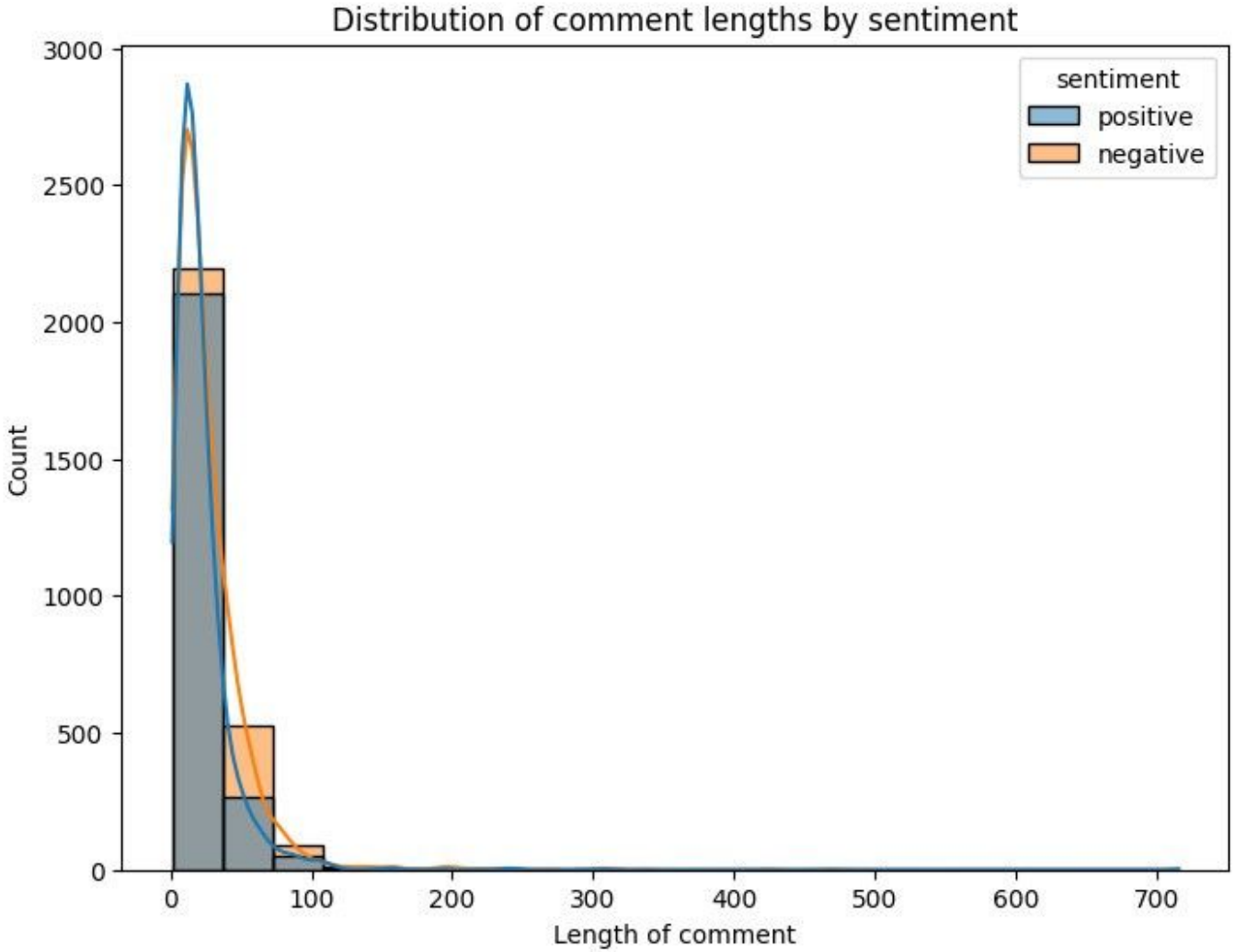
Giai đoạn	Annotators	Agreement
1	Khải - Bắc – Đức (500 cmts)	90,2%
2	Bắc – Khải (1000 cmts)	85.32%
	Đức – Khải (1000 cmts)	90,6%

Data statistics

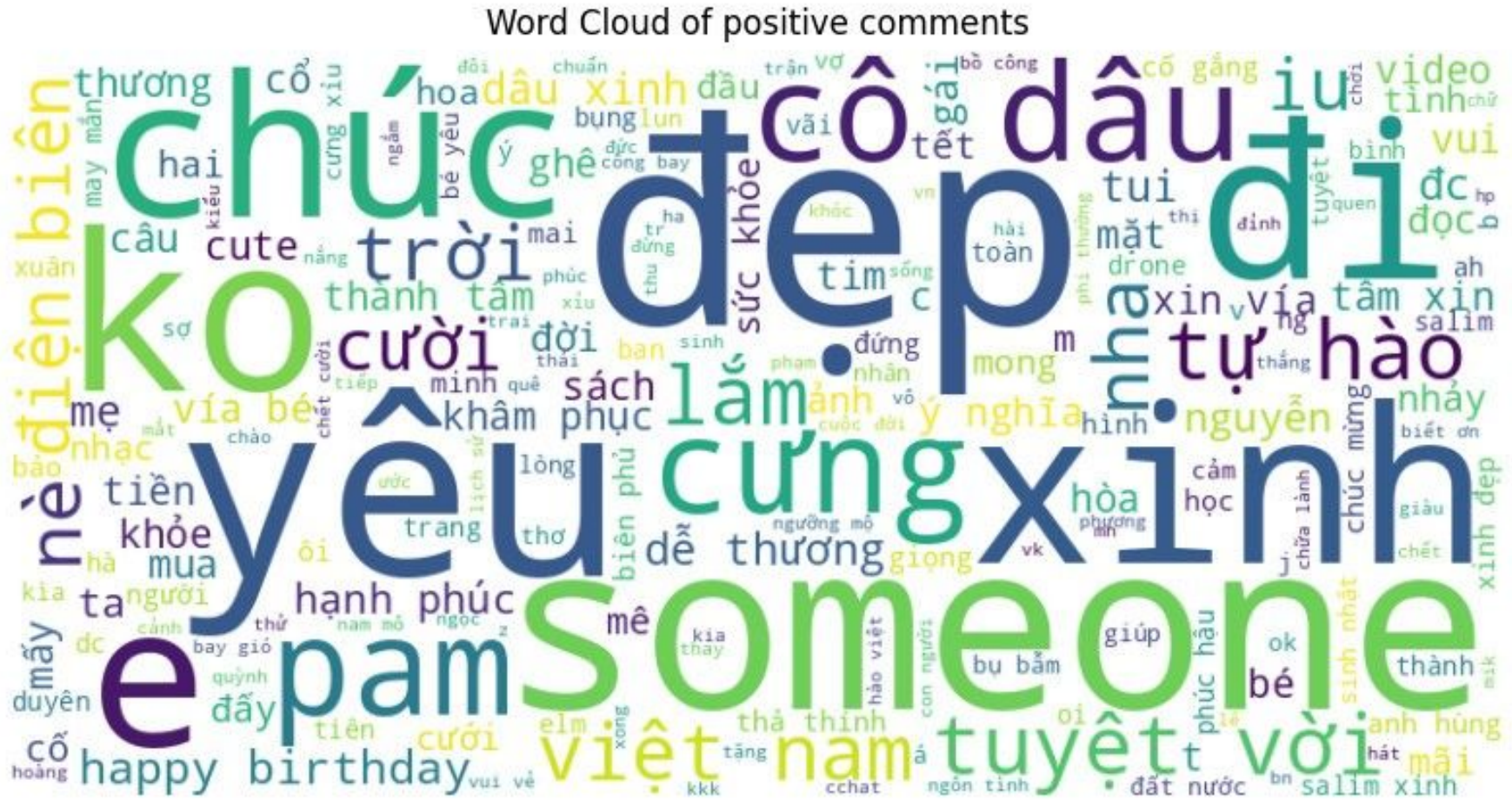
	Training set	Valid set	Test set	Tổng
Số lượng	5126	1050	257	6433

Data visualization





Data visualization



Several samples of data

	Comment	Tokenized	Tensor	Label
1	“iu nhất thế giới😊”	['iu', 'nhất', 'thế_ giới', '😊']	tensor([672, 153, 1, 1850])	1
2	“béo có tội gì chứ😔”	['béo', 'có', 'tội_gì', 'chứ', '😔']	tensor([1,4, 1, 35, 1011])	0
3	“hay😅”	['hay', '😅']	tensor([47, 555])	1
4	“tao hiểu :)) và tao đã chấm dứt ngay😉”	['tao', 'hiểu', ':))', 'và', 'tao', 'đã', 'chấm_dứt', 'ngay', '😉']	tensor([187, 179, 49, 41, 187, 60, 1, 473, 3662])	0

Models

RNN

Cấu trúc mạng:

RNN(

(**embedding**): Embedding(vocab_size=6250, embedding_dim=100, padding_idx)

(**rnn**): LSTM(embedding_dim = 100, hidden_dim=256, num_layers=2, dropout, bidirectional=True)

(**dropout**): Dropout(p, inplace=False))

(**fc**): Linear(in_features=512, out_features=1, bias=True)

Models

CNN

Cấu trúc mạng:

CNN(

(**embedding**): Embedding(input_dim, embedding_dim=100, padding_idx)

(**conv**): Conv1d(in_channels=100,out_channels=256 ,kernel_size=3)

(**Pool**): F.max_pool1d(conved, conved.shape[2])

(**fc**): Linear(in_features=256, out_features=1, bias=True)

Tuning CNN

Tuning hidden_dim, n_layers, dropout

	Hidden_dim	n_layers	Dropout	Best results	
				Loss Val	Acc Val
1	128	2	0.3	0.484	79.82%
2	128	2	0.5	0.45	80.27%
3	128	3	0.3	0.461	79.45%
4	128	3	0.5	0.457	80.55%
5	256	2	0.3	0.491	80.45%
6	256	2	0.5	0.481	80.45%
7	256	3	0.3	0.521	80.00%
8	256	3	0.5	0.568	80.00%

Tuning CNN

Tuning hyperparameter: CNN(hidden_dim = 128,n_layers = 2, dropout = 0.3)

	Loss Func	Optim	Learning-rate	Best results	
				Loss Val	Acc Val
1	BCEWithLogitsLoss()	Adam	0.001	0.443	79.91%
2	BCEWithLogitsLoss()	RMSprop	0.001	0.673	58.82%
3	BCEWithLogitsLoss()	RMSprop	0.001	0.452	79.55%
4	BCEWithLogitsLoss()	Adam	0.005	0.645	79.27%
5	BCEWithLogitsLoss()	SGD	0.005	0.646	63.00%
6	BCEWithLogitsLoss()	RMSprop	0.005	0.767	73.64%
7	BCEWithLogitsLoss()	Adam	0.01	0.831	80.36%
8	BCEWithLogitsLoss()	SGD	0.01	0.625	65.91%
9	BCEWithLogitsLoss()	RMSprop	0.01	0.683	79.36%

Tuning RNN

Tuning hidden_dim, n_layers, dropout

	Hidden-dim	N-layers	Dropout	Best Results	
				Loss Val	Acc Val (%)
1	128	2	0.3	.440	79.87
2	128	2	0.5	.434	81.53
3	128	3	0.3	.430	79.58
4	128	3	0.5	.431	80.44
5	**256**	**2**	**0.3**	**.422**	**81.15**
6	256	2	0.5	.426	80.72
7	256	3	0.3	.429	81.42
8	256	3	0.5	.437	81.25

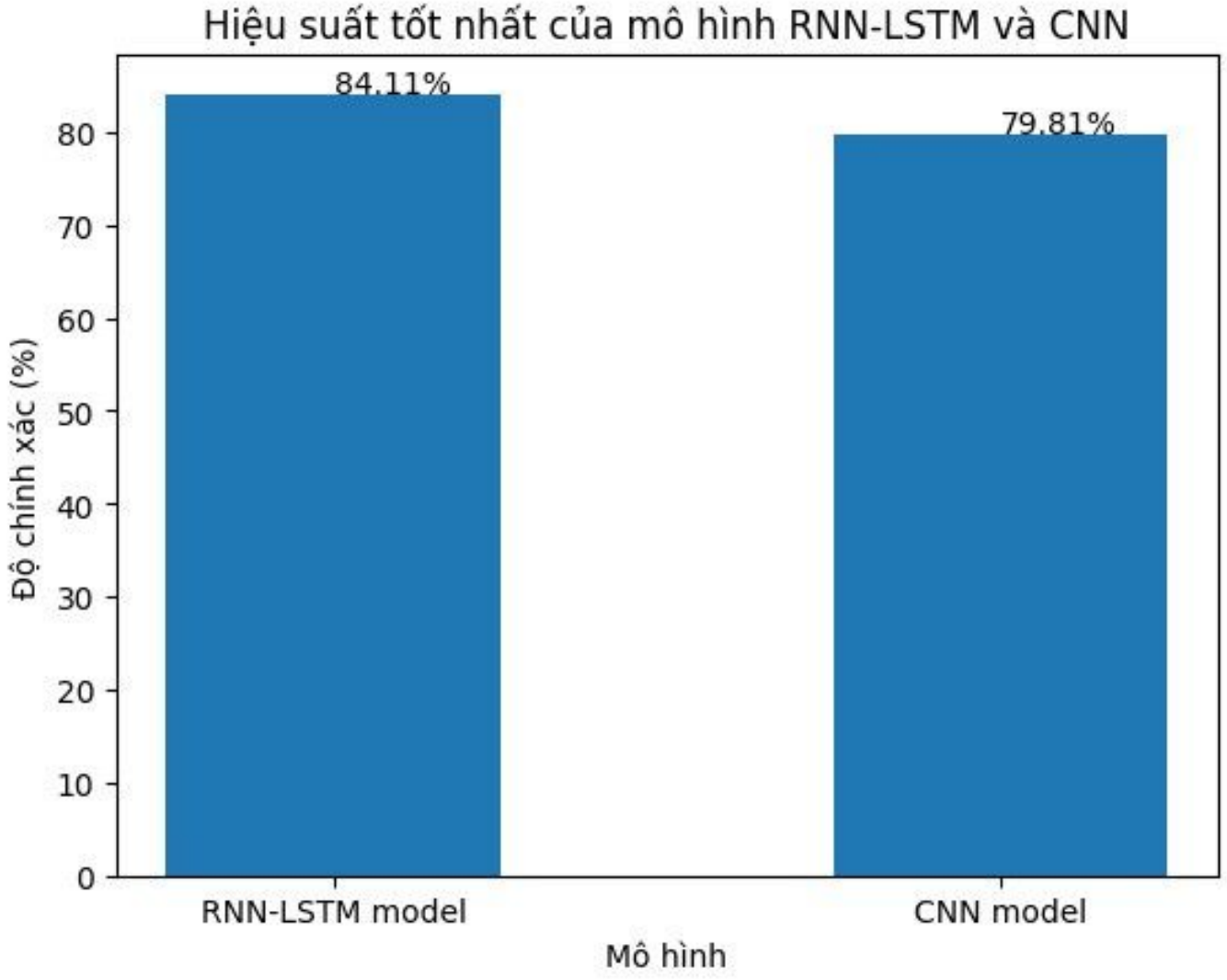
Tuning RNN

Tuning hyperparameter: RNN(hidden_dim = 256,n_layers = 2, dropout = 0.3)

	Loss Func	Optim	Learning-rate	Best Results	
				Loss Val	Acc Val
1	BCEWithLogitsLoss()	Adam	0.001	0.413	80.82%
2	BCEWithLogitsLoss()	SGD	0.001	0.6926	52.18%
3	BCEWithLogitsLoss()	RMSprop	0.001	0.421	80.09%
4	BCEWithLogitsLoss()	Adam	0.005	0.444	80.09%
5	BCEWithLogitsLoss()	SGD	0.005	0.6937	48.64%
6	BCEWithLogitsLoss()	RMSprop	0.005	0.465	77.64%
7	BCEWithLogitsLoss()	Adam	0.01	0.451	79.73%
8	BCEWithLogitsLoss()	SGD	0.01	0.493	49.09%
9	BCEWithLogitsLoss()	RMSprop	0.01	0.488	77.45%

Best model

Mô hình RNN-LSTM mang lại hiệu quả tốt hơn so với CNN



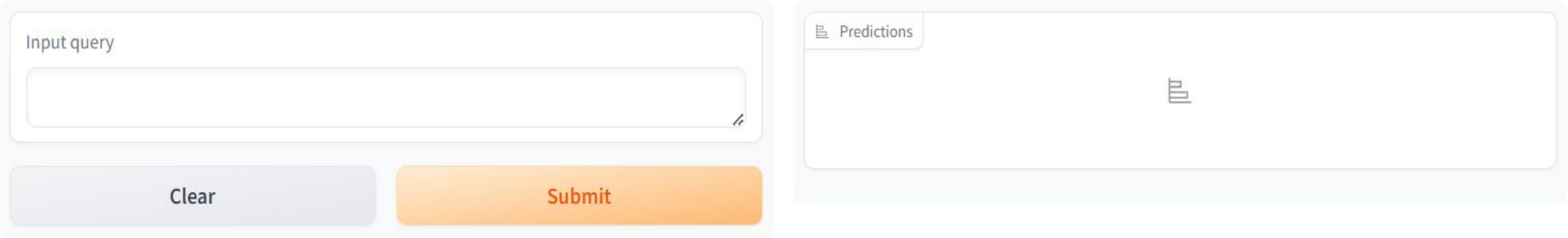
Sample prediction

Một số lý do dẫn đến model dự đoán sai:

- Gán nhãn sai cho một số dữ liệu
- Nguyên nhân lớn nhất là thiếu dữ liệu. Dữ liệu nhỏ dẫn đến số từ có trong từ điển thấp, vì vậy trong nhiều câu cho vào dự đoán thì có nhiều từ không xuất hiện trong từ điển, dẫn đến dự đoán sai
- Tập data training vẫn còn hạn chế, và chưa thực sự chất lượng. Một số từ “cười”, “khóc” hay cái icon “❤️”, “ =))” ,”((“ xuất hiện trong tập train nhiều về hẳn một lớp, vì vậy có xu hướng đánh xác suất cao cho lớp đó.

	Comment	Label	Pred
1	“Đúng vậy sao phải buồn vì thứ bỏ đi”	1	0
2	pam với khuôn mặt smiling **((** heart	1	0
3	Inter nay gà v	0	1
4	**cười** nhạt một cái rồi mai tính tiếp	0	1
5	Over thinh king	0	1
6	Cố lên ta ơi cuộc đời cho ta ngàn lý do để **khóc** .Nhưng ta cũng có ngàn lý do để **cười**	1	0
7	nghe phúc cười giòn như pháo tết ❤️	1	1
8	cưng quá 🥰	1	1
9	Không phải tự hào nhưng để nói không khí đón Tết và sự vui tươi hạnh phúc của nhân dân thì ko một đất nước nào đẹp hơn Việt Nam,nét truyền thống càng đc gìn giữ và ngày càng đẹp.	1	1
10	:) bỏ đi chị	0	0
11	nếu tha thứ đc thì tha thứ còn ko thì buông.. phụ nữ chúng mình khổ nhiều rồi.mạnh mẽ lên nhé 😭😭😭😭😭😭	0	0

Demo



Link: demo

Contribution

	Các đầu việc
Bắc	Crawl và xử lý data, Visualize, Train model, Train word2vec
Khải	Crawl và xử lý data, Visualize, Tuning hyper-parameters, Làm giao diện
Đức	Crawl và xử lý data, Tuning hyper-parameters, Làm slide

Limitations

- Việc crawl còn gặp bất lợi ở một số trang mạng xã hội, không sử dụng được API, selenium,...
- Vẫn còn nhãn chưa được gán đúng
- Data còn hạn chế, chất lượng chưa tốt
- Có một số câu câu gây bối rối, khó xác định được nhãn
- Số lượng models sử dụng còn ít, tuning chưa được nhiều

Conclusion

- Phân loại cảm xúc chính xác: Hệ thống có khả năng phân loại các comment chứa icons, emoticon, ký hiệu,.. thành các loại cảm xúc bao gồm tích cực, tiêu cực một cách tương đối chính xác.

- RNN cho thấy sự hiệu quả hơn CNN trong xử lý dữ liệu tuần tự như văn bản vì có khả năng trong việc truyền trạng thái ẩn qua các bước thời gian, giúp mô hình hiểu mối quan hệ giữa các phần tử liên tiếp.

Future work

- Tìm thêm cách crawl data hiệu quả hơn
- Update Annotation Guideline
- Tăng cường dataset đa dạng, phù hợp
- Thêm một số lớp phù hợp
- Tiến hành training, đánh giá trên nhiều models nữa
- Xây dựng giao diện tốt hơn, đáp ứng thêm nhiều nhu cầu của khách hàng.

Thank you!