**Báo cáo dự án Xử lý ngôn ngữ tự nhiên – Nhóm 8**

| **Thành viên:**  Phan Trung Thuận  Nguyễn Quang Tùng  Trần Trăng Sáng  Ung Khánh Như | **Công nghệ sử dụng:**  Pytorch, Scikit-learn  Pandas, matplotlib |
| --- | --- |

# I. Bài toán:

Phân loại báo tiếng Việt dựa trên nội dung của bài báo (content) vào 12 lớp được định nghĩa sẵn bao gồm:

1 Chính trị 2 Xã hội 3 Kinh tế 4 Văn hóa

5 Sức khỏe 6 Pháp luật 7 Thế giới 8 KHCN

9 Thể thao 10 Giải trí 11 Du lịch 12 Giáo dục

Chúng em tiếp cận theo hướng Sequence Classification và so sánh 4 mô hình bao gồm 1D CNN, Bidirectional LSTM, Bidirection GRU và MiniBERT do chúng em tự cài đặt.

# II. Bộ dữ liệu:

Bộ dữ liệu đã được cào từ 3 nguồn bao gồm: Vnexpress, Dantri và Vietnamnet, chủ yếu tập trung vào 2 thuộc tính đó là content và categories. Tuy nhiên, mỗi bạn cào dữ liệu theo các phương pháp khác nhau nên chúng em phải xử lý bằng cách:

* Tiền xử lý nội dung: Đầu tiên chúng em sẽ xóa các stop-words của tiếng Việt (danh sách stop-words có thể được tìm thấy [tại đây](https://github.com/stopwords/vietnamese-stopwords)). Tiếp theo, chúng em sẽ chia token sử dụng tokenizer của [PhoBERT](https://github.com/VinAIResearch/PhoBERT). Mặc dù đã giảm được chiều dài sequence (từ hơn 11000 xuống 8000) nhưng vẫn chưa đáng kể, chính vì vậy chúng em cần phải xây dựng các mô hình có thể xử lý được sequence rất dài.
* Đồng bộ thể loại bài báo: Do mỗi nguồn báo có danh sách thể loại khác nhau nên chúng em phải đồng bộ hóa chúng bằng cách dưới đây:

'khoa học' => 'khcn' 'bất động sản' => 'kinh tế' 'kinh doanh' => 'kinh tế'

'số hóa' => 'khcn' 'các bệnh' => 'sức khỏe' 'đời sống' => 'xã hội'

'xe' => xóa 'an sinh' => 'xã hội' 'lao động - việc làm' => 'kinh tế'

'thời sự' => 'xã hội' 'bạn đọc' => xóa 'bảo vệ người tiêu dùng' => 'kinh tế'

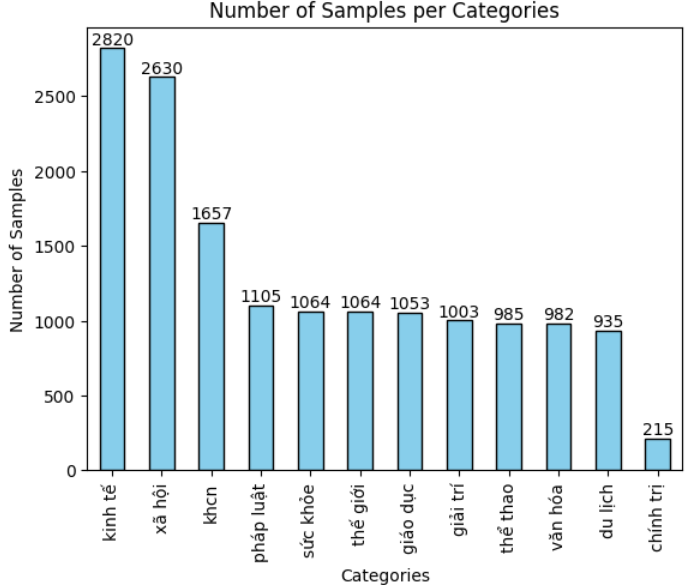
'công nghiệp hỗ trợ' => 'khcn' 'tuần việt nam' => 'văn hóa'

'thông tin và truyền thông' => xóa 'nội dung chuyên đề' => xóa

'dân tộc - tôn giáo' => 'xã hội' 'nông thôn mới' => 'xã hội'

'dân tộc thiểu số và miền núi' => 'văn hóa' 'tình yêu - giới tính' => 'văn hóa'

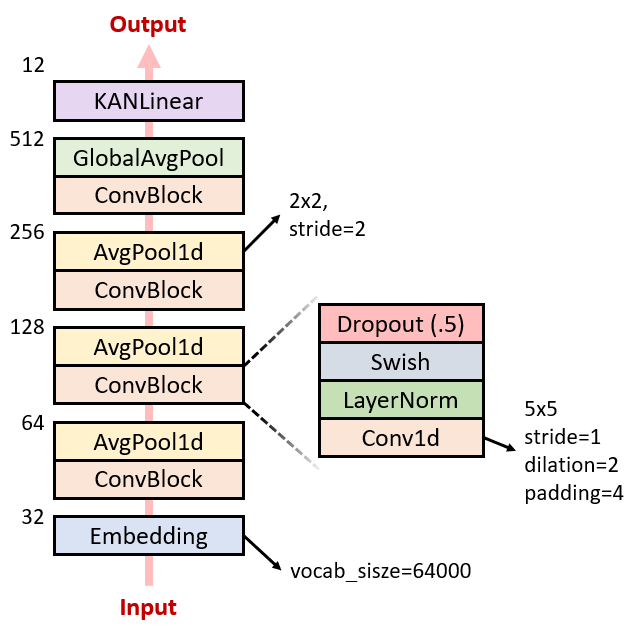
Sau khi tiền xử lý dữ liệu chúng em thu được dataset gồm 15513 bài báo đã được chia token với mỗi bài đã được padding để đạt độ dài sequence là 8000. Tuy nhiên do tính chất đặc thù của ngành báo chí nên bộ dữ liệu của chúng em bị mất cân bằng ở thể loại chính trị khi so sánh với các thể loại khác. Chúng em phân chia tập dữ liệu với tỉ lệ 6:4 cho huấn luyện và đánh giá. Vì chúng em muốn huấn luyện một mô hình với bộ dữ liệu ít nhưng đánh giá phức tạp để đo lường được sự tổng quát hóa một cách đáng tin cậy.



Hình 1: Biểu đồ số lượng bài báo trên mỗi thể loại trong tập dữ liệu

# III. Xây dựng mô hình:

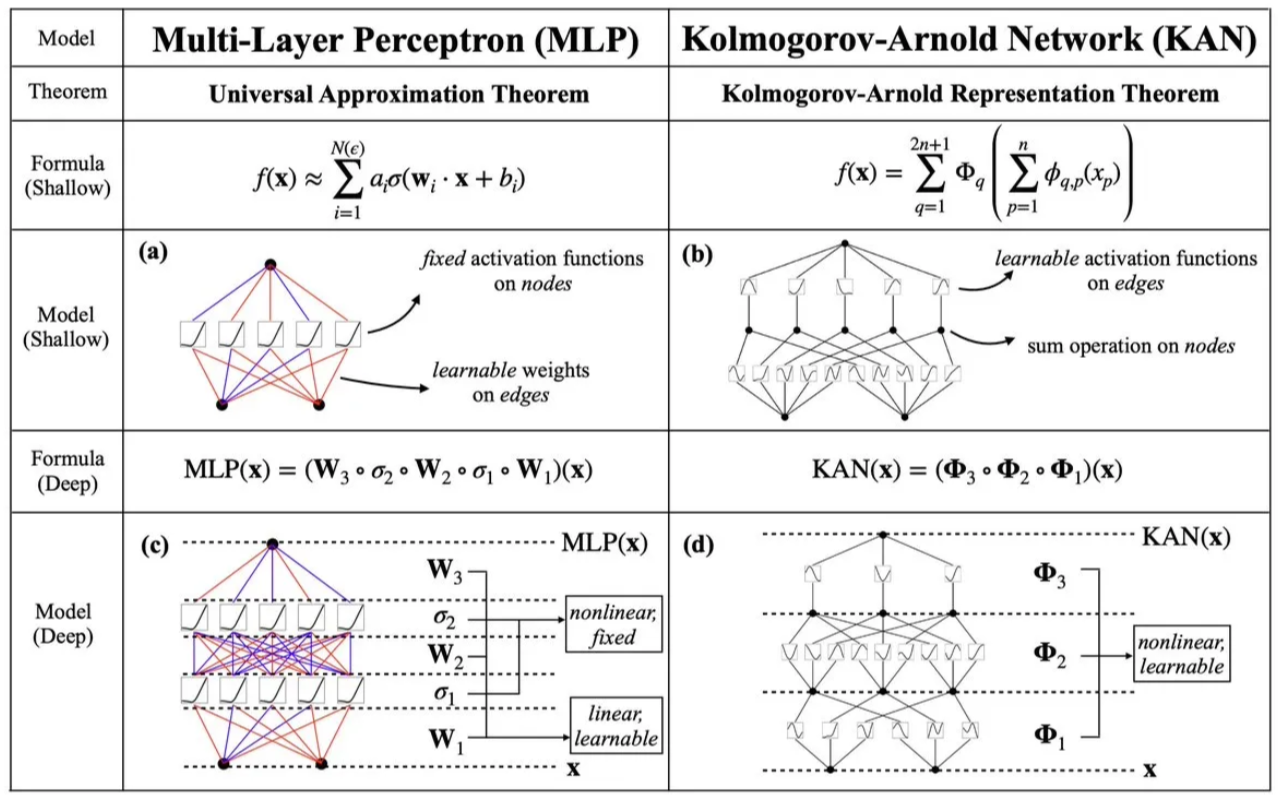
## 1D CNN:



Hình 2: Kiến trúc mô hình 1D CNN

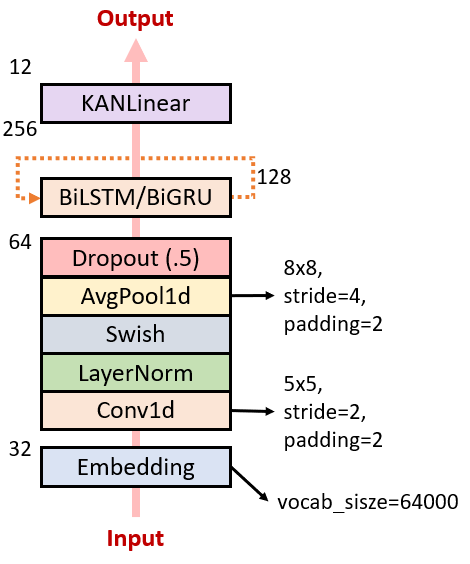
Chúng em sử dụng lớp Convolution 1d để trích xuất đặc trưng giữa các token, dilation được áp dụng để mô hình có thể tìm được mối quan hệ giữa các token liền kề một vùng rộng hơn, thưa hơn và cũng có khả năng bỏ qua những token ko cần thiết, padding được sử dụng để giữ độ dài sequence cố định, thuận tiện cho việc tính toán. Lớp Layer Normalization được áp dụng tiếp theo để chuẩn hóa dữ liệu về phân phối chuẩn dựa trên trung bình và độ lệch chuẩn của cả sequence. Activation function chúng em sử dụng là hàm Swish vốn có đạo hàm mượt mà hơn ReLU nhưng chi phí tính toán lại nhẹ hơn GELU. Chúng em còn áp dụng kỹ thuật Dropout để giúp mô hình đạt được sự tổng quát cao. Sau mỗi ConvBlock chúng em sẽ giảm độ dài sequence xuống một nửa. Cuối cùng là đi qua một lớp Global Average Pooling để chuyển thành vector và đưa qua một lớp Kolmogorov-Arnold Layer (KANLinear) để phân loại cho sequence trên.

Kolmogorov-Arnold Network (KAN) lần đầu được giới thiệu vào tháng 4 năm 2024 ([đường dẫn bài báo](https://arxiv.org/abs/2404.19756)) để thay thế MLP. Chúng em sử dụng KAN vì khả năng ánh xạ phi tuyến quá tốt, một lớp KAN có thể hội tụ nhanh hơn, hiệu quả hơn một mô hình MLP nhiều lớp.



Hình 3: So sánh mô hình MLP và KAN

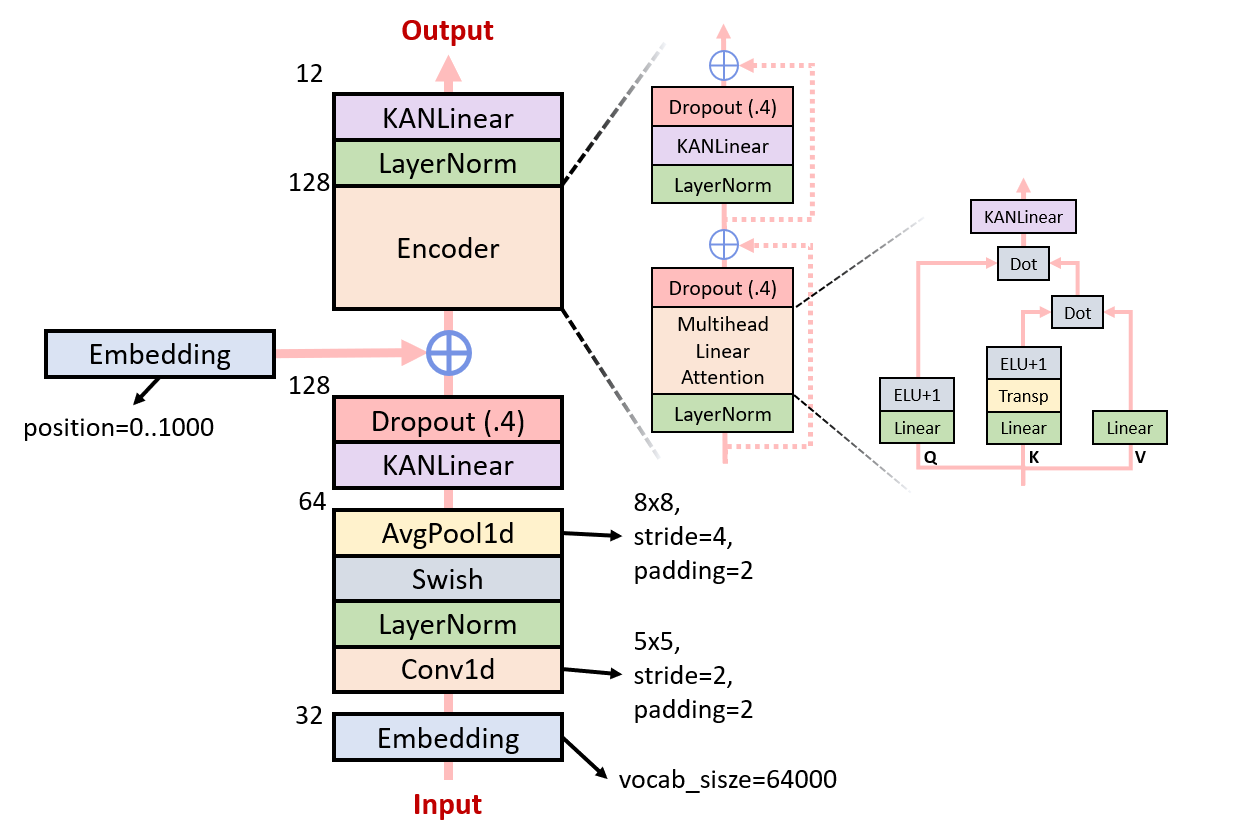
## BiLSTM và BiGRU:



Hình 4: Kiến trúc mô hình BiLSTM và BiGRU

Do mỗi bài báo được tokenizer phân chia tạo ra một sequence rất dài nên các mô hình RNN gặp rất nhiều khó khăn trong việc phân loại chính vì vậy chúng em phải giảm độ dài sequence từ 8000 xuống 1000 nhưng vẫn giữ được các đặc trưng cơ bản của bài báo để phân loại. Đầu tiên chúng em sẽ cho các embedding sequence qua một lớp Convolution 1d 5x5 với bước nhảy là 2 không chỉ giúp tìm được mối quan hệ của các token ở gần mà còn giảm độ dài sequence xuống một nửa. Sau đó chúng em sẽ đưa qua các lớp Layer Normalization và Swish activation function tương tự như 1D CNN. Để tiếp tục giảm độ dài sequence, chúng em áp dụng một lớp Average Pooling 1d để giảm xuống còn 1000. Sau đó, sequence mới có độ dài 1000 token sẽ được truyền qua một lớp Bidirectional LSTM hoặc Bidirectional GRU để trích xuất đặc trưng ngữ nghĩa của sequence. Do tính chất của bài toán phân lớp nên chúng em muốn một một hình có thể hiểu mối quan hệ giữa các token theo cả 2 chiều nên buộc phải dùng Bidirectional RNN. Sau khi mô hình BiLSTM tính toán hoàn tất, hidden state sẽ đi qua một lớp KANLinear để phân chia thể loại bài báo.

## MiniBERT:



Hình 5: Tổng quan về kiến trúc của mô hình MiniBERT

Vì độ dài sequence quá lớn và sự giới hạn về tài nguyên nên ở mô hình này, chúng em rút gọn sequence tương tự như ở các mô hình BiLSTM và BiGRU từ 8000 xuống 1000. Theo sau cách lớp để giảm độ dài sequence là 2 lớp KANLinear đóng vai trò là Input Embedding thứ 2 và Dropout hỗ trợ sự tổng quá hóa của mô hình. Sau đó chúng em sẽ cung cấp thêm thông tin về vị trí để khối Encoder có thể hiểu rõ đặc trưng về vị trí của các token. Ở khối Encoder này, chúng em sử dụng kiến trúc Pre-LayerNorm vì sự ổn định của nó trong quá trình huấn luyện ([bài báo](https://arxiv.org/pdf/2002.04745)) và sử dụng 1 encoder layer để có được sự so sánh công bằng nhất với các mô hình khác. Do cơ chế attention ở bài báo [“Attention is all you need!” (2017)](https://arxiv.org/abs/1706.03762) có độ phức tạp là O(N2d) với N là độ dài sequence (1000) và d là số chiều của các vector đầu vào (128) thì độ phức tạp này là quá lớn để có thể áp dụng. Chính vì vậy chúng em sử dụng cơ chế Linear Attention với Kernel function ([bài báo](https://arxiv.org/pdf/2006.16236)) đã giảm độ phức tạp xuống còn O(Nd2). Điều này đã tăng tốc đáng kể cho mô hình của chúng em. Sau cùng, chúng em sử dụng lớp LayerNorm và KANLinear cuối để phân loại các thể loại báo.

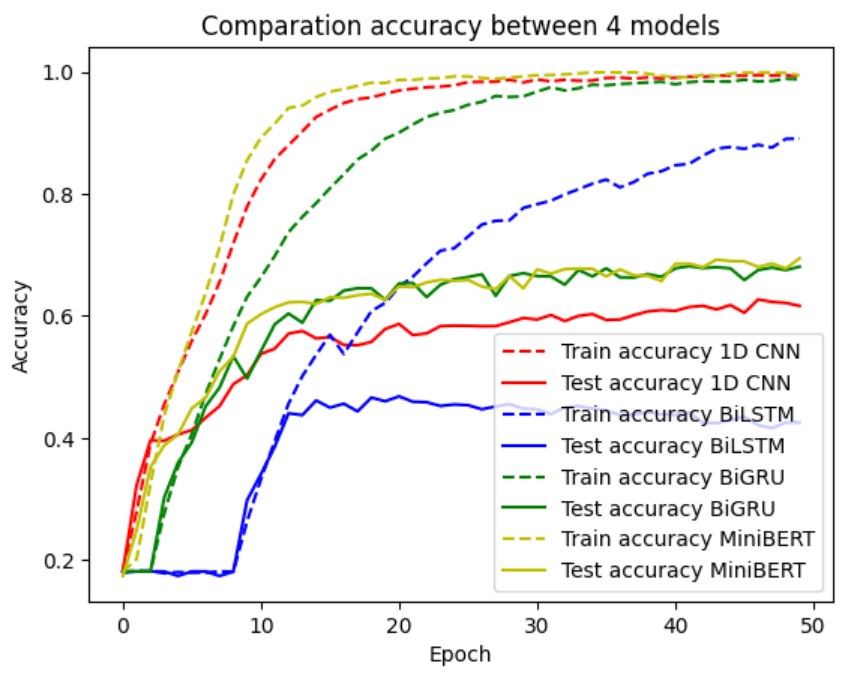
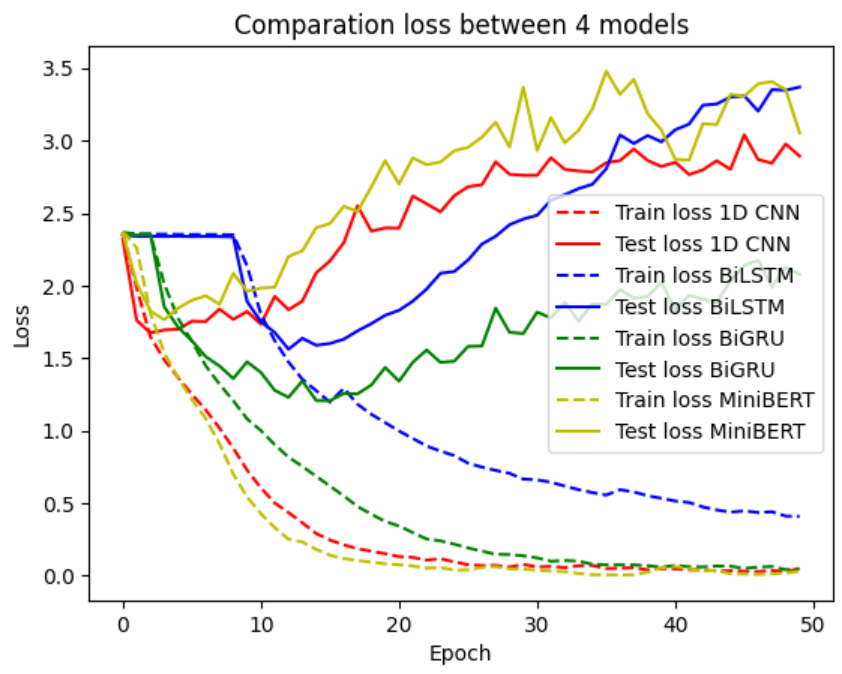
| **Dot-product Attention** | **Linear Attention** |
| --- | --- |
| Time complexity:  Memory complexity: | With  Time complexity:  Memory complexity: |

Bảng 1: So sánh giữa 2 cơ chế Attention

# IV. Các siêu tham số huấn luyện:

|  | 1D CNN | BiLSTM | BiGRU | MiniBERT |
| --- | --- | --- | --- | --- |
| Optimizer | AdamW | | | |
| Learning rate | 0.005 | | | |
| Weight decay | 0.01 | | | |
| Batch size | 64 | | | |
| Num epochs | 50 | | | |

# V. Kết quả thực nghiệm:



Hình 6: So sánh Loss và Accuracy giữa 4 mô hình đề xuất

Sau quá trình huấn luyện cho thấy tất cả các mô hình đều bị overfitting. MiniBERT mà chúng em đề xuất có performance tốt nhất nhưng cũng bị overfitting nặng nề nhất. BiLSTM là mô hình cho performance tệ nhất và hội tụ chậm nhất thậm chí khoảng 10 epoch đầu tiên mô hình có vẻ bị gradient vanishing, do LSTM là một mô hình khá cũ và khó có thể xử lý sequence dài (1000 tokens) và phức tạp. Không quá ngạc nhiên khi BiGRU là mô hình đạt hiệu quả chỉ thấp hơn MiniBERT khoảng 1% và là mô hình ít overfitting nhất vì nó đã được cải tiến rất nhiều so với LSTM. Điều bất ngờ đó là mô hình 1D CNN hội tụ nhanh tương đương với MiniBERT với tốc độ tính toán song song của các lớp tích chập, yêu cầu ít tài nguyên hơn để lưu trữ computational graph mà vẫn tốt hơn mô hình BiLSTM. Các lớp KAN classifier đã chứng minh tính ưu việt, khả năng ánh xạ phi tuyến cực kỳ tốt và khả năng hội tụ nhanh của nó bằng hiện tượng overfitting của các mô hình. Tuy nhiên chi phí tính toán vẫn khá cao khi so với MLP.

| Metrics | 1D CNN | BiLSTM | BiGRU | MiniBERT |
| --- | --- | --- | --- | --- |
| Accuracy | 0.61634 | 0.42539 | 0.68047 | **0.69417** |
| F1 score (Weighted) | 0.61809 | 0.43237 | 0.68302 | **0.69359** |

Bảng 2: So sánh các metrics đánh giá trên các mô hình đề xuất

|  |  |
| --- | --- |
|  |  |

Hình 7: So sánh confusion matrix giữa 4 mô hình đề xuất

# VI. Kết luận:

Đây vốn là một bài toán khó khi độ dài của các sequence rất lớn, một bài báo có thể dài cả ngàn từ nhưng mô hình BiGRU đã chứng minh được hiệu quả tuyệt vời của nó. Tuy nhiên chúng em cần scale up mô hình để có thể so sánh một cách công bằng hơn, mở rộng bộ dữ liệu hơn và giải quyết sự mất cân bằng dữ liệu đặc biệt ở thể loại chính trị. Mô hình KAN cần được hiệu chỉnh để tạo được một siêu phẳng tổng quát, tối ưu toàn cục, nếu không hiện tượng overfitting sẽ xảy ra ngày càng nhiều. Đây có vẻ là một đề tài nghiên cứu hấp dẫn. Hơn thế nửa bài toán long sequence classification cũng cần cải thiện bằng cách sử dụng các mô hình RNN hiện đại hơn như Mamba, RWKV, xLSTM hay TTT để có thể xử lý độ dài sequence lớn với lượng tài nguyên giới hạn.