

T5 cho bài toán Aspect-Based Sentiment Analysis (ABSA) trên bộ dữ liệu Tiếng Việt

Đặng Bá Quý¹, Trần Minh Quân²

Trường Đại học Nông Lâm, TP Hồ Chí Minh, Việt Nam
21130500@st.hcmuaf.edu.vn¹, 21130494@st.hcmuaf.edu.vn²

Tóm tắt

Trong những năm gần đây với việc bùng nổ những đánh giá của người dùng trên Internet, việc phân tích cảm xúc của những bình luận dựa trên những khía cạnh đang trở thành một nhiệm vụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên, cho phép hiểu rõ hơn về những cảm xúc được thể hiện đối với các khía cạnh cụ thể trong câu văn bản. Trong bài này chúng tôi sẽ trình bày một phương pháp tiếp cận bài toán ABSA (Aspect-Based Sentiment Analysis) bằng cách sử dụng mô hình T5 (Text-to-Text Transfer Transformer), một SOTA (state-of-the-art) framework có khả năng chuyển đổi các nhiệm vụ NLP khác nhau thành nhiệm vụ tạo văn bản. Cách tiếp cận của chúng tôi đơn giản hóa quy trình phức tạp của các mô hình ABSA truyền thống bằng cách tạo trực tiếp các cặp khía cạnh và cảm xúc trong một bước duy nhất, giảm thiểu nhu cầu cho các mô-đun truy xuất khía cạnh và phân loại cảm xúc riêng biệt. Chúng tôi thử nghiệm trên bộ dữ liệu tiêu chuẩn của VLSP (Vietnamese Language and Speech Processing) cho restaurant dựa trên mô hình ViT5 đạt được F1-scores 84.04% trong nhiệm vụ phát hiện khía cạnh và 72.29% trong cả hai nhiệm vụ trên.

1 Giới thiệu

Aspect-based sentiment analysis (ABSA) là nhiệm vụ trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Thay vì chúng ta chỉ phân loại một câu trong nhiệm vụ Sentiment Analysis (SA), ABSA sẽ phân tích câu văn bản sâu hơn bằng cách phân tích cảm xúc cho từng khía cạnh trong một câu văn bản. Ví dụ, trong thương mại điện tử, khách hàng thường có xu hướng thể hiện những cảm xúc tích cực hoặc tiêu cực của mình đối với các khía cạnh khác nhau của sản phẩm và dịch vụ. Với sự gia tăng nhanh chóng của các bình luận từ người dùng trên các trang web thương mại điện tử, Khai Thác Ý Kiến đã trở thành một lĩnh vực thu hút nhiều sự chú ý với mục tiêu phát triển các kỹ thuật khai thác văn bản. Ví dụ, trong thương mại điện tử, khách hàng thường bày tỏ cảm xúc tích cực hoặc tiêu cực của họ về các khía cạnh khác nhau của sản phẩm và

dịch vụ. Những đánh giá này được xem là nguồn tài nguyên miễn phí quý giá và là thước đo chất lượng cho cả tổ chức và người tiêu dùng. Trong các nghiên cứu trước, chúng tôi nhận thấy rằng việc phân tích các đánh giá của khách hàng là rất quan trọng. Nếu những đánh giá này được xử lý tốt, thông tin trích xuất sẽ đóng góp vào việc cải thiện sản phẩm và các chiến dịch chăm sóc khách hàng. Bên cạnh đó, các chiến dịch nghiên cứu thị trường, khảo sát và phản hồi hiện nay cũng rất phổ biến. Vì vậy, nghiên cứu các kỹ thuật có thể phân tích tự động các bình luận này để trích xuất thông tin hữu ích là rất cần thiết. Phân tích cảm xúc dựa trên khía cạnh (ABSA) là một trong những nhiệm vụ khám phá cảm xúc theo từng khía cạnh được đề cập trong bình luận.

Nhiệm vụ ABSA nhằm mục đích rút ra sự phân cực về tình cảm (ví dụ: positive, negative và neutral) đối với một mục tiêu, đó là khía cạnh của một thực thể cụ thể. Trong ABSA có thể được chia làm ba nhiệm vụ con (Pontiki et al., 2016): Aspect Detection, Opinion Target Expression, Sentiment Polarity. Tuy nhiên trong bài này chúng tôi chỉ tập trung vào nhiệm vụ chính là Aspect Detection và Sentiment Polarity trong bộ dữ liệu tiêu chuẩn VLSP 2018¹. Mục tiêu của của nhiệm vụ Aspect Detection là xác định được một hoặc nhiều khía cạnh khác nhau trong một tập khía cạnh đã được định nghĩa sẵn. Còn nhiệm vụ còn lại là Aspect Polarity, mục tiêu là phải xác định được sự phân cực tình cảm đối với các khía cạnh được phát hiện trong bài đánh giá. Ví dụ, cho một câu đánh giá “Về thức ăn và giá cả: Mình ăn phần ăn 279k nên được ăn khá nhiều các món, khá đa dạng, món nướng ướp ngon, đậm đà vừa phải.” với nhiệm vụ Aspect Detection sẽ có output là “food#prices”, “food#quality”, và “drinks#style&options”; trong khi đó nhiệm vụ Aspect Polarity, phân loại cảm xúc cho “food#prices” là *neutral*, “food#quality” là *positive* và “drinks#style&option” là *positive*. Từ ví dụ trên dùng chúng tôi thấy rằng những việc khai thác thông tin từ những bình luận của người dùng đóng vai trò quan trọng trong việc phát triển của các tổ chức kinh doanh.

Trong bài này chúng tôi sẽ trình bày việc sử dụng

¹<https://vlsp.org.vn/vlsp2018/eval/sa>

mô hình ViT5 (Phan et al., 2022) để giải quyết hai nhiệm vụ phụ cho vấn đề phân tích tình cảm dựa trên khía cạnh trên bộ dữ liệu tiếng việt cho 2 miền chính: aspect detection và aspect polarity. Chúng tôi sẽ đánh giá mô hình dự trên bộ dữ liệu tiêu chuẩn của VLSP cho lĩnh vực nhà hàng (restaurant). Bên cạnh đó chúng tôi cũng thử nghiệm đa dạng các phương pháp khác để so sánh hiệu suất của các phương pháp. Kết quả thử nghiệm cho thấy rằng phương pháp của chúng tôi vượt trội hơn về hiệu suất khi so với các phương pháp khác mà chúng tôi đã thử nghiệm.

2 Các công trình liên quan

Aspect-based sentiment analysis (ABSA) là nhiệm vụ đã thu hút được rất nhiều sự quan tâm của cộng đồng nghiên cứu từ rất lâu. Một vài workshop như SemEval2014 (Pontiki et al., 2014), SemEval2015 (Pontiki et al., 2015), SemEval2016 (Pontiki et al., 2016), VSLP2018 (Nguyen et al., 2019b), họ đã tổ chức giải này ra nhằm tìm được những các giải quyết vấn đề cho bài toán của họ một cách tốt nhất, và đã có rất nhiều phương pháp đã giải quyết được bài toán. Do bài toán bao gồm nhiệm vụ con là AD (Aspect dection) và SP (Sentiment Polarity), (Jihan et al., 2017) đã mô tả một hệ thống dùng máy học để trích xuất các khía cạnh từ các bài đánh giá khách hàng. Hệ thống này có thể trích xuất cả các khía cạnh rõ ràng (explicit) và ngầm (implicit), sử dụng Support Vector Machine (SVM) one-vs-rest (Zhang, 2001) (Xu et al., 2009), kết hợp với một pipeline tiền xử lý văn bản tiên tiến. Nó sử dụng các đặc trưng mới như vector trung bình (mean embedding) để cải thiện đáng kể độ chính xác của bộ phân loại SVM trên bộ dữ liệu của SemEval2016. (Bao et al., 2019) sử dụng mô hình AT-LSTM (Wang et al., 2016) làm baseline, đây là mô hình LSTM (Hochreiter and Schmidhuber, 1997) với cơ chế attention để có thể tập trung vào các từ quan trọng hơn liên quan đến từng khía cạnh (aspect) cần phân đoán, kết hợp thêm thông tin từ các từ điển sentiment (lexicon) vào mô hình, để tăng độ linh hoạt và sức mạnh của mô hình đồng thời họ cũng thêm regularizer cho attention vector để tránh tập trung quá vào một vài từ nhất định mà phân bỏ sự chú ý rộng hơn sang các từ khác cũng có thông tin quan trọng.

Trong tiếng Việt có một số bộ dữ liệu về phân tích tình cảm trong nhiều lĩnh vực. (Dang et al., 2022) đã sử dụng PhoBERT (Nguyen and Tuan Nguyen, 2020) làm mô hình ngôn ngữ được đào tạo trước cho tiếng Việt theo hai cách: Đa tác vụ (Multitask) và Đa tác vụ với cách tiếp cận đa nhánh (Multitask with Multi-branch approach), cả hai phương pháp đều cho kết quả tốt cho tác vụ AD và SP. Đặc biệt với phương pháp Multitask thì mô hình đạt được kết quả tốt nhất khi ghép bốn lớp cuối cùng của BERT (Devlin et al., 2019)

lại với nhau. Với bộ dữ liệu VSLP2018 ABSA trên Hotel domain họ đạt được kết quả cho F1-score là 82.55% trên nhiệm vụ AD và 77.32% cho nhiệm vụ SP, với miền Restaurant domain F1-score 83.29% và 71.55%. Ngoài ra còn (Dang et al., 2019) đề xuất phương pháp biến đổi (Transformation method) để giải quyết bài toán, họ tiếp cận bài toán như bài toán phân loại nhiều nhãn (Multi-label classification) và áp dụng phương pháp biến đổi để chuyển nó thành nhiều bài toán phân loại nhị phân (binary classification), để huấn luyện các bộ phân loại nhị phân, họ trích xuất các đặc trưng khác nhau từ các bài đánh giá và sử dụng tuyến tính SVM (Linear SVM) để phát hiện các khía cạnh và xác định cảm xúc của chúng. Qua phương pháp này họ đã đạt được F1-scores trên nhiệm vụ AD 77% và 69% trên hai bộ dữ liệu Restaurant domain và Hotel domain VLSP.

3 Phát biểu bài toán

3.1 Bài toán

Với thử thách từ aspect-based sentiment analysis chúng tôi xác định bài toán như sau: Cho một câu văn bản s và một tập đầu ra $\{a, p\}_{t=1}^T$ với p là polarity (positive, neutral, negative) của một aspect a . Aspect được xác định là một thực thể và một thuộc tính. Trong một câu văn bản có thể một hoặc nhiều hơn một tập T . Dưới đây là một ví dụ cho một câu văn bản trong bộ dữ liệu trong Bảng 1.

| Input | Output |
|--|---|
| Cá ba sa chiên giòn 45k: cá mềm, không bị khô nha, cần ra có hơi bốc khói nhẹ, loại này cũng có sốt riêng chấm luôn. | {food#price,neutral} {Food#Quality,positive} {Food#Price&style,neutral} |

Bảng 1: Một ví dụ về đầu vào và đầu ra trong tập dữ liệu.

3.2 Thuật toán

3.2.1 Suport Vector Machine

SVM là một thuật toán mạnh mẽ cho các nhiệm vụ phân loại. Trong nhiều ứng dụng thực tế, chúng ta cần xử lý các vấn đề phân loại nhiều lớp. Một cách tiếp cận phổ biến để mở rộng SVM nhị phân cho các bài toán đa lớp là chiến lược one-vs-rest (OvR). Trong bài này chúng tôi sẽ trình bày về Linear SVM sử dụng Hinge Loss với OvR.

SVM nhằm mục đích tìm ra một siêu phẳng giúp phân tách dữ liệu thành các lớp khác nhau một cách tốt nhất. Chiến lược OvR bao gồm việc đào tạo một bộ phân loại nhị phân riêng biệt cho mỗi lớp, trong đó mỗi bộ phân loại sẽ phân biệt một lớp với các lớp còn lại được định nghĩa bởi công thức 1.

Để phân loại nhiều lớp bằng OvR, chúng tôi đào

tạo k bộ phân loại nhị phân (Chi tiết vui lòng xem tại đây):

$$L(X, y, W) = \frac{1}{N} \sum_{n=1}^N \sum_{j \neq y_n} \max(0, \Delta - w_j^T x_n + w_j^T x_n) + \frac{\lambda}{2} \|x\|_F^2 \quad (1)$$

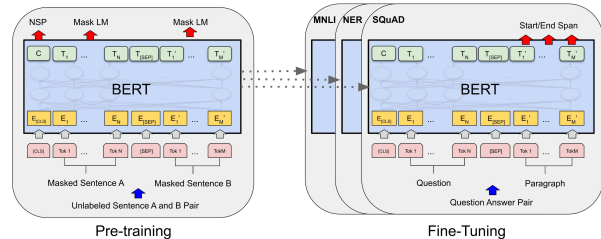
Ở công thức 1 được định nghĩa $X=[x_1, x_2, \dots, x_n]$ là các điểm dữ liệu, $y = [y_1, y_2, \dots, y_n]$ là vector chứa correct class, N là số training set trong tập dữ liệu, W là ma trận trọng số, x_n với y_n là các correct class trong điểm dữ liệu, w_j là cột thứ j của ma trận W , Δ và λ là các hyperparameter giúp cân bằng giữa data loss và regularization loss, $\|\cdot\|_F$ là Frobenius norm.

Multi-class SVM được coi là bộ phân loại tuyến tính như **Softmax Regression** vì đường phân chia đường phân chia giữa các lớp là các đường tuyến tính. Kernel SVM cũng cho kết quả khá tốt, nhưng việc tính toán ma trận kernel có thể tiêu tốn nhiều thời gian và tài nguyên bộ nhớ. Hơn nữa, mở rộng Kernel SVM cho bài toán phân loại đa lớp thường kém hiệu quả hơn so với Multi-class SVM. Một ưu điểm khác của Multi-class SVM là nó có thể được tối ưu hóa bằng (Stochastic) Gradient Descent, do đó thích hợp cho các bài toán có số chiều lớn như là văn bản. Trong thử nghiệm của chúng tôi thì Linear SVM hoạt động tốt hơn Kernel SVM về cả thời gian và hiệu suất.

3.2.2 PhoBERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) là một mô hình đào tạo trước (pre-trained model) mạnh mẽ được tinh chỉnh trên nhiều các tác vụ được minh họa trong Hình 2 nhưng nhiều ngôn ngữ có nguồn tài nguyên thấp, trong đó có tiếng Việt, vẫn gặp thách thức do thiếu dữ liệu để được đào tạo trước. Vì vậy PhoBERT (Nguyen and Tuan Nguyen, 2020) ra đời để giải quyết khoảng trống này bằng cách cung cấp một mô hình ngôn ngữ được đào tạo trước mạnh mẽ dành riêng cho tiếng Việt, tận dụng dữ liệu văn bản lớn không nhãn để cải thiện các nhiệm vụ NLP khác nhau.

PhoBERT được dựa trên kiến trúc của RoBERTa (Liu et al., 2019) là một phiên bản được tối ưu hóa của BERT. Nó được đào tạo trước trên một kho văn bản lớn bằng tiếng Việt (Vietnamese news corpus²) bằng cách sử dụng Masked Language Modeling (MLM) và nhiệm vụ Next Sentence Prediction (NSP). Mô hình này sử dụng kiến trúc transformer với nhiều lớp attention, tương tự như BERT gốc. Điểm khác biệt chính của PhoBERT là quá trình tiền huấn luyện trên một tập dữ liệu tiếng Việt lớn, bao gồm các bài báo, văn



Hình 1: BERT sử dụng cùng một kiến trúc cho cả quá trình huấn luyện trước và tinh chỉnh, chỉ khác ở lớp đầu ra. Các tham số ban đầu trong quá trình tinh chỉnh được lấy từ mô hình đã huấn luyện trước. Trong quá trình tinh chỉnh, tất cả các tham số đều được điều chỉnh. Ký hiệu đặc biệt [CLS] được thêm vào đầu mỗi ví dụ và [SEP] được dùng để phân tách các câu hoặc đoạn văn.

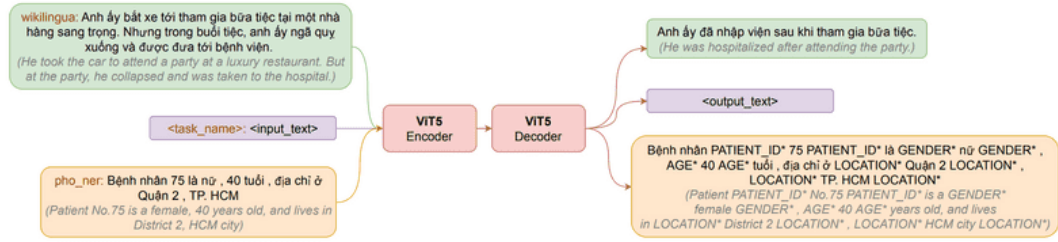
bản web và Wikipedia. Điều này giúp PhoBERT hiểu sâu sắc các đặc điểm ngôn ngữ Việt. Để xử lý hiệu quả việc tách từ tiếng Việt, PhoBERT sử dụng kỹ thuật RDRSegmenter (Nguyen et al., 2018) từ VNCORENLP (Vu et al., 2018), để thực hiện phân đoạn từ và câu trên tập dữ liệu huấn luyện trước, sau đó họ áp dụng kỹ thuật fastBPE (Sennrich et al., 2016) phân tách từ con (subword tokenization) được thiết kế để tối ưu hóa cả tốc độ và hiệu quả trong việc xử lý ngôn ngữ tự nhiên. Do PhoBERT có hai phiên bản *PhoBERT_{base}* và *PhoBERT_{large}*. Việc huấn luyện PhoBERT sử dụng nhiều tài nguyên tính toán, tận dụng GPU để tăng tốc quá trình huấn luyện, batch size 1024, learning rate 0.0004 cho phiên bản *BERT_{base}* và 512, 0.0002 cho phiên bản *PhoBERT_{large}*. Quá trình tiền huấn luyện diễn ra qua nhiều lượt trên tập dữ liệu tiếng Việt, giúp mô hình học sâu các mẫu và cấu trúc của ngôn ngữ.

3.2.3 ViT5

Các mô hình ngôn ngữ được đào tạo trước như T5 đã cách mạng hóa trong xử lý ngôn ngữ tự nhiên bằng cách chuyển đổi các tác vụ khác nhau thành định dạng chuyển văn bản thành văn bản. ViT5 là phiên bản tiếng Việt của T5, được thiết kế để nâng cao xử lý ngôn ngữ tự nhiên tiếng Việt bằng cách tận dụng dữ liệu văn bản không nhãn quy mô lớn.

ViT5 được tuân theo kiến trúc encoder-decoder (Wang et al., 2016) và T5 (Raffel et al., 2020). Các tác phẩm ban đầu của T5 đề xuất năm cấu hình kích thước mô hình khác nhau: small, base, large, 3B và 11B. Vì trong nghiên cứu nên những tác giả chỉ sử dụng những mô hình có kích thước base (310M parameter) và large (866M parameter) cho ViT5. Họ huấn luyện nhiều mô hình ViT5 trên nhiều nhiệm vụ khác nhau với hai độ dài đầu vào và đầu ra khác nhau: 256 and 1024-length. Họ đã thử nghiệm kỹ lưỡng với hai mô hình này để hiểu

²<https://github.com/binhvq/news-corpus>



Hình 2: Tổng quan về kiến trúc encoder-decoder ViT5, với các ví dụ đầu vào-đầu ra của hai downstream task. Đối với Named Entity Recognition, decoder sẽ xây dựng lại câu có chèn thêm các Entity tags.

rõ hơn về tầm quan trọng của độ dài dữ liệu tiền huấn luyện đối với các tác vụ tóm tắt văn bản. Cụ thể hơn, trong quá trình huấn luyện tự giám sát (self-supervised training), họ sử dụng một phương pháp gọi là "span-corruption" (làm hỏng một đoạn văn bản) với tỷ lệ làm hỏng là 15%. Điều này có nghĩa là họ sẽ ngẫu nhiên xóa 15% nội dung trong mỗi đoạn văn bản, và nhiệm vụ của mô hình là dự đoán lại những phần bị xóa đó. Kết quả cho thấy mô hình lớn hơn với ngữ cảnh lớn hơn (độ dài 1024) tối ưu tốt hơn đáng kể, dẫn đến hiệu suất tốt hơn trong các tác vụ sau này (như tóm tắt văn bản).

4 Thực nghiệm

4.1 Dữ liệu

Trong nghiên cứu này, chúng tôi đã đánh giá hiệu suất của các mô hình đề xuất trên bộ dữ liệu chuẩn dành cho tác vụ VSLP shared task năm 2018 (Nguyen et al., 2019a). Bộ dữ liệu này được thu thập từ các trang thương mại điện tử và được chú thích thủ công. Dữ liệu được chia thành ba tập: Training, development và testing. Bộ dữ liệu bao gồm 12 khía cạnh (aspect) và 3 cảm xúc (sentiment) khác nhau liên quan đến lĩnh vực nhà hàng. Bảng 2 sẽ trình bày thống kê tóm tắt chi tiết bộ dữ liệu lĩnh vực nhà hàng. Hình 3 sẽ minh họa phân phối cho mỗi aspect và sentiment trong tập dữ liệu.

| | N | l | $ V $ | $ I $ | A | $+$ | o | $-$ |
|-------|------|-----|-------|-------|-------|-------|-------|------|
| Train | 7028 | 71 | 7629 | - | 9,458 | 5179 | 2,533 | 1746 |
| Test | 1938 | 71 | 3712 | 1115 | 2629 | 1,503 | 691 | 435 |
| Dev | 771 | 71 | 2191 | 481 | 1053 | 579 | 278 | 196 |

Bảng 2: Tóm tắt thống kê cho bộ dữ liệu nhà hàng VSLP sau khi chúng tôi tiền xử lý dữ liệu. N là số lượng đánh giá, l biểu thị độ dài trung bình của đánh giá, $|V|$ là kích thước của từ điển, $|I|$ là số lượng từ vựng không nằm trong tập huấn luyện, A biểu thị số lượng aspect, (+, o, -) tương ứng với số lượng (positive, neutral, negative).

Preprocessing: Do dữ liệu của VLSP được thu thập từ các trang thương mại điện tử nên dữ liệu chưa được làm sạch, dẫn đến một số vấn đề như: lỗi chính tả, lỗi khoảng cách, ký tự đặc biệt, hashtag, URL, từ viết tắt, và nhiều vấn đề khác. Những yếu tố này có thể gây khó khăn trong việc xử lý và phân tích dữ liệu. Vì vậy, cần phải tiến hành các bước tiền xử lý để làm sạch và chuẩn hóa dữ liệu trước khi áp dụng các mô hình mà chúng tôi sẽ thử nghiệm. Các bước tiền xử lý bao gồm:

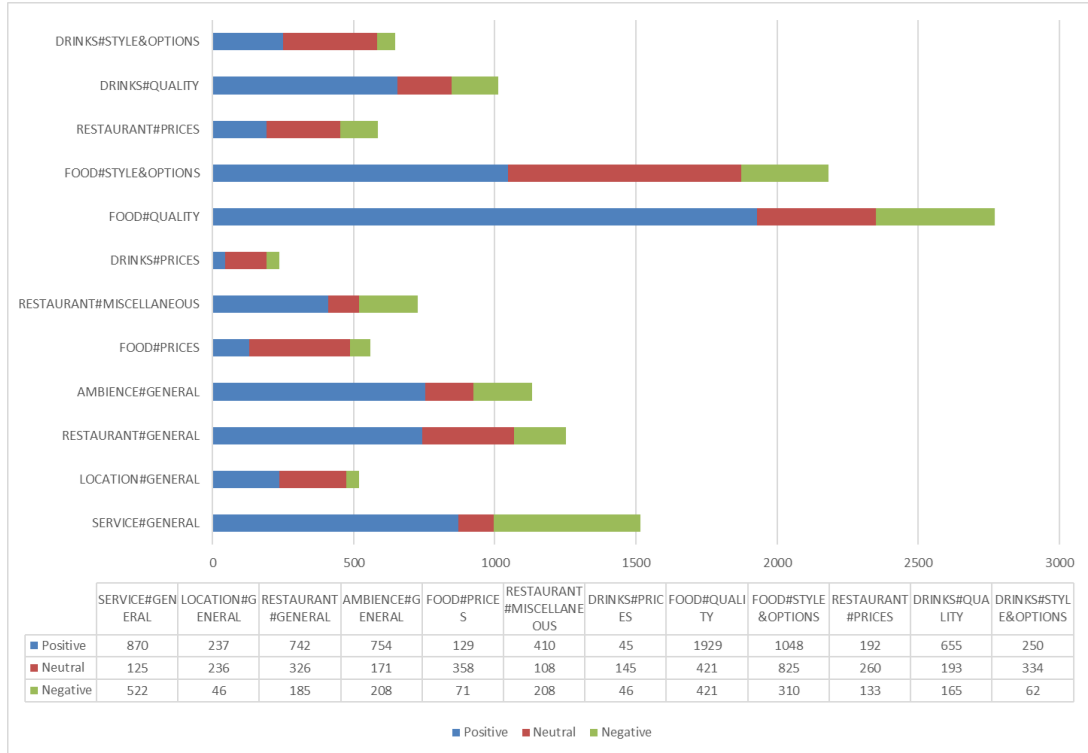
- **Bước 1:** HTML tags và những icon được loại bỏ khỏi câu.
- **Bước 2:** Chuẩn hóa bộ ký tự từ Windows-1252 sang UTF-8 và chuẩn hóa cách gõ tiếng Việt như chuẩn hóa kiểu gõ, chính tả và dấu câu.
- **Bước 3:** Các chữ viết tắt tiếng Việt thông dụng được chuẩn hóa cho từng miền trong tập dữ liệu như "khong", "ko", "khg", v.v., vì vậy những từ này sẽ được thay thế bằng từ được sử dụng thường xuyên nhất như là "không".
- **Bước 4:** Elongations (các từ có sự lặp lại của các chữ cái) đã được chuẩn hóa thành dạng từ thực sự của chúng (e.g., "ngooon" thì được chuẩn hóa thành "ngon").
- **Bước 5:** Trong tiếng Việt, một từ có thể được tạo thành từ hai hoặc nhiều từ khác (từ ghép), chẳng hạn như "nhà hàng" (mỗi từ có nghĩa riêng khi đứng một mình và có nghĩa khác khi kết hợp). Vì lý do này, cần phải phân đoạn văn bản và ghép chúng lại thành dạng như "nhà_hàng". Trong bài này, chúng tôi sử dụng Vietnamese Core NLP Toolkit Pyvi³ để phân đoạn từ.

4.2 Phương pháp

4.2.1 Multitask classification

Trong phần này, chúng tôi sẽ mô tả multi-task nhằm giải quyết hai nhiệm vụ con: AD (Aspect Detection) và AP (Aspect Polarity) trong ABSA. Đã

³<https://github.com/trungtv/pyvi>



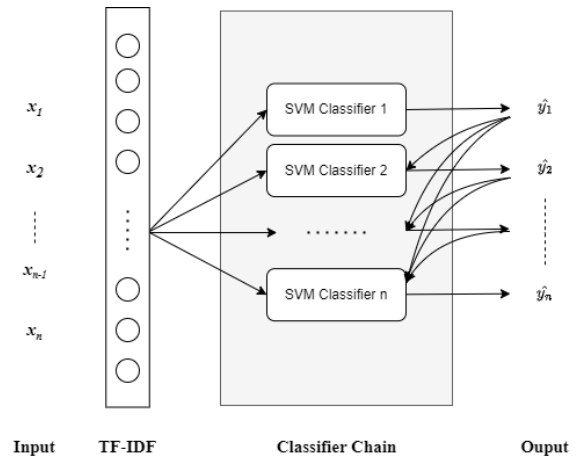
Hình 3: Phân phối aspect và sentiment trong cả ba tập dữ liệu training, testing, development

có nhiều nghiên cứu trước đây đề xuất các phương pháp giải quyết đồng thời cả hai nhiệm vụ này (Thin et al., 2020). Chúng tôi dựa trên những ý tưởng này để chuyển đổi bài toán thành một bài toán multi-label classification, nhằm xử lý hiệu quả cả hai nhiệm vụ AD và AP trong cùng một mô hình. Họ mô hình hóa đầu ra dưới dạng một danh sách các vector A trong đó A là số lượng khía cạnh của dữ liệu nhà hàng, với tổng cộng 12 khía cạnh. Mỗi vector v_i (với i từ 1 đến 12) có bốn phần tử, được ký hiệu là $v_i = [v_{i1}, v_{i2}, v_{i3}, v_{i4}]$ trong đó chỉ có một phần tử được gán giá trị 1 và các phần tử còn lại được gán giá trị 0. Bốn trạng thái của vector được giải mã thành các giá trị: None, Positive, Neutral hoặc Negative cho một khía cạnh cụ thể. "None" chỉ ra rằng đầu vào không chứa khía cạnh đó. Như vậy, với mỗi aspect i vector v_i sẽ chỉ ra trạng thái cảm xúc tương ứng của khía cạnh đó trong dữ liệu đầu vào. Ví dụ, nếu $v_1 = [0, 1, 0, 0]$, điều đó có nghĩa là khía cạnh đầu tiên có cảm xúc positive. Nếu $v_2 = [1, 0, 0, 0]$, điều đó có nghĩa là khía cạnh thứ hai không xuất hiện trong đầu vào.

4.2.2 SVM với Multitask classification

Chúng tôi áp dụng hai mô hình phân loại đa nhiệm là Multioutput Classifier và Classifier Chain từ thư viện Sklearn⁴ để giải quyết bài toán phân loại đa nhiệm. Quy trình thực hiện bao gồm các bước sau: tiền xử lý dữ liệu 4.1, xây dựng mô hình, huấn

luyện mô hình, và cuối cùng là đánh giá mô hình trên tập dữ liệu kiểm tra. Hình 4 sẽ mô hình hóa toàn bộ quy trình của chúng tôi. Trong quá trình



Hình 4: Kết hợp SVM Architecture và Classifier Chain cho phương pháp Multi-task Classification.

xây dựng mô hình, chúng tôi sử dụng Linear SVM làm mô hình cơ sở. Để trích xuất đặc trưng cho bộ phân loại, chúng tôi áp dụng phương pháp TF-IDF với N-gram, bao gồm unigram, bigram và trigram. Phương pháp này giúp nắm bắt được ngữ cảnh và quan hệ giữa các từ trong văn bản. Các tham số của Linear SVM được sử dụng mặc định.

Hiệu quả của phương pháp được đánh giá bằng các độ đo phổ biến như: tỷ lệ dự đoán chính xác

⁴<https://scikit-learn.org/stable/modules/multiclass>

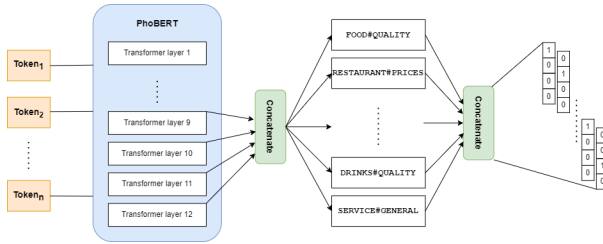
(Precision), tỷ lệ dự đoán chính xác trên tổng số nhân thực sự (Recall), và trung bình điều hòa của Precision và Recall (F1-score). Các độ đo này giúp đánh giá và so sánh hiệu quả của hai phương pháp Multioutput Classifier và Classifier Chain trong việc giải quyết bài toán multi-label classification.

4.2.3 PhoBERT với Multitask classification

Chúng tôi hiện thực lại phương pháp Multi-task với PhoBERT_{base} bằng cách kết hợp từ bốn lớp cuối cùng của mô hình BERT lại với nhau (Dang et al., 2022). Kỹ thuật này đã được chứng minh bài báo gốc của BERT (Devlin et al., 2019) và mang lại kết quả tốt nhất trong các thử nghiệm của họ.

| Feature-based approach BERT _{base} | Dev F1 |
|---|--------|
| Embeddings | 91.0 |
| Second-to-Last Hidden | 95.6 |
| Last Hidden | 94.9 |
| Weighted Sum Last Four Hidden | 95.9 |
| Concat Last Four Hidden | 96.1 |
| Weighted Sum All 12 Layers | 95.5 |

Bảng 3: Kết quả với các phương pháp tiếp cận dựa trên tính năng khác nhau của BERT.



Hình 5: PhoBERT bốn lớp cuối cùng với Multi-task.

Để tinh chỉnh được mô hình PhoBERT, chúng tôi thiết kế mô hình multitask để xử lý đồng thời hai nhiệm vụ AD và AP được minh họa ở hình 5. Đầu vào là một câu văn bản được mã hóa thành một vector $x_i \in \mathbb{R}^d$ với d là độ dài của một câu và $x_i \in X$, $X \in \mathbb{R}^{l \times d}$ với l là số input có trong tập dữ liệu. Sau đó đưa qua mô hình PhoBERT và lấy ra bốn last hidden state $H = [h_{-4}, h_{-3}, h_{-2}, h_{-1}]$. Tại đây chúng tôi concatenate các hidden states để tạo ra vector embedding kết hợp H_{pooled} và được truyền qua một lớp fully connected được tạo bằng cách ghép các lớp Dense A tương ứng với one-hot-vector A. Vì vậy chúng tôi sẽ thu được một lớp dense có 48 neuron cho tập dữ liệu nhà hàng (12×4) gọi là generated feature (g) và để có thể tính score cho giá trị dự đoán $\hat{y}^{(a)}$ của mô hình cho mỗi aspect $a \in A$ với softmax function:

$$\hat{y}^{(a)} = \text{softmax}(W^{(a)} \cdot g + b^{(a)}) \quad (2)$$

Vì vậy chúng tôi có thể dự đoán một aspect a cùng với polarity của nó chỉ trong một bước duy nhất:

$$\text{output}^{(a)} = \arg \max_i \hat{y}_i^{(a)}, \quad i = 0, 1, 2, 3 \quad (3)$$

Chúng tôi lấy Cross Entropy làm Loss function và nó được tính tổng trên tất cả các khía cạnh ở A.

$$\text{Loss}(y, \hat{y}) = - \sum_{a \in A} \sum_i y_i^{(a)} \cdot \log(\hat{y}_i^{(a)}) \quad (4)$$

Trong quá trình huấn luyện chúng tôi sử dụng mô hình pre-trained PhoBERT_{base} với batch size 32 và 16 cho quá trình xác thực. Độ dài cố định của input được đặt là 256 và số epoch cho quá trình huấn luyện là 10. Thuật toán tối ưu hóa AdamW (Loshchilov and Hutter, 2017) được sử dụng với learning rate khởi tạo là $1e-5$.

4.2.4 ViT5

Chúng tôi xem bài toán ABSA như một bài toán text-generation để có thể giải quyết đồng thời cả hai vấn đề AD và AP. Chiến lược của chúng tôi là từ một câu input, áp dụng mô hình ViT5 để tạo ra một câu văn bản khác phản ánh kết quả mong muốn. Ý tưởng này bắt nguồn từ (Raffel et al., 2020) họ tạo ra một Unified Framework chuyển đổi mọi vấn đề ngôn ngữ sang định dạng văn bản như tóm tắt văn bản, trả lời câu hỏi, phân loại văn bản, v.v. Chúng tôi muốn tạo ra một câu output có chứa tất cả các aspect và polarity tương ứng $Y = \{y_1, y_1, \dots, y_m\}$ với $Y \in \mathbb{R}^{1 \times m}$ và m là độ dài chưa được xác định, từ một câu input X có n từ với $X = \{x_1, x_1, \dots, x_n\}$. Để có thể xây dựng được mô hình Seq2Seq chúng tôi cần chuyển những nhãn mục tiêu thành một câu có thể hiểu theo ngôn ngữ tự nhiên. Vì nhãn của dữ liệu của chúng tôi đang có định dạng (e.g. FOOD#QUALITY, positive) vì vậy chúng tôi sẽ chuyển những output sang một câu hoàn chỉnh.

Slot-filling Template: Chúng tôi sẽ tạo một câu hoàn chỉnh bằng cách sử dụng định dạng Output Template gọi là Slot-filling (SF) để lấy nhãn cho các aspect và polarity tương ứng. Ví dụ cho một câu “*Mình uống trà sữa lài hạnh nhân khá thơm nha, vị hạnh nhân không đậm lắm nhưng mùi lài khá thơm, đậm vị trà hơi đắng nhẹ, độ ngọt vừa phải.*” và chúng tôi sẽ thu được câu “*Đánh giá thể hiện ý kiến [positive] cho [Drinks#Quality], [neutral] cho [Drinks#Style&Options].*”.

Unified Seq2Seq Model (US2S) : Chúng tôi cũng sử dụng ý tưởng của (Thin and Nguyen, 2023) họ ánh xạ những aspect và polarity sang một từ điển. Họ xây dựng từ điển bằng cách thống kê bằng cách áp dụng kỹ thuật TF-IDF trích xuất những mẫu trên aspect trên tập training set và

| Acronym | Enity#Attribute | Output Form | Acronym | Enity#Attribute | Output Form |
|---------|--------------------------|--------------------|---------|---------------------|--------------------|
| asp#1 | Restaurant#General | Nhà hàng nói chung | asp#7 | Drinks#Quality | Chất lượng đồ uống |
| asp#2 | Restaurant#Prices | Giá tiền nhà hàng | asp#8 | Drinks#Prices | Giá tiền đồ uống |
| asp#3 | Restaurant#Miscellaneous | Vấn đề khác | asp#9 | Drinks#Style&Option | Lựa chọn đồ uống |
| asp#4 | Food#Quality | Chất lượng đồ ăn | asp#10 | Location#General | Địa chỉ |
| asp#5 | Food#Prices | Giá tiền đồ ăn | asp#11 | Ambience#General | Không gian |
| asp#6 | Food#Style&Option | Lựa chọn đồ ăn | asp#12 | Service#General | Phục vụ |

Bảng 4: Sơ đồ thể hiện các danh mục aspect khác nhau cho dữ liệu nhà hàng.

development set. Từ vựng có điểm cao nhất được sử dụng để đại diện cho thể loại khía cạnh trong từ vựng tiếng việt. Bảng 4 là cách họ biểu diễn sơ đồ của các dạng khác nhau trong các aspect. Còn với những sentiment polarity họ chuyển chúng thành những từ tương ứng: “tốt” nếu “positive”, “tạm” nếu “neutral” và “tệ” nếu “negative”. Từ một câu cho một thể loại aspect và polarity tương ứng sẽ được biểu diễn dưới dạng định dạng output của danh mục aspect kết hợp với sentiment polarity. Ví dụ “Food#Quality, positive” thì sẽ được chuyển đổi thành “Chất lượng đồ ăn tốt” nếu có nhiều hơn một cặp aspect và polarity thì họ sẽ nối các cặp aspect và polarity lại bằng một từ nối “và”.

Trong bài này chúng tôi sẽ sử dụng lại kiến trúc ViT5(Phan et al., 2022). Một đầu vào chuỗi X được đưa vào encoder để thu được hidden states:

$$h^{encoder} = Encoder(X)$$

Tại mỗi bước thứ i của encoder mã thông báo được tạo y_i được tính toán dựa trên các hidden states và đầu ra $y_{0:i-1}$ trước đó để mang lại biểu diễn:

$$h_i^{encoder} = Decoder(h^{encoder}, y_{0:i-1})$$

Đầu ra của decoder được áp dụng vào hàm softmax để tính xác suất phân bố trên toàn bộ từ vựng cho mã thông báo tiếp theo. Hàm loss cho input-target được xây dựng:

$$Loss = \sum_i^m \log p_\theta(y_i | h^{encoder}, y_{0:i-1})$$

Với p_θ là phân phối xác suất của token kế tiếp y_i , θ được khởi tạo với các trọng số tham số được huấn luyện trước đó và được tinh chỉnh trong quá trình huấn luyện.

Chúng tôi sử dụng mô hình ViT5_{base} Để kiểm tra hiệu suất của mô hình. Do ViT5⁵ có hai phiên bản được tung ra cho cộng đồng nghiên cứu là ViT5_{base} và ViT5_{large}. Chúng tôi sử dụng độ dài dài nhất của input là 140 và độ dài output 30. Batch size mà chúng tôi thử nghiệm trên tập training set 16 và 8 cho tập development set với epoch là 10. Mô hình được tối ưu hóa bằng AdamW optimizer với learning rate khởi tạo là 2e-5.

⁵<https://github.com/vietai/ViT5>

4.3 Kết quả

Bảng 5 là kết quả chúng tôi đã thử nghiệm trên bộ dữ liệu nhà hàng của VSLP được đo bằng precision, recall, F1-score. Từ kết quả cho thấy phương pháp sử dụng mô hình ViT5 của chúng tôi vượt trội hơn các mô hình mà chúng tôi thử nghiệm trước đó.

| Task | Model | Precision | Recall | F1-score |
|-----------------|-----------|--------------|--------------|--------------|
| Aspect Dection | SVM | 80.50 | 63.56 | 69.14 |
| | PhoBERT | 89.23 | 79.39 | 83.86 |
| | ViT5-SFT | 85.71 | 82.6 | 83.98 |
| | ViT5-US2S | 84.91 | 83.34 | 84.04 |
| Aspect+Polarity | SVM | 69.07 | 52.28 | 55.86 |
| | PhoBERT | 77.58 | 59.94 | 61.29 |
| | ViT5-SFT | 72.17 | 69.48 | 70.17 |
| | ViT5-US2S | 73.59 | 71.68 | 72.30 |

Bảng 5: Các kết quả chính cho miền nhà hàng. Chúng tôi sử dụng các kết quả được báo cáo như SVM, PhoBERT và ViT5.

Trong nhiệm vụ AD, mô hình PhoBERT và các biến thể ViT5 (ViT5-SFT và ViT5-US2S) thể hiện hiệu quả vượt trội so với phương pháp học máy truyền thống như SVM. Cụ thể, PhoBERT đạt F1-score là 83.86%, trong khi ViT5-SFT và ViT5-US2S đạt lần lượt là 83.98% và 84.04%. Điều này cho thấy các mô hình dựa trên transformer, đặc biệt là ViT5-SFT có điểm F1-score và recall cao nhất lần lượt là 83.34% và 84.04%. Nhưng PhoBERT thể hiện điểm Precision vượt trội hơn hai mô hình ViT5 với 89.23%.

Trong nhiệm vụ AP, các mô hình transformer tiếp tục thể hiện ưu thế. ViT5-US2S đạt F1-score cao nhất là 72.29%, theo sau là ViT5-SFT với 70.17%. PhoBERT đạt F1-score là 61.29%, vẫn tốt hơn nhiều so với SVM 55.86%, nhưng không bằng các biến thể ViT5. Nhưng trong nhiệm vụ này mô hình PhoBERT vẫn thể hiện được tốt điểm precision 77.58% vượt trội so với hai biến thể vừa ViT5.

5 Kết luận

Trong bài này chúng tôi đã trình bày các phương pháp để giải quyết vấn đề của VSLP cho bài toán ABSA bằng các phương pháp học máy truyền

thống như SVM và sử dụng các mô hình pretrained như PhoBERT, ViT5. Qua đó ta thấy được rằng mô hình ViT5 cho được hiệu suất tốt nhất cho cả hai nhiệm vụ AD và AP, nhưng để cho mô hình T5 đạt hiệu suất tốt thì cũng phụ thuộc vào cách mà chúng tôi chuẩn hóa output. ViT5-US2S với output được chuẩn hóa theo phương pháp của tác giả (Thin and Nguyen, 2023) thì cho hiệu suất tốt hơn khi chúng tôi tự chuẩn bị output bằng phương pháp SFT. Tuy vậy chúng tôi cũng phải đánh đổi về thời gian và chi phí tính toán cho mô hình ViT5. Trong tương lai, chúng tôi dự định thử nghiệm các kỹ thuật như Instruction và fine-tuning với các mô hình hiện đại như GPT. Large Language Model (LLM) hiện đang là một chủ đề nổi bật, và chúng tôi sẽ tiến hành khảo sát bằng các kỹ thuật few-shot và zero-shot để đánh giá hiệu suất của các mô hình này.

Tài liệu tham khảo

- Lingxian Bao, Patrik Lambert, and Toni Badia. 2019. [Attention and lexicon regularized LSTM for aspect-based sentiment analysis](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 253–259, Florence, Italy. Association for Computational Linguistics.
- Hoang-Quan Dang, Duc-Duy-Anh Nguyen, and Trong-Hop Do. 2022. [Multi-task solution for aspect category sentiment analysis on vietnamese datasets](#). In *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, pages 404–409.
- Thin Dang, Duc-Vu Nguyen, Kiet Nguyen, and Nguyen Ngan. 2019. [A transformation method for aspect-based sentiment analysis](#). *Journal of Computer Science and Cybernetics*, 34:323–333.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Nadheesh Jihan, Yasas Senarath, Dulanjaya Tennekoon, Mithila Wickramaratne, and Surangika Ranathunga. 2017. Multi-domain aspect extraction using support vector machines. In *Proceedings of the 29th conference on computational linguistics and speech processing (ROCLING 2017)*, pages 308–322.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pre-training approach](#). *ArXiv*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2017. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Dat Quoc Nguyen, Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. 2018. A Fast and Accurate Vietnamese Word Segmenter. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2582–2587.
- Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. [PhoBERT: Pre-trained language models for Vietnamese](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, Online. Association for Computational Linguistics.
- Huyen Nguyen, Hung Nguyen, Quyen Ngo, Luong Vu, Vu Mai, Ngo Xuan Bach, and Cuong Le. 2019a. [Vlsp shared task: Sentiment analysis](#). *Journal of Computer Science and Cybernetics*, 34:295–310.
- Huyen T M Nguyen, Hung V Nguyen, Quyen T Ngo, Luong X Vu, Vu Mai Tran, Bach X Ngo, and Cuong A Le. 2019b. [Vlsp shared task: Sentiment analysis](#). *Journal of Computer Science and Cybernetics*, 34(4):295–310.
- Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H. Trinh. 2022. [ViT5: Pretrained text-to-text transformer for Vietnamese language generation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 136–142, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryigit. 2016. [SemEval-2016 task 5: Aspect based sentiment analysis](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [SemEval-2015 task 12: Aspect based sentiment analysis](#). In *Proceedings of the*

9th International Workshop on Semantic Evaluation (SemEval 2015), pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [SemEval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Dang Thin, Duc-Vu Nguyen, Kiet Nguyen, Ngan Nguyen, and Tu-Anh Hoang. 2020. [Multi-task Learning for Aspect and Polarity Recognition on Vietnamese Datasets](#), pages 169–180.

Dang Thin and Ngan Nguyen. 2023. [Aspect-category based sentiment analysis with unified sequence-to-sequence transfer transformers](#). *VNU Journal of Science: Computer Science and Communication Engineering*, 39(2).

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. [Vn-CoreNLP: A Vietnamese natural language processing toolkit](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. [Attention-based LSTM for aspect-level sentiment classification](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Lei Xu, Guangzhou Zhao, and Hong Gu. 2009. [Novel one-vs-rest classifier based on svm and multi-spheres](#). *Journal of Zhejiang University. Engineering Science*, 43.

Tong Zhang. 2001. [An introduction to support vector machines and other kernel-based learning methods](#). *AI Magazine*, 22(2):103.