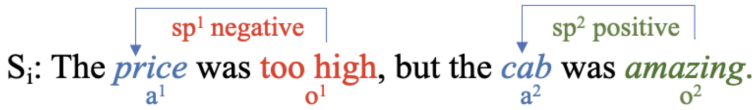
AI VIET NAM – AIO COURSE 2024

Project: Aspect-based Sentiment Analysis

Quoc-Thai Nguyen và Quang-Vinh Dinh

Ngày 7 tháng 2 năm 2025

Phần 1. Giới thiệu



Subtask	Input	Output
Aspect Term Extraction (ATE)	S_i	a^1, a^2
Aspect Term Sentiment Classification (ATSC)	$S_i + a^1$, $S_i + a^2$	sp^1, sp^2
Aspect Sentiment Pair Extraction (ASPE)	S_i	(a^1, sp^1), (a^2, sp^2)
Aspect Oriented Opinion Extraction (AOOE)	$S_i + a^1$, $S_i + a^2$	o^1, o^2
Aspect Opinion Pair Extraction (AOPE)	S_i	(a^1, o^1), (a^2, o^2)
Aspect Opinion Sentiment Triplet Extraction (AOSTE)	S_i	(a^1, o^1, sp^1), (a^2, o^2, sp^2)

Hình 1: Phân tích cảm xúc mức khía cạnh (Aspect-Based Sentiment Analysis)

.

Phân tích cảm xúc mức khía cạnh (Aspect-Based Sentiment Analysis) là bài toán có nhiều ứng dụng hiện nay trong việc phân tích các khía cạnh khác nhau của các bình luận, đánh giá về các sản phẩm, dịch vụ,... Ví dụ mình hoạ được minh hoạ như hình 1, với câu đầu vào: "The price was too high, but the cab was amazing.". Trong câu này, có hai khía cạnh được đánh giá là: "price" và "amazing". Với khía cạnh "price" được đánh giá là "too high" nên sẽ là "negative" và với khía cạnh "positive" được đánh giá là "amazing" nên sẽ là "positive".

Dựa vào việc xác định các giá trị đầu ra của mô hình, chúng ta có thể có một số bài toán nhỏ hơn như sau:

1. Aspect Term Extraction (ATE) hoặc Aspect-Based Term Extraction: trích xuất các khía cạnh được đánh giá trong bình luận

2. Aspect Term Sentiment Classification (ATSC): dựa vào câu đầu vào và các khía cạnh được đánh giá trong bình luận để dự đoán cảm xúc của bình luận
3. Aspect Sentiment Pair Extraction (ASPE): dựa vào câu đầu vào, trính xuất ra khía cạnh và dự đoán cảm xúc của các khía cạnh
4. Aspect Oriented Opinion Extraction (AOOE): dựa vào câu đầu vào và khía cạnh, dự đoán đoạn văn bản thể hiện cảm xúc của khía cạnh
5. Aspect Opinion Pair Extraction (AOPE): dựa vào câu đầu vào trích xuất thông tin về khía cạnh và đoạn văn bản thể hiện cảm xúc của khía cạnh
6. Aspect Opinion Sentiment Triplet Extraction (AOSTE): dựa vào câu đầu vào, trích xuất các thông tin về khía cạnh, đoạn văn bản thể hiện cảm xúc của khía cạnh và cảm xúc của khía cạnh trong bình luận

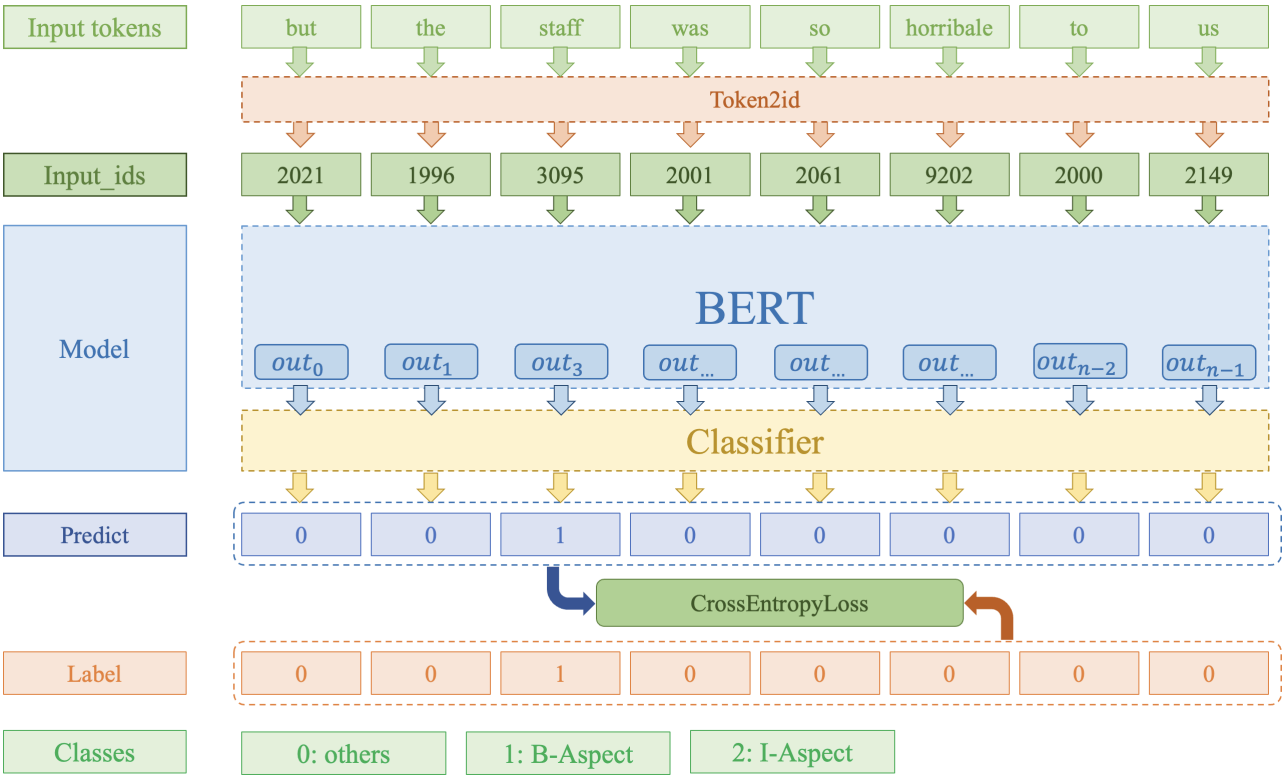
Trong phần này, chúng ta sẽ xây dựng mô hình giải quyết vấn đề cho bài toán ASPE. Dựa vào 2 bước chính:

1. Bước 1: Dự đoán từ trong văn bản thể hiện khía cạnh, hay chính là bài toán ATE
2. Bước 2: Dựa vào văn bản đầu vào và vị trí từ thể hiện khía cạnh dự đoán cảm xúc cho khía cạnh, hay là bài toán ATSC

Các thực nghiệm được xây dựng trên bộ dữ liệu SemEval-2014 Task 4: Aspect Based Sentiment Analysis và đã qua tiền xử lý cơ bản gồm: xoá bỏ dấu câu, chuẩn hoá và tách dựa vào khoảng trắng.

Phần 2. Aspect Term Extraction

Ở trong phần này, chúng ta xây dựng mô hình ATE dựa vào các phương pháp giải quyết bài toán sequence classification được sử dụng trong POS Tagging, NER,...



Hình 2: Aspect term extraction pipeline.

1. Build Dataset

```
# install lib
!pip install -q datasets==3.2.0

from datasets import load_dataset

ds = load_dataset("thainq107/abte-restaurants")
```

2. Tokenization

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("distilbert/distilbert-base-uncased")

def tokenize_and_align_labels(examples):
    tokenized_inputs = []
    labels = []
    for tokens, tags in zip(examples['Tokens'], examples['Tags']):
        tokens = tokens.replace("'", "").strip("][").split(', ')
        tags = tags.strip(']['').split(', ')

```

```
        bert_tokens = []
        bert_tags = []
        for i in range(len(tokens)):
            t = tokenizer.tokenize(tokens[i])
            bert_tokens += t
            bert_tags += [int(tags[i])]*len(t)

        bert_ids = tokenizer.convert_tokens_to_ids(bert_tokens)

        tokenized_inputs.append(bert_ids)
        labels.append(bert_tags)

    return {
            'input_ids': tokenized_inputs,
            'labels': labels
        }

preprocessed_ds = ds.map(tokenize_and_align_labels, batched=True)
```

3. Data Collator

```
from transformers import DataCollatorForTokenClassification

data_collator = DataCollatorForTokenClassification(tokenizer=tokenizer)
```

4. Evaluate

```
# Install lib
!pip install -q seqeval==1.2.2
import numpy as np
from seqeval.metrics import accuracy_score

def compute_metrics(p):
    predictions, labels = p
    predictions = np.argmax(predictions, axis=2)
    true_predictions = [
        [str(p) for (p, l) in zip(prediction, label) if l != -100]
        for prediction, label in zip(predictions, labels)
    ]
    true_labels = [
        [str(l) for (p, l) in zip(prediction, label) if l != -100]
        for prediction, label in zip(predictions, labels)
    ]
    results = accuracy_score(true_predictions, true_labels)
    return {"accuracy": results}
```

5. Model

```
from transformers import AutoModelForTokenClassification
id2label = {
    0: "O",
    1: "B-Term",
    2: "I-Term"
}
label2id = {
    "O": 0,
    "B-Term": 1,
    "I-Term": 2
}
```

```
12
13 model = AutoModelForTokenClassification.from_pretrained(
14     "distilbert/distilbert-base-uncased",
15     num_labels=3, id2label=id2label, label2id=label2id
16 )
```

6. Training

```
1  import os
2  os.environ['WANDB_DISABLED'] = 'true'
3  from transformers import TrainingArguments, Trainer
4
5  training_args = TrainingArguments(
6      output_dir="abte-restaurants-distilbert-base-uncased",
7      learning_rate=2e-5,
8      per_device_train_batch_size=16,
9      per_device_eval_batch_size=16,
10     num_train_epochs=5,
11     weight_decay=0.01,
12     eval_strategy="epoch",
13     save_strategy="epoch",
14     load_best_model_at_end=True
15 )
16
17 trainer = Trainer(
18     model=model,
19     args=training_args,
20     train_dataset=preprocessed_ds["train"],
21     eval_dataset=preprocessed_ds["test"],
22     processing_class=tokenizer,
23     data_collator=data_collator,
24     compute_metrics=compute_metrics,
25 )
26
27 trainer.train()
```

Kết quả training và accuracy trên tập test là 81.73%

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.476034	0.802080
2	No log	0.473619	0.817293
3	0.339000	0.511703	0.814453
4	0.339000	0.557133	0.811468
5	0.167300	0.581737	0.810120

Hình 3: Quá trình huấn luyện mô hình ATE.

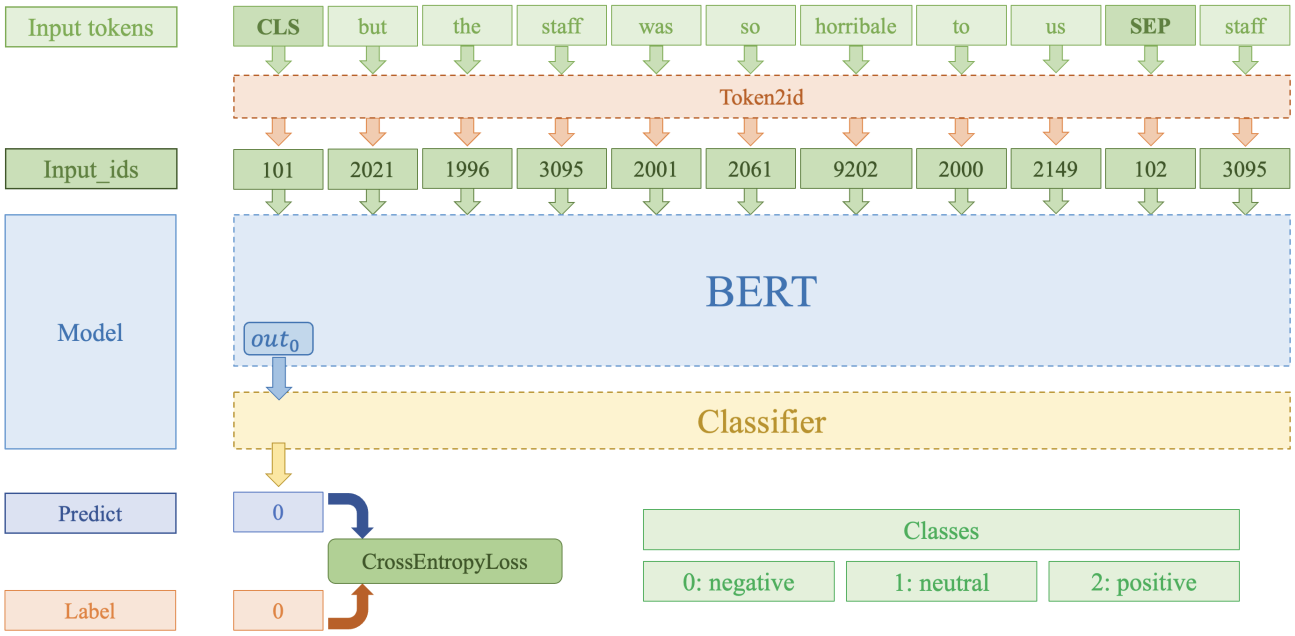
7. Prediction

Sau quá trình huấn luyện và mô hình có độ đo đánh giá tốt nhất được sử dụng để gán nhãn cho các câu đầu vào

```
from transformers import pipeline

token_classifier = pipeline(
    model="thainq107/abte-restaurants-distilbert-base-uncased",
    aggregation_strategy="simple"
)

test_sentence = 'The bread is top notch as well'
results = token_classifier(test_sentence)
results => bread
```

Phần 3. Aspect Term Sentiment Classification

Trong phần 1, chúng ta đã tình được từ là khía cạnh trong văn bản đầu vào, ở phần này, chúng ta xây dựng mô hình dựa vào câu đầu vào và khía cạnh được dự đoán xây dựng mô hình dự đoán cảm xúc.



Hình 4: Aspect term sentiment classification pipeline.

1. Build Dataset

```
# Install lib
!pip install -q datasets==3.2.0

from datasets import load_dataset

ds = load_dataset("thainq107/abte-restaurants")
```

2. Tokenization

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("distilbert/distilbert-base-uncased")

def tokenize_and_align_labels(examples):
    sentences, sentence_tags = [], []
    labels = []
    for tokens, pols in zip(examples['Tokens'], examples['Polarities']):
        tokens = tokens.replace("'", "").strip("][").split(', ')
        pols = pols.strip(']['').split(', ')

        bert_tokens = []
        bert_att = []
```

```
        pols_label = 0
        for i in range(len(tokens)):
            t = tokenizer.tokenize(tokens[i])
            bert_tokens += t
            if int(pols[i]) != -1:
                bert_att += t
                pols_label = int(pols[i])

        sentences.append(" ".join(bert_tokens))
        sentence_tags.append(" ".join(bert_att))
        labels.append(pols_label)

    tokenized_inputs = tokenizer(sentences, sentence_tags, padding=True, truncation=True, return_tensors="pt")
    tokenized_inputs['labels'] = labels
    return tokenized_inputs

preprocessed_ds = ds.map(tokenize_and_align_labels, batched=True)
```

3. Evaluate

```
# Install lib
!pip install -q evaluate==0.4.3

import evaluate
import numpy as np

accuracy = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    predictions = np.argmax(predictions, axis=1)
    return accuracy.compute(predictions=predictions, references=labels)
```

4. Model

```
from transformers import AutoModelForSequenceClassification

id2label = {0: 'Negative', 1: 'Neutral', 2: 'Positive'}
label2id = {'Negative': 0, 'Neutral': 1, 'Positive': 2}

model = AutoModelForSequenceClassification.from_pretrained(
    "distilbert/distilbert-base-uncased", num_labels=3, id2label=id2label, label2id=label2id
)
```

5. Training

```
import os
from transformers import TrainingArguments, Trainer

os.environ['WANDB_DISABLED'] = 'true'

training_args = TrainingArguments(
    output_dir="absa-restaurants-distilbert-base-uncased",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=5,
```

```
    weight_decay=0.01,
    eval_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=preprocessed_ds["train"],
    eval_dataset=preprocessed_ds["test"],
    processing_class=tokenizer,
    compute_metrics=compute_metrics,
)
trainer.train()
```

Kết quả training và accuracy trên tập test là 79.18

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	0.711144	0.733691
2	No log	0.651741	0.744415
3	0.694500	0.629071	0.760500
4	0.694500	0.598540	0.783735
5	0.425200	0.584476	0.791778

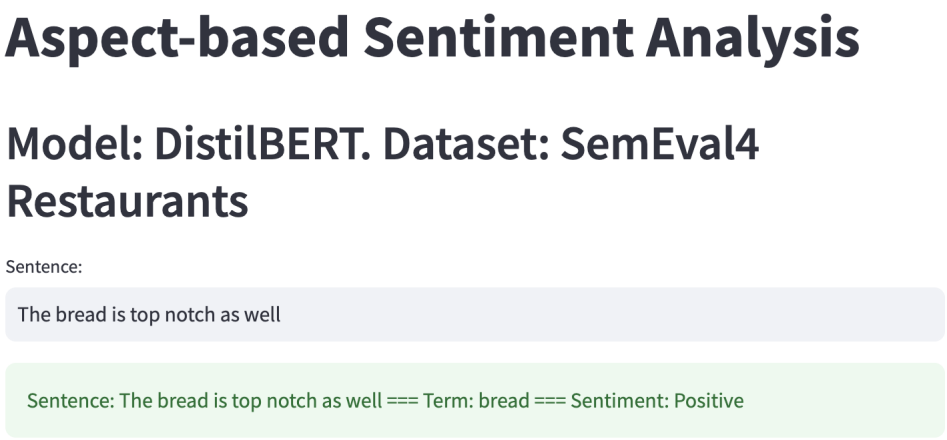
Hình 5: Quá trình huấn luyện ATSC.

6. Prediction

```
from transformers import pipeline

token_classifier = pipeline(
    model="thainq107/abte-restaurants-distilbert-base-uncased",
    aggregation_strategy="simple"
)

classifier = pipeline(
    model="thainq107/absa-restaurants-distilbert-base-uncased"
)

test_sentence = 'The bread is top notch as well'
results = token_classifier(test_sentence)
sentence_tags = " ".join([result['word'] for result in results])
pred_label = classifier(f'{test_sentence} [SEP] {sentence_tags}')
sentence_tags, pred_label
# ('bread', [{'label': 'Positive', 'score': 0.9864555597305298}])
```

7. Deployment

Triển khai ứng dụng sử dụng streamlit tham khảo mã nguồn tại đây, thử nghiệm ứng dụng tại đây. Giao diện và kết quả:



Hình 6: Triển khai ứng dụng.

Phần 4. Câu hỏi trắc nghiệm

Cho câu bình luận như sau: "The food is very fresh."

Câu hỏi 1 Dựa vào câu bình luận trên xác định giá trị dự đoán mô hình giải quyết bài toán ATE?
a) food
b) food, positive
c) food, very, fresh
d) food, very fresh, positive

Câu hỏi 2 Dựa vào câu bình luận trên xác định giá trị dự đoán mô hình giải quyết bài toán ASPE?
a) food
b) food, positive
c) food, very, fresh
d) food, very fresh, positive

Câu hỏi 3 Dựa vào câu bình luận trên xác định giá trị dự đoán mô hình giải quyết bài toán AOPE?
a) food
b) food, positive
c) food, very, fresh
d) food, very fresh, positive

Câu hỏi 4 Dựa vào câu bình luận trên xác định giá trị dự đoán mô hình giải quyết bài toán AOSTE?
a) food
b) food, positive
c) food, very, fresh
d) food, very fresh, positive

Câu hỏi 5 Mô hình giải quyết bài toán ATE ở phần 2 trên sử dụng bộ dữ liệu nào?
a) IMDB-Review
b) Penn Tree Bank
c) SemEval-2014 Task 4
d) SQuAD

Câu hỏi 6 Số lượng samples được sử dụng cho tập train trong phần thực nghiệm là
a) 3600
b) 3601
c) 3602
d) 3603

Câu hỏi 7 Mô hình giải quyết bài toán ATE ở phần 2 trên dựa vào bài toán tổng quát nào sau đây?
a) Sequence Labeling
b) Machine Translation

c) Topic Modeling
d) Sumarization

Câu hỏi 8 Mô hình tiền huấn luyện sử dụng cho bài toán ATE ở phần 2 trên là?
a) BERT
b) RoBERTA
c) DEBERTA
d) DistilBERT

Câu hỏi 9 Mô hình tiền huấn luyện sử dụng cho bài toán ASTC ở phần 3 trên là?
a) BERT
b) RoBERTA
c) DEBERTA
d) DistilBERT

Câu hỏi 10 Hàm mục tiêu để giải quyết bài toán ASTC ở phần 3 là
a) Masked Language Model
b) Next Sentence Prediction
c) Cả 2 đáp án trên đều đúng
d) Cả 2 đáp án trên đều sai

Phần 5. Phụ lục

1. **Hint**: Dựa vào file tải về Term Extraction và Sentiment Analysis để hoàn thiện các đoạn code.
2. **Solution**: Các file code cài đặt hoàn chỉnh và phần trả lời nội dung trắc nghiệm có thể được tải về tại đây (Lưu ý: Sáng thứ 3 khi hết deadline phần project, ad mới copy các nội dung bài giải nêu trên vào đường dẫn).
3. **Rubric**:

Phần	**Kiến Thức**	**Đánh Giá**
1	- Hiểu rõ bài toán Aspect-Based Sentiment Analysis - Sử dụng pretrained model cho các bài toán Aspect-Based Sentiment Analysis	- Các giá trị dự đoán khác nhau cho các bài toán Aspect-Based Sentiment Analysis
2.	- Hiểu rõ bài toán Aspect Term Extraction	- Xây dựng giải quyết bài toán bài toán Aspect Term Extraction
3.	- Hiểu rõ bài toán Aspect Term Sentiment Classification. Từ đó xây dựng mô hình giải quyết bài toán Aspect Sentiment Pair Extraction	- Xây dựng giải quyết bài toán bài toán Aspect Term Sentiment Classification

- Hết -