فاز اول: آماده سازی داده ها و تحلیل اکتشافی

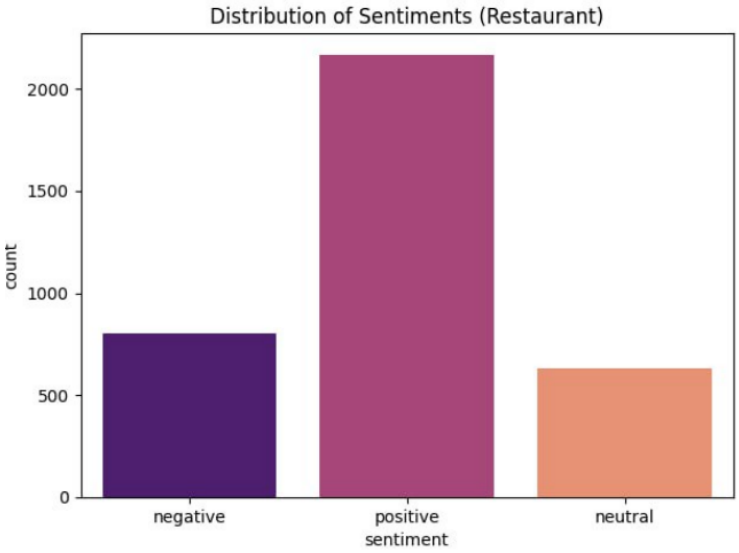
دریافت و استخراج داده:

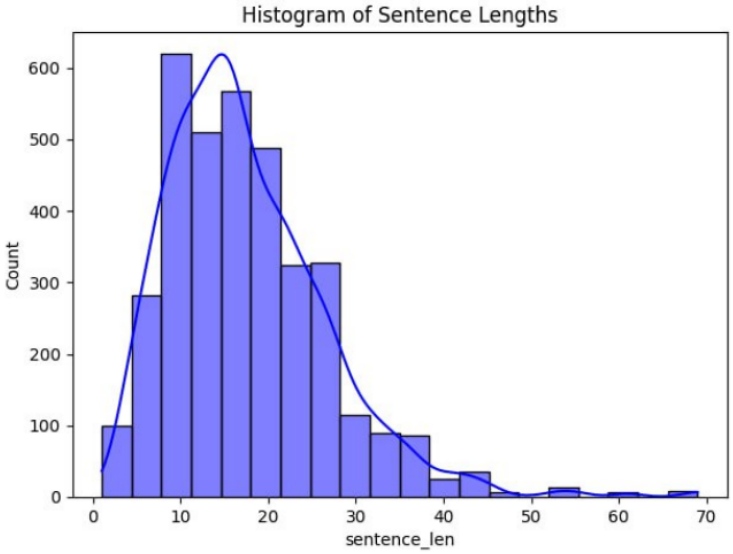
داده های استاندارد SemEval-2014 Task-4 (Restaurant) از فایل XML خوانده شده اند. از آنجا که یک جمله ممکن است چند Aspect Term داشته باشد، در مرحله استخراج، به ازای هر جنبه یک رکورد جدا ساخته شده است تا هر نمونه دقیقا یک زوج (text, aspect) و یک برچسب sentiment داشته باشد. همچنین برچسب های conflict حذف شده اند و فقط سه کلاس positive، negative و neutral نگه داشته می‌شوند. خروجی این مرحله یک فایل تمیز به نام restaurant_cleaned_data.csv است که ستون های اصلی آن شامل id، text، aspect و sentiment است.

تحلیل اکتشافی :(EDA)

برای تحلیل داده و شناخت بهتر توزیع آن، سه نمودار زیر ساخته و در گزارش قرار داده شده اند:

- **توزیع کلاس ها:** این نمودار نشان می‌دهد که داده ها از نظر برچسب ها متوازن نیستند و معمولا تعداد نمونه های مثبت بیشتر از منفی و خنثی است. این نکته در فاز آموزش مدل مهم است و به همین دلیل در آموزش از وزن دهی کلاسی در تابع هزینه استفاده شده است.
- **هیستوگرام طول جمله ها:** طول جمله ها بر حسب تعداد کلمات محاسبه شده است. توزیع طول جمله ها معمولا یک بخش اصلی در طول های متوسط دارد و یک دنباله بلند برای جمله های طولانی تر دیده می‌شود. این نمودار کمک می‌کند مقدار max_len را منطقی انتخاب کنیم (در این پروژه max_len=128 انتخاب شده تا حتی جمله های طولانی هم بدون مشکل پردازش شوند).
- **ابرکلمات جنبه ها:** با قرار دادن تمام aspect term ها کنار هم، پرتکرارترین جنبه ها مشخص می‌شوند. خروجی نشان می‌دهد جنبه هایی مثل food، service، place، staff، price و موارد مشابه، سهم زیادی از داده را تشکیل می‌دهند و در عمل، تمرکز مجموعه داده بیشتر روی همین موضوعات است.





شکل ۱: (از بالا به پایین) توزیع کلاس ها، هیستوگرام طول جمله ها، و ابرکلمات Aspect Term ها

فاز دوم: طراحی و پیاده سازی مدل

مرحله ۱: پیش پردازش داده

در این مرحله یک پیش پردازش ساده برای آماده سازی متن انجام شده است و خروجی در فایل restaurant_preprocessed_final.csv ذخیره می‌شود. روند کار به این صورت است که داده تمیز مرحله قبل (restaurant_cleaned_data.csv) خوانده می‌شود و یک ستون جدید به نام processed_text تولید می‌گردد. این ستون با یک تابع پاکسازی ساخته می‌شود که مراحل زیر را انجام می‌دهد:

- تبدیل متن به حروف کوچک
- حذف علائم نگارشی و کاراکترهای خاص با regex
- توکنایز کردن متن با NLTK
- حذف stopword های انگلیسی
- اتصال توکن های باقی مانده و تولید یک رشته تمیز

هدف این مرحله کاهش نویز و ساده سازی متن برای استفاده در مدل های مختلف است. (در مرحله آموزش RoBERTa، علاوه بر این مرحله، یک پاکسازی سبک داخل کد آموزش هم انجام شده است که شامل html unescape، حذف تگ های HTML و یکسان سازی فاصله ها است.)

مرحله ۲: آموزش مدل RoBERTa

در این مرحله، مدل roberta-base برای دسته بندی سه کلاسه (negative/neutral/positive) fine-tune شده است. روند آموزش به شکل خلاصه ولی دقیق به صورت زیر است:

خواندن داده و فیلتر برچسب ها:

فایل های train.csv و val.csv خوانده می‌شوند. سپس فقط نمونه هایی نگه داشته می‌شوند که برچسب آن ها یکی از سه حالت negative، neutral یا positive باشد و بقیه (مثل conflict) حذف می‌شوند. همچنین سطرهایی که text یا aspect یا sentiment خالی باشد حذف می‌شوند.

ساخت ورودی مدل و توکنایز:

ورودی مدل شامل جمله و کلمه جنبه است، بنابراین tokenizer به صورت ورودی دوتایی استفاده می‌شود:

(text, aspect)

توکنایز با max_len=128 و truncation="only_first" انجام می‌شود (یعنی اگر نیاز به کوتاه کردن باشد، جمله کوتاه می‌شود). در RoBERTa از token_type_ids استفاده نمی‌شود. در پیاده سازی، توکنایز در همان ابتدای ساخت Dataset انجام شده است تا در طول آموزش، فشار محاسباتی روی __getitem__ کمتر باشد.

ساخت batch و padding به صورت dynamic:

در DataLoader از یک collate_fn استفاده شده است که با tokenizer.pad، padding را به صورت پویا انجام می‌دهد. یعنی طول دنباله ها در هر batch برابر با بیشینه طول همان batch می‌شود و فقط input_ids و attention_mask ساخته می‌شود. همچنین برای تکرارپذیری، seed تنظیم شده و برای worker های DataLoader هم مقداردهی جدا انجام شده است.

مدل و تابع هزینه:

مدل RobertaForSequenceClassification با num_labels=3 استفاده شده است. برای کاهش اثر نامتوازن بودن کلاس ها، وزن های کلاسی از روی فراوانی برچسب های آموزش محاسبه شده و در CrossEntropyLoss استفاده می‌شود (وزن کلاس ها به صورت معکوس تعداد نمونه های همان کلاس است).

بهینه سازی و زمان بندی نرخ یادگیری:

در آموزش اولیه، از AdamW با weight_decay=0.01 استفاده شده است. پارامترهای bias و LayerNorm.weight از weight decay معاف هستند. نرخ یادگیری lr=2e-5 است و زمان بندی آن با linear warmup و سپس linear decay انجام می‌شود. تعداد کل گام ها برابر epochs * len(train_loader) است و نسبت warmup برابر warmup_ratio=0.1 در نظر گرفته شده است.

پایداری و سرعت آموزش:

آموزش با mixed precision انجام شده است (torch.amp.autocast و GradScaler). همچنین gradient clipping با مقدار grad_clip=1.0 اعمال می‌شود تا از انفجار گرادیان جلوگیری شود.

ارزیابی، انتخاب مدل و خروجی ها:

در هر epoch، معیارهای loss، accuracy و macro-F1 برای train و validation محاسبه می‌شوند. معیار اصلی برای انتخاب بهترین

مدل val macro-F1 است. اگر بهبود رخ دهد وزن های مدل در best_model.pt ذخیره می‌شود. همچنین early stopping با patience=3 فعال است و اگر بهبود متوالی رخ ندهد آموزش متوقف می‌شود. در پایان، بهترین مدل بارگذاری می‌شود و روی validation، classification report و confusion matrix چاپ می‌گردد. تاریخچه آموزش در train_history.csv ذخیره می‌شود و نمودارهای loss.png، acc.png و f1.png برای بررسی روند آموزش ساخته و در پوشه خروجی ذخیره می‌شوند.

تنظیمات اولیه اجرای آموزش (مقادیر پیش فرض کد):

- model_name=roberta-base
- max_len=128
- batch_size=16
- epochs=5
- lr=2e-5
- weight_decay=0.01
- warmup_ratio=0.1
- grad_clip=1.0
- patience=3
- num_workers=4
- seed=42
- output_dir=absa_output

هایپرپارامترها و نقش آن ها در آموزش

در fine-tuning مدل های زبانی مثل RoBERTa چند هایپرپارامتر اصلی داریم که مستقیما روی سرعت همگرایی، پایداری آموزش و میزان overfitting اثر می‌گذارند. از آنجا که مدل با چند تنظیم مختلف آموزش داده شده است، در این بخش فقط نقش هر هایپرپارامتر توضیح داده می‌شود و مقادیر دقیق و نتایج آن ها (برای هر تنظیم) گزارش می‌شود.

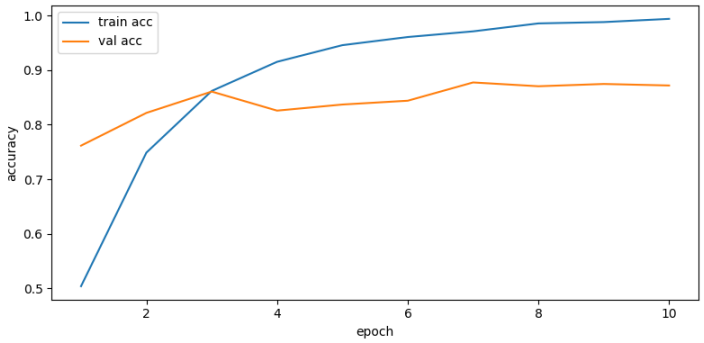
- **Learning Rate (lr):** نرخ یادگیری مشخص می‌کند وزن ها در هر گام چه مقدار تغییر کنند. مقدار بزرگ تر معمولا باعث یادگیری سریع تر می‌شود ولی می‌تواند نوسان ایجاد کند یا باعث شود مدل از بهینه عبور کند. مقدار کوچک تر آموزش را پایدارتر می‌کند اما ممکن است همگرایی را کند کند یا مدل در یک نقطه نه چندان خوب متوقف شود. در fine-tuning معمولا نرخ یادگیری حساس ترین هایپرپارامتر است.
- **Weight Decay:** نوعی منظم سازی است که از بزرگ شدن بیش از حد وزن ها جلوگیری می‌کند و معمولا به کاهش overfitting کمک می‌کند. ایده اصلی این است که مدل از راه حل های خیلی پیچیده که فقط داده های آموزش را حفظ می‌کنند فاصله بگیرد. در مدل های transformer معمولا bias و LayerNorm.weight را از weight decay مستقل می‌کنند چون اعمال آن روی این پارامترها می‌تواند اثر منفی روی پایداری آموزش داشته باشد.
- **Warmup Ratio:** در ابتدای آموزش اگر نرخ یادگیری از همان ابتدا بزرگ باشد، مخصوصا در مدل های از پیش آموزش دیده، ممکن است گرادیان ها ناپایدار شوند. warmup باعث می‌شود نرخ یادگیری در چند درصد اول گام ها به صورت خطی از مقدار کوچک به مقدار اصلی برسد و بعد وارد روند اصلی زمان بندی شود (مثلا کاهش خطی). warmup ratio تعیین می‌کند چه سهمی از کل گام ها صرف این مرحله شود.
- **Gradient Clipping (grad_clip):** برای جلوگیری از انفجار گرادیان، نرم گرادیان ها محدود می‌شود. این کار باعث می‌شود گام های بروزرسانی خیلی بزرگ نشوند و آموزش پایدارتر شود. معمولا در fine-tuning و مخصوصا هنگام استفاده از mixed precision این تنظیم کمک می‌کند که آموزش از کنترل خارج نشود.
- **Patience:** این پارامتر مربوط به early stopping است و مشخص می‌کند اگر معیار انتخاب مدل روی validation (مثلا macro-F1) برای چند epoch پشت سر هم بهتر نشود، آموزش متوقف شود. هدف این است که آموزش بی دلیل ادامه پیدا نکند و مدل وارد overfitting نشود. مقدار کمتر باعث توقف سریع تر و مقدار بیشتر باعث فرصت بیشتر برای بهبود می‌شود.

تنظیمات هایپرپارامترها (1 Run)
مقادیر هایپرپارامترها:

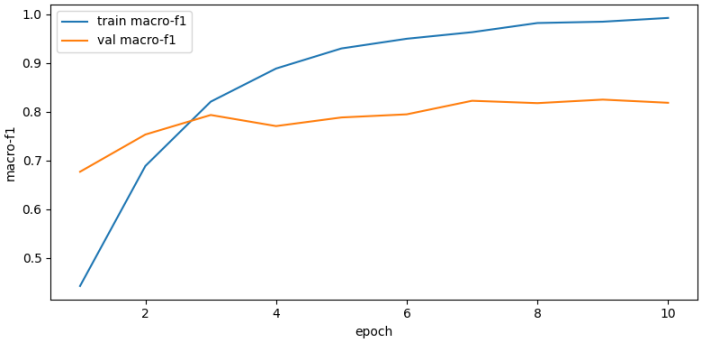
- model_name: roberta-base
- max_len: 128
- batch_size: 16
- epochs: 10
- lr: 2e-5
- weight_decay: 0.005
- warmup_ratio: 1.0
- grad_clip: 0.1
- patience: 3
- num_workers: 2
- seed: 42

نتیجه نهایی روی validation
در پایان آموزش، بهترین مدل روی validation ارزیابی شده و مقادیر زیر گزارش شده است:

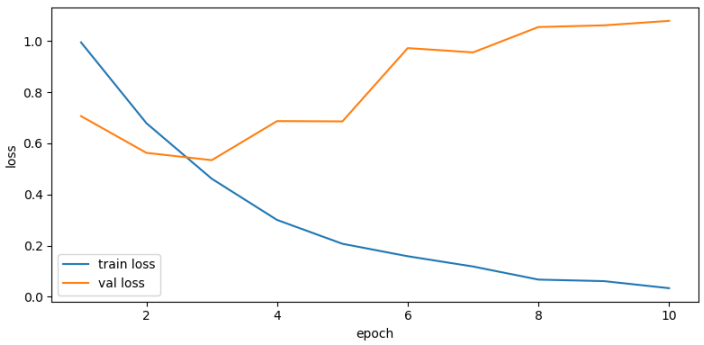
$$\text{Loss Val} = 1.0614, \qquad \text{Acc Val} = 0.8745, \qquad \text{Macro-F1 Val} = 0.8248.$$



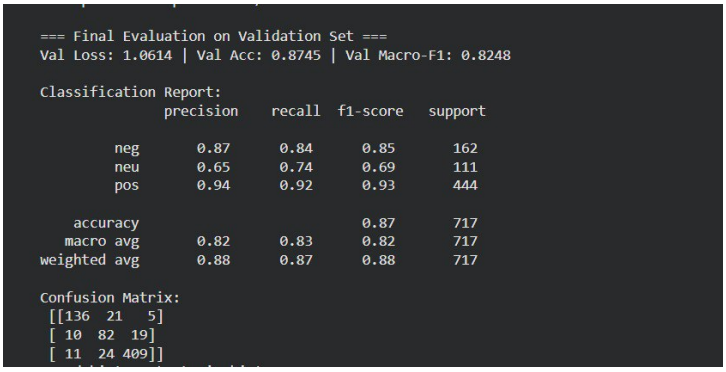
(الف) نمودار accuracy برای train و validation



(ب) نمودار macro-F1 برای train و validation



(پ) نمودار loss برای train و validation



```
=== Final Evaluation on Validation Set ===
Val Loss: 1.0614 | Val Acc: 0.8745 | Val Macro-F1: 0.8248

Classification Report:
              precision    recall  f1-score   support

         neg       0.87      0.84      0.85       162
         neu       0.65      0.74      0.69       111
         pos       0.94      0.92      0.93       444

    accuracy                           0.87       717
   macro avg       0.82      0.83      0.82       717
weighted avg       0.88      0.87      0.88       717

Confusion Matrix:
 [[136  21   5]
 [ 10  82  19]
 [ 11  24 409]]
```

(ت) classification report و confusion matrix

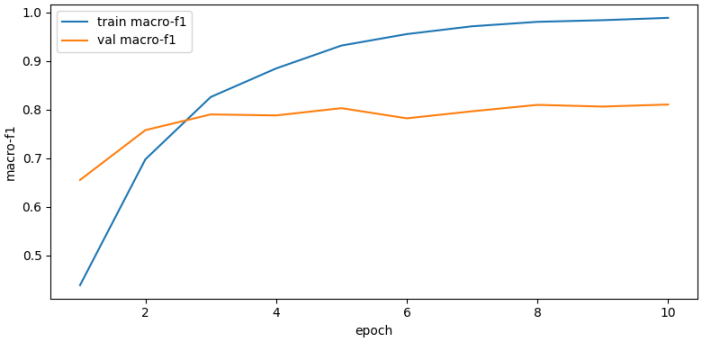
شکل ۲: خروجی های آموزشی و ارزیابی برای تنظیم هایپرپارامتر 1 Run

تنظیمات هایپرپارامترها (2 Run)
مقادیر هایپرپارامترها:

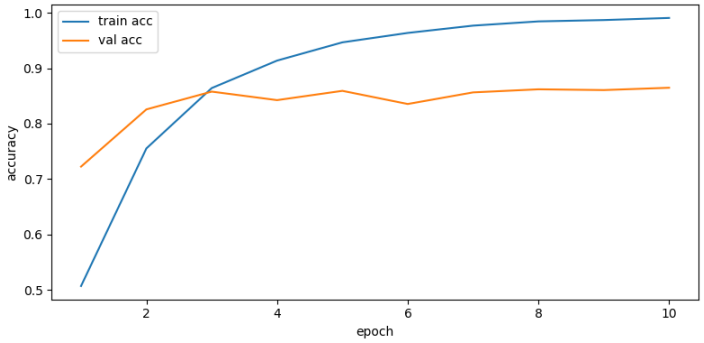
- max_len: 168
- batch_size: 16
- epochs: 10
- lr: 2e-5
- weight_decay: 02.0
- warmup_ratio: 1.0
- grad_clip: 0.1
- patience: 3

نتیجه نهایی روی validation
در پایان آموزش، بهترین checkpoint همین Run روی validation ارزیابی شده و مقادیر زیر گزارش شده است:

$$\text{Loss Val} = 1.1942, \qquad \text{Acc Val} = 0.8647, \qquad \text{Macro-F1 Val} = 0.8103.$$



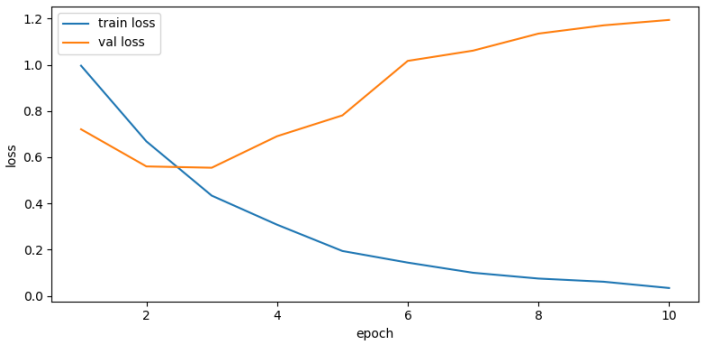
(ب) نمودار macro-F1 برای train و validation



(الف) نمودار accuracy برای train و validation

```
=== Final Evaluation on Validation Set ===
Val Loss: 1.1942 | Val Acc: 0.8647 | Val Macro-F1: 0.8103

Classification Report:
              precision    recall  f1-score   support

         neg       0.82      0.85      0.83       162
         neu       0.65      0.69      0.67       111
         pos       0.94      0.91      0.93       444

    accuracy                           0.86       717
   macro avg       0.80      0.82      0.81       717
weighted avg       0.87      0.86      0.87       717

Confusion Matrix:
 [[137  18   7]
 [ 16  77  18]
 [ 15  23 406]]
Saved history to train_history.csv
Plots saved to absa_output
```

(ت) classification report و confusion matrix



(پ) نمودار loss برای train و validation

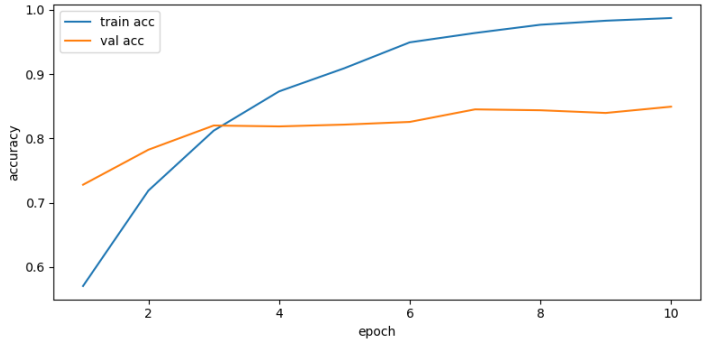
شکل ۳: خروجی های آموزشی و ارزیابی برای تنظیم هایپرپارامتر 2 Run

تنظیمات هایپرپارامترها (3 Run)
مقادیر هایپرپارامترها:

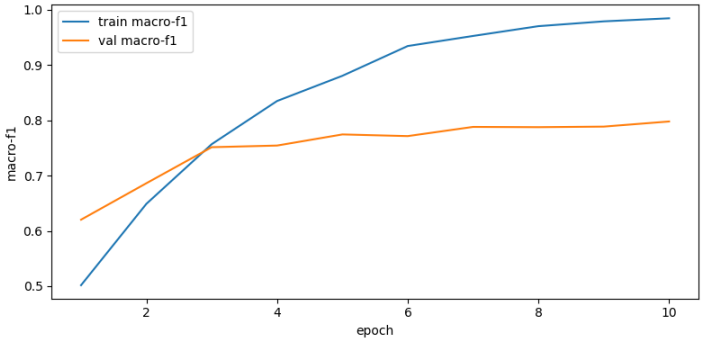
- max_len: 128
- batch_size: 8
- epochs: 10
- lr: 5e-5
- weight_decay: 01.0
- warmup_ratio: 1.0
- grad_clip: 0.1
- patience: 3

نتیجه نهایی روی validation
در پایان آموزش، بهترین checkpoint همین Run روی validation ارزیابی شده و مقادیر زیر گزارش شده است:

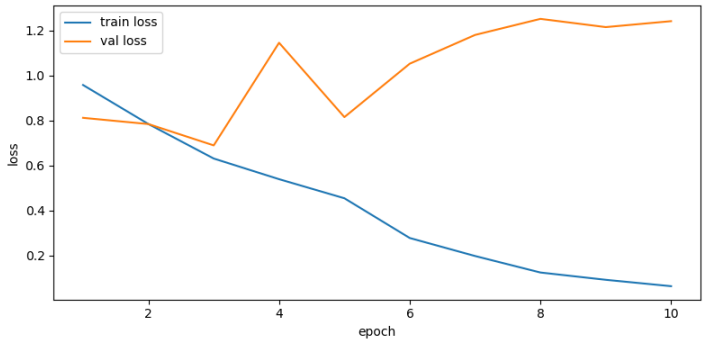
$$\text{Loss Val} = 1.2422, \qquad \text{Acc Val} = 0.8494, \qquad \text{Macro-F1 Val} = 0.7978.$$



(الف) نمودار accuracy برای train و validation



(ب) نمودار macro-F1 برای train و validation



(پ) نمودار loss برای train و validation

(ت) classification report و confusion matrix

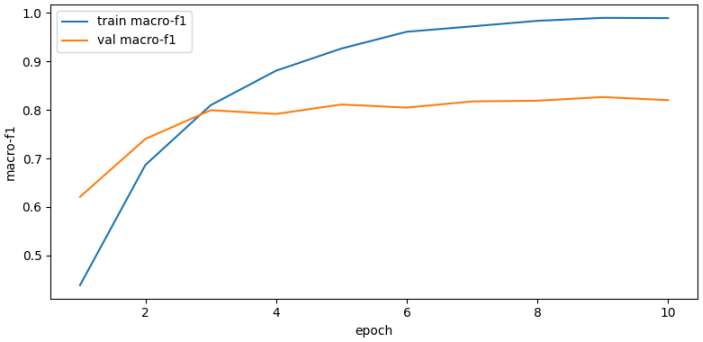
شکل ۴: خروجی های آموزشی و ارزیابی برای تنظیم هایپرپارامتر 3 Run

تنظیمات هایپرپارامترها (Run 4)
مقادیر هایپرپارامترها:

- max_len: 128
- batch_size: 16
- epochs: 10
- lr: 2e-5
- weight_decay: 001.0
- warmup_ratio: 1.0
- grad_clip: 0.1
- patience: 3

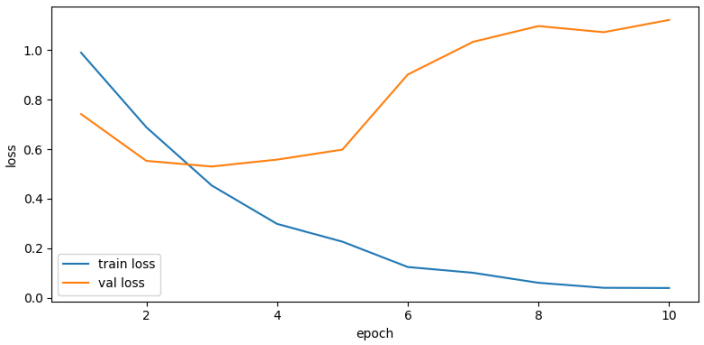
نتیجه نهایی روی validation
در پایان آموزش، بهترین checkpoint همین Run روی validation ارزیابی شده و مقادیر زیر گزارش شده است:

$$\text{Loss Val} = 1.0727, \qquad \text{Acc Val} = 0.8759, \qquad \text{Macro-F1 Val} = 0.8266.$$



(الف) نمودار accuracy برای train و validation

(ب) نمودار macro-F1 برای train و validation



(پ) نمودار loss برای train و validation

```
=== Final Evaluation on Validation Set ===
Val Loss: 1.0727 | Val Acc: 0.8759 | Val Macro-F1: 0.8266

Classification Report:
              precision    recall  f1-score   support

         neg       0.86      0.88      0.87       162
         neu       0.66      0.70      0.68       111
         pos       0.94      0.92      0.93       444

    accuracy                           0.88       717
   macro avg       0.82      0.83      0.83       717
weighted avg       0.88      0.88      0.88       717

Confusion Matrix:
 [[143  13   6]
 [ 14  78  19]
 [ 10  27 407]]
```

(ت) classification report و confusion matrix

شکل ۵: خروجی های آموزشی و ارزیابی برای تنظیم هایپرپارامتر Run 4

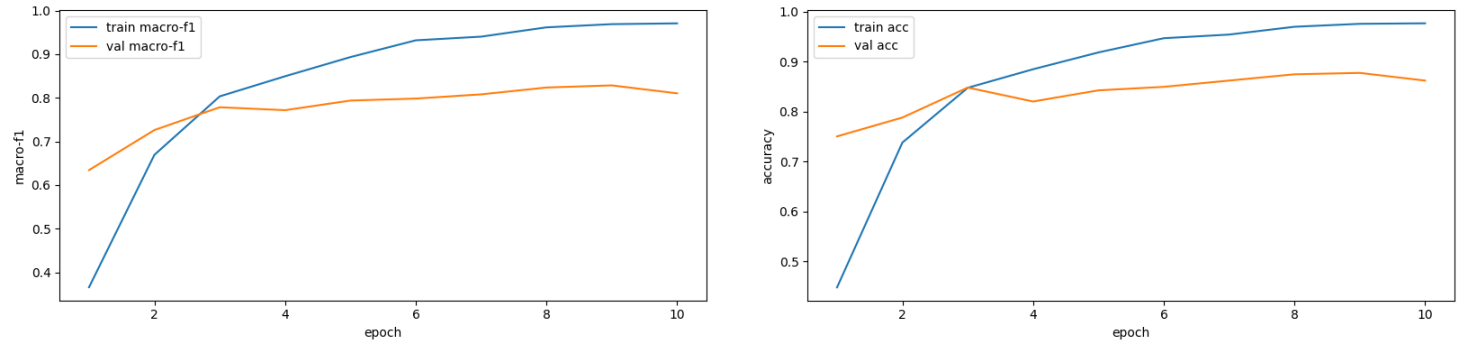
تنظیمات هایپرپارامترها (5 Run)
مقادیر هایپرپارامترها:

- max_len: 128
- batch_size: 16
- epochs: 10
- lr: 1e-5
- weight_decay: 0003.0
- warmup_ratio: 1.0
- grad_clip: 0.1
- patience: 3

نتیجه نهایی روی validation
در پایان آموزش، بهترین checkpoint همین Run روی validation ارزیابی شده و مقادیر زیر گزارش شده است:

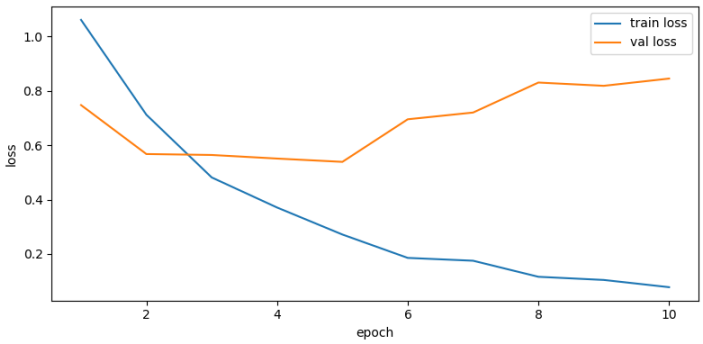
$$\text{Loss Val} = 0.8183, \qquad \text{Acc Val} = 0.8773, \qquad \text{Macro-F1 Val} = 0.8285.$$

این Run در بین 5 تنظیم اجرا شده، بهترین نتیجه را (بر اساس Val Macro-F1) داشته است.



(الف) نمودار accuracy برای train و validation

(ب) نمودار macro-F1 برای train و validation



(پ) نمودار loss برای train و validation

```
=== Final Evaluation on Validation Set ===
Val Loss: 0.8183 | Val Acc: 0.8773 | Val Macro-F1: 0.8285

Classification Report:
              precision    recall  f1-score   support

         neg       0.85      0.88      0.86       162
         neu       0.67      0.71      0.69       111
         pos       0.95      0.92      0.93       444

    accuracy                           0.88       717
   macro avg       0.82      0.84      0.83       717
weighted avg       0.88      0.88      0.88       717

Confusion Matrix:
 [[143  14   5]
 [ 14  79  18]
 [ 12  25 407]]
```

(ت) classification report و confusion matrix

شکل ۶: خروجی های آموزشی و ارزیابی برای تنظیم هایپرپارامتر 5 Run (بهترین نتیجه)

فاز سوم: ارزیابی و تحلیل خطا

در این فاز، مدل نهایی (بهترین تنظیم هایپرپارامتر) روی داده های ارزیابی اجرا شده و معیارهای Precision، Recall، F1-Score و Macro-F1 گزارش شده اند. همچنین برای تحلیل رفتار مدل، ماتریس آشفتگی رسم شده و 5 نمونه از خطاهای مدل بررسی شده است.

نتایج کلی (بر اساس ماتریس آشفتگی)

با استفاده از ماتریس آشفتگی زیر، دقت کلی مدل برابر با

$$\text{Accuracy} = 0.8705$$

و Macro-F1 برابر با

$$\text{Macro-F1} = 0.8002$$

به دست آمده است. معیارهای هر کلاس نیز به صورت زیر است:

کلاس	Precision	Recall	F1
negative	8283.0	8367.0	8325.0
neutral	7589.0	5459.0	6350.0
positive	9014.0	9670.0	9331.0

طبق جدول بالا، عملکرد مدل روی کلاس positive بهتر از بقیه است و بیشترین ضعف مدل مربوط به کلاس neutral است (به خصوص Recall پایین تر).

ماتریس آشفتگی و نمونه خطاها

تحلیل خطا (5 نمونه)

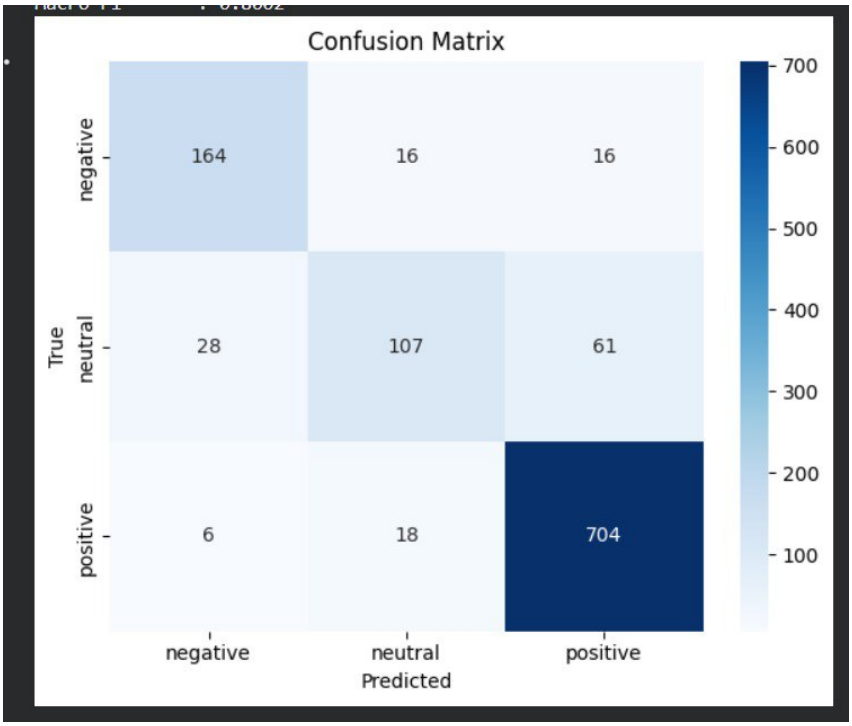
در ادامه، 5 نمونه خطا که در خروجی مدل گزارش شده اند آورده شده و دلیل احتمالی خطا به صورت کوتاه بیان شده است:

- **نمونه 1:**
  Text: The portions of the food that came out were mediocre.
  Aspect: portions of the food True: neutral Pred: negative
  واژه هایی مثل mediocre بار منفی دارند و مدل به طور طبیعی آن را منفی برداشت می‌کند، در حالی که برچسب داده به صورت خنثی ثبت شده است. این نوع ابهام برچسب گذاری معمولا باعث کاهش Recall کلاس neutral می‌شود.

- **نمونه 2:**
  Text: How pretentious and inappropriate for MJ Grill to claim that it provides power lunch and dinners!
  Aspect: lunch True: negative Pred: positive
  ساختار جمله حالت کنایه دارد و ممکن است مدل به علت وجود عبارت های عمومی مثل power lunch یا الگوی جمله، برداشت اشتباه مثبت داشته باشد.

- **نمونه 3:**
  Text: How pretentious and inappropriate for MJ Grill to claim that it provides power lunch and dinners!
  Aspect: dinners True: negative Pred: neutral
  در این نمونه، مدل منفی بودن کلی جمله را به صورت کامل روی جنبه dinners منتقل نکرده و آن را خنثی دیده است. این نوع خطا معمولا وقتی رخ می‌دهد که جمله بیشتر حالت کلی داشته باشد و اتصال مستقیم به جنبه ضعیف باشد.

- **نمونه 4:**
  Text: Entrees include classics like lasagna, fettuccine Alfredo and chicken parmigiana.
  Aspect: Entrees True: neutral Pred: positive
  ذکر اسم غذاها و ساختار معرفی منو می‌تواند توسط مدل به عنوان نشانه مثبت برداشت شود، در حالی که جمله واقعا فقط توصیفی است و احساس مشخصی ندارد.




```
=== 5 Misclassified Examples ===

Text   : The portions of the food that came out were mediocre.
Aspect : portions of the food
True   : neutral
Pred   : negative
------------------------------------------------------------
Text   : How pretentious and inappropriate for MJ Grill to claim that it provides power lunch and dinners!
Aspect : lunch
True   : negative
Pred   : positive
------------------------------------------------------------
Text   : How pretentious and inappropriate for MJ Grill to claim that it provides power lunch and dinners!
Aspect : dinners
True   : negative
Pred   : neutral
------------------------------------------------------------
Text   : Entrees include classics like lasagna, fettuccine Alfredo and chicken parmigiana.
Aspect : Entrees
True   : neutral
Pred   : positive
------------------------------------------------------------
Text   : Entrees include classics like lasagna, fettuccine Alfredo and chicken parmigiana.
Aspect : lasagna
True   : neutral
Pred   : positive
------------------------------------------------------------
```

شکل ۷: (بالا) ماتریس آشفتگی مدل نهایی (پایین) 5 نمونه از خطاهای مدل

- **نمونه 5:**
  Text: Entrees include classics like lasagna, fettuccine Alfredo and chicken parmigiana.
  Aspect: lasagna True: neutral Pred: positive
  مشابه نمونه قبل، مدل روی نام غذا حساس است و آن را به سمت مثبت می‌برد، در حالی که این جمله صرفا معرفی منو است.

فاز چهارم: تست مدل و بررسی استحکام (Robustness)

1) تست جمله های چالش برانگیز (20 جمله)
در این بخش، خروجی مدل برای 20 جمله چالش برانگیز به صورت اسکرین شات های زیر در گزارش قرار داده شده است.

```
Enter your sentence (or type 'exit'):
> the food was not fatty
Enter aspect:
> food

Prediction: positive
Probabilities:
Negative: 0.0026
Neutral : 0.1692
Positive: 0.8282
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> If the service is always this slow, I won't return
Enter aspect:
> service

Prediction: negative
Probabilities:
Negative: 0.9983
Neutral : 0.0005
Positive: 0.0012
```

```
> The burger would be better if it was not oily.
Enter aspect:
> burger

Prediction: negative
Probabilities:
Negative: 0.9982
Neutral : 0.0007
Positive: 0.0011
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> If I were you ; absoultely choice pizza
Enter aspect:
> pizza

Prediction: positive
Probabilities:
Negative: 0.0008
Neutral : 0.0006
Positive: 0.9986
----------------------------------------
```

```
> If they don't have fish, I will eat chicken.
Enter aspect:
> fish

Prediction: negative
Probabilities:
Negative: 0.9938
Neutral : 0.0055
Positive: 0.0007
```

```
Enter your sentence (or type 'exit'):
> The restuarant was not bad at all but not best choice for next time.
Enter aspect:
> restuarant

Prediction: negative
Probabilities:
Negative: 0.9760
Neutral : 0.0015
Positive: 0.0225
```

```
Enter your sentence (or type 'exit'):
> this restuarant has also same quality with other restuarant
Enter aspect:
> restaurants

Prediction: positive
Probabilities:
Negative: 0.0007
Neutral : 0.0005
Positive: 0.9987
```

```
----------------------------------------
Enter your sentence (or type 'exit'):
> The coffee is bitter unless you add sugar.
Enter aspect:
> coffee

Prediction: negative
Probabilities:
Negative: 0.9981
Neutral : 0.0007
Positive: 0.0012
----------------------------------------
```

```
> If you arrive late, the kitchen is closed.
Enter aspect:
> kitchen

Prediction: negative
Probabilities:
Negative: 0.9978
Neutral : 0.0008
Positive: 0.0014
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> rice wasnt salty at all
Enter aspect:
> rice

Prediction: positive
Probabilities:
Negative: 0.0107
Neutral : 0.0012
Positive: 0.9881
```

```
Enter your sentence (or type 'exit'):
> If you love spicy food, this curry is for you.
Enter aspect:
> curry

Prediction: positive
Probabilities:
Negative: 0.0006
Neutral : 0.0007
Positive: 0.9987
```

```
Enter your sentence (or type 'exit'):
> Even if it is cheap, the quality should be better.
Enter aspect:
> quality

Prediction: negative
Probabilities:
Negative: 0.9977
Neutral : 0.0007
Positive: 0.0015
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> Unless you book a table, you will have to wait
Enter aspect:
> wait

Prediction: negative
Probabilities:
Negative: 0.9980
Neutral : 0.0010
Positive: 0.0009
```

```
> I would only come back if this were the last restaurant on Earth
Enter aspect:
> restaurant

Prediction: positive
Probabilities:
Negative: 0.0450
Neutral : 0.0366
Positive: 0.9183
```

```
Enter your sentence (or type 'exit'):
> If you haven't tried their rice yet, you haven't lived
Enter aspect:
> rice

Prediction: positive
Probabilities:
Negative: 0.0128
Neutral : 0.0299
Positive: 0.9573
```

```
Enter your sentence (or type 'exit'):
> The fish is fresh if that is what you want.
Enter aspect:
> fish

Prediction: positive
Probabilities:
Negative: 0.0005
Neutral : 0.0012
Positive: 0.9983
```

```
Enter your sentence (or type 'exit'):
> Don't eat the salad if you have allergies.
Enter aspect:
> salad

Prediction: negative
Probabilities:
Negative: 0.9982
Neutral : 0.0006
Positive: 0.0012
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> If you sit by the window, you can see the street
Enter aspect:
> window

Prediction: positive
Probabilities:
Negative: 0.0088
Neutral : 0.0418
Positive: 0.9494
----------------------------------------
```

```
Enter your sentence (or type 'exit'):
> If I wanted frozen food, I would have stayed at home
Enter aspect:
> frozen

Prediction: negative
Probabilities:
Negative: 0.9863
Neutral : 0.0128
Positive: 0.0009
```

```
Enter your sentence (or type 'exit'):
> That mixed pizza would be tha last choice for me
Enter aspect:
> pizza

Prediction: negative
Probabilities:
Negative: 0.9509
Neutral : 0.0474
Positive: 0.0017
```

حمله تخاصمی ساده

در این بخش یک جمله ای انتخاب شد که مدل آن را درست دسته بندی می‌کند، سپس با یک تغییر جزئی (بدون تغییر معنی اصلی از نظر انسان) تلاش شد مدل به اشتباه بیفتد. در مثال زیر، اضافه شدن کلمه like باعث تغییر برچسب پیش بینی شده شده است.

در جمله اول، مدل برچسب neutral را پیش بینی کرده است، اما با اضافه کردن like در جمله دوم، خروجی مدل به positive تغییر کرده است. دلیل احتمالی این خطا این است که مدل در برخی موارد وجود کلمه like را به عنوان سیگنال مثبت در نظر می‌گیرد، در حالی که در این جمله like به معنی «مشابه» است و بار احساسی مثبت ندارد.

```
----------------------------------------
Enter your sentence (or type 'exit'):
> this resturant was casual
Enter aspect:
> restaurants

Prediction: neutral
Probabilities:
Negative: 0.0009
Neutral : 0.9521
Positive: 0.0470
----------------------------------------
Enter your sentence (or type 'exit'):
> this restaurants was casual like others
Enter aspect:
> restaurants

Prediction: positive
Probabilities:
Negative: 0.0006
Neutral : 0.0017
Positive: 0.9978
----------------------------------------
Enter your sentence (or type 'exit'):
>
```

شکل ۸: نمونه حمله تخاصمی ساده با اضافه شدن کلمه like