

به نام خدا
درس مبانی یادگیری عمیق
گزارش پروژه پایانی

استاد درس : دکتر مرضیه داوودآبادی
دستیاران : مرتضی حاجی آبادی، سحر سرکار، فائزه
صادقی، مهسا موفق بهروزی، الناز رضایی، پریسا ظفری،
حسن حماد، سید محمد موسوی، کمیل فتحی، شایان
موسوی نیا، امیررضا ویشته

دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر
نیمسال اول تحصیلی ۱۴۰۲ - ۱۴۰۳



موضوع:

تحلیل احساسات در متن فارسی

ردیف	نام و نام خانوادگی	شماره دانشجویی
۱	ریحانه شاهرخیان	99521361
۲	آیسا میاهی نیا	99522149

جدول ۱: مشخصات اعضای گروه

۱ شرح موضوع و مجموعه دادگان

بیان و تشخیص احساسات نقش مهمی در زندگی اجتماعی انسان‌ها دارد. که آن ارتباط نزدیکی با توانایی‌های شناختی و مهارت‌های ارتباطی افراد دارد. در نتیجه، هوش هیجانی یکی از توانایی‌های ضروری برای حرکت از هوش مصنوعی محدود به هوش مصنوعی عمومی است. تشخیص احساسات (ED) یک حوزه تحقیقاتی فعال برای توانمندسازی ماشین‌ها برای درک مؤثر احساسات انسان است. مجموعه دادگانی که برای این پروژه انتخاب شده است، مجموعه دادگان ArmanEmo است. که این دیتاست شامل 7 کلاس به صورت مقابل است: "ANGRY", "FEAR", "HAPPY", "HATE", "SAD", "SURPRISE", "OTHER". این دیتاست برای تشخیص احساسات متن فارسی می‌باشد. در این پروژه هدف تشخیص احساسات در متون فارسی با استفاده از دیتاست ArmanEmo می‌باشد.

۲ پیش‌پردازش داده‌ها

طبق مقاله ARMANEMO: A PERSIAN DATASET FOR TEXT-BASED EMOTION DETECTION، ما از Parsivar، ابزاری برای پیش‌پردازش متون فارسی، استفاده کردیم. یک تابع به اسم cleaning تعریف کردیم که در آن مواردی که در مقاله هم پیشنهاد شده بود برای پیش‌پردازش و نرمالایز کردن پیاده‌سازی کردیم. به توضیح مراحل این تابع می‌پردازیم. در این تابع از کتابخانه cleantext و متود clean آن نیز استفاده کردیم. ابتدا Parsivar Normalizer و Parsivar Tokenizer initialize کردیم، سپس یک متود به اسم CleanHtml تعریف کردیم که با استفاده از regular expression، تگ‌های html موجود در متن را حذف می‌کند. تابع اصلی ما Cleaning می‌باشد که یک ورودی متن دریافت می‌کند، مراحل زیر را انجام می‌دهد:

- فضاهای خالی پیشرو و انتهایی را از متن حذف می‌کند.
- برای انجام cleaning کلی متن از تابع پاک از کتابخانه cleantext استفاده می‌کند.
- برای حذف تگ‌های HTML تابع cleanhtml را فراخوانی می‌کند.
- با استفاده از Parsiver Normalizer متن را نرمالایز می‌کند.
- الگوهای خاص یونیکد را با استفاده از یک Regular Expression حذف می‌کند.
- هشتگ‌ها را حذف می‌کند و چندین فضای خالی را با یک فاصله جایگزین می‌کند.
- سپس متن پاک شده برگردانده می‌شود.

طبق مطالعاتی و جستجوهای به‌عمل آمده و با توجه به مدل انتخابی‌مان، به این نتیجه رسیدیم که لیبل‌های دیتاست باید به صورت one-hot باشد، پس لیبل‌ها را به صورت لیستی از صفر و یک‌ها (one-hot) تبدیل کردیم. کلاس MultiLabelDataset، برای تسهیل آموزش یک مدل شبکه عصبی برای کار طبقه‌بندی متن multi label پیاده‌سازی شده است. متن و لیبل‌های مربوطه را می‌گیرد، متن را با استفاده از یک tokenizer مشخص (BertTokenizer) tokenize می‌کند و اطمینان می‌دهد که داده‌های ورودی به‌طور مناسب برای آموزش آماده می‌باشند. نمونه‌های برگشتی از این مجموعه داده را می‌توان به راحتی در یک PyTorch DataLoader برای بارگذاری به صورت کارآمد در طول فرآیند آموزش استفاده کرد.

۳ انتخاب مدل

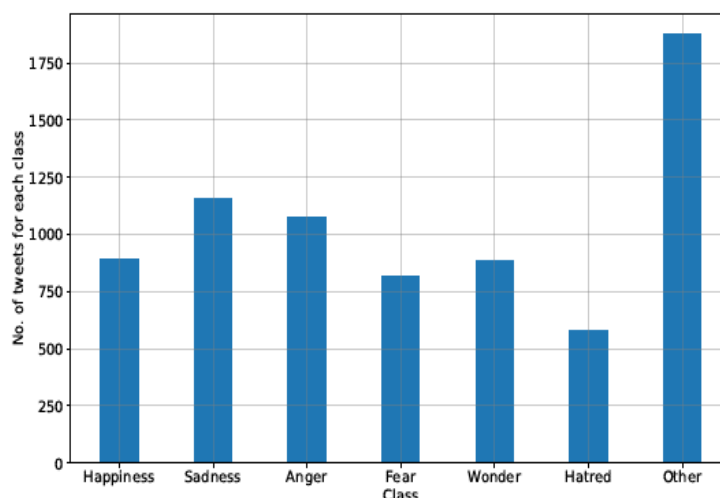
در این بخش، مدل‌های پایه برای طبقه‌بندی احساسات بر روی ArmanEmo ارائه می‌کنیم که از یادگیری انتقالی استفاده می‌کنند. از آنجایی که انجام این تسک بسیار هزینه‌بر است یادگیری انتقالی می‌تواند کمک حال ما باشد. یکی از این مدل‌های پایه‌ای، مدل ParsBert می‌باشد. حال به نکاتی در رابطه با مدل ParsBert می‌پردازیم.

- مدل تک زبانه: ParsBERT به طور خاص برای زبان فارسی طراحی شده است و آن را به یک مدل تک زبانه تبدیل می‌کند. یعنی روی متن فارسی آموزش داده شده و برای درک و تولید محتوا به زبان فارسی بهینه شده است.
 - معماری BERT: معماری زیربنایی ParsBERT بر اساس BERT است که مخفف Bidirectional Encoder Representation Transformer است. مدل‌های BERT به دلیل توانایی‌شان در گرفتن اطلاعات متنی به صورت دوطرفه، با در نظر گرفتن متن چپ و راست در یک دنباله از کلمات، شناخته شده‌اند.
 - مزایا نسبت به BERT چند زبانه: ParsBERT سبک‌تر از BERT چند زبانه است. با وجود معماری سبک‌تر، عملکرد بهتری دارد. این را می‌توان به آموزش آن بر روی مجموعه‌ای بزرگ‌تر و متنوع‌تر از مجموعه داده‌های فارسی از پیش آموزش دیده نسبت داد. تنوع در موضوعات و سبک‌های نوشتاری در داده‌های آموزشی به توانایی مدل برای رسیدگی مؤثر به طیف وسیعی از وظایف کمک می‌کند.
- مدل دیگری که در مقاله هم به آن اشاره شد، مدل XLM-RoBERTa می‌باشد. XLM-RoBERTa یکی دیگر از مدل‌های زبان پوشانده مبتنی بر ترانسفورماتور است که روی متن به ۱۰۰ زبان مختلف از قبل آموزش داده شده است.
- مدل دیگری که در مقاله به آن اشاره شد، مدل XLM-EMO می‌باشد. XLM-EMO یک مدل تشخیص احساسات چند زبانه برای متن رسانه‌های اجتماعی است. این بر روی مدل XLM-T ساخته شده است، که خود نسخه تنظیم شده‌ای از پایه XLM-RoBERTa است که به طور خاص برای داده‌های توییتر سازگار شده است. XLM-EMO روی مجموعه داده‌ها در 19 زبان مختلف تنظیم شده است و برچسب‌های احساسات را در این زبان‌ها برای ارزیابی بین زبانی مؤثرتر استاندارد می‌کند. این مدل برای ضبط و پیش‌بینی احساسات در محتوای رسانه‌های اجتماعی طراحی شده است، که انعطاف‌پذیری و سازگاری مدل‌های مبتنی بر ترانسفورماتور را برای تحلیل احساسات چند زبانه نشان می‌دهد.
- باتوجه به این که ParsBERT به طور خاص برای زبان فارسی طراحی شده است که ممکن است به آن مزیتی در درک نکات ظریف متن فارسی بدهد. موفقیت قبلی: در کارهای مختلف NLP فارسی، از جمله طبقه‌بندی متن و تحلیل احساسات، عملکرد برتر را نشان داده است. و به دلیل موجود بودن مدل در سایت huggingface از مدل ParsBert استفاده کردیم.

۴ اقدامات انجام شده

این مقاله درباره ED در داده‌های فارسی بوده. براساس این مقاله، دیتاست استفاده شده در این مقاله و در پروژه ما، از توثیت‌های فارسی، کامنت‌های اینستا و کامنت‌های دیجیکالا جمع‌آوری شده است. که برای این کار از api های مجزا استفاده شده است. بعد از جمع‌آوری دیتاها، لیبل گذاری آن‌ها به صورت ترکیبی دستی و اتوماتیک انجام شده است. (7500 از جمله‌ها به صورت دستی با کمک

یک بات تلگرامی annotate شده است). همچنین برای کنترل کیفیت لیبل‌ها، تعادل توزیع آن‌ها بین کلاس‌های مختلف می‌بایست بررسی می‌شد.



تصویر ۱: تعداد توئیت‌های هر کلاس

در این بخش از مقاله، مدل‌های پایه برای طبقه‌بندی احساسات بر روی ArmanEmo ارائه می‌شود که از یادگیری انتقالی استفاده می‌کنند. این مدل‌های پایه شامل ParsBert, XLM-RoBERTa, XLM-EMO می‌باشند که در قسمت انتخاب مدل براساس مطالب گفته شده در مقاله، به بررسی هریک پرداختیم. براساس مقاله، برای fine tune کردن زبان یا مدل‌های احساسی از پیش آموزش‌دیده شده برای این تسک، یعنی «تشخیص احساسات»، یک لایه dense fully connected اضافه شده است. همچنین یک cross-entropy loss function را برای انجام طبقه‌بندی چندکلاسه معرفی شده است. در حین fine tuning، تمام وزن‌های شبکه به جز وزن‌های لایه آخر را فریز می‌کنیم و برای لایه آخر از یک dense layer استفاده شده است. نتایج آزمایش‌های مختلف انجام شده در مقاله بر روی دیتاست ArmanEmo در جدول 1 آمده است.

به منظور مقایسه مدل‌های احساسی مبتنی بر مدل‌های زبانی تنظیم‌شده با سایر شبکه‌های عصبی، مدل‌های مبتنی بر CNN و RNN را در مجموعه داده‌های آموزش و آزمایش ارزیابی شده است. براساس مقاله، عملکرد بهترین مدل در بین مدل‌های پایه ما-XLM-RoBERTa-large می‌باشد. مدل به میانگین F1 score 75.39 در مجموعه آزمون دست یافته است و بهترین عملکرد را ارائه می‌دهد.

ما با استفاده از مطالب مقاله مذکور پس از پیش‌پردازش داده‌ها، ابتدا مدل را load کردیم و همان‌طور که گفته شده وزن‌ها را فریز کرده و از لایه‌های linear در لایه‌های ابتدایی و انتهایی مدل استفاده کردیم. به دلیل محدودیت منابع (gpu) ما مدل را در 5 اپوک آموزش دادیم و loss آن را در هر اپوک چاپ کردیم. بعد از آموزش مدل، داده‌های تست را ارزیابی کردیم که متریک‌های accuracy, precision, recall, F1 را برای هر کلاس محاسبه کردیم.

جدول 1: مقایسه عملکرد مدل ها

Model	Precision (Macro)	Recall (Macro)	F1 (Macro)
FastText [42]	54.82	46.37	47.24
HAN [43]	49.56	44.12	45.10
RCNN [44]	50.53	48.11	47.95
RCNNVariant	51.96	48.96	49.17
TextAttBiRNN [45, 46]	54.66	46.26	47.09
TextBiRNN	51.45	47.16	47.14
TextCNN [47]	58.66	51.09	51.47
TextRNN [48]	49.39	47.20	46.79
ParsBERT	67.10	65.56	65.74
XLM-Roberta-base	72.26	68.43	69.21
XLM-Roberta-large	75.91	75.84	75.39
XLM-EMO-t	70.05	68.08	68.57

۵ ارزیابی مدل

همان طور که در قسمت قبل اشاره شد، متریک های $F1$, $accuracy$, $precision$, $recall$ را برای هر کلاس محاسبه شد که به صورت زیر می باشد. همچنین $accuracy$ کل مقدار 99 درصد شد. به منظور بررسی دقیق تر و تحلیل ساده تر از عملکرد و خروجی مدل، $confusion matrix$ هر کلاس را نیز رسم کردیم که به صورت زیر می باشد.

Total accuracy is 0.9929096209912537

تصویر ۲: درصد محاسبه شده

Accuracy for OTHER: 0.9936326530612245
Precision for OTHER: 0.9742489270386266
Recall for OTHER: 0.9837486457204767
F1 for OTHER: 0.9789757412398922
Confusion Matrix for OTHER:
[[5178 24]
[15 908]]

تصویر ۳: موارد محاسباتی خواسته شده برای کلاس Other

```
Accuracy for SURPRISE: 0.9900408163265306
Precision for SURPRISE: 0.9460526315789474
Recall for SURPRISE: 0.972936400541272
F1 for SURPRISE: 0.9593062041360907
Confusion Matrix for SURPRISE:
[[5345  41]
 [  20 719]]
```

تصویر 4: موارد محاسباتی خواسته شده برای کلاس Surprise

```
Accuracy for SAD: 0.9923265306122449
Precision for SAD: 0.9851428571428571
Recall for SAD: 0.9620535714285714
F1 for SAD: 0.9734613212874081
Confusion Matrix for SAD:
[[5216  13]
 [  34 862]]
```

تصویر 5: موارد محاسباتی خواسته شده برای کلاس Sad

```
Accuracy for HATE: 0.9898775510204082
Precision for HATE: 0.9149722735674677
Recall for HATE: 0.9686888454011742
F1 for HATE: 0.94106463878327
Confusion Matrix for HATE:
[[5568  46]
 [  16 495]]
```

تصویر 6: موارد محاسباتی خواسته شده برای کلاس Hate

```
Accuracy for HAPPY: 0.9951020408163266
Precision for HAPPY: 0.9932885906040269
Recall for HAPPY: 0.9579288025889967
F1 for HAPPY: 0.9752883031301481
Confusion Matrix for HAPPY:
[[5503   4]
 [  26 592]]
```

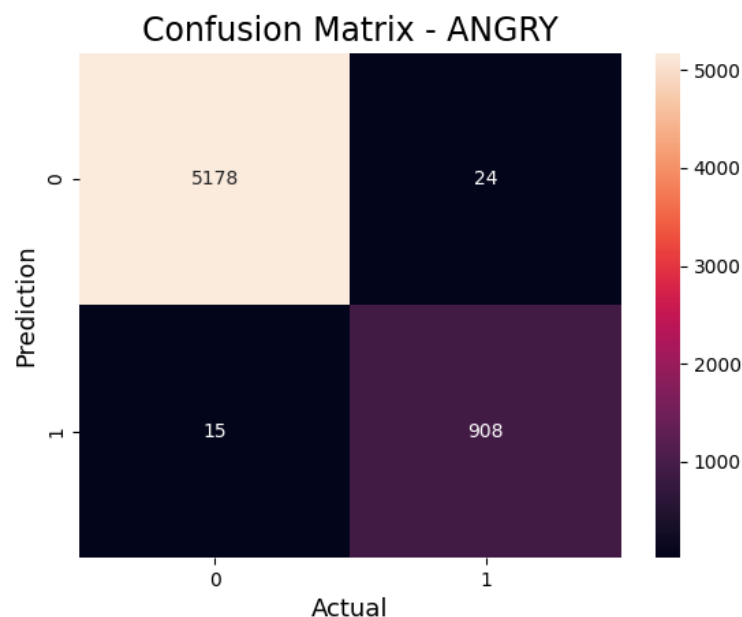
تصویر 7: موارد محاسباتی خواسته شده برای کلاس Happy

```
Accuracy for FEAR: 0.9957551020408163
Precision for FEAR: 0.9959294436906377
Recall for FEAR: 0.9696169088507266
F1 for FEAR: 0.9825970548862115
Confusion Matrix for FEAR:
[[5365    3]
 [  23   734]]
```

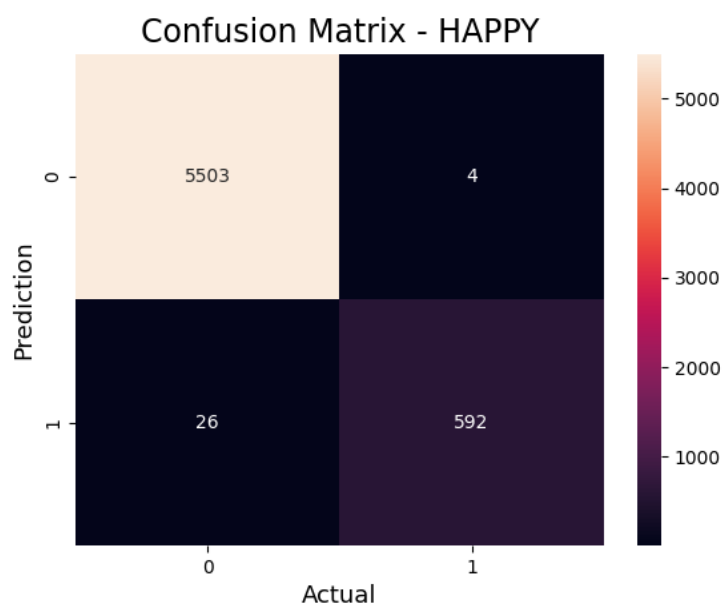
تصویر 8: موارد محاسباتی خواسته شده برای کلاس Fear

```
Accuracy for ANGRY: 0.9936326530612245
Precision for ANGRY: 0.9742489270386266
Recall for ANGRY: 0.9837486457204767
F1 for ANGRY: 0.9789757412398922
Confusion Matrix for ANGRY:
[[5178    24]
 [   15   908]]
```

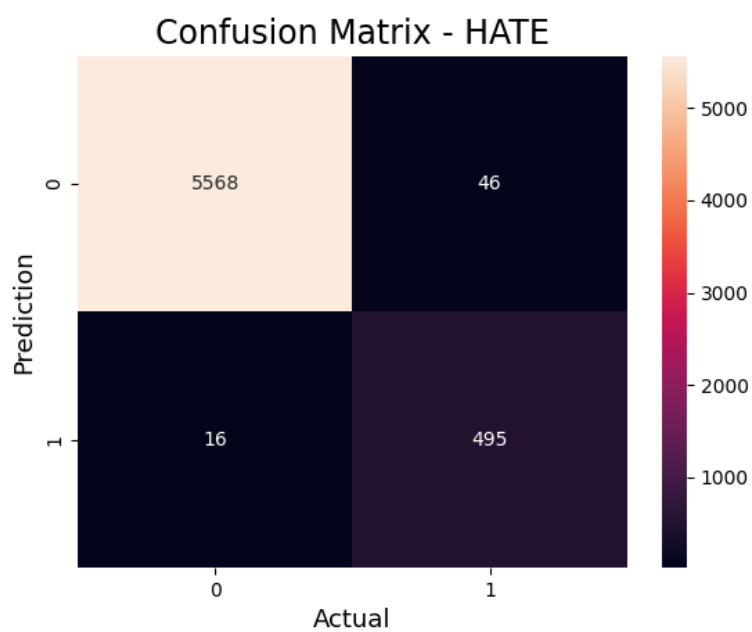
تصویر 9: موارد محاسباتی خواسته شده برای کلاس Angry



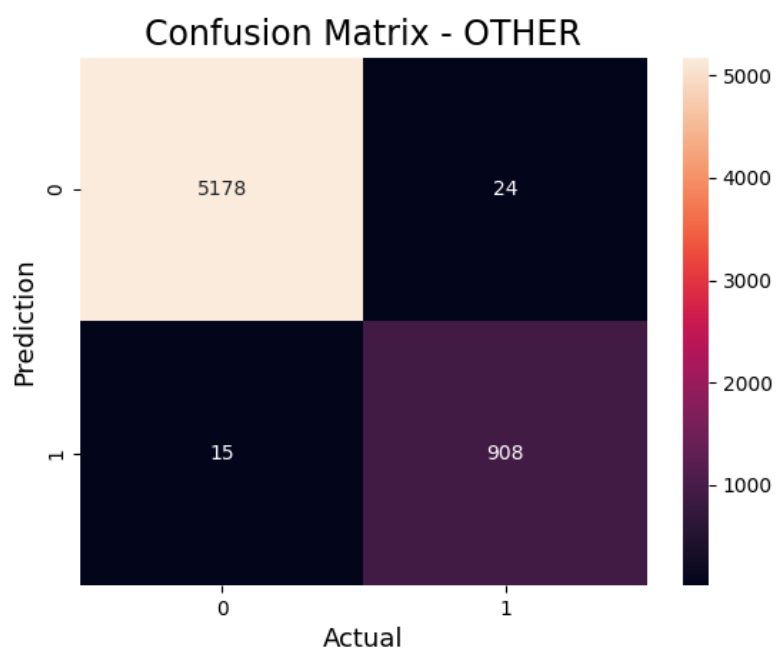
تصویر 10: Angry confusion matrix



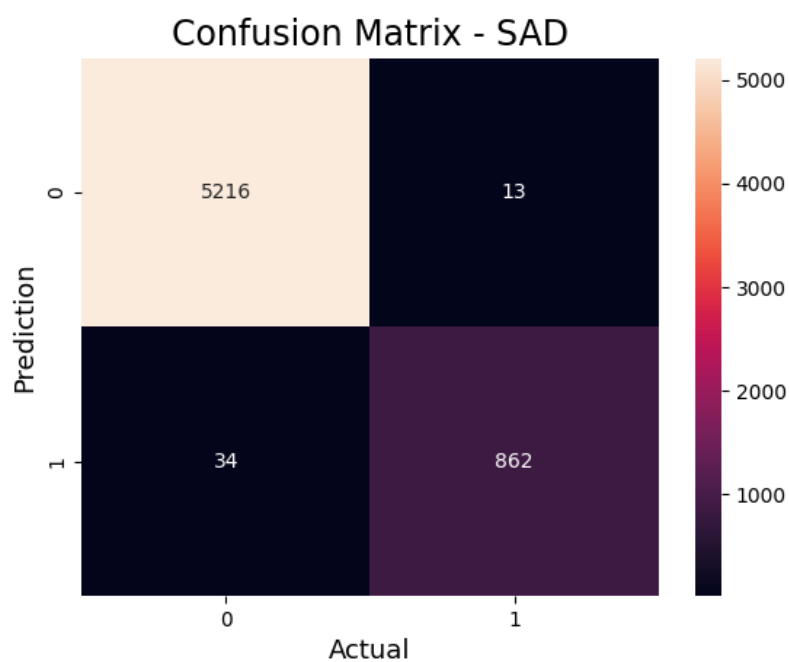
تصویر 10: Happy کلاس confusion matrix



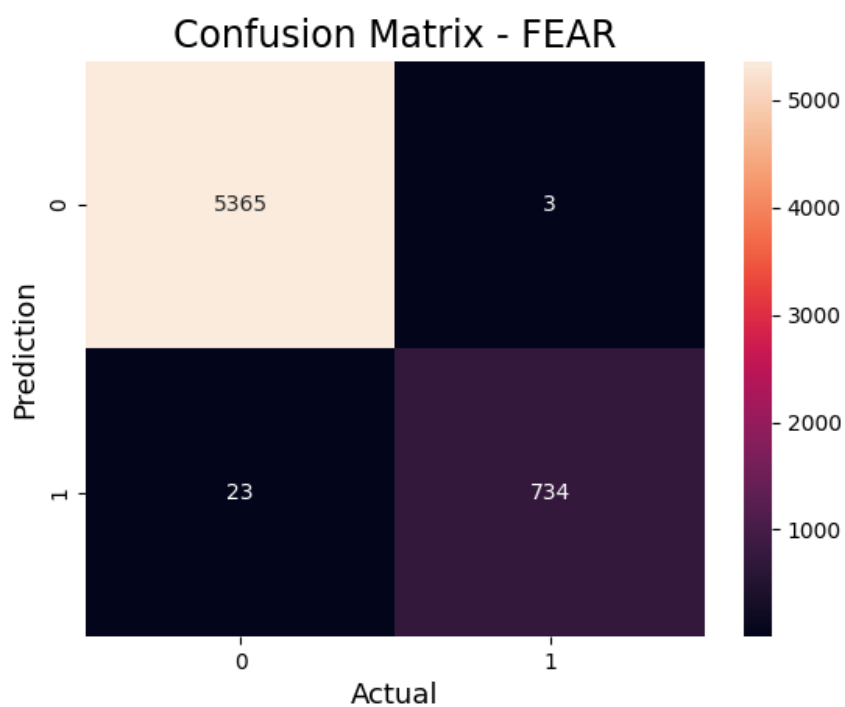
تصویر 11: Hate کلاس confusion matrix



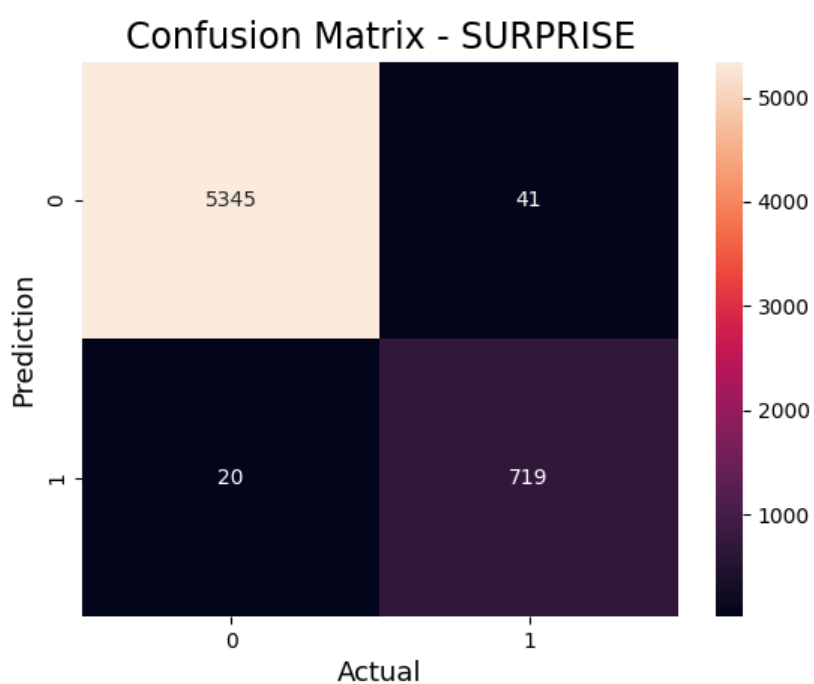
تصویر 12: Other کلاس confusion matrix



تصویر 13: Sad کلاس confusion matrix



تصویر 14: Fear کلاس confusion matrix



تصویر 15: Surprise کلاس confusion matrix

5 نمونه جمله را از دیتاست آزمایش به صورت رندوم انتخاب می کنیم و آن را به مدل می دهیم که خروجی به صورت زیر می باشد:

- The text is : قانون منع بکارگیری بازنشستگان شهرداران را معاف کرده ! این قانونگذاری مملکت خراب شده ما : خیلی عجیبه !
یه قانون تصویب میشه بعدیه تبصره میزنن قانون رو نفی میکنه . مثل این میمونه که بگن طبق قانون دزدی کار غیرقانونی است ولی تبصره معافیت از این قانون برای فقرا میاد . مجلس _ پی _ عرضه
The True Label is : ANGRY
The Predicted label is ANGRY
- The text is : خیلی با دقت هم دنبال می کنید:
The True Label is : OTHER
The Predicted label is OTHER
- The text is : دو روز به سالگرد ازدواجمون مونده و با رسیدن این کارت ، شور و هیجانی که سال پیش با ورق خوردن : تقویم و رسیدن به مراسم ازدواجمون داشتم رو دوباره حس کردم .
قراره به عنوان ساقدوش در کنار دو دوست همجنسگرای که توی شادی و غم کنارمون بودن حاضر بشیم . امیدوارم همه این حس رو تجربه کنن رامینما
The True Label is : HAPPY
The Predicted label is HAPPY
- The text is : اینقدر بدم ازن با حجاب اسلامی میاد نمی دونم چرا:
The True Label is : HATE
The Predicted label is HATE
- The text is : اینقدر که حال و هوای قبرستان بهم میچسبه شادی و ... بهم نمیچسبه !!!
حال و هوای غم و تنهایی !!!
دوست دارم !!! بعضی وقت ها تنها وی موقع میرم
The True Label is : SAD
The Predicted label is SAD

۷ مراجع

- [1] <https://arxiv.org/abs/2207.11808> - ARMANEMO: A PERSIAN DATASET FOR TEXT-BASED EMOTION DETECTION
- [2] <https://huggingface.co/HooshvareLab/bert-base-parsbert-uncased>
- [3] https://huggingface.co/docs/transformers/en/model_doc/distilbert
- [4] <https://huggingface.co/HooshvareLab/bert-fa-base-uncased>
- [5] <https://github.com/ICTRC/Parsivar>
- [6] <https://github.com/hooshvare/parsbert>
- [7] <https://github.com/AmirAbaskohi/Persian-Emotion-Detection-using-ParsBERT-and-Imbalanced-Data-Handling-Approaches/tree/main>
- [8] <https://medium.com/@ahmettsdmr1312/fine-tuning-distilbert-for-emotion-classification-84a4e038e90e>

پیاده سازی این پروژه با همکاری گروه ستاره باباجانی و ملیکا محمدی فخار صورت گرفته است.