به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس یادگیری عمیق با کاربرد در بینایی ماشین و پردازش صوت

تمرین شماره ۴

نام و نام خانوادگی : مهیار ملکی

شماره دانشجویی : ۸۱۰۱۰۰۴۷۶

تیر ماه ۱۴۰۱

فهرست

فهرست اشکال و جداول

مقدمه

این تمرین دو بخش متفاوت را شامل می‌شود. در بخش اول مفهموم شبکه‌های ترنسفورمری و کاربرد آنها در دو مسئله معروف حوزه پردازش گفتار یعنی تبدیل صوت به متن و تشخیص کلیدواژه بررسی می‌شود. همچنین در بخش دوم به بررسی کاربرد این شبکه ها برای داده های متنی خواهیم پرداخت و یک مدل ترنسفورمری را برای مسئله پرسش و پاسخ آموزش خواهیم داد.

توضیحات کد:

فایل‌های main و model سه مسئله تبدیل صوت به متن، تشخیص کلیدواژه و مسئله پرسش و پاسخ هر کدام در یک پوشه جداگانه قرار گرفته است.

برای هر مدل با دادن مقادیر دلخواه ورودی به شکل یک لیست[1] به آرگومان WholeDataset تابع data_loader می‌توان فقط بخش کوچکی از دادگان را بارگزاری کرد. همچنین با دادن مقدار True کل داده ها به عنوان ورودی در نظر گرفته خواهند شد.

[1] [num_train, num_validation, num_test] (question answering doesn't have any test dataset)

سوال اول :

بخش اول :

۱. چرا در آموزش HuBert یک روند جدای آموزشی برای خوشه‌بندی صوت داریم؟

استفاده از خوشه‌بندی صوت کمک می‌کند تا داده های صوتی ساختاری مشابه مدل های زبانی پیدا کنند. خوشه بندی صوت در جایگاه یک الگوریتم self-supervised عمل کرده و لیبل هایی[۱] برای داده های صوتی ایجاد می‌کند. لذا صوت می‌تواند به عنوان دنباله ای از اجزای گسسته‌ی لیبل‌دار در نظر گرفته شود و در نتیجه این امکان ایجاد می‌شود تا بتوان از مدل های قدرتمند حوزه پردازش زبان های طبیعی مانند Bert در کاربردهای تشخیص گفتار استفاده کرد.

۲. چرا در آموزش HuBert از تابع هزینه Cross Entropy استفاده می‌شود و هدف از این طبقه‌بندی چند کلاسه چیست؟

در فرایند آموزش HuBert به دلیل این که ما در واقع از خود مدل Bert استفاده کرده‌ایم لذا می‌توانیم به جای تابع هزینه‌ی پیچیده‌ای[۲] که در مدل wave2vec استفاده شد، از تابع هزینه cross entropy استفاده شده در مدل Bert استفاده کنیم. این تابع هزینه ساده‌تر بوده و کمک می‌کند تا فرایند آموزش پایدارتری داشته باشیم. همانطور که در سوال ۱ گفته شد استفاده از ایده HuBert به نوعی داده‌های صوتی را خوشه بندی کرده و این خوشه‌ها معادل لغات یا tokenها در دنباله های متنی می‌باشند، لذا از تابع هزینه cross entropy می‌توان برای پیش‌بینی این خوشه‌ها استفاده کرد.

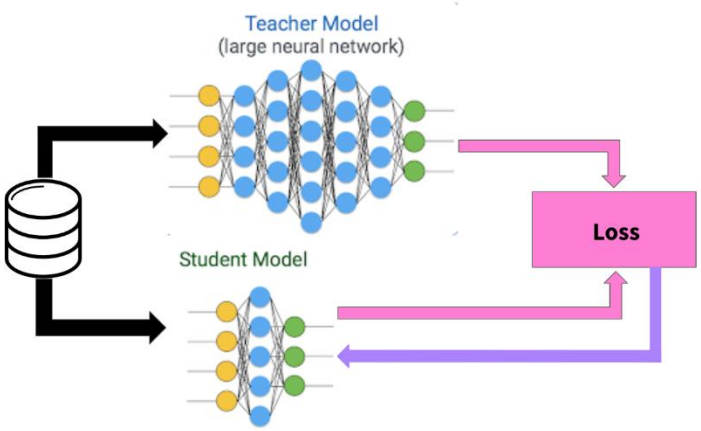
۳. فرآیند تقطیر را مطالعه کنید و توضیح دهید که چگونه این فرآیند در DistilHubert پیاده‌سازی شده است؟

مدل های عمیق مانند HuBert حافظه بزرگی نیاز داشته و هزینه‌های زیادی دارند. لذا برای کاربردهای آکادمیک و شرکت‌های کوچک غیر قابل استفاده می‌باشند. در فرایند تقطیر دانش[۳] به دنبال کاهش حداکثری اندازه مدل در برابر کاهش حداقلی توانایی آن هستیم. فرایند تقطیر از دو ساختار دانش‌آموز و آموزگار[۴] برای این کار استفاده می‌کند. ابتدا ساختار آموزگار که همان مدل اصلی و پرهزینه می‌باشد،

---

۱ Hidden units
۲ contrastive loss + diversity loss
۳ Knowledge Distilation
۴ Student & Teacher

با تمامی پارامترها آموزش می‌یابد. سپس مدل تقلیل یافته دانش آموز تعریف می‌شود. در نهایت مطابق شکل ۱ تابع هزینه بر خروجی هر دو مدل اعمال شده و عملیات پس‌انتشار[۱] فقط در مدل دانش‌آموز انجام شده و پارامترهای آن به روزرسانی می‌شوند.



شکل ۱- فرایند تقطیر دانش

٤. با مطالعه اسناد کتابخانه Huggingface، ورودی‌های اصلی مدل HubertForCTC را برای انجام یک پیش‌بینی توضیح دهید و بگویید که هر کدام چه نقشی ایفا می‌کنند و در کجای ساختار مدل به صورت ورودی مورد استفاده قرار می‌گیرند؟

جدول ۱- ورودی‌های اصلی مدل HubertForCTC

ورودی	مورد استفاده
input_values	مقادیر خام شکل موج صوت ورودی می‌باشد که قبل از استفاده در مدل، توسط تابع Wav2Vec2Processor پیش‌پردازش‌هایی چون padding و تبدیل به tensor روی آن اعمال می‌شود
inputs_ids	ایندکس متناظر با هر توکن یا لغت را به شبکه می‌دهد
vocab_size	تعداد کل توکن‌های متمایز
attention_mask	یک لیست از 0 و 1 که وظیفه ماسک کردن ورودی را به عهده دارد تا توابع مدل مثل convolution و attention روی padها اعمال نشوند
hidden_size	ابعاد لایه‌های انکدر
num_hidden_layers	تعداد لایه‌های مخفی انکدر

[۱] Backpropagation

بخش دوم :

در این بخش مدل DistilHubert بارگزاری و وزن‌های بخش از پیش آموزش دیده، ثابت شد. سپس مدل برای مسئله تبدیل صوت به متن، بر روی بخش asr مجموعه داده Librispeech با مشخصات ذکر شده در جدول ۲ آموزش داده شد.

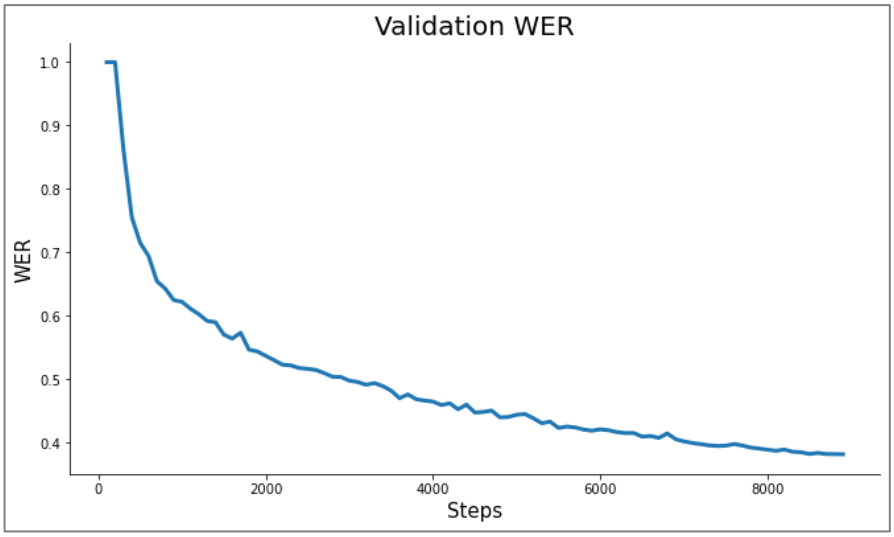
جدول ۲- پارامترهای مدل تبدیل صوت به متن

پارامتر	مقدار
Learning Rate	1e-3
Weight Decay	5e-3
Num Warmup Steps	250
Sampling Rate	16Khz
Num Train samples	28539
Num Validation samples	2703
Num Test samples	2620
Batch Size	16
Num Epochs	5

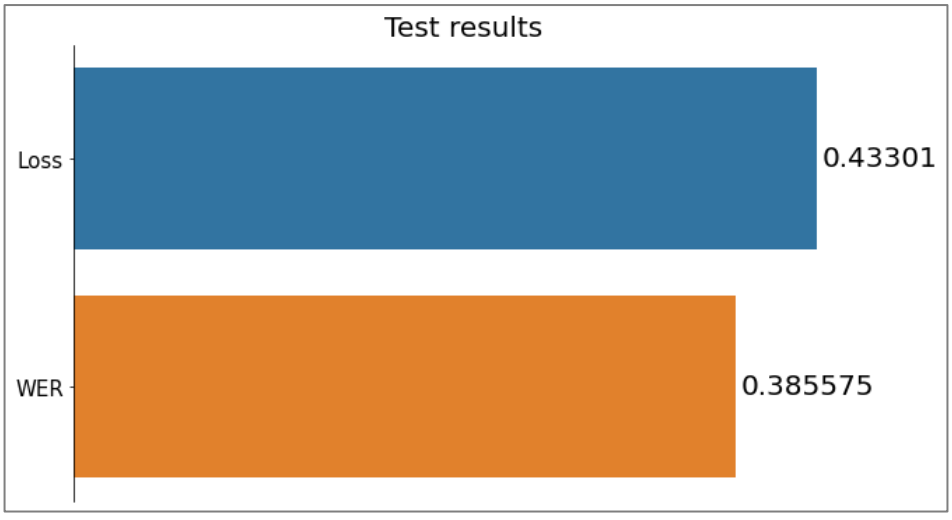
چنانچه در نمودار خطای شکل ۲ قابل مشاهده است، دو منحنی دادگان آموزش و اعتبارسنجی به خوبی همگرا شده اند. البته به نظر می‌رسد در انتها، مدل تمایل به فرابرازش دارد و با ادامه فرایند آموزش فرابرازش مدل امری محتمل است. در مقابل در نمودار شکل ۳ مشاهده می‌شود که روند منحنی خطای wer بر روی دادگان اعتبار سنجی به خوبی کاهشی بوده ولی هنوز به طور کامل همگرا نشده است. لذا با توجه به نمودار شکل ۳ می‌توان نتیجه گرفت آموزش مدل را برای چند دوره دیگر می‌توان ادامه داد.

در نهایت در شکل ۴ نتایج نهایی مدل آموزش دیده شده، روی دادگان ارزیابی نیز قابل مشاهده است. چنانچه انتظار میرفت نتایج بدست آمده در قسمت های قبلی، به روی دادگان ارزیابی نیز حاصل شده‌اند.

شکل ۲- نمودار خطا مسئله تبدیل صوت به متن



شکل ۳- معیار wer دادگان اعتبارسنجی تبدیل صوت به متن



شکل ٤- نتایج آزمون دادگان ارزیابی مسئله تبدیل صوت به متن

بخش سوم :

در این بخش نیز با استفاده از مدل DistilHubert و ثابت کردن وزن‌های بخش از پیش آموزش دیده، این بار مدل برای مسئله تشخیص کلیدواژه، بر روی بخش ks مجموعه داده SpeechCommands با مشخصات ذکر شده در جدول ۳ آموزش داده شد.

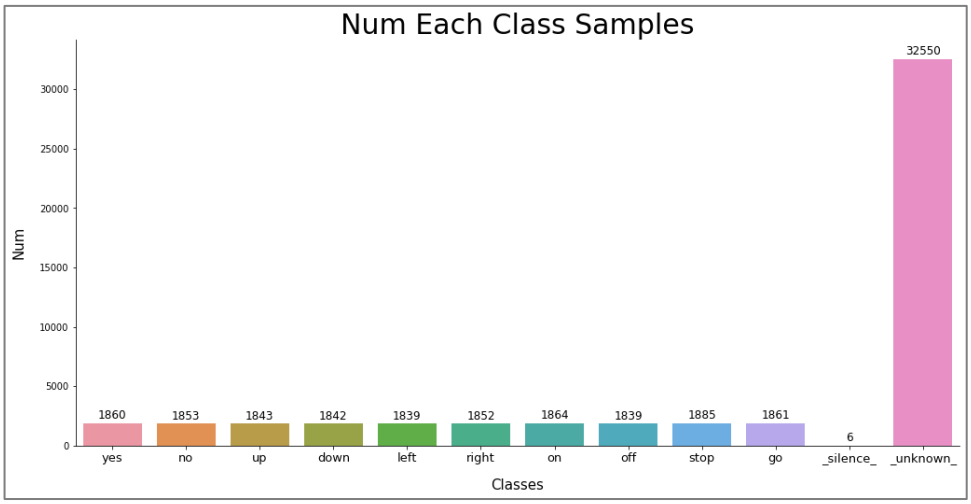
جدول ۳- پارامترهای مدل تشخیص کلیدواژه

پارامتر	مقدار
Learning Rate	1e-3
Weight Decay	5e-3
Num Warmup Steps	250
Sampling Rate	16Khz
Num Train samples	51094
Num Validation samples	6798
Num Test samples	3081
Batch Size	64
Num Epochs	10

۱. با توجه به اینکه این دادگان دچار مشکل عدم توازن فراوانی هستند، چه راهکارهایی را میتوان برای آموزش بهتر مدل اتخاذ کرد؟

با توجه به شکل ۵ مشاهده می‌شود که فراوانی دادگان آموزش در دسته‌های مختلف دچار عدم توازن زیادی است. برای مثال دسته _silence_ فقط ۶ عدد داده دارد که نسبت به بقیه دسته‌ها ناچیز است، یا دسته _unknown_ بیش از ۱۶ برابر باقی دسته ها داده دارد. لذا این عدم توازن باید مدیریت شود. برای این کار می‌توان دسته _silence_ را به طور کلی حذف کرد زیرا عملا داده ای نداشته و برای آن آموزشی انجام نخواهد شد. همچنین در ادامه برای بهبود فرایند آموزش و حل مشکل عدم توازن سایر دسته ها

می‌توان در تابع خطا برای هر دسته وزن متفاوتی برای محاسبات در نظر گرفت. برای مثال دسته _unknown_ که تعداد داده های بسیار بیشتری دارد وزن کمتری خواهد داشت. همچنین می‌توان با استفاده از روش‌هایی مثل over/under sampling یا augmentation یا شبکه‌های autogenerative نیز این عدم توازن دادگان را مدیریت کرد.



شکل ۵- فراوانی دسته های مختلف دادگان آموزش

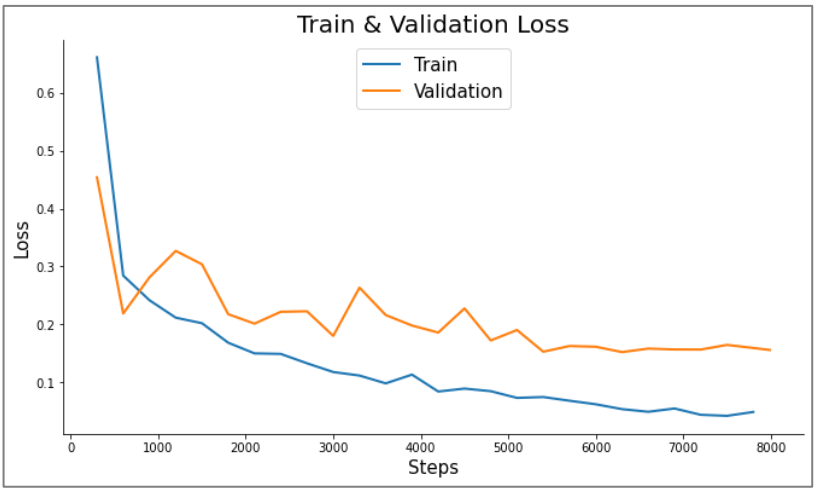
۲. در این مسئله با پیاده‌سازی یک تابع هزینه‌ی وزن‌دار، مشکل عدم توازن را بهبود ببخشید.

چنانچه در شکل ۶ قابل مشاهده است، وزن های هر دسته با استفاده از کتابخانه sklearn محاسبه شده‌اند. سپس این وزن‌ها در یک تابع cross entropy اعمال شده و با تعریف یک زیرکلاس جدید در تابع Trainer قرار گرفته است.

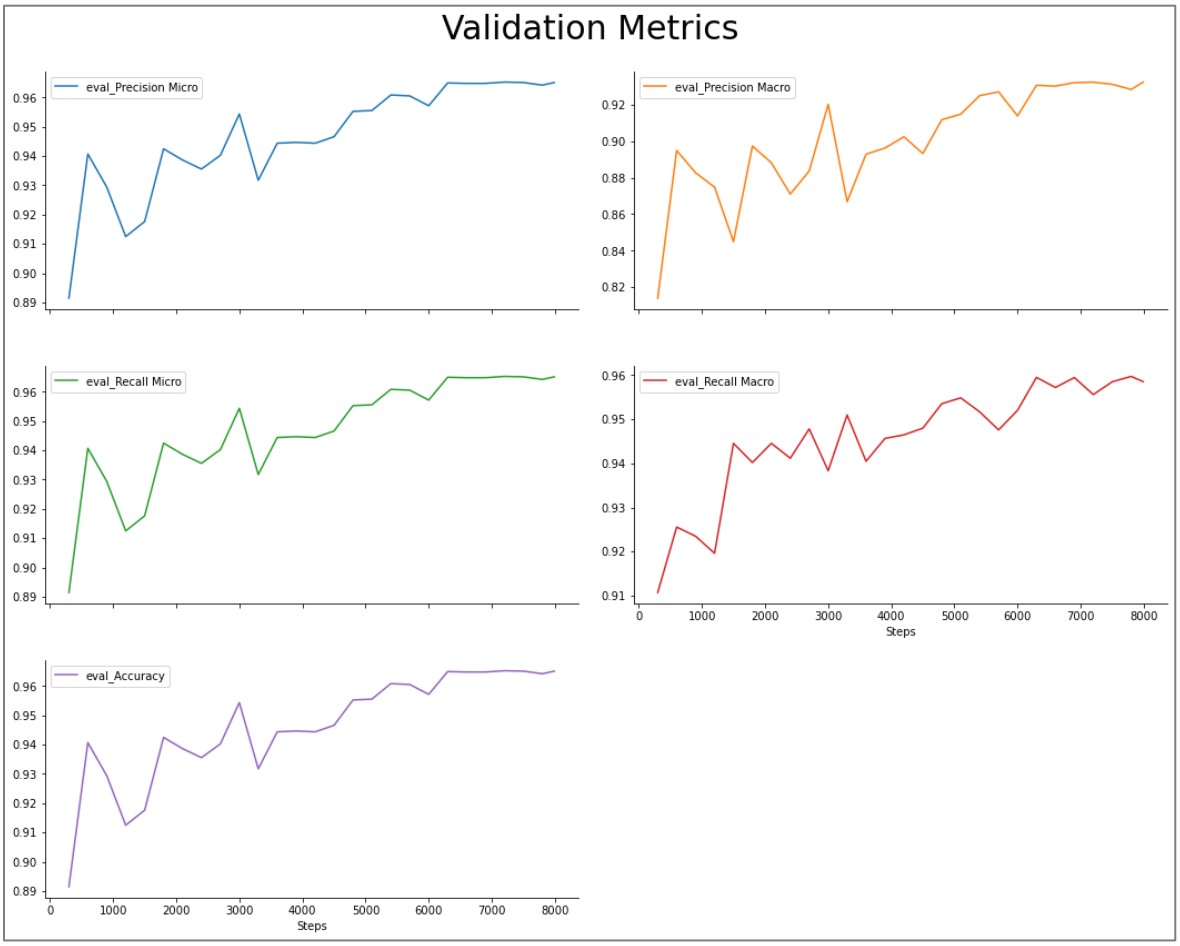
```
from transformers import Trainer
from sklearn.utils import class_weight
import numpy as np
import torch

y = dataset['train']['label']
class_weights = class_weight.compute_class_weight(classes=np.unique(y), y=np.array(y),
                                                  class_weight='balanced')
class_weights = torch.tensor(class_weights,dtype=torch.float).to('cuda:0')
loss_fct = torch.nn.CrossEntropyLoss(weight=class_weights,reduction='mean')

class CustomTrainer(Trainer):
    def compute_loss(self, model, inputs, return_outputs=False):
        labels = inputs.get("labels")
        # forward pass
        outputs = model(**inputs)
        logits = outputs.get("logits")
        # compute custom loss (suppose one has 3 labels with different weights)
        loss = loss_fct(logits.view(-1, self.model.config.num_labels), labels.view(-1))
        return (loss, outputs) if return_outputs else loss
```

شکل ٦- تابع هزینه وزن‌دار

میزان خطای مدل برای هر دو دادگان آموزش و اعتبارسنجی در شکل ۷ قابل مشاهده است. با توجه به شکل به نظر می‌رسد هر دو منحنی همگرا شده و ادامه فرایند آموزش بهبود چندانی را نتیجه نخواهد داد. همچنین در نمودار خطای دادگان اعتبارسنجی نویز زیادی مشاهده می‌شود که البته با توجه به این که ارزیابی دادگان اعتبارسنجی انتهای هر دوره صورت نگرفته و هر ۳۰۰ قدم انجام می‌شود، امری طبیعی‌ست.

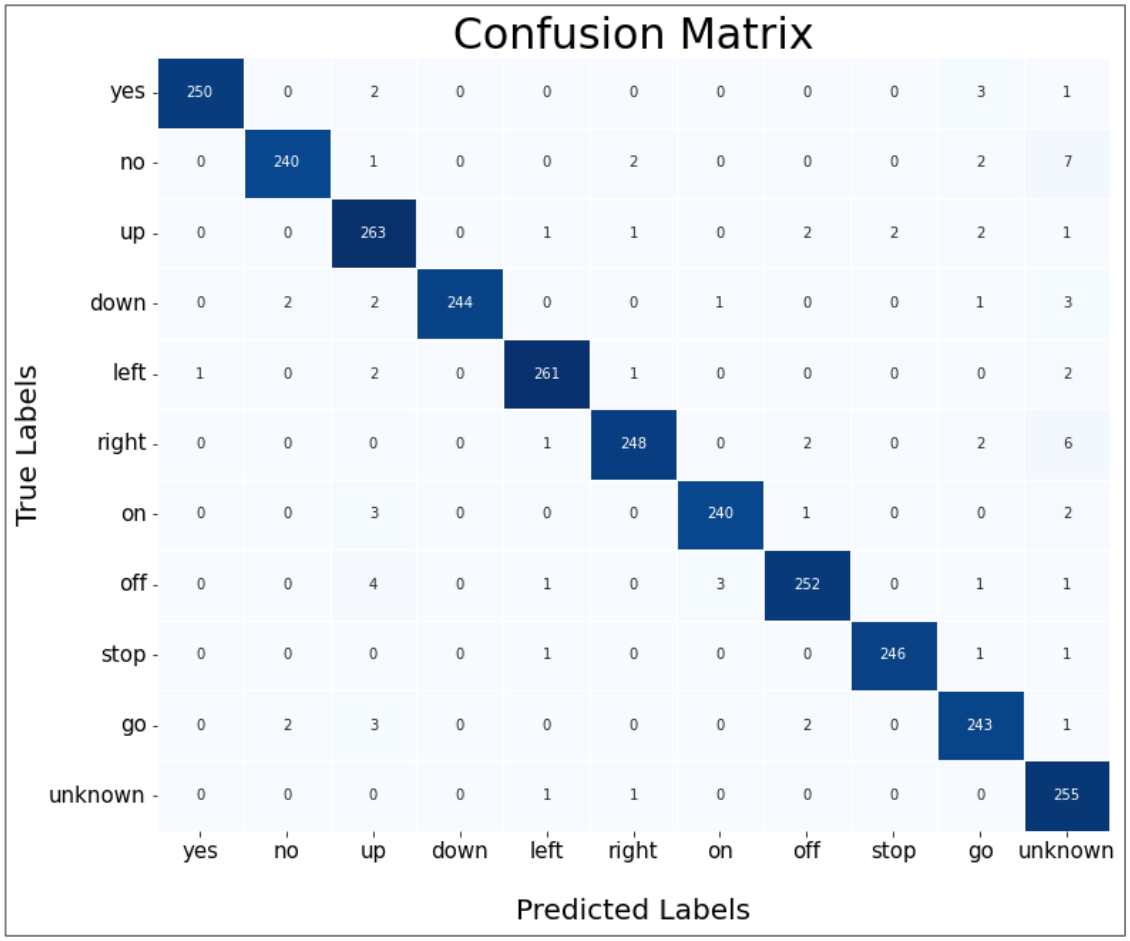


شکل ۷- نمودار خطای مسئله تشخیص کلیدواژه



شکل ۸- معیارهای ارزیابی دادگان اعتبارسنجی مسئله تشخیص کلیدواژه

همچنین نتایج حاصل از معیارهای ارزیابی دقت دادگان اعتبارسنجی نیز در شکل ۸ آورده شده است. چنانچه قابل مشاهده است تمامی معیارها به مقادیر بالای ۹۰ درصد رسیده و به نظر می‌رسد مدل به خوبی آموزش دیده‌باشد. نتایج حاصل از ماتریس درهم‌ریختگی شکل ۹ نیز این امر را تایید می‌کند.



شکل ۹- ماتریس درهم‌ریختگی مسئله تشخیص کلیدواژه

در نهایت در جدول ۴ نتایج نهایی مدل آموزش دیده شده، روی دادگان ارزیابی نیز قابل مشاهده است. چنانچه انتظار می‌رفت نتایج بدست آمده در قسمت های قبلی، به روی دادگان ارزیابی نیز حاصل شده و در تمامی معیارها به مقدار بسیار خوب ۹۷ درصد رسیده‌ایم.

جدول ٤- نتایج آزمون دادگان ارزیابی مسئله تشخیص کلیدواژه

Loss	Accuracy	Precision (Micro)	Precision (Macro)	Recall (Micro)	Recall (Macro)
0.1031	0.9709	0.9709	0.9720	0.9709	0.9710

سوال دوم :

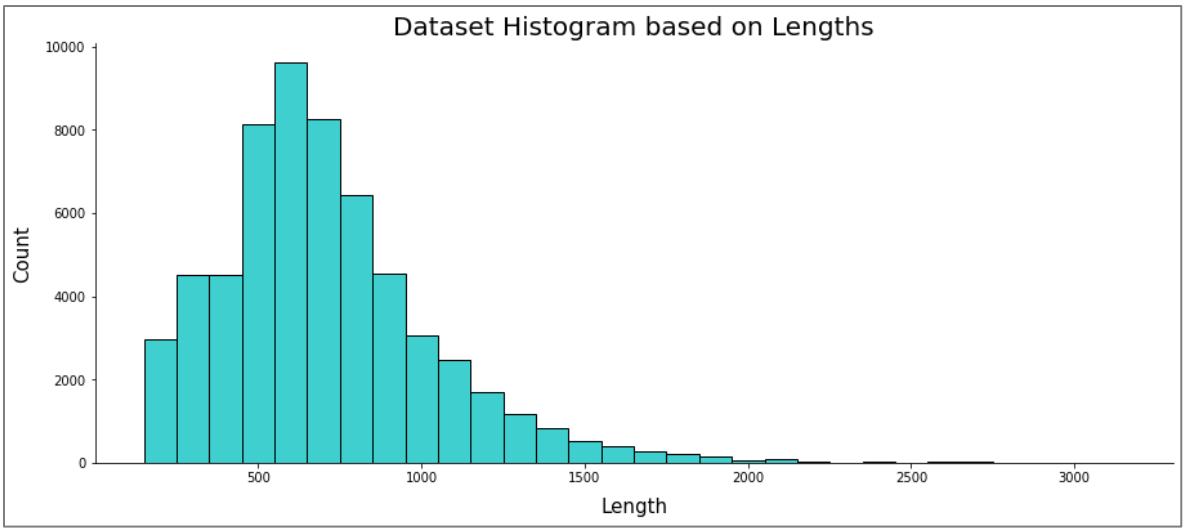
یکی از مسائلی که در پردازش زبان طبیعی مورد توجه قرار گرفته است، مسئله پرسش و پاسخ می‌باشد. در این مسئله یک متن داده می‌شود و با توجه به آن باید پاسخ سوال مورد نظر را پیدا کرد. یکی از معروفترین مجموعه دادگان برای مسئله پرسش و پاسخ SQuAD می‌باشد که ما در این تمرین از نسخه دوم این مجموعه داده استفاده می‌کنیم. این دادگان شامل سؤال‌هایی است که افراد آنها را از مجموعه متون سایت ویکیپدیا جمع‌آوری کرده‌اند.

بخش اول :

در این بخش دو مدل زبانی ترنسفورمری DistilBert و XLM انتخاب شده و برای مسئله پرسش و پاسخ بر روی مجموعه دادگان SQuAD V2 پیاده‌سازی و آموزش داده شدند.

لازم به ذکر است که به دلیل زمان آموزش بسیار طولانی و محدودیت‌های سخت‌افزاری موجود، از کل دادگان تنها ۶۰۰۰۰ داده از دادگان آموزش و ۵۰۰۰ داده از دادگان اعتبارسنجی انتخاب شدند.

با توجه به شکل ۱۰ با رسم فراوانی دادگان آموزش بر اساس طول آنها مشاهده شد که دادگان با طول تقریبی ۵۰۰ بیشترین تعداد را دارند، لذا عدد ۵۱۲ به عنوان طول بیشینه برای نمونه‌ها در نظر گرفته‌شد.



شکل ۱۰- فراوانی دادگان آموزش بر اساس طول داده‌ها

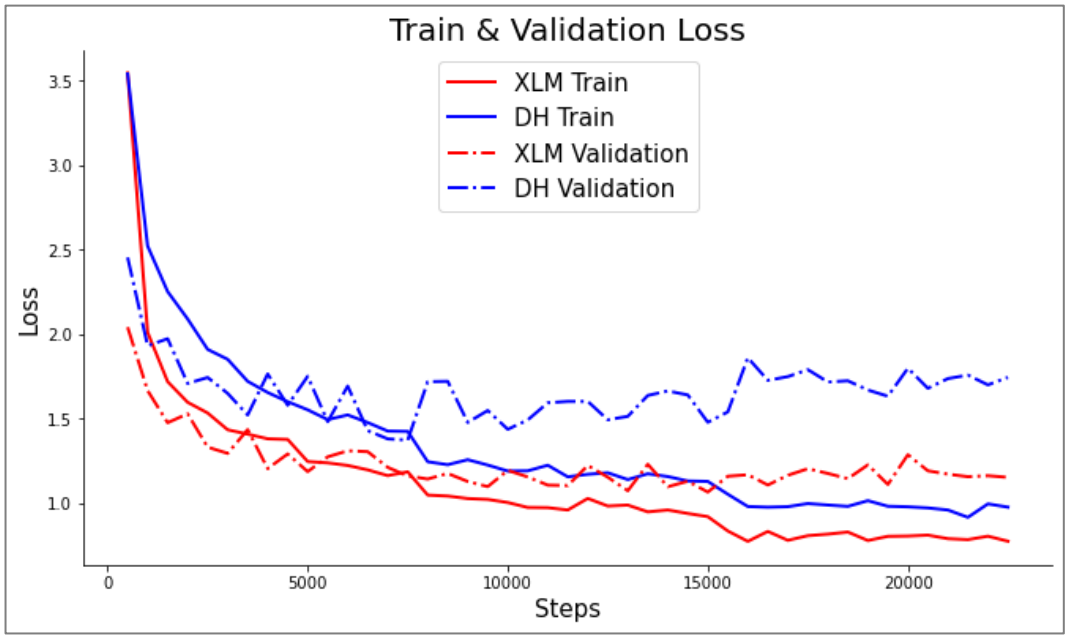
پارامترهای شبکه‌ی آموزش داده شده در جدول ۵ و مشخصات ساختاری دو مدل انتخاب شده نیز در جدول ۶ قابل مشاهده است.

جدول ۵- پارامترهای مدل پرسش و پاسخ

پارامتر	مقدار
Learning Rate	1e-5
Weight Decay	1e-4
Num Train samples	60000
Num Validation samples	5000
Batch Size	8
Num Epochs	3
Samples max length	512
Samples overlap	128

جدول ٦- مشخصات ساختاری دو مدل distilbert و xlm

Model name	DistilBert base	XLM Roberta base
Activation func.	gelu	gelu
Encoder layers dim.	768	768
Feed-forward layers size	3072	3072
Attention heads	12	12
Hidden layers	6	12
Vocab size	30522	250002



شکل ۱۱- نمودار خطای دو مدل distilbert و xlm

چنانچه در شکل ۱۱ قابل مشاهده است خطای هر دو مدل بر روی دادگان آموزش به خوبی نزولی بوده و همگرا شده‌است. البته مدل XLM عملکرد بهتری داشته و به مقادیر کمتری رسیده است، این امر چنانچه در جدول ۶ بیان شده است می‌تواند ناشی از بزرگتر بودن این مدل باشد، زیرا تعداد لایه‌های پنهان مدل XLM دو برابر مدل DistilBert بوده و همچنین تعداد واژگان بسیار بیشتری را نیز پشتیبانی می‌کند. این عملکرد بهتر مدل XLM در دادگان اعتبارسنجی نیز به وضوح قابل مشاهده است. چنانچه در شکل مشخص است مدل DistilBert دچار فرابرازش[۱] شده و خطای دادگان اعتبارسنجی آن پس از حدود ۱۰۰۰۰ قدم روندی صعودی به خود گرفته است، این در حالی‌ست که مدل XLM پس از این تعداد قدم تقریبا همگرا شده و کاهش یا افزایش خاصی در مقادیر خطای آن در ادامه مشاهده نمی‌شود. همچنین لازم به ذکر است که زمان آموزش مدل DistilBert به دلیل کوچکتر بودن آن بسیار کمتر از مدل XLM است.

[۱] Overtfitting

بخش دوم :

دو معیار پرکاربرد و مورد استفاده برای ارزیابی مسائل پرسش و پاسخ F1-score و exact match هستند.

- exact match :

این معیار مشخص می‌کند که آیا لغات جواب پیش‌بینی شده دقیقا مشابه لغات جواب درست است یا خیر. لذا برای هر داده مقدار آن ۱ یا صفر خواهدشد.

- F1-score :

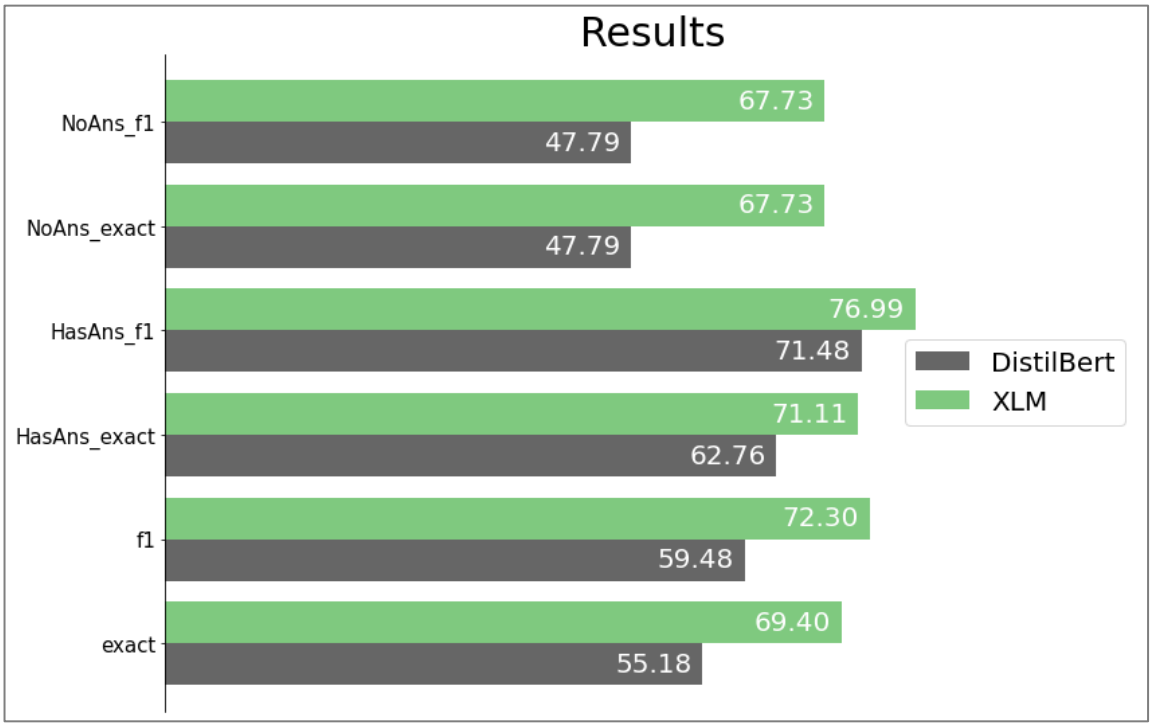
این معیار که در واقع میانگین دو معیار recall و precision می‌باشد، با توجه به لغاتی که درست یا غلط پیش‌بینی شده‌اند، اندازه‌گیری می‌شود: $F1 = 2.\frac{precision*recall}{precision+recall}$

در نهایت میانگین هر یک از این معیارها روی تمام داده‌ها گزارش خواهدشد. همچنین اگر سوالی وجود داشته‌باشد که چند جواب داشته‌باشد، این معیارها برای تمام جواب‌ها محاسبه شده و بیشترین مقدار آن برخواهدگشت. در اینجا این دو معیار را به سه شکل گزارش خواهیم‌کرد: برای پرسش‌های دارای جواب، پرسش‌های بی‌جواب و به صورت میانگین روی تمام پرسش‌ها.

جدول ۷- تعداد پرسش های بی‌جواب و باجواب دادگان اعتبارسنجی

پرسش‌های دارای جواب	پرسش‌های بی‌جواب	مجموع
2468	2532	5000

چنانچه در شکل ۱۲ قابل مشاهده است به طور کلی عملکرد هر دو مدل به خصوص در بخش پرسش‌های بدون جواب، خوب نیست، البته این امر می‌تواند ناشی از کاهش تقریبی تعداد داده‌ها به نصف تعداد دادگان اصلی باشد که با توجه محدودیت‌های موجود امری اجتناب ناپذیر بود. در مقایسه عملکرد دو مدل نیز مشاهده می‌شود که طبق انتظار مدل XLM بسیار بهتر عمل کرده‌است. دلیل آن طبق آنچه در بخش قبل گفته شد ناشی از بزرگتر بودن این مدل است، زیرا تعداد لایه‌های پنهان مدل XLM دو برابر مدل DistilBert بوده و همچنین تعداد واژگان بسیار بیشتری را نیز پشتیبانی می‌کند.



شکل ۱۲- معیارهای ارزیابی مسئله پرسش و پاسخ

منابع

[1] https://github.com/huggingface/notebooks/blob/main/examples/audio_classification.ipynb

[2] https://huggingface.co/blog/fine-tune-wav2vec2-english

[3] https://github.com/huggingface/notebooks/blob/main/examples/question_answering.ipynb

[4] https://huggingface.co/course/chapter7/7

[5] https://qa.fastforwardlabs.com/no%20answer/null%20threshold/bert/distilbert/exact%20match/f1/robust%20predictions/2020/06/09/Evaluating_BERT_on_SQuAD.html

[6] https://jonathanbgn.com/2021/10/30/hubert-visually-explained.html

[7] https://arxiv.org/abs/2106.07447

[8] https://arxiv.org/abs/2110.01900