شناسایی نظرات سمی با استفاده از یادگیری انتقالی*

محمد لشکری

۱ چکیده

با گسترش شبکه‌های اجتماعی در میام مردم، یکی از دغدغه‌های اصلی امروز در دنیا تبدیل این فضا به مکانی آرام و ایمن برای تبادل اطلاعات و به اشتراک گذاری نظرات است. یکی از عوامل برهم زدن نظم و آرامش فضای مجازی، منتشر کردن پیام‌های توهین‌آمیز، نژادپرستانه، تهدید و... است. ما این گونه پیام‌ها را سمی می‌نامیم. نیاز امروز ما شناسایی این نظرات سمی و جلوگیری از انتشار آن‌هاست؛ از آنجا که مدل‌های هوش مصنوعی کلاسیک دقت خوبی در این مسئله ندارند، به‌کارگیری مدل‌های یادگیری ماشین پردازش زبان‌های طبیعی برای حل این مسئله لازم است. در این مقاله، مدل‌هایی مبتنی بر BERT را با استفاده از یادگیری انتقالی، برای یافتن نظرات سمی/غیر سمی روی دادگان مسئله تنظیم [۱] کردیم. برای تنظیم، هیچ یک از لایه‌های BERT ثابت فرض نشد و همه لایه‌ها روی دادگان مسئله تنظیم شدند. مدل‌هایی که ساخته و بررسی شدند از نظر مفهومی به دو دسته کلی تقسیم می‌شوند: ۱. یادگیری توالی [۲] ۲.استخراج ویژگی [۳]. یادگیری توالی مبتنی بر شبکه عصبی تکرارشونده LSTM و استخراج ویژگی به کمک شبکه عصبی پیچشی [۴] یک بُعدی انجام شد. ما دو مدل مبتنی بر یادگیری توالی پیشنهاد می‌دهیم که علاوه بر LSTM از یک لایه CNN نیز استفاده می‌کند تا هم یادگیری توالی و هم استخراج ویژگی را داشته باشیم. دو مدل دیگر پیشنهادی ما فقط مبتنی بر مفهوم استخراج ویژگی ساخته شده‌اند و تنها از CNN استفاده می‌کنند. بهترین مدل به دست آمده BERT+2CNN-1D نام دارد که دو لایه CNN یک بُعدی به BERT اضافه می‌کند. این مدل، مصرف حافظه معقول و روی دادگان تست دقت بالایی دارد.

۲ مقدمه

مردم روزانه از شبکه‌های اجتماعی برای به اشتراک گذاری نظرات، تبادل اطلاعات، یافتن دوست و... استفاده می‌کنند. اما به اشتراک گذاری برخی نظرات که آن‌ها را **نظرات سمی** می‌نامیم، شبکه‌های اجتماعی را به بستری برای فعالیت‌های تنفرآمیز تبدیل می‌کند. نظرات سمی شامل نظرات توهین‌آمیز، غیر محترمانه و یا غیر منطقی می‌شوند که کوچک‌ترین پیامد ناشی از به اشتراک‌گذاری آن‌ها در شبکه‌های اجتماعی، غم‌انگیز کردن کاربران است. از این رو در

*Toxic comment detection using transfer learning

[1] Fine-tune

[2] Sequential learning

[3] Feature extraction

[4] Convolutional neural network (CNN)

طی چند سال گذشته، محققین زیادی روی موضوع تشخیص این نوع نظرات و جلوگیری از انتشار آن‌ها کار کرده‌اند. کوتاهی جملات، مهم‌بودن ترتیب کلمات، شناسایی ارتباطات بین کلمات در فضای پنهان از جمله چالش‌هایی هستند که باعث می‌شوند روش‌های مبتنی بر فراوانی کلمات (مانند TF-IDF) عملکرد خوبی برای حل این مسئله نداشته باشند [۲].

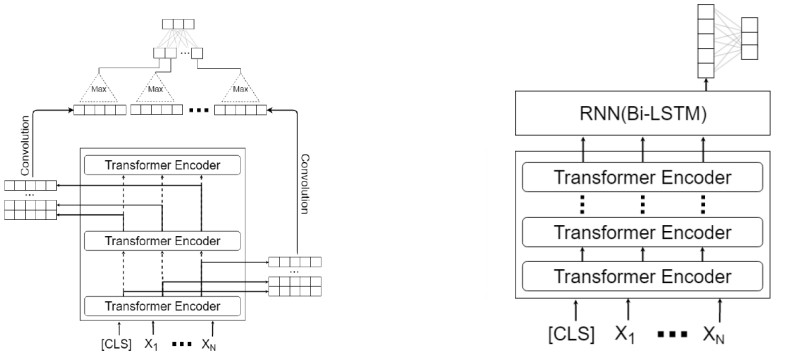
سند D ورودی مسئله و $C = \{۰, ۱\}$ مجموعه برچسب‌هاست که ۰ نشان‌دهنده سمی نبودن و ۱ نشان‌دهنده سمی بودن یک نظر است. مدل، $c \in C$ را به سند D تخصیص می‌دهد. مدل زبانی BERT [۵] می‌تواند انتخاب خوبی برای حل این مسئله باشد. زیرا دادگانی که BERT روی آن از پیش آموزش دیده است شامل برخی متن‌ها و عبارات سمی بوده است [۵]. برای یافتن نظرات سمی، روش‌های متفاوتی با استفاده از پردازش زبان‌های طبیعی، یادگیری ژرف، گراف‌های دانش [۶] ارائه شده است. پرکاربردترین شبکه‌های عصبی برای شناسایی این گونه نظرات CNN و LSTM هستند که به ترتیب به استخراج ویژگی (در اینجا شناسایی عبارات توهین‌آمیز) و یافتن ارتباطات کلمات می‌پردازند. اگرچه هنوز در کشف نظرات سمی عملکرد عمومی‌سازی خوبی ندارند [۶]. در مقاله [۶] استراتژی‌های مختلف تنظیم [۷] BERT روی دادگان مسئله مطرح شده است. یکی از مدل‌های پیشنهادی آنان اضافه کردن یک لایه LSTM به مدل BERT بوده است که طبق نتایج گزارش شده در مقاله، عملکرد خوبی داشته است (نام آن را BERT+LSTM گذاشتند). مدل پیشنهادی دیگری که در [۶] مطرح شده است، خروجی تمام لایه‌های BERT را دریافت می‌کند و به یک لایه CNN دو بُعدی می‌دهد که بهترین مدل معرفی شده در این مقاله است (نام آن را BERT+CNN گذاشتند). ما در این مقاله چهار استراتژی برای تنظیم BERT پیشنهاد می‌دهیم. اولین مدل، خروجی BERT را به یک لایه LSTM-CNN می‌دهد و مدل دوم آن را به یک لایه CNN-LSTM می‌دهد؛ در هر دو مدل از CNN یک بُعدی استفاده شده است؛ این دو مدل جایگزین‌های BERT+LSTM هستند. دو مدل دیگر پیشنهادی ما، به ترتیب یک و دو لایه CNN یک بُعدی به BERT اضافه می‌کنند و ورودی آن‌ها خروجی لایه آخر BERT است؛ در مدل دوم، برای لایه ترکیب [۸] یک هسته [۹] در نظر گفته می‌شود تا بعد از یک لایه شبکه عصبی پیچشی یک بُعدی، همچنان یک ماتریس داشته باشیم؛ این دو مدل جایگزین‌های BERT+CNN هستند. دلیل پیشنهاد دو مدل اول، بهبود نتایج مدل BERT+LSTM است. دو مدل بعدی به دلیل پیچیدگی حافظه کمتر آن‌ها نسبت به مدل BERT+CNN پیشنهاد شده است.

۳ کارهای گذشتگان

در [۱] از شبکه‌های عصبی پیچشی برای تعیین نظرات سمی استفاده شد که تعبیه کلمات [۱۰] مبتنی بر word2vec بوده است. دقت کار آن‌ها روی دادگان تست حدود ۰٫۹۱ گزارش شده است. مدل دیگری مبتنی بر LSTM با تعبیه کلمات SpaCy ارائه شده است که دقت آن حدود ۰٫۹۵ است. اما دادگان استفاده شده، منشتر نشده است و چون از یادگیری انتقالی و مدل‌های از پیش آموزش دیده در آن استفاده نشده است، نمی‌توان به عملکرد خوب عمومی‌سازی مدل مطمئن بود. در [۵] از شبکه‌های CNN و LSTM استفاده شده که تعبیه کلمات آن مبتنی بر FastText و

[5] Bidirectional Encoder Representations from Transformers
[6] Knowledge graphs
[7] fine-tune
[8] Pooling layer
[9] Kernel
[10] Word Embedding

BERT بوده است. دقت هر دو مدل در حدود ۰٫۹۱ بوده و دادگان آن‌ها از پیکره توییتر استخراج شده است. مدل دیگری با اضافه کردن یک لایه LSTM به انتهای BERT پیشنهاد شده است(شکل ۱ قسمت (آ)) [۶]. دقت این مدل ۰٫۹۲ گزارش و پایگاه داده‌ی مورد استفاده آن‌ها از توییتر جمع‌آوری شده است. مدل دیگری مبتنی بر CNN (شکل ۱ قسمت (ب)) برای تنظیم BERT ارائه شده است [۶] که دقت آن حدود ۰٫۹۱ گزارش شده است. اما این مدل، پیچیدگی محاسباتی بالایی دارد و در مرحله پیش‌بینی، حافظه زیادی مصرف می‌کند.



(آ) استراتژی تنظیم BERT با اضافه کردن یک لایه LSTM (BERT+LSTM)

(ب) استراتژی تنظیم BERT با اضافه کردن یک لایه CNN (BERT+CNN)

شکل ۱: معماری‌های پیشنهادی [۶] برای تنظیم BERT

۴ روش پیشنهادی

مدل زبانی BERT در سال ۲۰۱۸ توسط گوگل معرفی شد که یک مدل از پیش آموزش‌دیده روی BookCurpos و Wikipedia Corpora است. از BERT می‌توان در طبقه‌بندی متن، سیستم‌های پرسش و پاسخ و کشف روابط معنایی کلمات استفاده کرد. همانطور که در بخش مقدمه گفته شد، مدل‌هایی مبتنی بر BERT با اضافه کردن یک لایه LSTM و CNN [۱۱] به آن برای حل این مسئله پیشنهاد شده است که نام آن‌ها به ترتیب BERT + LSTM و BERT + CNN است [۶]. ما چهار استراتژی برای تنظیم BERT روی دادگان مسئله پیشنهاد می‌دهیم:

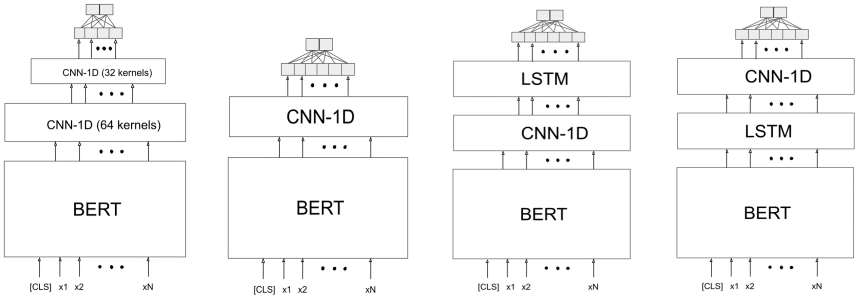
۱. اضافه کردن یک لایه LSTM-CNN به BERT که لایه پیچشی [۱۲] آن یک بُعدی است و جایگزین مدل BERT + LSTM است. نام آن را BERT + LSTM-CNN می‌گذاریم.

۲. اضافه کردن یک لایه CNN-LSTM به انتهای BERT که لایه پیچشی آن یک بُعدی است. در این مدل ابتدا استخراج ویژگی و سپس یادگیری توالی انجام می‌شود. این مدل جایگزین BERT+LSTM است و آن را BERT+CNN-LSTM می‌نامیم.

[۱۱] ما در این مقاله CNN را شبکه عصبی پیچشی ترجمه می‌کنیم. این شبکه از دو لایه تشکیل شده است:۱.لایه پیچشی ۲. لایه ترکیب
[12] Convolutional layer

۳. اضافه کردن یک لایه CNN به BERT که ورودی آن تنها خروجی لایه آخر BERT است و از یک لایه پیچشی و یک لایه ترکیب یک بُعدی تشکیل شده است. این مدل جایگزین مدل BERT+CNN است و آن را BERT+CNN-1D می‌نامیم.

۴. مدل پیشنهادی بعدی یک لایه CNN یک بُعدی دیگر به مدل قبل اضافه می‌کند. اگر اعمال کردن لایه ترکیب به جای کل توالی خروجی لایه پیچشی، روی هسته‌هایی از آن اعمال شود، بعد از یک لایه CNN یک بُعدی، همچنان یک ماتریس خواهیم داشت و می‌توانیم روی آن یک لایه CNN یک بُعدی دیگر اعمال کنیم. این مدل را BERT+2CNN-1D می‌نامیم.



(آ) BERT+LSTM-CNN (ب) BERT+CNN-LSTM (ج) BERT+CNN-1D (د) BERT+2CNN-1D

شکل ۲: مدل‌های پیشنهادی

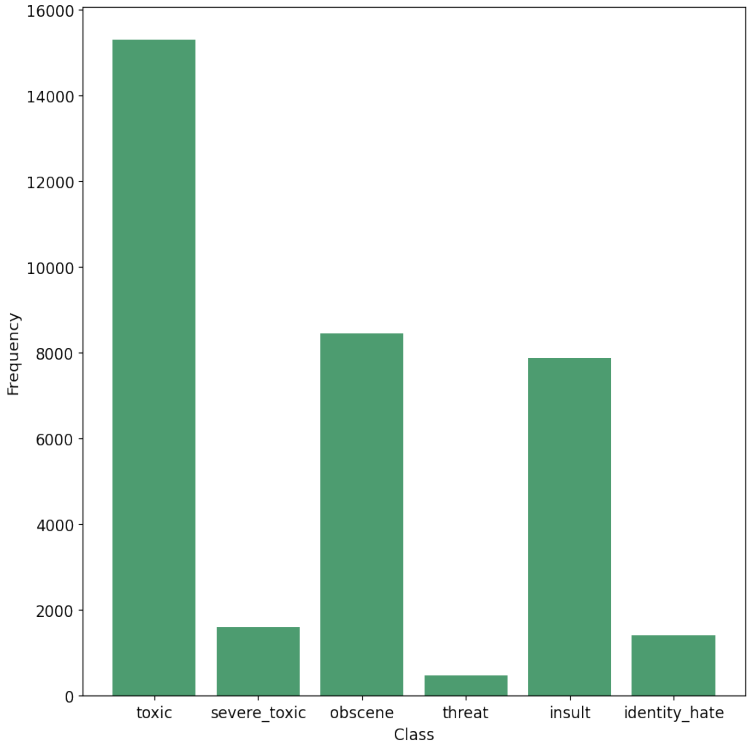
در شکل ۴ در مدل‌های (آ) و (ب) و (ج) تعداد هسته‌های لایه پیچشی ۶۴ است. در مدل‌های ۱ و ۲، بُعد ورودی و خروجی LSTM برابر است.

۵ بخش آزمایشی

۱.۵ دادگان

دادگان مسابقه شناسایی نظرات سمی، برگزار شده توسط گوگل در Kaggle را در نظر گرفته‌ایم که شامل ۱۵۹,۵۷۱ متن نظرات جمع‌آوری شده از توییتر با میانگین ۶۷ کلمه برای جملات است [۴]. برچسب‌های نمونه‌ها به ترتیب toxic، severe_toxic، obscene، threat، insult و identity_hate هستند. اسنادی که یکی از ویژگی‌های قبل برای آنها ۱ بوده است را با برچسب ۱ (سمی) و اسنادی که هیچ یک از ویژگی‌های ذکر شده برای آنها ۱ نبوده با برچسب ۰(غیر سمی) علامت‌گذاری می‌کنیم و مسئله را به یک مسئله طبقه‌بندی دودویی [۱۳] تبدیل می‌کنیم. در اشکال ۳ و ۴ فراوانی کلاس‌ها قبل و بعد از تبدیل مسئله به طبقه‌بندی دودویی قابل مشاهده است. چون بعد از تبدیل مسئله به حالت دودویی، نامتوازن بودن شدیدی در دادگان مشاهده می‌شود، با استفاده از روش under-sampling دادگان را متوزان می‌کنیم. جزئیات این قسمت در بخش ۳.۵ تشریح می‌گردد.

[13] Binary class classification



شکل ۳: فراوانی کلاس‌ها قبل از تبدیل به طبقه‌بندی دودویی

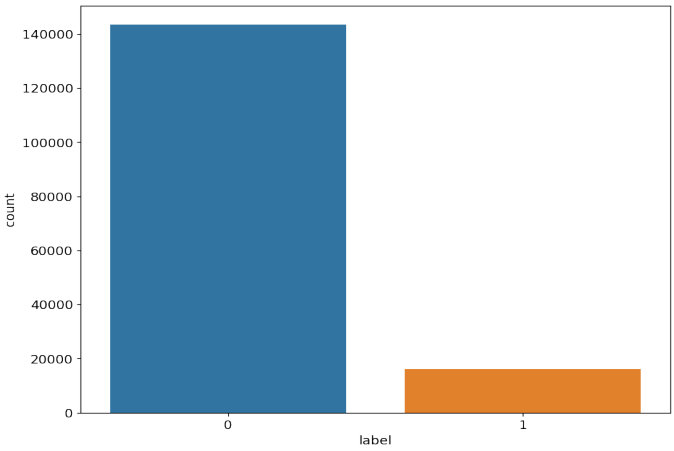
۲.۵ پیش‌پردازش دادگان

کاراکترهای تمام پیام‌ها به حروف کوچک تبدیل شدند. تمام اسامی کاربران، اعداد، هشتگ‌ها، رایانشانی‌ها [۱۴] و علامت‌های احساسات به ترتیب با <user> و <number> و <hashtag> و <url> و <emoticon> جایگزین شدند. تمام کاراکترهای اضافی و uni-code ها از متون حذف شدند. چون مدل BERT توالی کلمات را یاد می‌گیرد، کلمات ایست [۱۵] حذف نشدند و از ساده‌سازی [۱۶] کلمات نیز صرف نظر شد.

۳.۵ پیاده‌سازی و نتایج

در قسمت پیاده‌سازی، برای استفاده از مدل BERT، از کتابخانه transformers و برای شبکه‌های LSTM و CNN از کلاس‌های آماده pyTorch استفاده کردیم. از محیط Google colab به عنوان بستر پیاده‌سازی استفاده شد. توکن‌سازی دادگان به وسیله کلاس BertTokenizer از کتابخانه transformers با حداکثر طول جمله ۳۶

[14] URLs
[15] Stop words
[16] Stemming



شکل ۴: فراوانی کلاس‌ها بعد از تبدیل به طبقه‌بندی دودویی

انجام شد. دادگان با نسبت ۸۰ و ۱۰ و ۱۰ درصد به آموزش، اعتبارسنجی و تست تقسیم شدند. اندازه هسته [۱۷] برای لایه‌های پیچشی و ترکیب ۳ و اندازه گام [۱۸] ۱ انتخاب شد. وزن‌های هر لایه با نسبت ۰٫۱ حذف شدند. مدل در ۳ مرحله [۱۹] با اندازه دسته ۳۲ به وسیله الگوریتم AdamW [۲۰] با نرخ یادگیری ۲×۱۰^{-۵} آموزش داده شد. وزن تمام لایه‌های BERT در تمام لایه‌ها روی دادگان مسئله تنظیم شد. جهت متوازن‌سازی دادگان، به تعداد ۲ برابر تعداد نمونه‌های کلاس یک از کلاس صفر نمونه گرفته شد و سایر پیام‌ها نادیده گرفته شد. با انجام این کار، فراوانی کلاس یک ۱۶,۲۲۵ و فراوانی کلاس صفر ۳۲,۴۵۰ به دست آمد. جزئیات پیاده‌سازی مدل‌ها به شرح زیر است:

۱. **مدل BERT+LSTM-CNN**: از آنجا که حداکثر طول جمله را ۳۶ قرار دادیم؛ خروجی BERT به صورت (۳۶, ۷۶۸) است. این ماتریس به یک لایه LSTM با بُعد خروجی ۳۶ داده می‌شود و بعد از ترانهاده کردن، به یک شبکه پیچشی با تعداد ۶۴ هسته داده می‌شود. خروجی نهایی یک ماتریس با بُعد (۶۴, ۳۴) است که بعد از مسطح‌سازی به یک لایه چگال داده می‌شود.

۲. **مدل BERT+CNN-LSTM**: پیاده‌سازی این مدل مشابه مورد اول است با این تفاوت که ابتدا یک شبکه پیچشی و بعد LSTM اعمال می‌شود. خروجی نهایی، یک ماتریس با بُعد (۶۴, ۳۴) است که بعد از مسطح‌سازی به یک لایه چگال داده می‌شود.

۳. **مدل BERT+CNN-1D**: خروجی BERT به یک شبکه پیچشی با ۶۴ هسته داده می‌شود؛ بُعد خروجی نهایی (۶۴, ۳۴) است. تفاوت این مدل با دو مدل قبل در این است که یادگیری توالی در آن انجام نمی‌شود و فقط یک لایه CNN جهت استخراج ویژگی به کار گرفته می‌شود.

[17] Kernel size
[18] Stride
[19] Epochs
[20] Adam with decoupled weight decay

۴. **مدل BERT+2CNN-1D**: از آنجا که لایه ترکیب را به جای اعمال روی کل توالی ورودی، روی هسته‌هایی با اندازه ۳ از آن با اندازه گام ۱ اعمال می‌کنیم، بعد از یک شبکه پیچشی یک بُعدی با ۶۴ هسته، همچنان یک ماتریس خواهیم داشت که بُعد آن $(۶۴, ۳۴)$ است. این ماتریس به یک شبکه پیچشی دیگر با ۳۲ هسته پیچشی و لایه ترکیب با اندازه ۳ و طول گام ۱ داده می‌شود. خروجی یک ماتریس $(۳۲, ۳۲)$ است که بعد از مسطح‌سازی به یک لایه چگال داده می‌شود.

از آنجا که ارزش هر دو کلاس برابر است، از دو معیار Accuracy و F1 استفاده می‌کنیم که F1 میانگین ماکروی امتیاز F1 برای هر دو کلاس است. در کنار این دو معیار از دو معیار Recall و Precision به عنوان معیارهای ثانوی استفاده می‌کنیم. فرض کنیم $c \in \{۰, ۱\}$ و T_c و F_c به ترتیب، نشان‌دهنده تعداد اعضایی از مجموعه آزمایشی هستند که به درستی و به اشتباه در کلاس c پیش‌بینی شده‌اند و F'_c تعداد اعضایی باشد که برچسب مشاهده شده آن‌ها c بوده اما در کلاس دیگری پیش‌بینی شده‌اند. P_c و R_c و F1_c که به‌ترتیب صحت [۲۱]، دقت [۲۲] و امتیاز F1 کلاس c هستند که به صورت زیر محاسبه می‌شوند:

$$P_c = \frac{T_c}{T_c + F_c},\ R_c = \frac{T_c}{T_c + F'_c},\ \text{F1}_c = ۲ \cdot \frac{P_c R_c}{P_c + R_c}$$

با توجه به فرمول‌های فوق، معیارهای ارزیابی به صورت زیر به دست می‌آیند:

$$\begin{aligned}
\text{Accuracy} &= \frac{T_۰ + T_۱}{T_۰ + T_۱ + F_۰ + F_۱}, \\
\text{F1} &= \frac{\text{F1}_۰ + \text{F1}_۱}{۲}, \\
\text{Precision} &= \frac{P_۰ + P_۱}{۲}, \\
\text{Recall} &= \frac{R_۰ + R_۱}{۲}
\end{aligned}$$

نتایج حاصل شده، در جدول زیر قابل مشاهده است. عملکرد BERT+LSTM-CNN و BERT+CNN-LSTM تقریباً به اندازه BERT+LSTM بوده است که نشان می‌دهد افزودن لایه CNN به مدل اولیه، بهبودی در نتایج حاصل نمی‌کند. مدل‌های بعدی که برای رفع مشکل پیچیدگی حافظه پیشنهاد شده اند تا اندازه خوبی این مشکل را رفع کرده‌اند، در حالی که کماکان دقت آن‌ها به خوبی مدل پیشنهادی [۶] است. مدل BERT+CNN برای طبقه‌بندی حدود ۴,۶۰۰ جمله مجموعه تست، نزدیک به ۶ گیگابایت حافظه نیاز دارد. اما مدل BERT+2CNN-1D با مصرف کمتر از ۲ گیگابایت حافظه، این وظیفه را انجام می‌دهد. به دلیل مصرف حافظه معقول این مدل و دقت بالای آن، تصمیم گرفتیم یک بار دیگر آن را با حداکثر طول جمله ۶۴ آموزش دهیم؛ نتایج ارزیابی این مدل در سطر آخر جدول ۱ آمده است. در این حالت، مصرف حافظه این مدل حدود ۳ گیگابایت بوده که همچنان از نظر پیچیدگی مصرف حافظه، از مدل پیشنهادی [۶] بهینه‌تر است.

[21] Precision
[22] Recall

Methods	Accuracy	F1	Precision	Recall
Naive Bayes [1]	0.72	0.70	0.75	0.70
SVM [1]	0.81	0.80	0.85	0.80
BERT + LSTM [6]	0.91	0.90	0.90	0.91
BERT + CNN-LSTM	0.92	0.91	0.91	0.91
BERT + LSTM-CNN	**0.92**	**0.91**	**0.92**	0.90
BERT + CNN [6]	0.92	0.91	0.90	0.91
BERT + CNN-1D	0.91	0.90	0.91	0.90
BERT + 2CNN-1D	0.92	0.91	0.91	0.91
BERT + 2CNN-1D (max len.=64)	**0.94**	**0.93**	**0.92**	**0.93**

Table 1: The results

مراجع

[1] Georgakopoulos, Spiros V., et al. ”Convolutional neural networks for toxic comment classification.” Proceedings of the 10th hellenic conference on artificial intelligence. 2018.

[2] Quan, Xiaojun, et al. ”Short text similarity based on probabilistic topics.” Knowledge and information systems 25.3 (2010): 473-491.

[3] Dubey, Krishna, et al. ”Toxic Comment Detection using LSTM.” 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC). IEEE, 2020.

[4] https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge?select=train.csv [Accessed: 4-23-2022]

[5] d’Sa, Ashwin Geet, Irina Illina, and Dominique Fohr. ”Bert and fasttext embeddings for automatic detection of toxic speech.” 2020 International Multi-Conference on:“Organization of Knowledge and Advanced Technologies”(OCTA). IEEE, 2020.

[6] Mozafari, Marzieh, Reza Farahbakhsh, and Noel Crespi. ”A BERT-based transfer learning approach for hate speech detection in online social media.” International Conference on Complex Networks and Their Applications. Springer, Cham, 2019.