بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيم



تشخیص متن تولیدشده توسط هوش مصنوعی در زبان فارسی

طراحی معماری چندمدلی، روش‌شناسی یادگیری ماشین و یادگیری عمیق، و تحلیل تعمیم‌پذیری

استاد راهنما

دکتر مرضیه ورپشتی

نگارنده

احسان ترابی فارسانی

مهر ۱۴۰۵

تشکر و قدردانی

از استاد راهنمای محترم، سرکار خانم دکتر مرضیه ورپشتی، که با راهنمایی‌های علمی و دیدگاه‌های ارزشمند خود مسیر اجرای این پژوهش را روشن ساختند، صمیمانه سپاس‌گزارم. همچنین از پژوهشگران و توسعه‌دهندگان جامعهٔ متن‌باز — به‌ویژه سازندگان کتابخانه‌های scikit-learn، PyTorch، Transformers و shekar، و گروه پژوهشی HooshvareLab که مدل‌های زبانی پایهٔ فارسی را در اختیار جامعهٔ علمی قرار دادند — قدردانی می‌کنم.

فهرست مطالب

تشکر و قدردانی آ

چکیده د

۱ مقدمه ۱

۱.۱ انگیزه و اهمیت مسئله ۱

۲.۱ چالش‌های ویژهٔ زبان فارسی ۲

۳.۱ معماری کلی سامانه ۲

۴.۱ ساختار گزارش ۳

۲ داده‌ها ۵

۱.۲ گردآوری داده‌ها در نسخهٔ نخست ۵

۲.۲ گسترش چنددامنه‌ای و پالایش داده در نسخهٔ دوم ۶

۳.۲ تفکیک گروه‌آگاه داده‌ها ۸

۴.۲ مجموعهٔ حملهٔ بازنویسی ۱۰

۳ پیش‌پردازش و استخراج ویژگی ۱۲

۱.۳ دو خط‌لولهٔ موازی پیش‌پردازش ۱۲

۲.۳ بازنمایی متون با TF-IDF و بهینه‌سازی فضای ویژگی ۱۳

۳.۳ ویژگی‌های سبک‌شناسی محاسباتی ۱۴

۴ مدل‌های یادگیری ماشین کلاسیک ۱۶

۱.۴ فرمولاسیون مدل‌ها و تنظیم ابرپارامترها ۱۶

۲.۴ نتایج ارزیابی روی تفکیک مرجع ۱۷

۵ مدل‌های یادگیری عمیق آموزش‌دیده از صفر ۱۹
۱.۵ زیرساخت مشترک شبکه‌های واژگانی ۱۹
۲.۵ شبکهٔ بازگشتی دوسویه (BiLSTM) ۱۹
۳.۵ شبکهٔ کانولوشنی TextCNN و شبکهٔ Attention-BiLSTM ۲۱
۴.۵ شبکه‌های عصبی تکمیلی: MLP، CharCNN و Transformer کوچک . . . ۲۲

۶ مدل‌های زبانی پیش‌آموزش‌دیده و لایهٔ تصمیم ۲۴
۱.۶ تنظیم ظریف مدل زبانی GPT-۲ فارسی ۲۴
۲.۶ تنظیم ظریف مدل ParsBERT ۲۵
۳.۶ تحلیل شاخص سرگشتگی (Perplexity) بدون آموزش ۲۵
۴.۶ لایهٔ تصمیم‌گیری ساختاریافته (System-One) ۲۶
۵.۶ تنظیم ظریف مدل تصمیم Laya با الگوریتم RLCD ۲۸
۶.۶ کاربردهای یادگیری تقویتی در لایهٔ تصمیم و آشکارساز ۲۸
۷.۶ جمع‌بندی مدل‌های مستقر در سامانه ۲۹

۷ معماری نرم‌افزاری و استقرار سامانه ۳۱
۱.۷ معماری سرویس تشخیص و مسیردهی پویا ۳۱
۲.۷ جریان پردازش درخواست و تضمین کیفیت ۳۲
۳.۷ رابط کاربری وب و مقایسه با سرویس ابری TypeSafe ۳۲

۸ ارزیابی سخت‌گیرانه و تحلیل خطا ۳۴
۱.۸ پروتکل ارزیابی، کالیبراسیون و سیاست تصمیم‌گیری ۳۴
۲.۸ نتایج ارزیابی درون‌دامنه‌ای گروه‌آگاه ۳۵
۳.۸ ارزیابی تعمیم‌پذیری: تولیدکننده‌ها، دامنه‌ها و سبک‌های جدید ۳۶
۴.۸ ارزیابی مقاومت در برابر حملهٔ بازنویسی ۳۷
۵.۸ ارزیابی در حالت سه‌کلاسه ۳۹
۶.۸ تحلیل کمّی و کیفی خطاها ۳۹
۷.۸ محدودیت‌های پژوهش ۴۰

۹ نتیجه‌گیری و پیشنهادها ۴۲
۱.۹ جمع‌بندی دستاوردها و یافته‌های کلیدی ۴۲

۲.۹ پیشنهادها برای کارهای آینده ۴۳

آ راهنمای اجرا و بازتولید نتایج **۴۴**

آ.۱ پیش‌نیازهای نرم‌افزاری و سخت‌افزاری ۴۴

آ.۲ زنجیرهٔ بازتولید با نوت‌بوک‌های شش‌گانه ۴۴

آ.۳ راه‌اندازی سرور تشخیص، رابط وب و آزمون‌ها ۴۷

آ.۴ اجرای آموزش و ارزیابی کامل روی Colab Google ۴۷

مراجع **۴۹**

چکیده

این گزارش به طراحی، پیاده‌سازی و ارزیابی سامانه‌ای جامع برای تشخیص متون تولیدشده توسط هوش مصنوعی در زبان فارسی می‌پردازد. هستهٔ تحلیلی سامانه شامل شانزده مدل در پنج خانوادهٔ الگوریتمی است: (۱) پنج مدل یادگیری ماشین کلاسیک مبتنی بر بازنمایی TF-IDF و ده ویژگی سبک‌شناسی محاسباتی؛ (۲) شش شبکهٔ عصبی آموزش‌دیده از صفر شامل BiLSTM پایه، TextCNN، Attention-BiLSTM، پرسپترون چندلایه (MLP)، شبکهٔ نویسه‌سطح CharCNN و رمزگذار Transformer کوچک؛ (۳) دو مدل ترنسفورمر پیش‌آموزش‌دیدهٔ فارسی (GPT-2 و ParsBERT)؛ (۴) دو مدل تصمیم‌گیری ساختاریافته (System-One و Laya) که سه خروجی نوع‌بندی‌شده شامل برچسب منبع، درجهٔ ماشینی بودن و نیاز به بازبینی انسانی تولید می‌کنند؛ و (۵) رویکردهای مبتنی بر یادگیری تقویتی شامل تنظیم ظریف RLCD و سیاست تصمیم باندیت با الگوریتم REINFORCE. متون ماشینی این پژوهش با استفاده از ۷۰ مدل زبانی در ۵ سبک نگارشی تولید شده‌اند. در نسخهٔ دوم مجموعه‌داده، با افزودن شش دامنهٔ متنی انسانی و حذف سه‌مرحله‌ای نمونه‌های تکراری، پیکره‌ای شامل ۳۳٫۶۸۷ سند در سه کلاس (انسان، ماشین خام، و ماشین بازنویسی‌شده) شکل گرفته است (فصل ۲).

این پژوهش همچنین چالش‌های روش‌شناختی رایج در ارزیابی سامانه‌های تشخیص متن را بررسی و اصلاح می‌کند. در آزمایش‌های اولیه، استفادهٔ انحصاری از سه‌گرم‌های واژگانی بدون سقف ابعاد (فضایی با ۹٫۴ میلیون ویژگی) و نشت موضوعی در تفکیک تصادفی داده‌ها، به امتیازهای ظاهری بالای ۰٫۹۹ اما تعمیم‌پذیری ضعیف در محیط واقعی منجر شده بود. با جداسازی کامل مجموعهٔ اعتبارسنجی، محدودسازی فضای TF-IDF به ۱۵۰٫۰۰۰ ویژگی (یک‌گرم تا سه‌گرم)، طراحی دو خط‌لولهٔ پیش‌پردازش مجزا برای مدل‌های واژگانی و ترنسفورمرها، و تنظیم اصولی ابرپارامترها، مدل‌ها از نو آموزش دیدند. روی تفکیک مرجع، شبکه‌های TextCNN ($F_1 = ۰٫۹۹۰۵$) و Attention-BiLSTM ($F_1 = ۰٫۹۸۸۱$) از BiLSTM پایه (۰٫۹۸۳۱) پیشی گرفتند و مدل ParsBERT با امتیاز ۰٫۹۹۸۳ بالاترین عملکرد درون‌توزیعی را ثبت کرد (فصل‌های ۳ تا ۶).

در فصل ۸، چارچوب ارزیابی گروه‌آگاه (Group-Aware Splitting) معرفی می‌شود که در آن هیچ موضوع یا خوشهٔ معنایی مشترکی میان داده‌های آموزش و آزمون وجود ندارد. پایداری مدل‌ها در

برابر ۳۷ تولیدکنندهٔ دیده‌نشده، دامنه‌های متنی جدید، سبک‌های نگارشی متفاوت و حملات بازنویسی سنجیده شده است. نتایج نشان می‌دهد که مدل پایه در ارزیابی درون‌دامنه‌ایِ گروه‌آگاه به F_1 ماکرو برابر با ۰/۹۳۹۲ و در برابر تولیدکننده‌های دیده‌نشده به ۰/۹۶۵۹ دست می‌یابد. همچنین مدل‌های مبتنی بر واژگان و سبک‌شناسی در برابر حملات بازنویسی سطحی کاملاً مقاوم هستند (افت حداکثر ۰/۰۰۱۱)، در حالی که خط پایهٔ مبتنی بر طول متن دچار افت شدید ۰/۱۹۵۰ می‌شود. در نهایت، با کالیبراسیون احتمالات و تعریف سیاست تصمیم‌گیری دومرحله‌ای، نرخ پرچم‌گذاری نادرست متون انسانی در زیر آستانهٔ ٪۰/۵ مهار شده و موارد مرزی به بازبینی انسانی ارجاع داده می‌شوند.

کلیدواژه‌ها: تشخیص متن هوش مصنوعی؛ پردازش زبان طبیعی فارسی؛ سبک‌شناسی محاسباتی؛ شبکه‌های ترنسفورمر؛ کالیبراسیون احتمال؛ ارزیابی گروه‌آگاه.

فصل ۱

مقدمه

۱.۱ انگیزه و اهمیت مسئله

گسترش سریع مدل‌های زبانی بزرگ (LLMs) و دسترسی همگانی به ابزارهای تولید متن، تشخیص منشأ متون را به یکی از مسائل مهم پردازش زبان طبیعی تبدیل کرده است. امروزه دانشگاه‌ها برای ارزیابی اصالت نوشته‌های علمی، رسانه‌ها برای پالایش محتوا و مدیران سکوهای انتشار متن، نیازمند ابزارهایی هستند که نوشته‌های انسانی را از متون تولیدشده توسط ماشین تفکیک کنند. بااین‌حال، اغلب سامانه‌های تشخیص موجود با تمرکز بر زبان انگلیسی توسعه یافته‌اند و در مواجهه با متون فارسی دچار افت دقت یا نرخ بالای خطای مثبت کاذب می‌شوند. تفاوت‌های ساختاری و صرفی زبان فارسی، چالش‌های خط و نیم‌فاصله، و کمبود مجموعه‌داده‌های مرجع فارسی، عوامل اصلی این شکاف عملکردی هستند.

هدف این پروژه، طراحی و پیاده‌سازی یک سامانهٔ بومی و چندمدلی برای زبان فارسی است؛ سامانه‌ای که طیف متنوعی از الگوریتم‌ها — از ماشین بردار پشتیبان خطی تا ترنسفورمرهای پیش‌آموزش‌دیده — را روی داده‌های متوازن فارسی آموزش می‌دهد، آن‌ها را در شرایط تغییر توزیع (Distribution Shift) می‌آزماید و در قالب یک وب‌سرویس و رابط کاربری در دسترس قرار می‌دهد. در این گزارش، هر معیار عددی در بستر شرایط اندازه‌گیری خود گزارش شده و نقاط ضعف سامانه در مواجهه با متون کوتاه یا محاوره‌ای به‌صورت شفاف تحلیل گردیده است (بخش ۷.۸).

۲.۱ چالش‌های ویژهٔ زبان فارسی

پردازش زبان فارسی در مسئلهٔ تشخیص متن، چهار چالش مشخص دارد که طراحی خط‌لولهٔ پیش‌پردازش و انتخاب معماری مدل‌های این سامانه را شکل داده‌اند:

- **چندگانگی کدگذاری نویسه‌ها:** در متون وب فارسی، اختلاط نویسه‌های عربی و فارسی (مانند

«ي» و «ك» عربی در برابر «ی» و «ک» فارسی) رایج است و بدون یکسان‌سازی اولیه، یک واژهٔ واحد به دو شناسهٔ متفاوت در واژگان مدل نگاشت می‌شود.

- **ناهمگونی در کاربرد نیم‌فاصله (ZWNJ):** افعال و واژه‌های مرکب ممکن است به صورت‌های متفاوتی نوشته شوند (مانند «می‌روم»، «می روم» و «میروم») که مرز توکن‌ها را تغییر داده و نیازمند نرمال‌سازی پیش از توکن‌سازی است.
- **ساختار صرفی و اشتقاقی غنی:** در زبان فارسی، افزودن پیشوندها، پسوندها و شناسه‌های فعلی، صورت‌های ظاهری متعددی از یک ریشهٔ واحد می‌سازد. بدون ریشه‌یابی در مدل‌های کیسه‌واژه (Bag-of-Words)، ابعاد فضای ویژگی متورم و ماتریس داده‌ها تنک (Sparse) می‌شود.
- **محدودیت مدل‌های پایهٔ پیش‌آموزش‌دیده:** در این پژوهش از مدل‌های پیش‌آموزش‌دیدهٔ GPT-2-fa و ParsBERT استفاده شده است که دانش زبانی آن‌ها نقش تعیین‌کننده‌ای در برتری ترنسفورمرها نسبت به شبکه‌های آموزش‌دیده از صفر دارد (بخش ۳.۵).

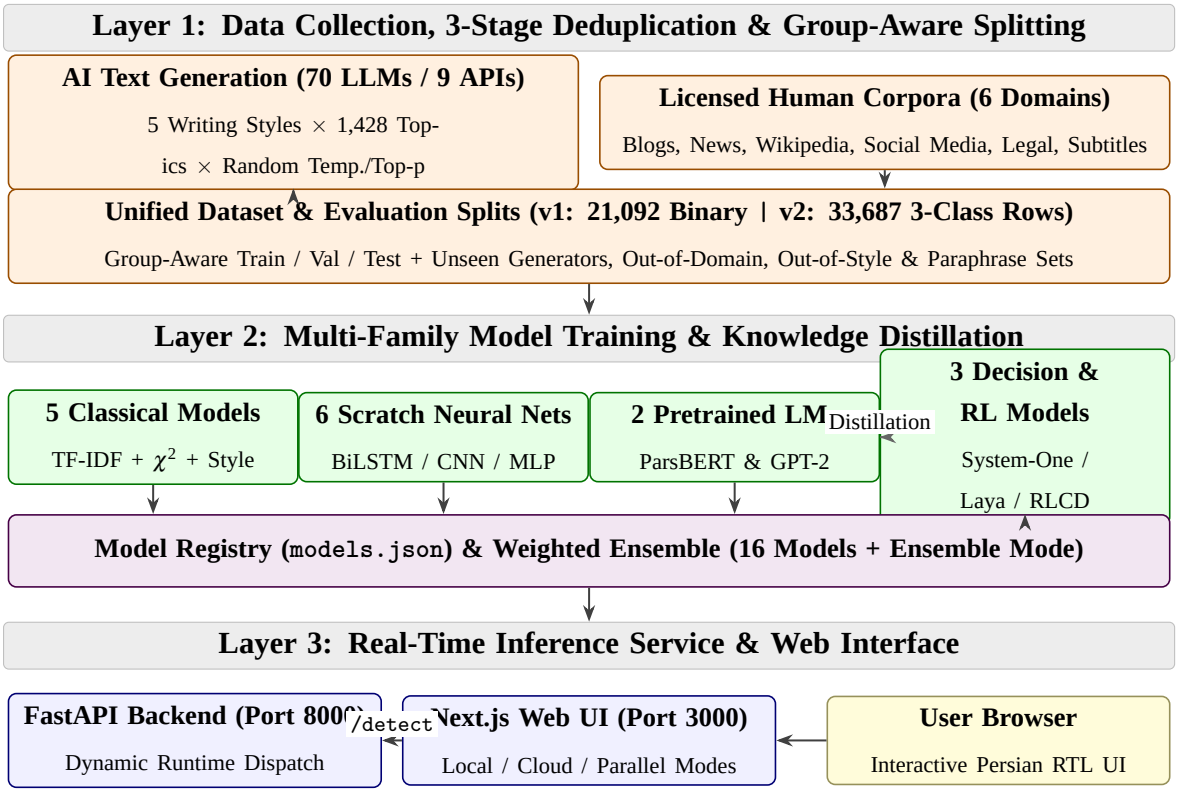
نکتهٔ کلیدی آن است که یک پیش‌پردازش واحد برای تمامی خانواده‌های مدل بهینه نیست. در حالی که ریشه‌یابی (Lemmatization) برای مدل‌های کلاسیک و واژگانی ضروری است، اعمال آن روی ورودی مدل‌های ترنسفورمر زیان‌بار خواهد بود؛ زیرا توکن‌سازهای زیرواژه‌ای (Subword Tokenizers) در ترنسفورمرها بر اساس صورت طبیعی و صرف‌شدهٔ جملات آموزش دیده‌اند. به همین دلیل، در این سامانه دو خط‌لولهٔ پیش‌پردازش مجزا و موازی طراحی شده است (بخش ۱.۳).

۳.۱ معماری کلی سامانه

شکل ۱.۱ معماری کلی سامانه را در سه لایهٔ اصلی نشان می‌دهد: (۱) لایهٔ گردآوری و تفکیک گروه‌آگاه داده‌ها، (۲) لایهٔ آموزش پنج خانوادهٔ الگوریتمی و تجمیع آن‌ها در دفتر ثبت مدل‌ها، و (۳) لایهٔ سرویس‌دهی برخط (FastAPI) و رابط کاربری وب (Next.js).

۴.۱ ساختار گزارش

ادامهٔ این گزارش در هشت فصل و یک پیوست تنظیم شده است: فصل ۲ گردآوری داده‌ها، حذف تکرار و تفکیک گروه‌آگاه را شرح می‌دهد؛ فصل ۳ به دو خط‌لولهٔ پیش‌پردازش، فضای TF-IDF و ویژگی‌های سبک‌شناسی می‌پردازد؛ فصل ۴ مدل‌های یادگیری ماشین کلاسیک و فصل ۵ شبکه‌های



شکل ۱.۱: معماری کلی سه‌لایهٔ سامانه: (۱) گردآوری و تفکیک گروه‌آگاه داده‌ها، (۲) آموزش پنج خانوادهٔ الگوریتمی و تجمیع در دفتر ثبت مدل‌ها، و (۳) سرویس‌دهی برخط و رابط کاربری.

عمیق آموزش‌دیده از صفر را بررسی می‌کنند؛ فصل ۶ تنظیم ظریف ترنسفورمرها، تحلیل سرگشتگی و لایهٔ تصمیم‌گیری سیستم‌ـیک و یادگیری تقویتی را پوشش می‌دهد؛ فصل ۷ معماری نرم‌افزاری و استقرار سرویس را تشریح می‌کند؛ فصل ۸ نتایج ارزیابی سخت‌گیرانه و تحلیل خطاها را گزارش می‌نماید؛ فصل ۹ به جمع‌بندی و کارهای آینده اختصاص دارد و پیوست آ راهنمای بازتولید نتایج است.

فصل ۲

داده‌ها

۱.۲ گردآوری داده‌ها در نسخهٔ نخست

نسخهٔ نخست مجموعه‌داده از ترکیب دو منبع متنی شکل گرفت: متون تولیدشده توسط مدل‌های زبانی بزرگ (کلاس مثبت) و نوشته‌های وبلاگی انسانی (کلاس منفی). برای ایجاد توازن کامل میان دو کلاس، به همان تعداد نمونه‌های ماشینی، از پیکرهٔ وبلاگ‌های فارسی به‌صورت تصادفی نمونه‌برداری شد تا مجموعه‌داده‌ای متوازن شامل ۲۱,۰۹۲ سند (۱۰,۵۴۶ نمونه در هر کلاس) شکل گیرد.

در طراحی خط‌لولهٔ تولید متون ماشینی، اصل بنیادین «تنوع حداکثری» بوده است تا مدل تشخیص‌دهنده به جای حفظ کردن لحن یک مدل خاص، ویژگی‌های عمومی نوشتار ماشینی را بیاموزد. برای تحقق این هدف، تنوع در چهار محور اعمال شد:

۱. **تنوع مدل تولیدکننده:** درخواست‌های تولید متن به‌طور ناهمگام به ۹ درگاه API ارسال شدند که در مجموع ۷۰ مدل زبانی از خانواده‌های GPT، Claude، Gemini، Gemma، DeepSeek، Llama و Qwen را پوشش می‌دادند.

۲. **تنوع سبک نگارش:** هر درخواست با یکی از پنج دستورالعمل سیستمی شامل «مقالهٔ رسمی»، «پست وبلاگی محاوره‌ای»، «پاسخ آموزشی گام‌به‌گام»، «متن تحلیلی‌ـانتقادی» و «خلاصهٔ ساختاریافته» همراه شد.

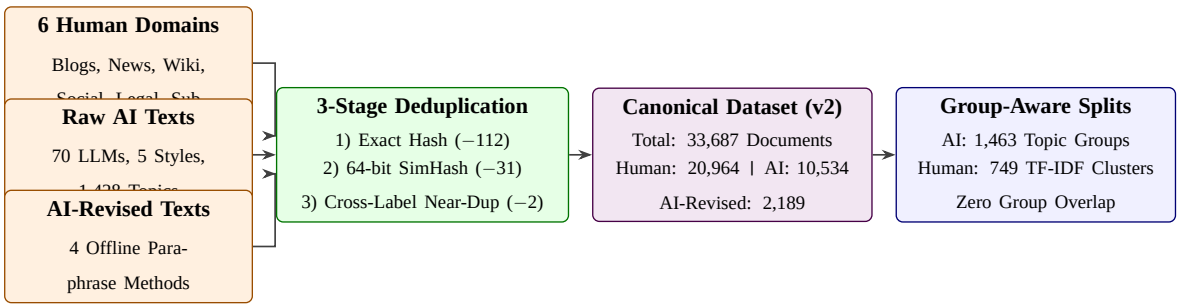
۳. **تنوع موضوعی:** موضوع هر متن از میان ۱,۴۲۸ عنوان یکتای فارسی انتخاب گردید.

۴. **تنوع پارامترهای نمونه‌برداری:** برای هر درخواست، پارامتر دما (Temperature) از بازهٔ یکنواخت [۰٫۳۰, ۰٫۹۵] و پارامتر Top-p از بازهٔ [۰٫۸۰, ۱٫۰۰] نمونه‌برداری شدند تا خروجی‌ها طیف گسترده‌ای از پراکندگی واژگانی را پوشش دهند.

این مجموعه‌داده به همراه فرادادهٔ کامل هر سند با مجوز متن‌باز Apache 2.0 در مخزن Hugging Face Datasets منتشر شده است [۱۴].

۲.۲ گسترش چنددامنه‌ای و پالایش داده در نسخهٔ دوم

بررسی دقیق نسخهٔ نخست نشان داد که چون تمامی متون انسانی آن از یک دامنه (وبلاگ‌ها) گردآوری شده‌اند، خطر آن وجود دارد که طبقه‌بند به‌جای یادگیری مرز واقعی «انسان در برابر ماشین»، ویژگی‌های سبکی «وبلاگ در برابر غیروبلاگ» را بیاموزد. برای رفع این کاستی، در نسخهٔ دوم (v2) چارچوب داده از پایه بازطراحی شد (شکل ۱.۲). در این نسخه، بخش انسانی به **شش دامنهٔ متنی متنوع** گسترش یافت (جدول ۱.۲): وبلاگ‌ها، اخبار (ایسنا)، دانشنامه (ویکی‌پدیای فارسی)، شبکهٔ اجتماعی (توییتر فارسی)، متون اداری و حقوقی، و زیرنویس‌های ترجمه‌شده. همچنین طرح برچسب‌گذاری به **سه‌کلاسه** ارتقا یافت: نوشتهٔ انسان (`human`)، تولید مستقیم ماشین (`ai`) و متن ماشینیِ بازنویسی‌شده (`ai_revised`).



شکل ۱.۲: خط‌لولهٔ ساخت داده در نسخهٔ دوم: تجمیع شش دامنهٔ انسانی، متون ماشینی خام و متون بازنویسی‌شده، عبور از صافی حذف تکرار سه‌مرحله‌ای و تولید تفکیک‌های گروه‌آگاه.

جدول ۱.۲: دامنه‌های انسانی: منبع، مجوز و شمار ردیف‌های حاضر در قاب داده (شناسهٔ کامل منابع و درهم‌ساز فایل‌ها در data/human/provenance.json)

دامنه	منبع	مجوز	اصل	ردیف
وبلاگ	پیکرهٔ وبلاگ پروژه	پیکرهٔ پروژه (بدون مجوز بیرونی)	فارسی	۹,۷۹۶
خبر	isna-news	apache-2.0	فارسی	۲,۹۶۳
دانشنامه	wikipedia:20231101.fa	cc-by-sa-4.0	فارسی	۲,۹۹۹
شبکهٔ اجتماعی	PersianTwitterDataset	odbl	فارسی	۲,۲۰۶
اداری و حقوقی	iran-legal-persian-qa	cc	فارسی	۲,۹۳۷
زیرنویس ترجمه‌ای	persian-subtitles	cc-by-4.0	ترجمه‌ای	۶۳

پیش از تفکیک داده‌ها، عملیات **حذف تکرار در سه سطح** اجرا شد: (۱) حذف تکرار دقیق بر پایهٔ تطابق چکیدهٔ رمزی متن؛ (۲) حذف تکرار نزدیک (Near-Duplicate) با محاسبهٔ اثر انگشت ۶۴ بیتی SimHash روی کلوخه‌های چهارواژه‌ای و حذف اسناد با فاصلهٔ همینگ حداکثر ۳؛ و (۳) حذف تشابه میان‌برچسبی (Cross-Label) برای حذف اسنادی که میان دو کلاس متضاد هم‌پوشانی واژگانی بسیار نزدیکی دارند. در این فرآیند، از مجموع ۳۳٬۸۳۲ ردیف اولیه، تعداد ۱۱۲ تکرار دقیق، ۳۱ تکرار نزدیک و ۲ نمونهٔ مشابه میان‌برچسبی حذف شدند و پیکرهٔ نهایی شامل ۳۳٬۶۸۷ سند (۲۰٬۹۶۴ سند انسانی، ۱۰٬۵۳۴ سند ماشینی خام و ۲٬۱۸۹ سند بازنویسی‌شده) شکل گرفت (جدول‌های ۲.۲ و ۳.۲).

جدول ۲.۲: ترکیب قاب دادهٔ نسخهٔ دوم (پس از حذف تکراری‌ها)

برچسب	ردیف	سهم
انسان	۲۰٬۹۶۴	۶۲٫۲٪
ماشین (خام)	۱۰٬۵۳۴	۳۱٫۳٪
بازنویسی‌شده	۲٬۱۸۹	۶٫۵٪
مجموع	۳۳٬۶۸۷	۱۰۰٫۰٪

جدول ۳.۲: ترکیب طول، فرمالیته و اصلِ زبان در قاب دادهٔ نسخهٔ دوم

سنجه	مقدار	ردیف	سهم
طول	بلند (≥ ۵۰۰ واژه)	۱۴٬۵۸۸	۴۳٫۳٪
طول	متوسط	۱۲٬۱۳۹	۳۶٫۰٪
طول	کوتاه (≤ ۱۰۰ واژه)	۶٬۹۶۰	۲۰٫۷٪
فرمالیته	خنثی	۲۵٬۲۵۶	۷۵٫۰٪
فرمالیته	خودمانی	۴٬۷۵۰	۱۴٫۱٪
فرمالیته	رسمی	۳٬۶۸۱	۱۰٫۹٪
اصلِ زبان	فارسی	۳۳٬۶۲۴	۹۹٫۸٪
اصلِ زبان	ترجمه‌ای	۶۳	۰٫۲٪

در تحلیل این پیکره، دو نکته مد نظر قرار گرفته است: نخست آنکه دامنهٔ زیرنویس‌ها ماهیتاً متونی ترجمه‌شده به فارسی هستند و حجم آن‌ها تنها ۶۳ سند است؛ بنابراین این زیرمجموعه صرفاً به‌عنوان

یک محک چالشی برای سنجش نرخ خطای مثبت کاذب به کار می‌رود. دوم آنکه در داده‌های شبکهٔ اجتماعی (توییتر)، نام‌های کاربری و ارجاعات با عبارات منظم پاک‌سازی شده‌اند.

۳.۲ تفکیک گروه‌آگاه داده‌ها

در تفکیک تصادفی در سطح تک‌سند، اسناد مختلفی که دربارهٔ یک موضوع واحد تولید شده‌اند می‌توانند هم در بخش آموزش و هم در بخش آزمون قرار گیرند و باعث نشت موضوعی شوند. برای بستن این راه نشت، در نسخهٔ دوم از **تفکیک گروه‌آگاه (Group-Aware Splitting)** استفاده شده است تا تمامی اسناد هم‌گروه یکجا در بخش آموزش یا در بخش آزمون قرار گیرند. در سمت متون ماشینی، گروه همان موضوع پرامپت است (۱,۴۶۳ گروه موضوعی) و در سمت متون انسانی، از خوشه‌بندی TF-IDF متون برای تشکیل ۷۴۹ گروه معنایی استفاده شده است. تفکیک اصلی داده‌ها شامل ۲۳,۵۶۴ سند آموزش، ۳,۴۷۸ سند اعتبارسنجی و ۶,۶۴۵ سند آزمون است (جدول ۴.۲).

جدول ۴.۲: شرط‌های ارزیابی و اندازهٔ بخش‌های هر شرط (ردیف)

شرط	آموزش	اعتبارسنجی	آزمون
درون‌دامنه (گروه‌آگاه)	۲۳,۵۶۴	۳,۴۷۸	۶,۶۴۵
برون‌تولیدکننده	۲۰,۴۶۹	۳,۰۴۷	۹۴۰
برون‌دامنه: خبر	۱۹,۸۹۶	—	۴,۲۱۴
برون‌دامنه: دانشنامه	۱۹,۸۹۰	—	۴,۲۱۴
برون‌دامنه: شبکهٔ اجتماعی	۲۰,۴۲۵	—	۴,۲۱۴
برون‌دامنه: اداری و حقوقی	۱۹,۹۱۴	—	۴,۲۱۴
برون‌دامنه: وبلاگ	۱۵,۱۶۷	—	۴,۲۱۴
برون‌دامنه: زیرنویس ترجمه‌ای	۲۱,۹۲۶	—	۱۲۶
برون‌سبک: وبلاگِ گفت‌وگویی	۲۰,۵۹۴	—	۷۶۲
برون‌سبک: مقالهٔ رسمی	۲۰,۶۵۰	—	۷۵۸
برون‌سبک: یادداشت انتقادی	۲۰,۶۳۶	—	۸۰۶
برون‌سبک: پرسش و توضیح	۲۰,۰۴۳	—	۱,۰۸۴
برون‌سبک: خلاصهٔ مروری	۲۰,۵۴۹	—	۸۰۴
یادگیری با نمونهٔ کم (۸)	۱۶	—	۶,۶۴۵
یادگیری با نمونهٔ کم (۳۲)	۶۴	—	۶,۶۴۵
یادگیری با نمونهٔ کم (۱۲۸)	۲۵۶	—	۶,۶۴۵
طول: کوتاه (آزمون)	—	—	۱,۳۱۹
طول: بلند (آزمون)	—	—	۳,۳۳۳
آزمون نسخهٔ نخست	—	—	۴,۰۵۶
ذخیره‌شدهٔ حملهٔ بازنویسی	—	—	۳۰۰
انسان در برابر بازنویسی‌شده	۱۶,۱۹۱	۴۴۲	۷۴۶

همهٔ گزاره‌های سلامت در splits_v2.json برقرارند.

علاوه بر تفکیک اصلی، چهار سناریوی تخصصی برای سنجش تعمیم‌پذیری ساخته شده است:

- **تولیدکننده‌های دیده‌نشده (برون‌تولیدکننده):** اسناد آزمون در این برش هم توسط مدل‌های دیده‌نشده (۳۷ تولیدکننده از مجموع ۹۴ مدل) تولید شده‌اند و هم به موضوعات دیده‌نشده

تعلق دارند. با حذف ۱,۵۰۰ ردیف دارای هم‌پوشانی موضوعی، مجموعهٔ آزمونی متوازن شامل ۹۴۰ سند شکل گرفته است.

- **دامنه‌های دیده‌نشده (برون‌دامنه):** در هر نوبت، یکی از شش دامنهٔ انسانی به‌طور کامل از داده‌های آموزش حذف شده و مدل روی اسناد همان دامنهٔ حذف‌شده آزمایش می‌شود.
- **سبک‌های دیده‌نشده (برون‌سبک):** در هر نوبت یکی از پنج سبک نگارش ماشینی از آموزش کنار گذاشته می‌شود.
- **کمبود دادهٔ برچسب‌دار (Few-Shot):** آموزش مدل‌ها تنها با ۸، ۳۲ و ۱۲۸ سند به ازای هر کلاس.

در سناریوهای برون‌تولیدکننده، برون‌دامنه و برون‌سبک، ارزیابی به‌صورت دودویی (انسان در برابر ماشین خام) انجام می‌شود تا اثر تغییر دامنه یا تولیدکننده با اثر بازنویسی متن در هم نیامیزد. اثر بازنویسی به‌طور مستقل در سناریوی حمله و ارزیابی سه‌کلاسه بررسی شده است (بخش‌های ۴.۲ و ۵.۸).

۴.۲ مجموعهٔ حملهٔ بازنویسی

برای سنجش مقاومت سامانه در برابر بازنویسی متون ماشینی، مجموعهٔ حملهٔ بازنویسی از روی ۳۰۰ سند ماشینیِ کنارگذاشته‌شده ساخته شد. روی هر سند، چهار روش بازنویسی خودکار اعمال گردید: (۱) ترجمهٔ رفت‌وبرگشتی (Round-Trip Translation) بین فارسی و انگلیسی با مدل‌های OPUS-MT؛ (۲) جایگزینی واژگانی با مدل زبانی پوشیده (MLM) توسط ParsBERT؛ (۳) دست‌کاری ساختار و مرز جملات؛ و (۴) تغییرات نویسه‌ای و نیم‌فاصله‌ها. در ارزیابی حمله (بخش ۴.۸)، مقایسه به‌صورت **جفتی (Paired)** انجام می‌شود؛ یعنی عملکرد مدل روی هر سند حمله‌شده دقیقاً با عملکرد همان مدل روی نسخهٔ دست‌نخوردهٔ همان سند مقایسه می‌گردد.

فصل ۳

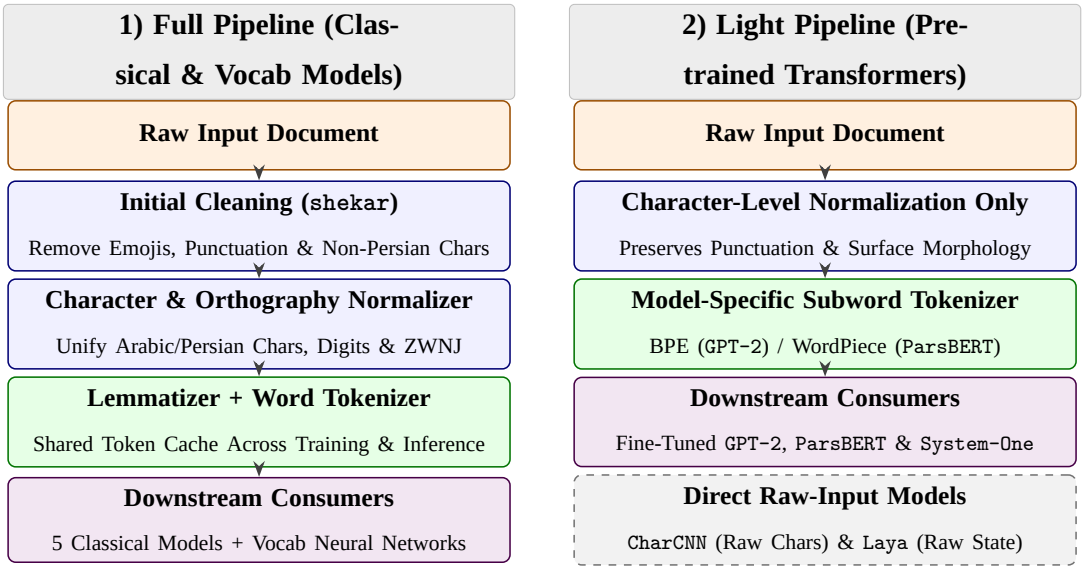
پیش‌پردازش و استخراج ویژگی

۱.۳ دو خط‌لولهٔ موازی پیش‌پردازش

مدل‌های مبتنی بر کیسه‌واژه و شبکه‌های عمیق واژگانی به شدت از ریشه‌یابی (Lemmatization) سود می‌برند، زیرا ریشه‌یابی صورت‌های صرف‌شدهٔ یک کلمه را به یک نماد واحد تبدیل کرده و از پراکندگی آماری واژگان می‌کاهد. در مقابل، مدل‌های ترنسفورمر پیش‌آموزش‌دیده دارای توکن‌سازهای زیرواژه‌ای (BPE یا WordPiece) هستند که روی متون طبیعی آموزش دیده‌اند؛ حذف علائم نگارشی و شکستن افعال به ریشهٔ آن‌ها، جملات را از توزیع طبیعی خارج کرده و کیفیت بردارهای بازنمایی ترنسفورمر را کاهش می‌دهد. بر همین اساس، در این سامانه دو خط‌لولهٔ پیش‌پردازش موازی طراحی شده است (شکل ۱.۳):

۱. **خط‌لولهٔ کامل (ویژهٔ مدل‌های کلاسیک و شبکه‌های واژگانی):** این مسیر بر پایهٔ کتابخانهٔ پردازش زبان فارسی shekar بنا شده و شامل حذف شکلک‌ها، حذف علائم نگارشی، حذف نویسه‌های غیرفارسی، نرمال‌سازی متن (یکسان‌سازی حروف عربی و فارسی، تبدیل ارقام، اصلاح نیم‌فاصله‌ها، حذف اعراب و کوتاه‌سازی حروف تکراری)، ریشه‌یابی واژه‌ها و در نهایت توکن‌سازی واژگانی است.

۲. **خط‌لولهٔ سبک (ویژهٔ مدل‌های ترنسفورمر):** در این مسیر، تنها گام نرمال‌سازی سطح نویسه اعمال می‌شود و علائم نگارشی و ساختار صرفی جملات حفظ می‌گردند. متن نرمال‌شده مستقیماً به توکن‌ساز زیرواژه‌ای اختصاصی هر مدل (BPE برای GPT-2 و WordPiece برای ParsBERT) تحویل داده می‌شود.



شکل ۱.۳: دو خط‌لولهٔ موازی پیش‌پردازش در سامانه: مسیر کامل مبتنی بر ریشه‌یابی برای مدل‌های کیسه‌واژه و واژگانی، و مسیر سبک با حفظ ساختار طبیعی جمله برای مدل‌های ترنسفورمر.

۲.۳ بازنمایی متون با TF-IDF و بهینه‌سازی فضای ویژگی

برای تغذیهٔ مدل‌های یادگیری ماشین کلاسیک، هر سند به یک بردار عددی بر پایهٔ وزن‌دهی TF-IDF نگاشت می‌شود [۱۰]. برای هر عبارت t در سند d از پیکره‌ای شامل N سند، وزن TF-IDF با مقیاس لگاریتمی تکرار واژه (Sublinear TF) مطابق روابط زیر محاسبه می‌شود:

$$\mathrm{tf}_{\mathrm{sub}}(t,d) = ۱ + \ln\big(\mathrm{tf}(t,d)\big), \tag{۱.۳}$$

$$\mathrm{idf}(t) = \ln\left(\frac{۱+N}{۱+\mathrm{df}(t)}\right) + ۱, \tag{۲.۳}$$

$$\mathrm{tfidf}(t,d) = \mathrm{tf}_{\mathrm{sub}}(t,d) \times \mathrm{idf}(t), \tag{۳.۳}$$

که در آن $\mathrm{df}(t)$ تعداد اسنادی است که عبارت t حداقل یک‌بار در آن‌ها ظاهر شده است.

در آزمایش‌های اولیه، بردارساز TF-IDF صرفاً روی سه‌گرم‌های واژگانی و بدون سقف ابعاد تنظیم شده بود که فضایی تنک با ۹,۳۸۵,۶۱۱ بُعد ایجاد می‌کرد و باعث بیش‌برازش روی عبارات نادر می‌شد. در طراحی اصلاح‌شده، بازهٔ nـ‌گرم‌ها به ترکیب یک‌گرم تا سه‌گرم گسترش یافت، عبارات تک‌رخدادی و واژه‌های حاضر در بیش از ۹۰ درصد اسناد حذف شدند و سقف واژگان روی ۱۵۰,۰۰۰ ویژگی پرتکرار تنظیم گردید. همچنین برای مدل‌های درختی، یک مرحله گزینش ویژگی با آزمون آماری کای‌ـدو ($\chi^۲$) اضافه شد تا ۳۰,۰۰۰ ویژگیِ دارای بیشترین همبستگی با برچسب کلاس انتخاب شوند.

۳.۳ ویژگی‌های سبک‌شناسی محاسباتی

برای پوشش ویژگی‌های ساختاری مستقل از موضوع متن، **ده ویژگی سبک‌شناسی (Stylometric Features)** طراحی شد:

۱. **آنتروپی شانون در سطح نویسه:** میزان تنوع کاراکترها نرمال‌شده بر حسب لگاریتم تعداد نویسه‌های یکتای سند؛

۲. **پراکندگی طول جملات:** ضریب تغییرات (CV، نسبت انحراف معیار به میانگین) طول جملات سند بر حسب تعداد واژه؛

۳. **نرخ تکرار سه‌گرم‌ها:** نسبت سه‌گرم‌های واژگانی تکرارشده به کل سه‌گرم‌های یکتای سند؛

۴. **انسجام موضعی میان‌جمله‌ای:** میانگین شباهت کسینوسی بردارهای TF-IDF هر دو جملهٔ متوالی در متن؛

۵. **غنای واژگانی (Type-Token Ratio):** نسبت تعداد واژه‌های یکتا به کل واژه‌های سند؛

۶. **میانگین طول جملهٔ نرمال‌شده:** متوسط تعداد واژه‌های هر جمله، تقسیم بر ۵۰ و محدودشده در بازهٔ $[0, 1]$؛

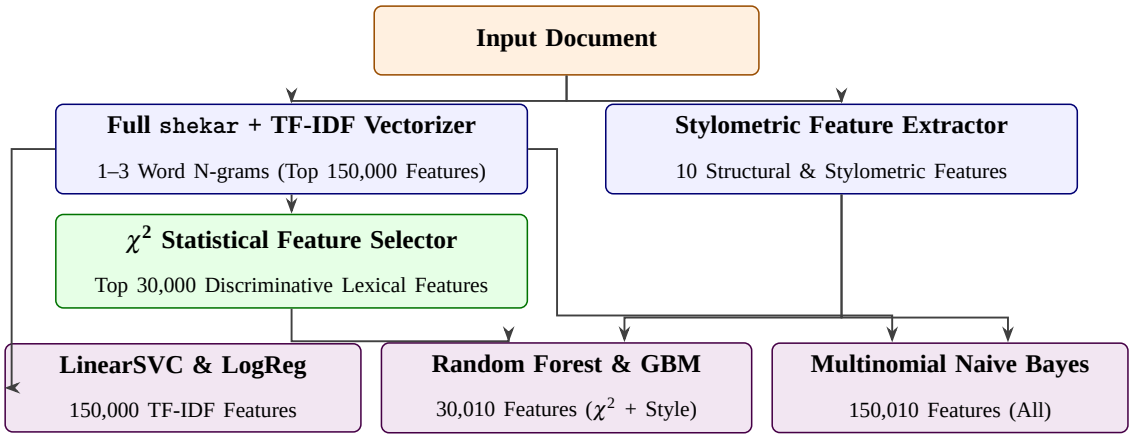
۷. **تراکم واژه‌های کارکردی:** نسبت فراوانی ۳۴ واژهٔ دستوری و ربطی پرتکرار فارسی به کل واژه‌های متن؛

۸. **تراکم علائم نگارشی:** نسبت تعداد علائم سجاوندی به کل نویسه‌ها؛

۹. **تجمع رخداد واژه‌ها (Lexical Burstiness):** ضریب تغییراتِ فاصلهٔ مکانی بین تکرارهای متوالی یک واژه در متن؛

۱۰. **تنوع طول جمله‌واره‌ها (Clauses):** ضریب تغییرات طول قطعات متنی جداشده با ویرگول و نقطه‌ویرگول.

آزمایش‌های حذفی نشان دادند که افزودن این ده ویژگی به بردار ۱۵۰٬۰۰۰ بُعدی در مدل‌های خطی (SVM و رگرسیون لجستیک) بهبودی ایجاد نمی‌کند، اما در مدل جنگل تصادفی ٪۷ و در گرادیان بوستینگ ٪۵۹ از کل اهمیت تصمیم‌گیری بر دوش همین ده ویژگی سبک‌شناسی است. بر این اساس، تخصیص ویژگی‌ها به خانواده‌های مختلف مدل‌های کلاسیک مطابق شکل ۲.۳ انجام شد.



شکل ۲.۳: معماری استخراج ویژگی و تخصیص بهینهٔ ویژگی‌های واژگانی (TF-IDF) و سبک‌شناسی به خانواده‌های مدل‌های کلاسیک.

فصل ۴

مدل‌های یادگیری ماشین کلاسیک

۱.۴ فرمولاسیون مدل‌ها و تنظیم ابرپارامترها

خانوادهٔ مدل‌های کلاسیک شامل پنج الگوریتم یادگیری ماشین است که روی فضای ویژگیِ تشریح‌شده در فصل قبل آموزش می‌بینند:

۱. **ماشین بردار پشتیبان خطی (LinearSVC):** ابرصفحه‌ای را می‌یابد که دو کلاس را با بیشترین حاشیهٔ امنیت (Margin) از هم جدا کند:

$$\min_{w,b} \frac{۱}{۲}\|w\|^۲ + C\sum_{i=۱}^{n} \max\bigl(۰,\ ۱ - y_i(w\cdot x_i + b)\bigr). \qquad (۱.۴)$$

برای استخراج امتیاز شبه‌احتمالی در بازهٔ $[۰,۱]$، فاصلهٔ نمونه از ابرصفحه یعنی $f(x) = w\cdot x + b$ از تابع سیگموید $\sigma(f(x))$ عبور داده می‌شود.

۲. **رگرسیون لجستیک (Logistic Regression):** احتمال تعلق سند به کلاس هوش مصنوعی را با تابع لجستیک مدل‌سازی کرده و با کمینه‌سازی زیان آنتروپی متقاطع همراه با جریمهٔ نرمال‌سازی $\ell_۲$ آموزش می‌بینند.

۳. **بیز سادهٔ چندجمله‌ای (Multinomial Naive Bayes):** با فرض استقلال شرطی ویژگی‌ها در هر کلاس، توزیع احتمال $P(t \mid c)$ را با ابرپارامتر هموارسازی لاپلاس (α) برآورد می‌کند.

۴. **جنگل تصادفی (Random Forest):** میانگین احتمالات پیش‌بینی‌شده توسط مجموعه‌ای از درخت‌های تصمیم مستقل را محاسبه می‌کند [۸].

۵. **گرادیان بوستینگ (Gradient Boosting):** یک مدل افزایشی ترتیبی است که در هر گام، درخت جدیدی را برای تقریب گرادیان منفی تابع زیانِ مرحلهٔ قبل آموزش می‌دهد ($F_m = F_{m-۱} + \eta\, h_m$) [۹].

برای یافتن بهترین پیکربندی هر مدل، از اعتبارسنجی متقاطع (Cross-Validation) صرفاً روی داده‌های آموزش با معیار هدف F_1 استفاده شد: جست‌وجوی تصادفی ۵ـبخشی روی پارامتر C، نوع تابع زیان و وزن کلاس‌ها برای LinearSVC و رگرسیون لجستیک؛ جست‌وجوی شبکه‌ای ۵ـبخشی روی پارامتر α برای بیز ساده؛ و جست‌وجوی تصادفی ۳ـبخشی روی تعداد درخت‌ها، حداکثر عمق، حداقل نمونه در برگ و نرخ یادگیری برای دو مدل درختی.

۲.۴ نتایج ارزیابی روی تفکیک مرجع

تفکیک مرجع در نسخهٔ نخست، یک برش لایه‌بندی‌شده به نسبت ۸۰ به ۲۰ با بذر ثابت ۴۲ است که شامل ۱۶,۸۷۳ سند آموزش و ۴,۲۱۹ سند آزمون با توازن کامل میان دو کلاس است. جدول ۱.۴ نتایج هر پنج مدل را نشان می‌دهد.

جدول ۱.۴: عملکرد پنج مدل یادگیری ماشین کلاسیک روی تفکیک مرجع (۱۶,۸۷۳ سند آموزش / ۴,۲۱۹ سند آزمون).

نام مدل	فضای ویژگی ورودی	دقت	دقت مثبت	بازیابی	F_1
LinearSVC	۱۵۰,۰۰۰ ویژگی TF-IDF	۰/۹۹۶۷	۰/۹۹۸۱	۰/۹۹۵۳	۰/۹۹۶۷
رگرسیون لجستیک	۱۵۰,۰۰۰ ویژگی TF-IDF	۰/۹۹۶۴	۰/۹۹۷۶	۰/۹۹۵۳	۰/۹۹۶۴
گرادیان بوستینگ	۳۰,۰۰۰ ویژگی χ^2 + ۱۰ سبکی	۰/۹۹۱۲	۰/۹۹۲۴	۰/۹۹۰۰	۰/۹۹۱۲
جنگل تصادفی	۳۰,۰۰۰ ویژگی χ^2 + ۱۰ سبکی	۰/۹۸۸۶	۰/۹۹۱۰	۰/۹۸۶۳	۰/۹۸۸۶
بیز سادهٔ چندجمله‌ای	۱۵۰,۰۰۰ ویژگی TF-IDF + ۱۰ سبکی	۰/۹۸۱۰	۰/۹۸۸۵	۰/۹۷۳۵	۰/۹۸۰۹

در این ارزیابی، مدل LinearSVC با امتیاز $F_1 =$ ۰/۹۹۶۷ و رگرسیون لجستیک با $F_1 =$ ۰/۹۹۶۴ بهترین عملکرد را ثبت کردند. مقایسهٔ این ارقام با دور اول آزمایش‌ها نشان می‌دهد که اصلاح فضای ویژگی و تنظیم ابرپارامترها، امتیاز F_1 رگرسیون لجستیک را از ۰/۹۴۰۴ به ۰/۹۹۶۴ و گرادیان بوستینگ را از ۰/۹۳۰۳ به ۰/۹۹۱۲ ارتقا داده است.

فصل ۵

مدل‌های یادگیری عمیق آموزش‌دیده از صفر

۱.۵ زیرساخت مشترک شبکه‌های واژگانی

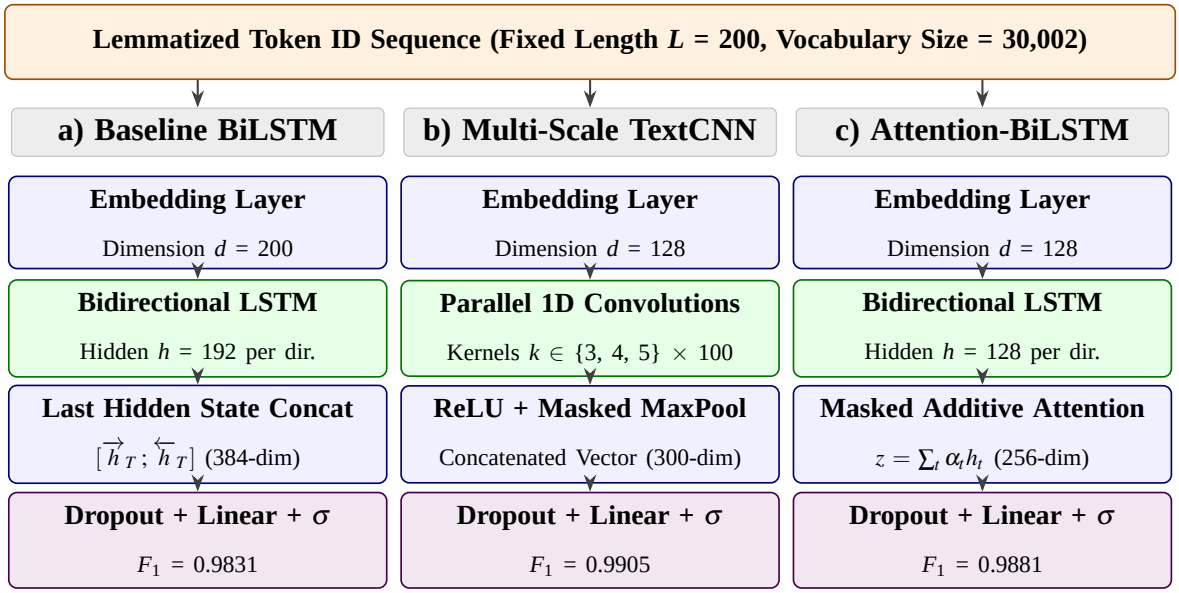
خانوادهٔ دوم مدل‌های این سامانه شامل شبکه‌های عصبی عمیقی است که بدون استفاده از وزن‌های پیش‌آموزش‌دیده، مستقیماً از صفر (From Scratch) آموزش می‌بینند. برای شبکه‌های مبتنی بر واژه، ابتدا از روی خروجی خط‌لولهٔ کامل shekar، واژگانی شامل ریشه‌هایی که حداقل ۲ بار در داده‌های آموزش تکرار شده‌اند با سقف ۳۰,۰۰۰ واژه ساخته می‌شود (که با افزودن دو توکن ویژهٔ <pad> و <unk> مجموعاً دارای ۳۰,۰۰۲ شناسه است). سپس هر سند به توالی ثابتی به طول ۲۰۰ توکن تبدیل می‌شود. تمامی مدل‌های این خانواده با تابع زیان آنتروپی متقاطع دودویی، بهینه‌ساز AdamW، اندازهٔ دستهٔ ۶۴ و توقف زودهنگام بر اساس امتیاز F_1 مجموعهٔ اعتبارسنجی آموزش دیده‌اند. شکل ۱.۵ معماری سه شبکهٔ اصلی واژگانی (BiLSTM پایه، TextCNN و Attention-BiLSTM) را نشان می‌دهد.

۲.۵ شبکهٔ بازگشتی دوسویه (BiLSTM)

مدل پایهٔ این خانواده، یک شبکهٔ حافظهٔ کوتاه‌مدت بلندمدت دوسویه (BiLSTM) [۷] است. در هر گام زمانی t، بردار تعبیهٔ واژهٔ ورودی x_t وارد سلول LSTM شده و وضعیت پنهان h_t و حافظهٔ سلول c_t از طریق سه دروازهٔ کنترلی به‌روزرسانی می‌شوند:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \qquad i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \qquad o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{۱.۵}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, x_t] + b_c), \qquad h_t = o_t \odot \tanh(c_t). \tag{۲.۵}$$



شکل ۱.۵: مقایسهٔ معماری سه شبکهٔ عصبی واژگانی آموزش‌دیده از صفر: (الف) BiLSTM پایه متکی بر آخرین گام زمانی، (ب) TextCNN مبتنی بر فیلترهای کانولوشنی موازی، و (ج) Attention-BiLSTM مجهز به مکانیزم توجه روی کل توالی.

در ساختار دوسویه، دو لایهٔ LSTM توالی را در دو جهت پیمایش می‌کنند و آخرین حالت پنهان هر دو جهت به یکدیگر الحاق می‌شود: $h_{\text{final}} = [\overrightarrow{h}_T ; \overleftarrow{h}_T]$. روی تفکیک مرجع، این مدل به دقت ۰٫۹۸۳۲ و امتیاز $F_1 = ۰٫۹۸۳۱$ دست یافت. تحلیل خطاهای این مدل سه گلوگاه را مشخص کرد: (۱) فقدان دانش زبانی پیش‌آموزش‌دیده در جدول تعبیهٔ واژگان، (۲) حساسیت به متون ماشینی دارای لحن محاوره‌ای، و (۳) گلوگاه خلاصه‌سازی کل سند ۲۰۰ واژه‌ای در آخرین گام زمانی (h_T).

۳.۵ شبکهٔ کانولوشنی TextCNN و شبکهٔ Attention-BiLSTM

برای رفع گلوگاهِ تکیه بر آخرین گام زمانی در BiLSTM پایه، دو معماری جایگزین طراحی و آموزش داده شدند:

- **شبکهٔ کانولوشنی متن (TextCNN):** هر واژه به برداری ۱۲۸ بُعدی نگاشت شده و سه دسته فیلتر کانولوشنی یک‌بعدی با اندازه‌های پنجرهٔ $k \in \{۳, ۴, ۵\}$ (هر کدام شامل ۱۰۰ فیلتر) به‌طور موازی روی متن حرکت می‌کنند تا الگوهای موضعی ۳ تا ۵ کلمه‌ای را استخراج نمایند. پس از اعمال تابع فعال‌ساز ReLU، بیشینه‌گیری سراسری (Global Max-Pooling) همراه با ماسک طول اعمال می‌شود تا برداری ۳۰۰ بُعدی حاصل گردد.
- **شبکهٔ BiLSTM مجهز به مکانیزم توجه (Attention-BiLSTM):** توالی ورودی پس از

عبور از لایهٔ تعبیهٔ ۱۲۸ بُعدی و یک BiLSTM با بُعد پنهان ۱۲۸، توسط یک لایهٔ توجه افزایشی (Additive Attention) در تمامی گام‌های زمانی وزن‌دهی و ترکیب می‌شود:

$$s_t = w_s^\top h_t + b_s, \qquad \alpha_t = \mathrm{softmax}_t(s_t), \qquad z = \sum_{t=1}^{L} \alpha_t h_t. \qquad (۳.۵)$$

جدول ۱.۵ نتایج هر سه مدل واژگانی را روی تفکیک مرجع نشان می‌دهد. هر دو معماری جدید از BiLSTM پایه پیشی گرفته‌اند و مدل TextCNN با امتیاز $F_1 = ۰/۹۹۰۵$ و خطای کالیبراسیون $\mathrm{ECE} = ۰/۰۰۵۷$ بهترین عملکرد را در میان شبکه‌های واژگانی ثبت کرده است.

جدول ۱.۵: مقایسهٔ عملکرد سه شبکهٔ عمیق واژگانی روی تفکیک مرجع (۱۶٫۸۷۳ سند آموزش / ۴٫۲۱۹ سند آزمون).

نام مدل	دقت	دقت مثبت	بازیابی	F_1	ROC-AUC	ECE
BiLSTM پایه	۰/۹۸۳۲	۰/۹۸۹۴	۰/۹۷۶۸	۰/۹۸۳۱	–	–
TextCNN چندمقیاسه	۰/۹۹۰۵	۰/۹۸۹۱	۰/۹۹۱۹	۰/۹۹۰۵	۰/۹۹۹۴	۰/۰۰۵۷
Attention-BiLSTM	۰/۹۸۸۱	۰/۹۹۰۵	۰/۹۸۵۸	۰/۹۸۸۱	۰/۹۹۸۳	۰/۰۰۶۶

۴.۵ شبکه‌های عصبی تکمیلی: MLP، CharCNN و Transformer کوچک

برای پوشش سه سطح بازنمایی متفاوت از متن، سه شبکهٔ دیگر نیز از صفر آموزش داده شدند:

۱. **پرسپترون چندلایه (MLP):** بردار TF-IDF کاهش‌یافته با آزمون χ^2 به ۱۰٬۰۰۰ بُعد را به همراه ده ویژگی سبک‌شناسی دریافت کرده و از دو لایهٔ پنهان (۵۱۲ و ۱۲۸ نورون) عبور می‌دهد.

۲. **شبکهٔ کانولوشنی نویسه‌سطح (CharCNN):** معماری TextCNN را به سطح نویسه می‌برد و با واژگانی شامل ۱٬۰۰۲ نویسه روی پنجره‌های ۵۱۲ کاراکتری از متن خام عمل می‌کند.

۳. **رمزگذار Transformer کوچک (Tiny-Transformer):** یک رمزگذار ترنسفورمر دولایهٔ آموزش‌دیده از صفر (۴ سر توجه، بُعد پیش‌رو ۲۵۶) روی واژگان ۳۰٬۰۰۲ تایی است که با میانگین‌گیری ماسک‌شده خلاصه‌سازی می‌شود.

جدول ۲.۵ عملکرد این سه شبکه را روی تفکیک مرجع نشان می‌دهد. همچنین در یک آزمایش اولیهٔ برون‌دامنه روی ۱٬۲۰۰ سند خبری، مدل‌های TextCNN، Attention-BiLSTM و MLP به دقت

۱٫۰۰۰، مدل Tiny-Transformer به دقت ۰٫۹۹۹ و مدل نویسه‌سطح CharCNN به دقت ۰٫۹۸۰ دست یافتند.

جدول ۲.۵: عملکرد سه شبکهٔ عصبی تکمیلی آموزش‌دیده از صفر روی تفکیک مرجع.

نام مدل	دقت	دقت مثبت	بازیابی	F_1	ROC-AUC	ECE
پرسپترون چندلایه (MLP)	۰٫۹۹۶۴	۰٫۹۹۴۳	۰٫۹۹۸۶	۰٫۹۹۶۵	۰٫۹۹۹۹	۰٫۰۰۳۶
شبکهٔ نویسه‌سطح (CharCNN)	۰٫۹۹۸۳	۱٫۰۰۰۰	۰٫۹۹۶۷	۰٫۹۹۸۳	۱٫۰۰۰۰	۰٫۰۰۲۹
Transformer کوچک	۰٫۹۸۹۱	۰٫۹۹۱۰	۰٫۹۸۷۲	۰٫۹۸۹۱	۰٫۹۹۸۹	۰٫۰۰۵۱

فصل ۶

مدل‌های زبانی پیش‌آموزش‌دیده و لایهٔ تصمیم

۱.۶ تنظیم ظریف مدل زبانی GPT-۲ فارسی

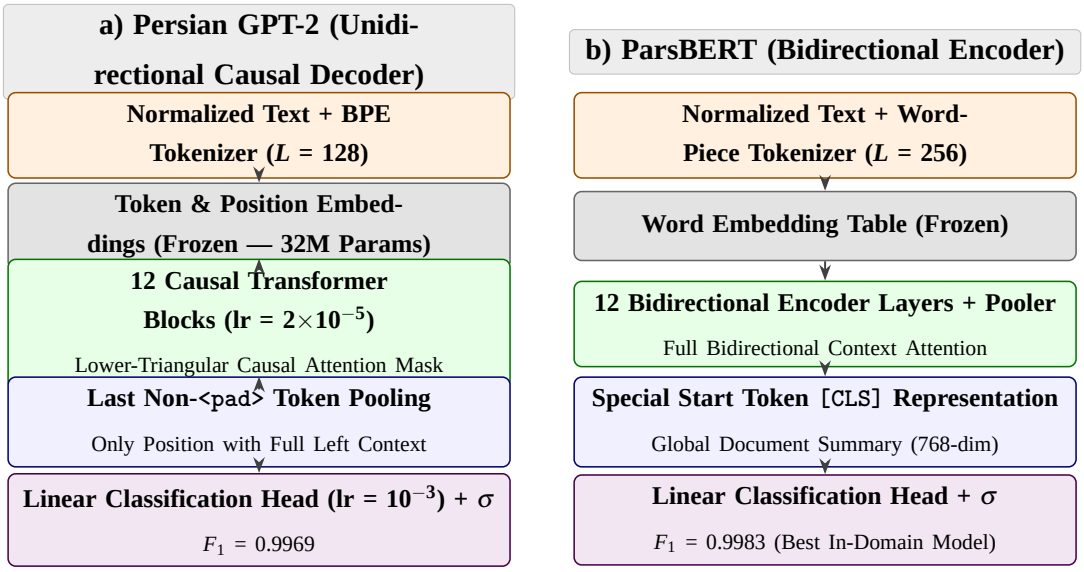
مدل GPT-2 [۶] مبتنی بر بخش رمزگشای (Decoder) معماری ترنسفورمر [۱، ۵] است و از مکانیزم **توجه علّی (Causal Attention)** استفاده می‌کند؛ یعنی در هر موقعیت زمانی، هر توکن تنها مجاز به دیدن توکن‌های پیش از خود است. در این پژوهش از مدل پیش‌آموزش‌دیدهٔ HooshvareLab/gpt2-fa استفاده شده است که دارای ۱۲ لایهٔ ترنسفورمر، بُعد پنهان ۷۶۸، ۱۲ سر توجه و واژگانی شامل ۴۲,۰۰۱ توکن (مجموعاً ۱۲۴ میلیون پارامتر) است [۱۳]. هستهٔ محاسبات در هر سر توجه، ضرب نقطه‌ای مقیاس‌شده است:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V. \tag{۱.۶}$$

همان‌طور که در شکل ۱.۶ (بخش الف) نشان داده شده است، برای تبدیل GPT-2 به یک طبقه‌بند متن، سرِ پیش‌بینی واژه با یک لایهٔ خطی تک‌خروجی جایگزین می‌شود. از آنجا که توجه در GPT-2 یک‌طرفه است، تنها **آخرین توکن واقعیِ هر دنباله** (پیش از توکن‌های لایه‌گذاری <pad>) به تمامی توکن‌های قبلی سند دسترسی داشته است؛ بنابراین بردار حالت پنهان آخرین توکن معتبر استخراج شده و به سر طبقه‌بند داده می‌شود:

$$P(\text{AI} \mid x) = \sigma\left(W_{\text{head}} \cdot h_{\text{last}}^{(۱۲)} + b_{\text{head}}\right). \tag{۲.۶}$$

در فرآیند تنظیم ظریف، لایه‌های تعبیهٔ واژه و موقعیت (۳۲ میلیون پارامتر) منجمد شدند و ۱۲ بلوک ترنسفورمر با استفاده از نرخ یادگیری تفکیکی [۱۱] آموزش دیدند: نرخ کوچک ۲×۱۰^{-۵} برای بدنهٔ پیش‌آموزش‌دیده و نرخ ۱×۱۰^{-۳} برای سر طبقه‌بند. این مدل با طول دنبالهٔ ۱۲۸ توکن روی تفکیک مرجع به دقت و امتیاز $F_۱$ برابر با ۰/۹۹۶۹ دست یافت.



شکل ۱.۶: مقایسهٔ تنظیم ظریف (Fine-Tuning) و استخراج بردار سند در دو معماری ترنسفورمر سامانه: (الف) GPT-2 با توجه علّی و استخراج آخرین توکن معتبر، و (ب) ParsBERT با توجه دوسویه و استخراج توکن [CLS].

۲.۶ تنظیم ظریف مدل ParsBERT

مدل ParsBERT [۱۲] نسخه‌ای از معماری BERT [۲، ۳، ۴] است که روی پیکرهٔ بزرگی از متون فارسی پیش‌آموزش دیده است. برخلاف GPT-2، معماری BERT یک رمزگذار **دوسویه (Bidirectional)** است؛ به این معنا که هر توکن در هر لایه، هم‌زمان به تمامی توکن‌های چپ و راست خود توجه می‌کند. تنظیم ظریف ParsBERT (شکل ۱.۶، بخش ب) با طول دنبالهٔ ۲۵۶ توکن، توکن‌ساز WordPiece و استفاده از بردار خروجی توکن ویژهٔ آغازین [CLS] به‌عنوان نمایندهٔ کل سند انجام شد:

$$P(\mathrm{AI} \mid x) = \sigma\left(W_{\mathrm{head}} \cdot h^{(\mathrm{CLS})} + b_{\mathrm{head}}\right). \qquad (۳.۶)$$

این مدل روی تفکیک مرجع به دقت و امتیاز F_1 برابر با ۰٫۹۹۸۳ (دقت مثبت ۰٫۹۹۸۱ و بازیابی ۰٫۹۹۸۶؛ تنها ۴ مثبت کاذب و ۳ منفی کاذب) دست یافت و بالاترین عملکرد درون‌دامنه‌ای را در میان تمامی مدل‌ها به ثبت رساند.

۳.۶ تحلیل شاخص سرگشتگی (Perplexity) بدون آموزش

یکی از رویکردهای بدون نیاز به آموزش نظارت‌شده، محاسبهٔ **سرگشتگی (Perplexity)** متن تحت یک مدل زبانی مرجع است. سرگشتگی برابر با نماییِ میانگین منفی لگاریتم احتمال توکن‌هاست و میزان غافلگیری مدل زبانی از دیدن جمله‌بندی متن را می‌سنجد. برای آزمودن این فرضیه در زبان فارسی، از

نسخهٔ پایه و تنظیم‌نشدهٔ GPT-2-fa روی ۱٫۵۰۰ سند آزمون استفاده شد. نتایج تجربی نشان داد که مساحت زیر نمودار ROC-AUC برای شاخص سرگشتگی ۰٫۳۳۸۷ است؛ یعنی جهت سیگنال کاملاً وارونه است (میانهٔ سرگشتگی متون ماشینی ۵۰٫۶ و متون انسانی ۳۰٫۷ بود) و حتی با معکوس کردن جهت شاخص نیز AUC از ۰٫۶۶۱۳ و دقت از ۰٫۵۲۶ فراتر نرفت. علت فنی این پدیده، «عدم تقارن میان مدل تولیدکننده و مدل سنجنده» است: متون ماشینی این پیکره توسط ۷۰ مدل زبانی مدرن و بزرگ تولید شده‌اند که از دید یک مدل کوچک ۱۲۴ میلیون پارامتری مانند GPT-2-fa، پیچیده‌تر و غافلگیرکننده‌تر از نوشته‌های وبلاگی به نظر می‌رسند. بر این اساس، شاخص سرگشتگی در سامانه صرفاً به‌عنوان یک معیار تحلیلی جانبی گزارش می‌شود و هیچ وزنی در رأی‌گیری نهایی ندارد.

۴.۶ لایهٔ تصمیم‌گیری ساختاریافته (System-One)

چارچوب تصمیم‌گیری «سیستم‌ـ‌یک» (System-One / JEV) [۱۵، ۱۶] متن ورودی را دریافت کرده و در **یک گذر رو به جلوی واحد (Single Forward Pass)**، به سه پرسش نوع‌بندی‌شده پاسخ‌های احتمالی و ساختاریافته می‌دهد:

۱. **انتخاب منبع (Choice Head):** توزیع احتمال روی دو گزینهٔ «نوشتهٔ انسان» و «تولید هوش مصنوعی»؛

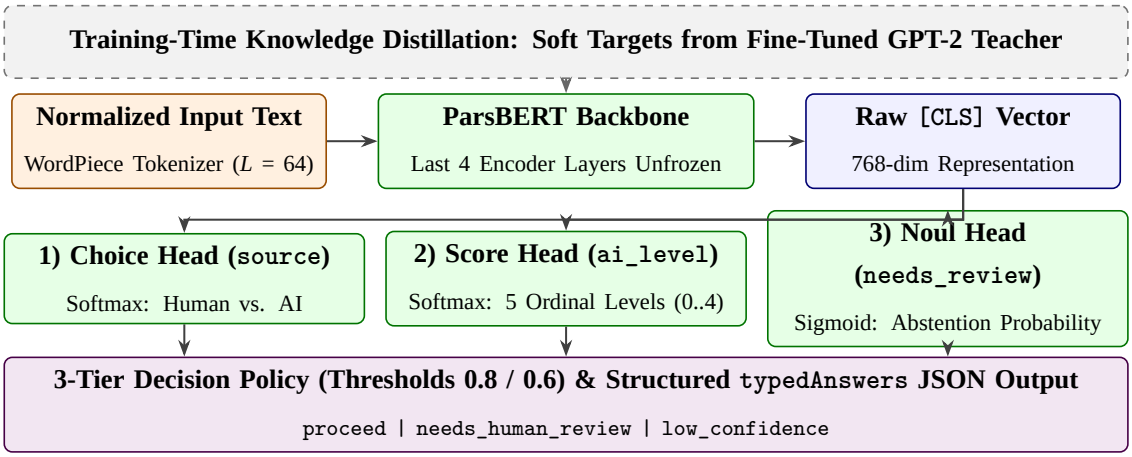
۲. **درجهٔ ماشینی بودن (Score Head):** توزیع احتمال روی یک طیف ترتیبی ۵ سطحی (از ۰ تا ۴) که امید ریاضی آن $\hat{s} = \sum_{k=0}^{4} k\, p_k$ یک امتیاز پیوسته در بازهٔ $[0, 4]$ می‌سازد؛

۳. **نیاز به بازبینی انسانی (Abstention / Noul Head):** یک خروجی احتمالی که مشخص می‌کند آیا نمونهٔ فعلی مبهم است و باید از تصمیم‌گیری خودکار خودداری شود یا خیر.

شکل ۲.۶ معماری این مدل را نشان می‌دهد. ستون فقرات شبکه، رمزگذار ParsBERT است که بردار توکن `[CLS]` آن پس از لایهٔ Dropout به سه سر خطیِ موازی تحویل داده می‌شود:

$$p^{\text{choice}} = \text{softmax}(W_c h), \qquad p^{\text{score}} = \text{softmax}(W_s h), \qquad p^{\text{noul}} = \sigma(W_n h). \quad (۴.۶)$$

برای آموزش این سه سر از روش **تقطیر دانش (Knowledge Distillation)** با استفاده از مدل معلم (GPT-2 تنظیم‌شده) استفاده شد: هدف سر انتخاب از ترکیب خطی برچسب واقعی $y \in \{0, 1\}$ و احتمال معلم به صورت $t_{\text{AI}} = \text{clip}(0.8\, y + 0.2\, p_{\text{teacher}},\ 0.02,\ 0.98)$، هدف سر امتیاز از توزیع گاوسی گسسته‌شده روی ۵ سطح به مرکزیت $4\, t_{\text{AI}}$، و هدف سر خودداری بر اساس حضور معلم در



شکل ۲.۶: معماری لایهٔ تصمیم‌گیری سیستم‌ـیک: استخراج بردار [CLS] از بدنهٔ مشترک ParsBERT و تغذیهٔ هم‌زمان سه سر تصمیم در یک گذر رو به جلو همراه با تقطیر دانش از مدل معلم.

ناحیهٔ تردید ($۰٫۱۵ \leq p_{\text{teacher}} \leq ۰٫۸۵$) تعریف گردید. روی مجموعهٔ آزمون ۱,۵۰۰ نمونه‌ای، این مدل به دقت و امتیاز F_1 برابر با ۰٫۹۸۹۳، میانگین خطای مطلق سر امتیاز برابر با ۰٫۰۴۹۳، خطای کالیبراسیون $\text{ECE} = ۰٫۰۱۵۶$ و مساحت زیر نمودار $\text{AUC} = ۰٫۷۱۹۹$ برای سر خودداری دست یافت.

۵.۶ تنظیم ظریف مدل تصمیم Laya با الگوریتم RLCD

مدل متن‌باز Laya [۱۸] یک معماری تصمیم‌گیری نوع‌بندی‌شده بر پایهٔ رمزگذار دوسویهٔ چندزبانهٔ mmBERT-base (با ۳۲۲ میلیون پارامتر) است. در این معماری، هر گزینهٔ تصمیم‌گیری یک توکن [MASK] اختصاصی در دنبالهٔ ورودی دریافت می‌کند و شبکه با الگوریتم **یادگیری تقویتی برای تصمیم‌گیری‌های کالیبره (RLCD)** بر پایهٔ قواعد نمره‌دهی اکیداً صحیح (Strictly Proper Scoring Rules) آموزش می‌بینند. ارزیابی مدل تنظیم‌شده روی ۱,۵۰۰ نمونهٔ آزمون، دقت و امتیاز F_1 برابر با ۰٫۹۹۹۳، مساحت زیر نمودار $\text{ROC-AUC} = ۱٫۰۰۰۰$ و خطای کالیبراسیون $\text{ECE} = ۰٫۰۰۹۲$ را در سر انتخاب منبع ثبت کرد؛ هرچند خطای مطلق سر امتیاز (۱٫۴۰۱۹) بالاتر از مدل سیستم‌ـیک محلی است و در متون بسیار کوتاه به سمت کلاس انسان سوگیری دارد.

۶.۶ کاربردهای یادگیری تقویتی در لایهٔ تصمیم و آشکارساز

دو رویکرد مکمل یادگیری تقویتی نیز پیاده‌سازی و بررسی شدند:

۱. **سیاست تصمیم باندیت با الگوریتم REINFORCE:** انتخاب میان «تصمیم خودکار» و

«ارجاع به داور انسانی» به صورت یک باندیت متنی روی ۱۱ ویژگی (احتمال مدل مرجع و ده ویژگی سبک‌شناسی) با تابع پاداش حساس به هزینه (تشخیص درست $+۱$، تشخیص نادرست -۳ و ارجاع به بازبینی $-۰/۱$) فرموله شد. روی مجموعهٔ آزمون، میانگین پاداش این سیاست به ۰/۴۲۰۴ رسید که هم‌تراز با سیاست دومرحله‌ای مبتنی بر آستانه‌های ثابت (۰/۴۲۲۱) است.

۲. **تنظیم ظریف RLCD روی شبکهٔ TextCNN:** حلقهٔ یادگیری تقویتی RLCD بر روی چک‌پوینت نظارت‌شدهٔ TextCNN با استفاده از اهداف نرم مدل معلم اجرا شد و روی مجموعهٔ آزمون مرجع به دقت ۰/۹۹۰۰ و ROC-AUC $=$ ۰/۹۹۹۴ دست یافت.

۷.۶ جمع‌بندی مدل‌های مستقر در سامانه

جدول ۱.۶ مشخصات کامل شانزده مدل مستقر در سامانه را به همراه خانوادهٔ الگوریتمی، وزن مشارکت در رأی‌گیری ترکیبی (Ensemble Weight) و معیارهای عملکرد درون‌دامنه‌ای نشان می‌دهد.

جدول ۱.۶: مشخصات شانزده مدل مستقر در سامانه به همراه وزن رأی در مدل ترکیبی و معیارهای عملکرد درون‌دامنه‌ای.

نام و شناسهٔ مدل	خانوادهٔ الگوریتمی	وزن رأی	دقت	$F_۱$
ماشین بردار پشتیبان خطی (svm)	یادگیری ماشین کلاسیک	۱/۰	۰/۹۹۶۷	۰/۹۹۶۷
رگرسیون لجستیک (logreg)	یادگیری ماشین کلاسیک	۱/۰	۰/۹۹۶۴	۰/۹۹۶۴
گرادیان بوستینگ (gradient_boosting)	یادگیری ماشین کلاسیک	۱/۰	۰/۹۹۱۲	۰/۹۹۱۲
جنگل تصادفی (random_forest)	یادگیری ماشین کلاسیک	۱/۰	۰/۹۸۸۶	۰/۹۸۸۶
بیز سادهٔ چندجمله‌ای (naive_bayes)	یادگیری ماشین کلاسیک	۱/۰	۰/۹۸۱۰	۰/۹۸۰۹
شبکهٔ بازگشتی دوسویه (bilstm)	شبکهٔ عمیق واژگانی	۰/۷	۰/۹۸۳۲	۰/۹۸۳۱
شبکهٔ کانولوشنی متن (textcnn)	شبکهٔ عمیق واژگانی	۰/۵	۰/۹۹۰۵	۰/۹۹۰۵
شبکهٔ بازگشتی توجهی (attention_bilstm)	شبکهٔ عمیق واژگانی	۰/۵	۰/۹۸۸۱	۰/۹۸۸۱
پرسپترون چندلایه (mlp)	شبکهٔ عصبی ویژگی‌محور	۰/۵	۰/۹۹۶۴	۰/۹۹۶۵
شبکهٔ کانولوشنی نویسه‌سطح (charcnn)	شبکهٔ عمیق نویسه‌ای	۰/۵	۰/۹۹۸۳	۰/۹۹۸۳
ترنسفورمر کوچک از صفر (tinytransformer)	شبکهٔ عمیق واژگانی	۰/۵	۰/۹۸۹۱	۰/۹۸۹۱
مدل زبانی GPT-2 فارسی (gpt2_transformer)	ترنسفورمر پیش‌آموزش‌دیده	۳/۰	۰/۹۹۶۹	۰/۹۹۶۹
مدل زبانی ParsBERT (parsbert)	ترنسفورمر پیش‌آموزش‌دیده	۳/۰	۰/۹۹۸۳	۰/۹۹۸۳
مدل تصمیم سیستم‌ـیک (system_one)	لایهٔ تصمیم ساختاریافته	۰/۶	۰/۹۸۹۳	۰/۹۸۹۳
مدل تصمیم چندزبانهٔ Laya (laya)	لایهٔ تصمیم ساختاریافته	۲/۰	۰/۹۹۹۳	۰/۹۹۹۳
شبکهٔ TextCNN تقویتی (textcnn_rlcd)	یادگیری تقویتی (RLCD)	۰/۰	۰/۹۹۰۰	۰/۹۹۰۰

فصل ۷

معماری نرم‌افزاری و استقرار سامانه

۱.۷ معماری سرویس تشخیص و مسیردهی پویا

سرویس بک‌اند سامانه با استفاده از چارچوب FastAPI طراحی شده و هستهٔ معماری آن بر الگوی **مسیردهی پویا بر اساس محیط اجرا (Runtime Dispatch)** استوار است. در این الگو، منطق سرور به نام یا تعداد مدل‌ها وابسته نیست؛ بلکه هر مدل در دفتر ثبت (`models.json`) دارای مشخصه‌ای است که محیط اجرای آن (نحوهٔ بارگذاری چک‌پوینت، خط‌لولهٔ پیش‌پردازش مورد نیاز و تابع استنتاج) را تعیین می‌کند.

همان‌طور که در شکل ۱.۷ مشاهده می‌شود، مدل‌های مستقر در چهار مسیر اجرایی اصلی سازمان‌دهی شده‌اند:

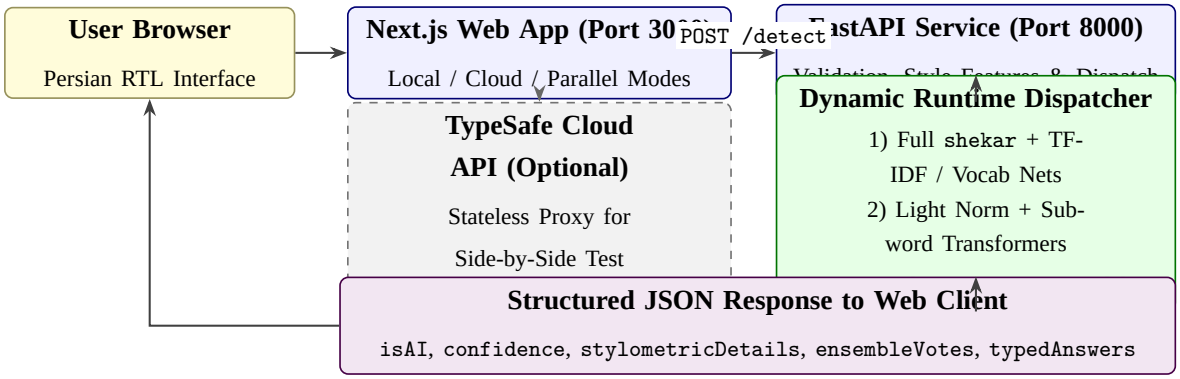
۱. **مسیر خط‌لولهٔ کامل shekar:** شامل مدل‌های کلاسیک مبتنی بر TF-IDF، پرسپترون چندلایه و شبکه‌های عمیق واژگانی (BiLSTM، TextCNN، Attention-BiLSTM و Tiny-Transformer)؛

۲. **مسیر خط‌لولهٔ سبک و توکن‌سازهای زیرواژه‌ای محلی:** شامل مدل‌های GPT-2، ParsBERT و مدل تصمیم System-One؛

۳. **مسیر نویسه‌سطح خام:** ویژهٔ شبکهٔ CharCNN که مستقیماً روی کاراکترهای متن خام عمل می‌کند؛

۴. **مسیر وضعیت خام (Raw State):** ویژهٔ مدل تصمیم چندزبانهٔ Laya.

در لحظهٔ راه‌اندازی سرور، تمامی چک‌پوینت‌ها، بردارسازها و توکن‌سازهای محلی یک‌بار در حافظه بارگذاری و گرم (Warm-up) می‌شوند تا درخواست‌های کاربران بدون تأخیر بارگذاری اولیه و بدون نیاز به اتصال اینترنت پاسخ داده شوند.



شکل ۱.۷: معماری سرویس‌دهی و جریان پردازش درخواست میان مرورگر کاربر، رابط وب Next.js، سرویس استنتاج FastAPI و مسیردهندهٔ پویای مدل‌ها.

۲.۷ جریان پردازش درخواست و تضمین کیفیت

هنگامی که یک متن برای تشخیص به سرور ارسال می‌شود، ابتدا اعتبارسنجی اولیه انجام شده و ده ویژگی سبک‌شناسی به همراه شاخص سرگشتگی به‌صورت مشترک استخراج می‌شوند. سپس بر اساس مدل انتخابی کاربر (یکی از ۱۶ مدل مجزا یا گزینهٔ هفدهم یعنی «مدل ترکیبی»)، متن به مسیر اجرایی مربوطه هدایت شده و خروجی نهایی شامل برچسب تشخیص، درصد اطمینان، جزئیات سبک‌شناختی، آرای تفکیکی مدل‌ها و پاسخ‌های نوع‌بندی‌شدهٔ لایهٔ تصمیم بازگردانده می‌شود.

برای تضمین پایداری نرم‌افزار و جلوگیری از واگرایی میان محیط آموزش و محیط عملیاتی (Training-Serving Skew)، مجموعهٔ آزمون‌های خودکار در دو سطح پیاده‌سازی شده است:

- **آزمون‌های واحد سریع (Unit Tests):** بررسی صحت توابع نرمال‌سازی، فرمول‌های ده‌گانهٔ سبک‌شناسی، ابعاد تنسورها و ساختار پاسخ‌ها در چند ثانیه بدون نیاز به بارگذاری وزن‌های حجیم؛
- **آزمون‌های رگرسیون طلایی (Golden Tests):** بارگذاری کامل تمامی ۱۶ مدل و مقایسهٔ خروجی عددی آن‌ها روی متون مرجع با مقادیر ذخیره‌شدهٔ استاندارد تا اطمینان حاصل شود که هیچ تغییری در کدها یا کتابخانه‌ها باعث تغییر ناخواسته در رفتار مدل‌ها نشده است.

۳.۷ رابط کاربری وب و مقایسه با سرویس ابری TypeSafe

رابط کاربری سامانه با چارچوب Next.js و React به‌صورت کاملاً راست‌به‌چپ طراحی شده است. کاربر می‌تواند مدل مورد نظر خود را از فهرست پویا انتخاب کرده، شناسنامهٔ عملکردی و نمودارهای مقایسه‌ای مدل‌ها را مشاهده نماید و جزئیات سبک‌شناختی و خروجی‌های سه‌گانهٔ لایهٔ تصمیم را بررسی

کند. همچنین رابط وب امکان مقایسهٔ هم‌زمان و موازی موتور محلی را با سرویس ابری تجاری TypeSafe System One فراهم می‌سازد.

برای ارزیابی تجربی سرویس ابری TypeSafe روی زبان فارسی، ۱۰۰ سند از مجموعهٔ آزمون با همان سه پرسش نوع‌بندی‌شده و آستانه‌های یکسان (۰٫۸ و ۰٫۶) آزمایش شدند. نتایج نشان داد که سرویس تجاری TypeSafe از نظر رتبه‌بندی نسبی متون انسانی و ماشینی عملکرد خوبی دارد (ROC-AUC = ۰٫۹۸۰)، اما به دلیل آنکه عمدتاً روی زبان انگلیسی آموزش دیده است، احتمالات خروجی آن روی زبان فارسی به‌شدت بدکالیبره است (ECE = ۰٫۲۵۹ و دقت خام ۰٫۷۰۰). این یافته، ضرورت بومی‌سازی و آموزش مدل‌های تصمیم‌گیری اختصاصی برای زبان فارسی (مدل‌های System-One و Laya در فصل ۶) را به‌خوبی نشان می‌دهد.

فصل ۸

ارزیابی سخت‌گیرانه و تحلیل خطا

۱.۸ پروتکل ارزیابی، کالیبراسیون و سیاست تصمیم‌گیری

در این فصل، عملکرد مدل‌ها بر روی نسخهٔ دوم مجموعه‌داده (۳۳,۶۸۷ سند) و تحت تفکیک‌های گروه‌آگاه (Group-Aware Splits) بررسی می‌شود تا اثر هرگونه نشت موضوعی حذف گردد. در این پروتکل، پنج مدل نماینده از خانواده‌های مختلف به همراه دو خط پایهٔ کنترلی روی تمامی برش‌ها آموزش دیده و با ۳ بذر تصادفی متفاوت (۴۲ تا ۴۴) و ۱,۰۰۰ بار بازنمونه‌گیری بوت‌استرپ لایه‌بندی‌شده ارزیابی می‌شوند:

۱. ماشین بردار پشتیبان خطی روی ویژگی‌های TF-IDF (`tfidf_svm`) — مدل پایهٔ ارزیابی)؛

۲. رگرسیون لجستیک روی همان فضای ویژگی (`tfidf_logreg`)؛

۳. بیز سادهٔ چندجمله‌ای (`tfidf_nb`)؛

۴. جنگل تصادفی صرفاً روی ۲۰ ویژگی سبک‌شناسی (`style_rf`)؛

۵. جنگل تصادفی روی ترکیب ویژگی‌های برتر χ^2 و سبک‌شناسی (`tfidf_style_rf`)؛

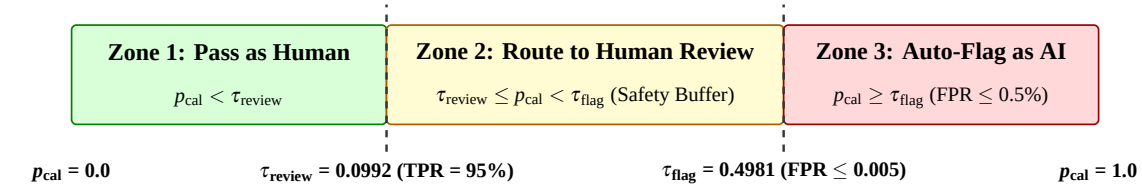
۶. **خط پایهٔ طول متن (`length_only`):** مدلی که تنها تعداد واژه‌ها و نویسه‌های سند را می‌بیند تا مشخص شود چه سهمی از دقت مدل‌ها صرفاً ناشی از تفاوت طول اسناد است؛

۷. **خط پایهٔ سرگشتگی (`perplexity_only`):** شاخص بدون آموزش سرگشتگی با جهت بهینه‌شده روی داده‌های آموزش.

کالیبراسیون احتمال و سیاست تصمیم‌گیری دومرحله‌ای: از آنجا که امتیازهای خام خروجی مدل‌ها الزاماً بیانگر احتمال کالیبره نیستند، روی مجموعهٔ اعتبارسنجی سه روش کالیبراسیون — مقیاس‌دهی

دما (Temperature Scaling)، مقیاس‌دهی پلات (Platt Scaling) و رگرسیون ایزوتونیک — برازش شده و روشی که کمترین زیان لگاریتمی را دارد انتخاب می‌شود. سپس، با توجه به اینکه در کاربردهای عملی **هزینهٔ اتهام نادرست به نویسندهٔ انسانی (خطای مثبت کاذب) بسیار سنگین‌تر از تشخیص ندادن یک متن ماشینی است**، یک سیاست تصمیم‌گیری دومرحله‌ای با دو آستانه روی مجموعهٔ اعتبارسنجی تعریف می‌شود (شکل ۱.۸):

- **آستانهٔ بازبینی (τ_{review})**: بزرگ‌ترین آستانه‌ای که نرخ بازیابی (TPR) حداقل ۹۵ درصد از متون ماشینی را تضمین می‌کند (برای مدل پایه برابر با ۰٫۰۹۹۲)؛
- **آستانهٔ پرچم‌گذاری خودکار (τ_{flag})**: کوچک‌ترین آستانه‌ای که نرخ خطای مثبت کاذب (FPR) روی متون انسانی را زیر **نیم‌درصد** (۰٫۵٪) نگه می‌دارد (برای مدل پایه برابر با ۰٫۴۹۸۱).



شکل ۱.۸: سیاست تصمیم‌گیری دومرحله‌ای بر روی احتمالات کالیبره‌شده (p_{cal}) در مدل پایه (`tfidf_svm`). هر دو آستانه صرفاً روی مجموعهٔ اعتبارسنجی تعیین می‌شوند تا نرخ پرچم‌گذاری نادرست متون انسانی زیر ۰٫۵٪ باقی بماند.

۲.۸ نتایج ارزیابی درون‌دامنه‌ای گروه‌آگاه

جدول ۱.۸ نتایج ارزیابی مدل‌ها را بر روی ۶٬۶۴۵ سند آزمونِ تفکیک گروه‌آگاه نشان می‌دهد. مدل پایه (`tfidf_svm`) به امتیاز F_1 ماکرو برابر با ۰٫۹۳۹۲ و ROC-AUC = ۰٫۹۴۳۴ دست می‌یابد. رگرسیون لجستیک (۰٫۹۳۳۷) و جنگل تصادفی ترکیبی (۰٫۹۳۴۱) عملکردی نزدیک به آن دارند و جنگل تصادفیِ صرفاً سبک‌شناختی نیز بدون دیدن واژگان متن، به امتیاز ۰٫۹۲۶۹ و بالاترین مساحت زیر نمودار (ROC-AUC = ۰٫۹۵۱۵) می‌رسد.

سه ستون پایانی جدول ۱.۸ رفتار عملیاتی سیاست دومرحله‌ای را نشان می‌دهند: در مدل پایه، ۸۶٫۰۵٪ از متون ماشینی به‌طور خودکار پرچم‌گذاری می‌شوند، ۴۳٫۸۸٪ از کل اسناد در نوار احتیاطیِ بازبینی قرار می‌گیرند، و تنها ۰٫۰۰۵۰ (**نیم‌درصد**) از متون انسانی به اشتباه پرچم خودکارِ هوش مصنوعی دریافت می‌کنند.

جدول ۱.۸: نتایج ارزیابی درون‌دامنه‌ای روی تفکیک گروه‌آگاه ($n = ۶,۶۴۵$). سه ستون پایانی بیانگر خروجی سیاست تصمیم‌گیری دومرحله‌ای هستند.

مدل	بذر	n	$F_۱$ ماکرو	بازیابی AI	FPR@TPR95	ROC-AUC	PR-AUC	ECE	پرچم AI	انسانیِ پرچم‌خورده	ارجاع به بازبینی
tfidf_svm	۳	۶۶۴۵	۰٫۹۳۹۲ ± ۰٫۰۰۰۰	۰٫۸۶۰۵	۰٫۰۰۰۵	۰٫۹۴۳۴	۰٫۹۳۹۶	۰٫۰۱۲۸	۰٫۸۶۰۵	۰٫۰۰۵۰	۰٫۴۳۸۸
tfidf_logreg	۳	۶۶۴۵	۰٫۹۳۳۷ ± ۰٫۰۰۰۰	۰٫۸۶۱۳	۰٫۰۰۰۵	۰٫۹۴۲۰	۰٫۹۳۰۷	۰٫۰۱۴۱	۰٫۸۴۱۵	۰٫۰۰۳۱	۰٫۴۳۶۰
tfidf_nb	۳	۶۶۴۵	۰٫۹۱۵۳ ± ۰٫۰۰۰۰	۰٫۸۱۰۹	۰٫۰۰۰۲	۰٫۹۲۹۲	۰٫۹۱۴۴	۰٫۰۲۰۹	۰٫۷۶۰۵	۰٫۰۰۳۸	۰٫۶۳۱۸
style_rf	۳	۶۶۴۵	۰٫۹۲۶۹ ± ۰٫۰۰۱۷	۰٫۸۶۰۵	۰٫۰۰۳۹	۰٫۹۵۱۵	۰٫۹۴۰۶	۰٫۰۲۱۲	۰٫۸۵۳۸	۰٫۰۱۷۷	۰٫۳۸۵۹
tfidf_style_rf	۱	۶۶۴۵	۰٫۹۳۴۱ ± ۰٫۰۰۰۰	۰٫۸۵۳۶	۰٫۰۰۰۵	۰٫۹۳۳۳	۰٫۹۲۶۴	۰٫۰۱۱۸	۰٫۸۵۳۶	۰٫۰۰۸۴	۰٫۴۲۸۳
length_only	۳	۶۶۴۵	۰٫۷۸۹۴ ± ۰٫۰۰۰۰	۰٫۶۳۴۳	۰٫۰۰۳۶	۰٫۸۵۱۴	۰٫۸۰۸۸	۰٫۰۱۹۹	۰٫۳۱۴۹	۰٫۰۰۷۹	۰٫۷۲۲۳
perplexity_only	۳	۶۶۴۵	۰٫۳۸۵۳ ± ۰٫۰۰۰۰	۰٫۰۰۰۰	۰٫۵۲۷۳	۰٫۵۴۴۱	۰٫۳۹۷۱	۰٫۰۵۲۴	۰٫۵۹۷۲	۰٫۵۲۷۳	۰٫۴۴۶۷

تفاوت میان امتیاز ۰٫۹۳۹۲ در این جدول و امتیاز ۰٫۹۹۶۷ در تفکیک تصادفی نسخهٔ نخست (جدول ۱.۶) نشان می‌دهد که حذف نشت موضوعی، افزودن پنج دامنهٔ انسانی جدید و حضور متون بازنویسی‌شده، حدود ۶ واحد درصد از خوش‌بینیِ تفکیک تصادفی کاسته و برآورد واقع‌بینانه‌تری از عملکرد سامانه ارائه می‌دهد. همچنین خط پایهٔ «تنها طول متن» به $F_۱ = ۰٫۷۸۹۴$ می‌رسد، اما مدل‌های اصلی بیش از ۱۵ واحد درصد بالاتر از طولِ صرف عمل می‌کنند.

۳.۸ ارزیابی تعمیم‌پذیری: تولیدکننده‌ها، دامنه‌ها و سبک‌های جدید

جدول ۲.۸ نتایج پایداری مدل‌ها را در مواجهه با تولیدکننده‌های دیده‌نشده، دامنه‌های متنی جدید، سبک‌های نگارشی متفاوت و کمبود دادهٔ آموزشی نشان می‌دهد:

- **تولیدکننده‌های دیده‌نشده (برون‌تولیدکننده):** هنگامی که متون ماشینی آزمون از ۳۷ مدل زبانیِ جدید و موضوعات دیده‌نشده تولید می‌شوند، امتیاز $F_۱$ مدل پایه ۰٫۹۶۵۹ و رگرسیون لجستیک ۰٫۹۶۸۱ است (با $\text{FPR@TPR}۹۵ = ۰٫۰۰۰۰$). این پایداری تأیید می‌کند که تنوع ۷۰ مدلی در زمان گردآوری داده، مانع از وابستگی طبقه‌بند به یک خانوادهٔ مدل خاص شده است.

- **سبک‌های نگارشی جدید (برون‌سبک):** مدل پایه در سبک‌های «مقالهٔ رسمی» (۰٫۹۸۹۴) و «پرسش و توضیح» (۰٫۹۸۴۳) بسیار دقیق است، اما در مواجهه با سبک دیده‌نشدهٔ **«وبلاگ گفت‌وگویی و محاوره‌ای»** دچار افت شده و امتیاز $F_۱$ آن به ۰٫۸۵۱۵ کاهش می‌یابد. در همین سناریو، مدل جنگل تصادفیِ سبک‌شناسی (style_rf) به امتیاز ۰٫۹۵۴۱ دست می‌یابد؛ یعنی وقتی واژگان متن محاوره‌ای می‌شوند و مدل‌های مبتنی بر TF-IDF را دچار خطا می‌کنند، ویژگی‌های ساختاری و سبک‌شناختی همچنان مرز انسان و ماشین را به‌خوبی تفکیک می‌نمایند.

- **دامنه‌های متنی جدید (برون‌دامنه):** مدل پایه در تمامی دامنه‌های کنارگذاشته‌شده امتیاز F_1 بالای ۰٫۹۵ کسب می‌کند (در دامنه‌های خبر، دانشنامه، شبکهٔ اجتماعی و حقوقی بالای ۰٫۹۸۹). در مقابل، مدل صرفاً سبک‌شناختی (`style_rf`) در دامنهٔ «زیرنویس‌های ترجمه‌ای» به ۰٫۶۰۳۴ افت می‌کند؛ زیرا جملات کوتاه و منقطع زیرنویس‌ها از نظر آماری با ساختار جملات در متون عادی فارسی تفاوت دارند.

- **کمبود دادهٔ برچسب‌دار و اثر طول سند:** حتی با ۸ سند آموزشی در هر کلاس، مدل پایه به $F_1 = ۰٫۸۳۲۶$ می‌رسد و با ۱۲۸ سند از مرز ۰٫۹۰ عبور می‌کند. در مقابل، تحلیل عملکرد بر حسب طول سند نشان می‌دهد که روی متون بلندتر از ۵۰۰ واژه، امتیاز F_1 مدل پایه ۰٫۹۹۲۵ است، اما روی **متون کوتاه‌تر از ۱۰۰ واژه**، امتیاز F_1 به ۰٫۶۴۹۱ و نرخ بازیابی متون ماشینی به ۰٫۲۰۸۶ کاهش می‌یابد؛ زیرا در متون بسیار کوتاه، شواهد آماری کافی شکل نمی‌گیرد. همچنین ارزیابی روی برش نگه‌داشته‌شدهٔ نسخهٔ نخست (۴٫۰۵۶ سند) امتیاز ۰٫۹۹۱۴ را نشان می‌دهد.

۴.۸ ارزیابی مقاومت در برابر حملهٔ بازنویسی

جدول ۳.۸ نتایج ارزیابی جفتی مدل‌ها را روی ۳۰۰ سند ماشینیِ کنارگذاشته‌شده، پیش و پس از اعمال چهار روش بازنویسی آفلاین نشان می‌دهد. تمامی پنج مدل اصلی در برابر این بازنویسی‌ها مقاوم هستند و افت نرخ بازیابی آن‌ها در محدودهٔ نویز آماری است (بیشترین افت میانگین تنها ۰٫۰۰۱۱ است و بازهٔ اطمینان ۹۵ درصد تمامی آن‌ها شامل عدد صفر می‌شود). در نقطهٔ مقابل، خط پایهٔ «تنها طول متن (`length_only`)» در برابر حملهٔ بازنویسی دچار سقوط شدید شده و نرخ بازیابی آن از ۰٫۷۸۰۰ به ۰٫۵۸۵۰ کاهش می‌یابد (افت جفتیِ ۰٫۱۹۵۰، که بیشترین سهم آن ناشی از تغییر طول جملات در روش ترجمهٔ رفت‌وبرگشتی است).

۵.۸ ارزیابی در حالت سه‌کلاسه

در جدول ۴.۸، مدل‌ها موظف شده‌اند هر سه کلاس «انسان (`human`)»، «ماشین خام (`ai`)» و «ماشین بازنویسی‌شده (`ai_revised`)» را به‌طور مستقل از هم تفکیک کنند. هر دو مدل در شناسایی کلاس انسان (F_1 برابر با ۰٫۹۴۲ و ۰٫۹۱۴) و کلاس ماشین خام (F_1 برابر با ۰٫۹۷۴ و ۰٫۹۷۸) دقیق عمل می‌کنند، اما در تفکیک کلاس سوم (`ai_revised`) به امتیاز کلاسی ۰٫۲۵۰ و ۰٫۳۰۰ می‌رسند.

جدول ۲.۸: نتایج تعمیم‌پذیری در سناریوهای برون‌تولیدکننده، برون‌دامنه، برون‌سبک و آموزش با نمونهٔ کم.

سناریوی ارزیابی	مدل	n	F_1 ماکرو	بازیابی AI	FPR@TPR95	ROC-AUC
برون‌تولیدکننده (۳۷ مدل جدید)	tfidf_svm	۹۴۰	۰٫۹۶۵۹	۰٫۹۳۱۹	۰٫۰۰۰۰	۰٫۹۹۶۸
برون‌تولیدکننده (۳۷ مدل جدید)	tfidf_logreg	۹۴۰	۰٫۹۶۸۱	۰٫۹۴۶۸	۰٫۰۰۰۰	۰٫۹۹۵۲
برون‌تولیدکننده (۳۷ مدل جدید)	style_rf	۹۴۰	۰٫۹۶۳۱ ± ۰٫۰۰۴۰	۰٫۹۴۴۷	۰٫۰۰۲۱	۰٫۹۸۹۰
برون‌تولیدکننده (۳۷ مدل جدید)	length_only	۹۴۰	۰٫۷۷۲۲	۰٫۶۵۵۳	۰٫۰۰۲۱	۰٫۸۹۸۴
برون‌دامنه: وبلاگ	tfidf_svm	۴۲۱۴	۰٫۹۵۷۸	۰٫۹۶۵۸	۰٫۰۰۰۰	۰٫۹۸۳۶
برون‌دامنه: وبلاگ	style_rf	۴۲۱۴	۰٫۹۵۶۶ ± ۰٫۰۰۱۲	۰٫۹۶۳۶	۰٫۰۰۲۲	۰٫۹۸۸۳
برون‌دامنه: دانشنامه (ویکی‌پدیا)	tfidf_svm	۴۲۱۴	۰٫۹۸۹۳	۰٫۹۹۵۳	۰٫۰۰۰۰	۰٫۹۹۹۳
برون‌دامنه: دانشنامه (ویکی‌پدیا)	style_rf	۴۲۱۴	۰٫۸۷۴۷ ± ۰٫۰۱۳۶	۰٫۹۱۳۶	۰٫۰۱۳۱	۰٫۹۴۱۲
برون‌دامنه: خبر (ایسنا)	tfidf_svm	۴۲۱۴	۰٫۹۹۷۹	۰٫۹۹۵۷	۰٫۰۰۰۰	۰٫۹۹۹۸
برون‌دامنه: خبر (ایسنا)	style_rf	۴۲۱۴	۰٫۹۸۸۸ ± ۰٫۰۰۰۹	۰٫۹۸۲۳	۰٫۰۰۱۴	۰٫۹۹۷۹
برون‌دامنه: شبکهٔ اجتماعی (توییتر)	tfidf_svm	۴۲۱۴	۰٫۹۹۳۴	۰٫۹۹۶۷	۰٫۰۰۰۰	۰٫۹۹۹۰
برون‌دامنه: اداری و حقوقی	tfidf_svm	۴۲۱۴	۰٫۹۹۶۹	۰٫۹۹۳۸	۰٫۰۰۰۰	۰٫۹۹۹۴
برون‌دامنه: زیرنویس ترجمه‌ای	tfidf_svm	۱۲۶	۱٫۰۰۰۰	۱٫۰۰۰۰	۰٫۰۰۰۰	۱٫۰۰۰۰
برون‌دامنه: زیرنویس ترجمه‌ای	style_rf	۱۲۶	۰٫۶۰۳۴ ± ۰٫۰۰۶۶	۰٫۹۸۴۱	۰٫۰۵۸۲	۰٫۹۲۵۹
برون‌سبک: وبلاگ گفت‌وگویی	tfidf_svm	۷۶۲	۰٫۸۵۱۵	۰٫۷۱۶۵	۰٫۰۰۰۰	۰٫۹۶۴۴
برون‌سبک: وبلاگ گفت‌وگویی	style_rf	۷۶۲	۰٫۹۵۴۱ ± ۰٫۰۰۱۳	۰٫۹۴۴۰	۰٫۰۱۰۵	۰٫۹۸۳۴
برون‌سبک: مقالهٔ رسمی	tfidf_svm	۷۵۸	۰٫۹۸۹۴	۰٫۹۹۲۱	۰٫۰۰۰۰	۰٫۹۹۷۷
برون‌سبک: پرسش و توضیح	tfidf_svm	۱۰۸۴	۰٫۹۸۴۳	۰٫۹۸۷۱	۰٫۰۰۰۰	۰٫۹۹۶۴
نمونهٔ کم: ۸ سند در هر کلاس	tfidf_svm	۶۶۴۵	۰٫۸۳۲۶	۰٫۷۲۲۶	۰٫۰۰۰۲	۰٫۸۶۱۰
نمونهٔ کم: ۳۲ سند در هر کلاس	tfidf_svm	۶۶۴۵	۰٫۸۸۵۴	۰٫۷۷۹۰	۰٫۰۰۰۵	۰٫۸۹۷۴
نمونهٔ کم: ۱۲۸ سند در هر کلاس	tfidf_svm	۶۶۴۵	۰٫۹۰۰۱	۰٫۸۲۱۴	۰٫۰۰۰۷	۰٫۹۰۷۰

جدول ۳.۸: ارزیابی مقاومت در برابر حملهٔ بازنویسی: مقایسهٔ جفتی نرخ بازیابی روی ۳۰۰ سند ماشینی پیش و پس از حمله.

مدل	بازیابی متن پاک	بازیابی پس از حمله	افت جفتی (بازهٔ اطمینان ۹۵٪)	P(drop > 0)
tfidf_svm	۰٫۹۹۶۷	۰٫۹۹۷۵	−۰٫۰۰۰۸ [−۰٫۰۰۲۵, ۰٫۰۰۰۰]	۰٫۰۰۰
tfidf_logreg	۱٫۰۰۰۰	۰٫۹۹۹۲	۰٫۰۰۰۸ [۰٫۰۰۰۰, ۰٫۰۰۲۵]	۰٫۶۴۷
tfidf_nb	۰٫۹۵۳۳	۰٫۹۵۸۳	−۰٫۰۰۵۰ [−۰٫۰۱۲۵, ۰٫۰۰۲۸]	۰٫۰۸۳
style_rf	۰٫۹۸۱۱	۰٫۹۸۴۴	−۰٫۰۰۳۳ [−۰٫۰۰۸۳, ۰٫۰۰۱۱]	۰٫۰۷۲
tfidf_style_rf	۰٫۹۸۷۸	۰٫۹۸۶۷	۰٫۰۰۱۱ [−۰٫۰۰۴۴, ۰٫۰۰۶۴]	۰٫۶۰۸
length_only	۰٫۷۸۰۰	۰٫۵۸۵۰	۰٫۱۹۵۰ [۰٫۱۷۱۱, ۰٫۲۲۰۸]	۱٫۰۰۰
perplexity_only	۰٫۰۰۰۰	۰٫۰۰۰۰	۰٫۰۰۰۰ [۰٫۰۰۰۰, ۰٫۰۰۰۰]	۰٫۰۰۰

هنگامی که خروجی همین مدل‌های سه‌کلاسه به حالت دودویی (انسان در برابر مجموعِ ماشین خام و بازنویسی‌شده) نگاشت می‌شود (ستون آخر جدول)، امتیاز F_1 ماکرو دوباره به ۰٫۹۱۷۷ و ۰٫۸۸۴۲ بازمی‌گردد. دلیل این پدیده آن است که بازنویسی‌های آفلاین، ردپای آماری متن را آن‌قدر تغییر نمی‌دهند که شبیه نوشتهٔ انسان شود، اما مرز میان «ماشین خام» و «ماشین بازنویسی‌شده» را هم‌پوشان می‌کنند.

جدول ۴.۸: نتایج طبقه‌بندی سه‌کلاسه روی آزمون گروه‌آگاه ($n = 6{,}645$، میانگین ۳ بذر). ستون آخر، امتیاز F_1 ماکرو پس از ادغام دو کلاس ماشینی در قالب مسئلهٔ دودویی است.

مدل سه‌کلاسه	دقت	F_1 ماکرو	human (P / R / F1)	ai (P / R / F1)	ai_revised (P / R / F1)	ROC-AUC OvR	F_1 دودویی
tfidf_logreg_3class	۰٫۹۲۵۸	۰٫۷۲۲۲	0.924 / 0.961 / 0.942	0.959 / 0.990 / 0.974	0.464 / 0.172 / 0.250	۰٫۹۳۱۲	۰٫۹۱۷۷
tfidf_style_logreg_3class	۰٫۸۹۳۶	۰٫۷۳۰۵	0.936 / 0.892 / 0.914	0.963 / 0.992 / 0.978	0.261 / 0.354 / 0.300	۰٫۹۳۹۵	۰٫۸۸۴۲

۶.۸ تحلیل کمّی و کیفی خطاها

برای کالبدشکافی موارد شکست مدل پایه (`tfidf_svm`) در ارزیابی درون‌دامنه‌ای، شدیدترین خطاها در هر دو جهت استخراج شدند: ۱۰۰ خطای مثبت کاذب (متون انسانی که ماشینی تشخیص داده شده‌اند) و ۳۴۶ خطای منفی کاذب (متون ماشینی که انسانی تشخیص داده شده‌اند).

بررسی عملکرد سیاست دومرحله‌ای روی این خطاها (جدول ۵.۸) کارایی نوار بازبینی را نشان می‌دهد: از میان ۱۰۰ خطای مثبت کاذبِ مدل در آستانهٔ پیش‌فرض، **۷۹ مورد در نوار احتیاطیِ «نیازمند بازبینی انسانی» متوقف شده‌اند** و تنها ۲۱ مورد از آستانهٔ پرچم‌گذاری خودکار (۰٫۴۹۸۱) عبور کرده‌اند. در سمت خطاهای منفی کاذب نیز ۲۷۱ مورد از ۳۴۶ خطا به داور انسانی ارجاع داده شده است.

خوشه‌بندی این خطاها یافته‌های بخش تعمیم‌پذیری را تأیید می‌کند: در میان خطاهای مثبت کاذب، سه دستهٔ اصلی شامل «اسناد کوتاه» (۲۷ مورد)، «متون انسانیِ رسمی و بلند» (۲۲ مورد) و «نوشته‌های وبلاگی خاص» (۲۱ مورد) هستند. در سمت خطاهای منفی کاذب نیز بیش از ۷۶ درصد خطاها در دو دسته متمرکز شده‌اند: **«متون ماشینی با لحن محاوره‌ای» (۱۵۶ مورد)** و **«متون ماشینی کوتاه» (۱۱۰ مورد)**.

۷.۸ محدودیت‌های پژوهش

برای تفسیر دقیق نتایج این گزارش، محدودیت‌های زیر باید مد نظر قرار گیرند:

- **هزینهٔ محاسباتی بازآموزی چندبذری ترنسفورمرها:** به دلیل محدودیت پردازندهٔ گرافیکی در

جدول ۵.۸: کالبدشکافی خطاهای مدل پایه (`tfidf_svm`) در آزمون درون‌دامنه‌ای گروه‌آگاه.

شاخص تحلیل خطا	مثبت کاذب (انسان → ماشین)	منفی کاذب (ماشین → انسان)
تعداد کل نمونه‌های خطای استخراج‌شده	۱۰۰	۳۴۶
متوقف‌شده در نوار «نیازمند بازبینی انسانی»	۷۹	۲۷۱
عبورکرده از مرز تصمیم خودکار	۲۱	۷۵
دامنهٔ متنی غالب در میان خطاها	وبلاگ (۵۷)	خبر (۱۱۵)
گروه طولی غالب	بلند (۵۵)	متوسط (۲۲۹)
سطح رسمیت غالب	خنثی (۶۶)	خنثی (۲۴۶)
خوشهٔ خطای اول	سند کوتاه (۲۷)	متن ماشینی محاوره‌ای (۱۵۶)
خوشهٔ خطای دوم	متن رسمیِ بلند (۲۲)	متن ماشینی کوتاه (۱۱۰)
خوشهٔ خطای سوم	لحن خودمانی (۲۱)	سایر موارد (۵۵)

محیط محلی، بازآموزی چندبارهٔ ParsBERT و GPT-2 به ازای تمامی برش‌های گروه‌آگاه و بذرهای مختلف در قالب نوت‌بوک‌های مکمل Google Colab (پیوست آ.۴) تدارک دیده شده است.

- **ماهیت آفلاین بازنویسی‌ها:** کلاس سوم (`ai_revised`) و مجموعهٔ حمله با چهار روش بازنویسی ماشینی آفلاین ساخته شده‌اند؛ ارزیابی در برابر بازنویسی‌های پیچیدهٔ مبتنی بر مدل‌های زبانی بزرگ نیازمند فراخوانی درگاه‌های API بیرونی است.
- **حجم محدود دامنهٔ زیرنویس:** دامنهٔ زیرنویس‌های ترجمه‌ای تنها شامل ۶۳ سند است و صرفاً به‌عنوان محک کاوش خطا استفاده شده است.
- **وابستگی زمانی به نسل مدل‌های زبانی:** متون ماشینی این پیکره نمایندهٔ نسل مدل‌های زبانی در زمان گردآوری داده هستند و با ظهور نسل‌های جدیدتر، به‌روزرسانی دوره‌ای داده‌های آموزش ضروری خواهد بود.
- **نبود تحلیلگر نحوی (Dependency Parser):** ویژگی‌های سبک‌شناسی فعلی در سطوح نویسه‌ای، واژگانی و جمله‌ای تعریف شده‌اند و ویژگی‌های درخت نحو در آن‌ها حضور ندارند.

فصل ۹

نتیجه‌گیری و پیشنهادها

۱.۹ جمع‌بندی دستاوردها و یافته‌های کلیدی

در این پروژه، یک سامانهٔ چندمدلی برای تشخیص متون تولیدشده توسط هوش مصنوعی در زبان فارسی طراحی، پیاده‌سازی و ارزیابی شد که شانزده مدل از پنج خانوادهٔ الگوریتمی را در قالب یک وب‌سرویس یکپارچه و رابط کاربری تعاملی گرد هم می‌آورد. مهم‌ترین یافتهٔ روش‌شناختی این پژوهش آن است که **در مسائل تشخیص متن، امتیازهای آزمایشگاهی بدون تعریف دقیق نحوهٔ تفکیک داده می‌توانند خوش‌بینانه و فریبنده باشند.** حذف نشت موضوعی، کنترل ابعاد فضای ویژگی TF-IDF (۱۵۰,۰۰۰ ویژگی ۱ تا ۳ گرم به‌جای ۹٫۴ میلیون سه‌گرم) و جداسازی کامل مجموعهٔ اعتبارسنجی، ستون‌های اصلی ارزیابی علمی این سامانه را تشکیل دادند.

در بعد مدل‌سازی، طراحی دو خط‌لولهٔ پیش‌پردازش موازی (مسیر کامل مبتنی بر ریشه‌یابی برای مدل‌های واژگانی و مسیر سبک برای ترنسفورمرها)، طراحی شبکه‌های TextCNN ($F_1 = ۰٫۹۹۰۵$) و Attention-BiLSTM ($F_1 = ۰٫۹۸۸۱$)، و تنظیم ظریف ترنسفورمرهای فارسی (GPT-2 با ۰٫۹۹۶۹ و ParsBERT با ۰٫۹۹۸۳) دقت درون‌دامنه‌ای سامانه را به حداکثر رساندند. همچنین با افزودن مدل‌های تصمیم‌گیری «سیستم‌ـیک» و Laya، خروجی سامانه از یک عدد خام به سه پاسخ ساختاریافته و کالیبره گسترش یافت.

در بعد ارزیابی سخت‌گیرانه (فصل ۸)، روی نسخهٔ دوم مجموعه‌داده شامل شش دامنهٔ انسانی و سه کلاس برچسب، امتیاز F_1 ماکرو در ارزیابی درون‌دامنه‌ایِ گروه‌آگاه برابر با ۰٫۹۳۹۲ و در مواجهه با ۳۷ مدل تولیدکنندهٔ دیده‌نشده برابر با ۰٫۹۶۵۹ ثبت شد. مدل‌های واژگانی و سبکی در برابر حملات بازنویسی سطحی کاملاً پایدار ماندند (افت حداکثر ۰٫۰۰۱۱)، و پیاده‌سازی سیاست تصمیم‌گیری دومرحله‌ای با نوار «نیازمند بازبینی انسانی» تضمین کرد که نرخ پرچم‌گذاری نادرست متون انسانی در زیر آستانهٔ ٪۰٫۵ مهار شود.

۲.۹ پیشنهادها برای کارهای آینده

بر پایهٔ یافته‌ها و تحلیل خطاهای این پژوهش، محورهای زیر برای ادامهٔ مسیر پیشنهاد می‌شوند:

۱. **غنی‌سازی پیکره با متون کوتاه و محاوره‌ای:** افزودن نمونه‌های کوتاه (زیر ۱۰۰ واژه) و پرامپت‌های محاوره‌ایِ بیشتر به داده‌های آموزش، جهت برطرف کردن دو گلوگاه اصلی شناسایی‌شده در تحلیل خطا؛

۲. **تکمیل ارزیابی گروه‌آگاه برای ترنسفورمرها:** اجرای کامل نوت‌بوک‌های Colab (بخش آ.۴) روی پردازندهٔ گرافیکی برای بازآموزی چندبذری ParsBERT و GPT-2 روی تمامی برش‌های نسخهٔ دوم؛

۳. **تولید مجموعهٔ حملهٔ بازنویسی با مدل‌های زبانی بزرگ:** استفاده از مدل‌های زبانی بزرگ برای بازنویسی عمیق معنایی و سنجش مقاومت سامانه در برابر حملات پیشرفته؛

۴. **استفاده از روش‌های مبتنی بر انحنای احتمال:** بهره‌گیری از مدل‌های زبانی پایهٔ بزرگ‌تر و الگوریتم‌هایی نظیر DetectGPT برای رفع مشکل عدم تقارن در شاخص سرگشتگی.

پیوست آ

راهنمای اجرا و بازتولید نتایج

آ.۱ پیش‌نیازهای نرم‌افزاری و سخت‌افزاری

تمامی دستورات این پیوست از مسیر ریشهٔ مخزن پروژه (https://github.com/ehsan-torabi/AITextDetection) اجرا می‌شوند. برای اجرای بک‌اند و نوت‌بوک‌ها، پایتون نسخهٔ ۱۱٫۳ یا بالاتر و مدیر بستهٔ uv مورد نیاز است. تمامی وابستگی‌ها در پروندهٔ pyproject.toml تعریف شده‌اند و با اجرای دستور uv sync در محیط مجازی venv. نصب می‌شوند. چک‌پوینت‌های آموزش‌دیده در مخزن https://huggingface.co/ehsantorabi/aitextd-models منتشر شده‌اند. برای اجرای رابط کاربری وب نیز محیط Node.js و مدیر بستهٔ pnpm لازم است.

آ.۲ زنجیرهٔ بازتولید با نوت‌بوک‌های شش‌گانه

تمامی مراحل آماده‌سازی داده، استخراج ویژگی و آموزش مدل‌ها در شش نوت‌بوک شماره‌گذاری‌شده در پوشهٔ notebooks/ سازمان‌دهی شده‌اند (شکل آ.۱):

۱. 01_data_preparation.ipynb: ساخت مجموعهٔ متوازن نسخهٔ نخست، تشکیل حافظهٔ نهان ریشه‌یابی shekar، محاسبهٔ ماتریس TF-IDF و ده ویژگی سبک‌شناسی، و ساخت کامل لایهٔ ارزیابی نسخهٔ دوم (حذف تکرار سه‌مرحله‌ای، تفکیک‌های گروه‌آگاه و مجموعهٔ حمله)؛

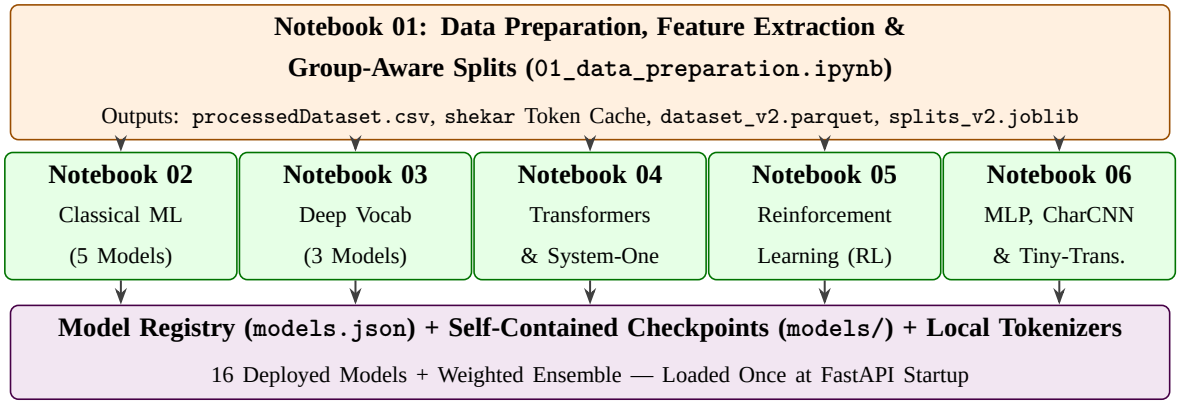
۲. 02_classical_models.ipynb: آموزش و تنظیم ابرپارامترهای پنج مدل یادگیری ماشین کلاسیک؛

۳. 03_deep_learning_models.ipynb: ساخت واژگان و آموزش سه شبکهٔ عمیق واژگانی (BiLSTM پایه، TextCNN و Attention-BiLSTM)؛

۴. `04_llm_models.ipynb`: تنظیم ظریف GPT-2 و ParsBERT، تولید داده‌های تقطیر دانش، آموزش لایهٔ تصمیم سیستم‌ـیک و تنظیم ظریف مدل Laya؛

۵. `05_reinforcement_learning.ipynb`: آموزش سیاست تصمیم باندیت با الگوریتم REINFORCE و تنظیم ظریف RLCD روی TextCNN؛

۶. `06_neural_networks.ipynb`: آموزش از صفر سه شبکهٔ تکمیلی شامل MLP، CharCNN و Tiny-Transformer.



شکل آ.۱: نمودار وابستگی و جریان بازتولید در نوت‌بوک‌های شش‌گانه: از آماده‌سازی داده‌های خام در نوت‌بوک ۰۱ تا تولید چک‌پوینت‌های نهایی در نوت‌بوک‌های ۰۲ تا ۰۶.

تمامی نوت‌بوک‌ها با رویکرد بارگذاری‌محور (Load-First) نوشته شده‌اند؛ یعنی در صورت وجود چک‌پوینت روی دیسک، همان فایل را بارگذاری و ارزیابی می‌کنند. برای اجبار به بازسازی داده‌ها یا آموزش مجدد مدل‌ها، پرچم‌های کنترلی `FORCE_*` در سلول نخست هر نوت‌بوک تعبیه شده‌اند (جدول آ.۱). دستورات اجرای ترتیبی نوت‌بوک‌ها به شرح زیر است:

```
uv sync

uv run jupyter nbconvert --to notebook --execute \
    notebooks/01_data_preparation.ipynb
uv run jupyter nbconvert --to notebook --execute \
    notebooks/02_classical_models.ipynb
uv run jupyter nbconvert --to notebook --execute \
    notebooks/03_deep_learning_models.ipynb
uv run jupyter nbconvert --to notebook --execute \
    notebooks/04_llm_models.ipynb
uv run jupyter nbconvert --to notebook --execute \
```

```
    notebooks/05_reinforcement_learning.ipynb
uv run jupyter nbconvert --to notebook --execute \
    notebooks/06_neural_networks.ipynb
```

جدول آ.۱: پرچم‌های کنترلی بازسازی (*_FORCE) و خروجی‌های هر نوت‌بوک.

نوت‌بوک	پرچم‌های کنترلی *_FORCE	آرتیفکت‌های خروجی
۰۱ آماده‌سازی داده	FORCE_REBUILD_V1، FORCE_REBUILD_V2، FORCE_REBUILD_ATTACK	مجموعه‌دادهٔ نسخهٔ اول، قاب و تفکیک‌های گروه‌آگاه نسخهٔ دوم و مجموعهٔ حمله
۰۲ مدل‌های کلاسیک	FORCE_RETRAIN	فایل‌های *.joblib و بردارساز TF-IDF
۰۳ شبکه‌های عمیق	FORCE_RETRAIN	lstmModel.pt، textcnnModel.pt و attnBilstmModel.pt
۰۴ مدل‌های زبانی	FORCE_TRAIN_GPT2، FORCE_TRAIN_PARSBERT، FORCE_TRAIN_S1، FORCE_TRAIN_LAYA	gpt2ClassifierModel.pt، parsbertClassifierModel.pt، systemOneModel.pt و laya_persian
۰۵ یادگیری تقویتی	FORCE_TRAIN_RLCD، FORCE_TRAIN_POLICY	textcnnRlcdModel.pt و rlPolicyModel.pt
۰۶ شبکه‌های عصبی	FORCE_RETRAIN	mlpModel.pt، charcnnModel.pt و tinyTransformerModel.pt

آ.۳ راه‌اندازی سرور تشخیص، رابط وب و آزمون‌ها

برای دریافت خودکار چک‌پوینت‌های مدل‌ها از مخزن Hugging Face و راه‌اندازی سرویس بک‌اند روی درگاه 8000، دستورات زیر اجرا می‌شوند:

```
# Fetch missing checkpoints from Hugging Face and start server:
.venv/bin/python server.py --download-online

# Standard production startup on 0.0.0.0:8000:
.venv/bin/python server.py
```

برای راه‌اندازی رابط کاربری وب روی درگاه 3000 و اجرای آزمون‌های خودکار:

```
# Start Next.js web interface on http://localhost:3000:
cd web-app && pnpm install && pnpm dev

# Run fast unit tests and full golden regression tests:
.venv/bin/python -m pytest tests/ -m "not slow"
.venv/bin/python -m pytest tests/ -m slow
```

افزودن یک مدل جدید به سامانه که از یکی از محیط‌های اجرای موجود پیروی می‌کند، تنها نیازمند قرار دادن فایل چک‌پوینت در پوشهٔ `/models` و ثبت مشخصات آن در پروندهٔ `models.json` است و نیازی به تغییر در کدهای سرور یا رابط کاربری ندارد.

آ.۴ اجرای آموزش و ارزیابی کامل روی Colab Google

برای اجرای آموزش‌های مبتنی بر پردازندهٔ گرافیکی بدون فشار بر سخت‌افزار محلی، بستهٔ مکمل AITextDetection-colab طراحی شده است. پس از قرار دادن پرونده‌های داده در فضای Google Drive، چهار نوت‌بوک به ترتیب زیر اجرا می‌شوند:

۱. `01_setup_and_data.ipynb`: نصب بسته‌ها، اتصال به Drive و بررسی سلامت تفکیک داده؛

۲. `02_classical_models.ipynb`: آموزش و ارزیابی کامل مدل‌های کلاسیک و خطوط پایه روی تمامی سناریوهای تعمیم، حملهٔ بازنویسی و حالت سه‌کلاسه؛

۳. `03_transformers_unsloth.ipynb`: بازآموزی ترنسفورمرها روی پردازندهٔ گرافیکی T4 با بهره‌گیری از شتاب‌دهندهٔ Unsloth؛

۴. `04_laya_finetune.ipynb`: تنظیم ظریف مدل Laya با حلقهٔ RLCD و کالیبراسیون دما.

کتاب‌نامه

[۱] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. *Attention Is All You Need*. Advances in Neural Information Processing Systems (NeurIPS), 2017, https://arxiv.org/abs/1706.03762.

[۲] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Proceedings of NAACL-HLT, 2019.

[۳] B. Muller. *BERT 101: State Of The Art NLP Model Explained*. Hugging Face Blog, March 2, 2022, https://huggingface.co/blog/bert-101.

[۴] Hugging Face. *BERT: Transformers Model Documentation*. https://huggingface.co/docs/transformers/model_doc/bert.

[۵] S. H. Díaz, B. Charpentier. *LLM Architectures Explained: What Powers Today's Top Models*. Hugging Face Blog (Pruna AI), March 4, 2026, https://huggingface.co/blog/PrunaAI/llm-architectures-overview.

[۶] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever. *Language Models are Unsupervised Multitask Learners*. OpenAI Technical Report, 2019.

[۷] S. Hochreiter, J. Schmidhuber. *Long Short-Term Memory*. Neural Computation, 9(8):1735–1780, 1997.

[۸] L. Breiman. *Random Forests*. Machine Learning, 45(1):5–32, 2001.

J. F. Friedman. *Greedy Function Approximation: A Gradient Boosting Machine*. Annals of Statistics, 29(5):1189–1203, 2001. [۹]

G. Salton, C. Buckley. *Term-Weighting Approaches in Automatic Text Retrieval*. Information Processing & Management, 24(5):513–523, 1988. [۱۰]

J. Howard, S. Ruder. *Universal Language Model Fine-tuning for Text Classification*. Proceedings of ACL, 2018. [۱۱]

M. Farahani, M. Gharachorloo, M. Farahani, M. Manthouri. *ParsBERT: Transformer-based Model for Persian Language Understanding*. Neural Processing Letters, 53(6):3831–3847, 2021. [۱۲]

HooshvareLab. *gpt2-fa: A GPT-2 Language Model Pretrained on Persian Text*. Hugging Face Model Hub, https://huggingface.co/HooshvareLab/gpt2-fa. [۱۳]

E. Torabi Farsani. *Persian AI-Generated Text Dataset*. Hugging Face Datasets, 2026, https://huggingface.co/datasets/ehsantorabi/persian-ai-generated-text. [۱۴]

TypeSafe. *Jev: A Practical Guide to System One Models and Typed AI*. Hugging Face Blog, 2026, https://huggingface.co/blog/paidaxccc/jev-system-one-model-a-practical-guide-to-typed-ai. [۱۵]

S. Chudmunge. *What is Jev? The System One Model and Architecture Blueprint*. Eulerfold, September 25, 2026, https://www.eulerfold.com/articles/what-is-jev-system-one-architecture. [۱۶]

Liuziyu77. *Valen: An Open-Source Multimodal Decision Model Inspired by Jev*. GitHub Repository, 2026, https://github.com/Liuziyu77/Valen. [۱۷]

Convai Innovations. *Laya: Multilingual, Non-Autoregressive System One Decision Model*. GitHub Repository and Hugging Face Model Hub, 2026, https: [۱۸]

//github.com/NandhaKishorM/laya / https://huggingface.co/convaiinnovations/laya.