Information Technology Innovations and Applied Communication

Vol. 3, No. 3, Autumn, 2024, Serial No. 7,pp.43-54

Presenting a method for detecting self-contradictory articles based on RWKV neural network in Persian encyclopedia

Sadegh Bejani*

*Assistant Professor, Imam Hossein University , Tehran, Iran.

(**Received:** 11 October 2025, **Revised:** 18 November 2025, **Accepted:** 27 November 2025, **Published:** 07 December 2025)

Abstract

In the Persian encyclopedia, one of the necessary capabilities is to recognize contradictory sentences in each text in order to provide appropriate services to users. So far, few research works have been done inside the country to detect contradictions in Persian texts, which do not have enough accuracy. This research investigates and evaluates a new method for detecting contradictory sentences in a Persian text using the RWKV neural network. The main goal of this research is to detect contradictory sentences in a Persian text using data from FarsTail dataset with high accuracy. Based on the research method, the appropriate neural network RWKV was selected as a deep learning model. Next, fine-tuning the mentioned model for the task of detecting contradictions in the text, loading the model and data, pre-processing the data, fine-tuning the model with the pre-processed data and then evaluating the model was done. It should be mentioned that the data was divided into three categories: inferential, contradictory and neutral, and after pre-processing, the model was trained and used to recognize contradictory sentences in a Persian text. The evaluation of the model was done by using the confusion matrix and calculating various criteria including accuracy, positive prediction accuracy, readability and F1 score. The evaluation results showed that the RWKV model with an accuracy of 95.06% and an F1 score equal to 92.03% has a high ability to identify contradictory sentences in a text, compared to previous similar works. Comparing the performance of the RWKV model with other models such as mBERT, ESIM, HBMP and DecompAtt indicates the superiority of RWKV in separating the contradiction classes and reducing the false positive error rate. In evaluating a text containing contradictory sentences, the model is able to correctly detect 8 out of 10 contradictions in the text.

Keywords: Recognize Contradictory Sentences, RWKV Neural Network, Deep Learning, FarsTail Dataset, Paradoxical Data, Inferential Data, Neutral Data.

* Corresponding Author E-mail: Sbejani@ihu.ac.ir

«نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی»
سال سوم، شماره ۳، فصل پاییز ۱۴۰۴؛ ص ۵۴-۴۳.

علمی - پژوهشی

ارائه روشی جدید مبتنی بر شبکه عصبی RWKV، برای تشخیص مقالات خود متناقض در دانشنامه فارسی

صادق بجانی [۱]

۱- استادیار، دانشگاه جامع امام حسین(ع)، تهران ، ایران

(**دریافت:** ۱۹مهر ۱۴۰۴، **بازنگری:** ۲۷ آبان ۱۴۰۴، **پذیرش:** ۰۶ آذر۱۴۰۴، **انتشار:** ۱۶ آذر ۱۴۰۴)

چکیده

در دانشنامه فارسی یکی از قابلیت‌های لازم تشخیص جملات متناقض در هر متن است تا از این طریق خدمات مناسب برای کاربران فراهم شود. تاکنون در داخل کشور برای تشخیص تناقض در متن‌های فارسی کارهای تحقیقاتی کمی انجام‌گرفته است که از دقت کافی برخوردار نیستند. این تحقیق به بررسی و ارزیابی یک روش نوین برای تشخیص جملات متناقض در یک متن فارسی با استفاده از شبکه عصبی RWKV پرداخته است. هدف اصلی این تحقیق، تشخیص جملات متناقض در یک متن فارسی با استفاده از داده‌های مجموعه‌داده FarsTail بادقت بالا است. بر اساس روش تحقیق، شبکه عصبی مناسب RWKV به‌عنوان یک مدل یادگیری عمیق انتخاب شد. در ادامه تنظیم دقیق[۱] مدل یادشده جهت وظیفه تشخیص تناقض در متن، بارگذاری مدل و داده‌ها، پیش‌پردازش داده‌ها، تنظیم دقیق مدل با داده‌های پیش‌پردازش شده و سپس ارزیابی مدل انجام گرفت. لازم به بیان است که داده‌ها به سه دسته‌ی استنتاجی[۲]، متناقض[۳] و خنثی[۴]تقسیم‌بندی شده و پس از پیش‌پردازش، مدل آموزش‌دیده و برای تشخیص جملات متناقض در یک متن فارسی به کار گرفته شد. ارزیابی مدل با استفاده از ماتریس درهم‌ریختگی[۵] و محاسبه‌ی معیارهای مختلفی ازجمله دقت، دقت پیش‌بینی مثبت، بازخوانی و امتیاز F1 انجام شد. نتایج ارزیابی نشان داد که مدل RWKV بادقت ۹۵/۰۶٪ و امتیاز F1 برابر با ۹۲/۰۳٪، توانایی بالایی در شناسایی جملات متناقض در یک متن، نسبت به کارهای مشابه قبلی دارد. مقایسه‌ی عملکرد مدل RWKV با مدل‌های دیگر مانند mBERT، ESIM، HBMP و DecompAtt حاکی از برتری RWKV در تفکیک رده‌های تناقض و کاهش نرخ خطای مثبت کاذب است. در ارزیابی یک متن شامل جملات متناقض، مدل قادر به تشخیص درست ۸ مورد از ۱۰ تناقض موجود در متن است.

کلیدواژه‌ها: تشخیص جملات متناقض، شبکه عصبی RWKV، یادگیری عمیق، مجموعه‌داده FarsTail، داده متناقض، داده استنتاجی، داده خنثی.

رایانامه نویسنده مسئول: E-mail: javadzade@ihu.ac.ir :

[1] Fine tune
[2] Entailment
[3] Contradiction
[4] Neutral
[5] Confusion Matrix

۱. مقدمه

تشخیص تناقض یک کار اساسی برای درک متن است و کاربردهای زیادی به‌ویژه در استنتاج متنی و تحلیل احساسات متنی دارد. رابطه تناقض بیانگر تقابل معنایی بین دو جمله است و به‌طورکلی شامل تمام تضادها، نفی و سایر تقابل‌های معنایی مانند دانش جهانی، واژگانی، عددی و ساختاری می‌شود. واژه‌های تناقض، تضاد، نفی، تعارض و تقابل در این حوزه، باوجود تفاوت‌هایی که دارند، گاهی به‌جای یکدیگر به کار می‌روند. تناقض معمولاً در دو سطح کلمه و ساختار رخ می‌دهد. معمولاً انواع تناقض‌ها در سطح کلمه، بدون درنظرگرفتن نقش هر یک از آن‌ها در جملات مشخص می‌شود، اما بر اساس نتایج بررسی‌های انجام‌گرفته، در بیشتر مواقع تمرکز بر روی تشخیص تناقض بین جملات است. تناقض عمدتاً بین دو جمله یا یک قطعه متن است، مانند «علی دیروز کشته شد» و «علی اکنون در حال خوردن ناهار در غذاخوری است» که دو جمله به طور هم‌زمان درست نیستند. باتوجه‌به انواع تناقض‌هایی که در دلالت‌های متنی رخ می‌دهد، می‌توان دسته‌بندی دیگری از تناقض‌ها در سطح جمله داشت که شامل هفت دسته می‌شود. کلمات متضاد (مانند سوپ داغ است و سوپ سرد است)، نفی (مانند سامان به مدرسه رفت و سامان به مدرسه نرفت)، عددی (مانند در جنگ بیش از ۱۰۰ نفر کشته شدند و در جنگ ۵۰ نفر کشته شدند)، واقعیت (مانند دردها قصد نداشتند که وارد بانک شوند و دردها وارد بانک شدند)، ساختاری (مانند استقلال در بازی دیروز پرسپولیس را شکست داد و پرسپولیس در بازی دیروز استقلال را شکست داد)، واژگانی (مانند آقای رحیمی فرزندانش را به روش مدرن تربیت می‌کند و آقای رحیمی فرزندش را کتک می‌زند) و دانش جهانی (مانند دانشگاه امیرکبیر سال ۱۳۳۷ تأسیس شد و پدر من در سال ۱۳۳۵ دانشجوی دانشگاه امیرکبیر بود)[۱].

مقالات خود - متناقض مقالاتی هستند که حاوی جملات در تقابل یکدیگر هستند. در دانشنامه انتظار این است که مقالات خود متناقض تشخیص داده شوند، برای تشخیص این نوع تناقض‌ها در دانشنامه قابلیتی وجود ندارد. تاکنون برای تشخیص تناقض در متن کارهای زیادی انجام‌گرفته و روش‌های متعددی مبتنی بر مدل‌های مختلفی از شبکه‌های عصبی ارائه‌شده است، اما درجه پیچیدگی آن‌ها عمدتاً n^2 و یا حتی مضربی از n^2 است و این ویژگی سبب شده است که دقت و سرعت تشخیص تناقض مطلوب نباشد. بر اساس نتایج پژوهش‌های انجام‌گرفته در این مقاله، آخرین کار تشخیص مقالات خود متناقض از شبکه عصبی به نام PCL[1] استفاده می‌کند که بر اساس نتایج منتشرشده، سرعت و دقت تشخیص مقالات خود متناقض از درجه مطلوبی برخوردار نیست[۲]. در این مقاله بر آنیم که روشی با استفاده از مدل شبکه عصبی RWKV[2] جهت تشخیص مقالات خود متناقض ارائه کنیم. تاکنون برای تشخیص مقالات خود متناقض از این مدل شبکه استفاده‌نشده است. نوآوری تحقیق جاری شناسایی قابلیت‌های شبکه عصبی RWKV و استفاده از آن برای تشخیص مقالات خود متناقض در دانشنامه فارسی است. شبکه عصبی RWKV از درجه پیچیدگی خطی برخوردار است و استفاده از آن موجب افزایش دقت و سرعت تشخیص تناقض می‌شود[۳].

در ادامه مقاله، ابتدا روش تحقیق بیان‌شده و سپس کارهای محققین قبلی دررابطه‌با روش تشخیص تناقض در انواع متون مورد بررسی قرار گرفته است تا ضرورت انجام تحقیق جاری مشخص گردد. آنگاه مدل پیشنهادی برای حل مسئله تحقیق به طور مفصل ارائه‌شده و سپس ارزیابی مدل پیشنهادی، تحلیل نتایج ارزیابی اجرای مدل، تحلیل اعتبار نتایج ارزیابی ارائه‌شده است؛ سپس نتیجه‌گیری فصل، شامل دستاوردهای تحقیق، نوآوری و کارهای آینده بیان‌شده است.

۲. روش تحقیق

روش تحقیق مبتنی بر استفاده از مدل شبکه عصبی RWKV جهت تشخیص مقالات خود متناقض و سپس ارزیابی نتایج استفاده از آن است. این شبکه عصبی تاکنون در تشخیص مقالات فارسی خود متناقض مورداستفاده قرار نگرفته است.

۲-۱. کارهای مرتبط پیشین

در مطالعات پیشین، روش‌های مختلفی برای تشخیص تناقض در زبان‌های کم منبع معرفی‌شده است. رحیمی و شمس‌فرد[۱] باتوجه‌به کمبود داده‌های مناسب برای یادگیری عمیق در زبان فارسی، یک سیستم مبتنی بر قانون و همچنین یک سیستم یادگیری عمیق بر پایه BERT را برای شناسایی تناقض در متون فارسی پیشنهاد کردند. سیستم مبتنی بر قانون با امتیاز F1 حدود ۷۶٪ عملکرد بهتری نسبت به روش‌های دیگر داشت.

وارگاس[3] و همکاران[۴] رویکردی مبتنی بر تحلیل قطبیت جملات را پیشنهاد دادند که تناقض‌ها را بر اساس تضاد قطبیت شناسایی می‌کند. توفیک و اسپرویت[4][۵] نیز یک مدل دوفازی برای تشخیص تناقض معرفی کردند که از ادعاهای بازیابی شده برای ارزیابی تناقض استفاده می‌کند.

[1] Pairwise contradiction learning

[2] Receptance Weighted Key Value (RWKV)

[3] Vargas

[4] Tawfik and Spruit

سپولودا تورس[1][۶] با استفاده از مدل BETO برای زبان اسپانیایی، توانست تناقضات معنایی بین جملات را بادقت بالایی شناسایی کند. لی[2] و همکاران[۷] با استفاده از تعبیه کلمات خاص تضاد، توانستند الگوریتمی برای تشخیص تضادها بادقت بالاتر از روش‌های مبتنی بر تعبیه‌های سنتی ایجاد کنند.

لینگام[3] و همکاران[۸] روشی را برای تشخیص تناقض در سه نوع مختلف شامل نفی، تضاد و عدم تطابق عددی پیشنهاد کردند. خدادادی و همکاران[۹] ترکیب چندین دسته‌بند برای تشخیص تناقض در متون فارسی استفاده کرده و دقت بالایی به دست آوردند.

دیبر[4] و همکاران[۱۰] از معماری ترنسفورمر برای تشخیص تناقض در گزارش‌های مالی استفاده کردند و با افزودن دانش بخش‌های گفتاری به مدل، دقت بالایی به دست آوردند. لندوای[5] و همکاران[۱۱] از ویژگی‌های شباهت متنی برای شناسایی تناقض در ادعاهای شایعات استفاده کردند و طبقه‌بندی‌کننده‌ای بر اساس نزدیکی متنی ارائه دادند.

سیفا[6] و همکاران[۱۲] با ترجمه ماشینی مجموعه‌داده استنباط زبان طبیعی استنفورد از انگلیسی به آلمانی، چهار طبقه‌بندی‌کننده را برای تشخیص تناقض ارزیابی کرده‌اند. آن‌ها اولین ارزیابی بزرگ‌مقیاس این مسئله به زبان آلمانی را انجام داده و کارایی ترجمه ماشینی برای تولید داده را تأیید کرده‌اند. همچنین، یک رویکرد جدید برای یادگیری تعبیه جملات از طریق حالت‌های پنهان مدل رمزگذار - رمزگشا ارائه داده‌اند که نتایج امیدوارکننده‌ای داشته و با تنظیم بیشتر مدل می‌تواند بهبود یابد.

خرات[7] و همکاران[۱۳] به بررسی تشخیص تناقض در عبارات نامطمئن پرداخته‌اند، جایی که کاربر علاوه بر اطلاعات واقعی، سرنخ‌هایی در مورد معقول بودن آن‌ها ارائه می‌دهد. آن‌ها رویکردی برای تشخیص تناقض با معیارهای اضافی جهت شناسایی ویژگی‌های معنایی زبانی و شواهد عدم قطعیت معرفی کرده‌اند. این رویکرد با تحلیل روابط معنایی و ارزیابی عدم قطعیت‌ها، تناقض‌ها را شناسایی می‌کند و بر روی چهار مجموعه‌داده آزمایش‌شده است که نتایج نشان‌دهنده اثربخشی آن است.

دنتان و موشک آزاد نیری[8][۱۴] به چالش‌های پردازش متن چندزبانه اشاره کرده‌اند، به‌ویژه در زبان‌هایی با داده‌های کم. آن‌ها الگوریتم‌هایی را برای طبقه‌بندی جفت جمله‌ها در سه کلاس مستلزم، تناقض و خنثی، باهدف به حداکثر رساندن دقت وزنی، آموزش داده‌اند. از مدل‌های زبان بزرگ مانند BERT و XLM-R استفاده کرده و با افزودن ویژگی‌های جدید، دقت پیش‌بینی‌ها را بهبود داده‌اند. سپس از طبقه‌بندی XGBoost برای ترکیب پیش‌بینی‌ها استفاده کرده و با استراتژی انباشتگی، دقت را ۳ درصد نسبت به خط پایه افزایش داده‌اند.

اسماعیل بداچه[9] و همکاران[۱۵] از قطبیت مرورها برای آموزش طبقه‌بندی‌کننده‌های نظارت شده به‌منظور پیش‌بینی شدت تناقض در بررسی‌های آیتم‌ها استفاده کرده‌اند. آن‌ها بر تشخیص جنبه‌ها و تحلیل احساسات این جنبه‌ها تمرکز دارند و با استفاده از ویژگی‌هایی مانند رتبه‌بندی و قطبیت جنبه‌ها، شدت تناقض را پیش‌بینی می‌کنند. الگوریتم J48 در مقایسه با بیز ساده و ماشین بردار پشتیبان عملکرد بهتری داشته است. بااین‌حال، وابستگی به کیفیت مدل‌های احساسات و جنبه‌ها یکی از نقاط ضعف این رویکرد است.

علمری[10] و استیونسون[11][۱۶] سیستمی خودکار برای تشخیص تناقضات بین ادعاهای علمی با استفاده از ماشین بردار پشتیبان و استخراج ویژگی‌های زبانی ارائه می‌دهند. این سیستم که بر روی مجموعه داده‌ای شامل ۲۵۸ ادعا از ۲۴ گروه آزمایش‌شده است، به محققان کمک می‌کند تا به‌صورت بصری ادعاهای متناقض را در بازبینی‌های سامان‌مند شناسایی کنند. طبقه‌بندی‌کننده آن‌ها به‌دقت ۸۷/۳٪ و ROC برابر با ۸۹٪ دست‌یافت که به طور قابل‌توجهی از خط پایه ۶۸٪ بهتر است.

۲-۲. مدل پیشنهادی

تشخیص تناقض یکی از وظایف مهم در پردازش متن است و در کاربردهایی نظیر استنتاج متنی و تحلیل احساسات نقش اساسی دارد. تناقض نشان‌دهنده تضاد معنایی بین دو جمله است و شامل انواع تقابل‌ها مانند تضاد، نفی، و اختلافات عددی یا ساختاری می‌شود. این مفهوم می‌تواند در سطح کلمه یا جمله رخ دهد و به دسته‌بندی‌های مختلفی ازجمله تناقض‌های واژگانی، نفی، عددی، و دانش جهانی تقسیم شود. در مقالات خود متناقض، جملات در تضاد با یکدیگر قرار دارند. در دانشنامه‌ها معمولاً انتظار می‌رود این تناقض‌ها شناسایی شوند، اما ابزاردقیقی برای تشخیص آن‌ها وجود ندارد. تاکنون روش‌های متعددی برای تشخیص تناقض در متن، اغلب با استفاده از شبکه‌های عصبی، پیشنهادشده است. اما این روش‌ها معمولاً پیچیدگی بالایی دارند و ازنظر دقت و سرعت عملکرد مطلوبی ارائه نمی‌دهند.

در این پژوهش، یک مدل جدید با استفاده از شبکه عصبی RWKV برای تشخیص مقالات خود متناقض در دانشنامه

[1] Sepúlveda-Torres
[2] Li
[3] Lingam
[4] Deußer
[5] Lendvai
[6] Sifa
[7] Kharrat
[8] Dentan and A. Rakotonirainy
[9] Badache
[10] Alamri
[11] Stevensony

فارسی ارائه‌شده است. علت اصلی استفاده از شبکه عصبی RWKV برای تشخیص مقالات خود متناقض پایین‌بودن درجه پیچیدگی آن است که جدول (۱) آن را نشان می‌دهد.

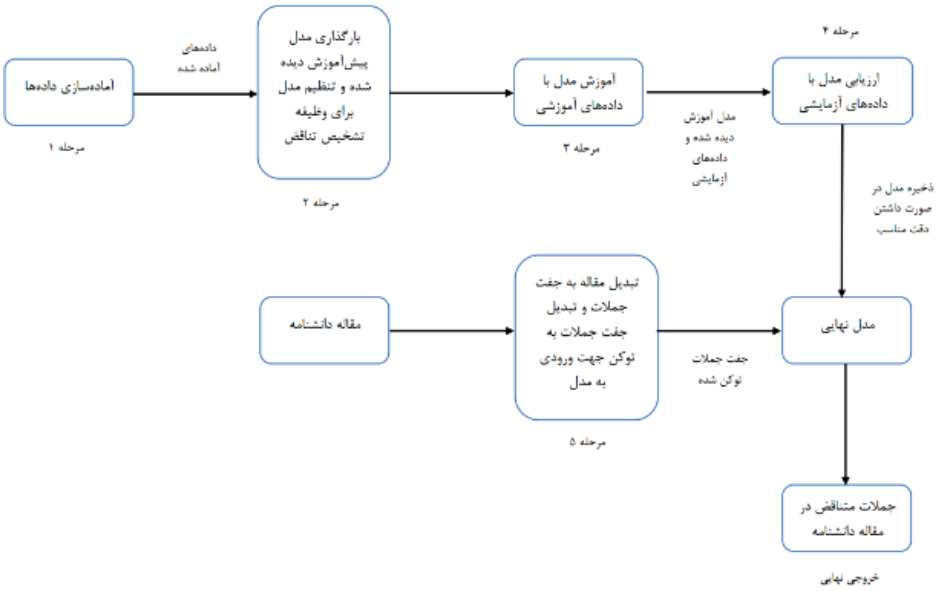
جدول (۱). پیچیدگی زمانی و حافظه‌ای مدل‌های مختلف[۳]

مدل	پیچیدگی زمانی	پیچیدگی حافظه
Transformer	$O(T^2d)$	$O(T^2 + Td)$
Reformer	O(TlogTd)	O(TlogT + Td)
Linear Transformer	$O(Td^2)$	$O(Td + d^2)$
Performer	$O(Td^2logd)$	$O(Tdlogd + d^2logd)$
AFT-full	$O(T^2d)$	O(Td)
MEGA	O(cTd)	O(cTd)
RWKV	O(Td)	O(d)

شبکه عصبی RWKV با پیچیدگی خطی، بهبودهایی در دقت و سرعت تشخیص تناقض نسبت به روش‌های پیشین نشان می‌دهد. مدل RWKV به دلیل برخورداری از پیچیدگی زمانی و حافظه‌ای خطی، قابلیت مقیاس‌پذیری بهتری دارد که منجر به افزایش دقت در عملکرد آن می‌شود که همین امر سبب تمایز آن با دیگر شبکه‌ها شده و به همین دلیل به‌عنوان راه‌حل برای این موضوع به‌کارگرفته‌شده است.

شبکه عصبی RWKV ، بهره‌وری موازی‌سازی ترانسفورمرها را با کارایی استنتاجی شبکه‌های عصبی بازگشتی ترکیب کرده و نقاط ضعف هر دو را پوشش می‌دهد.

معماری روش پیشنهادی در شکل (۱) ارائه‌شده است.



شکل (۱). معماری مدل پیشنهادی

در ادامه اجزاء معماری مدل پیشنهادی تشریح می‌شود.

۲-۲-۱- آماده‌سازی داده‌ها

اولین جزء در معماری مدل جدید پیشنهادی آماده‌سازی داده‌هاست، یعنی در این معماری اولین گام تبدیل برچسب‌های متنی به اعداد صحیح است. این برچسب‌ها که نوع رابطه بین دو جمله پیش‌فرض و فرضیه را نشان می‌دهند، به‌صورت دسته‌بندی‌های «استنتاج» (e)، «تناقض» (c) و «خنثی» (n) تعریف‌شده‌اند. برای اینکه مدل بتواند با این داده‌ها کار کند، برچسب‌ها به ترتیب به اعداد ۰، ۱، و ۲ تبدیل می‌شوند. سپس از توکن‌سازی[1] مدل پیش آموزش‌دیده شده شبکه عصبی RWKV برای پردازش داده‌های متنی استفاده می‌شود. توکن‌سازی وظیفه دارد جملات ورودی را به واحدهای کوچک‌تری به نام توکن تبدیل کند. توکن‌ساز در مدل شبکه عصبی RWKV ، جملات ورودی را به توکن‌های مربوط تقسیم کرده و هر توکن را به یک عدد مرتبط در واژگان مدل نگاشت می‌کند. این توکن‌ها می‌توانند شامل کلمات، بخش‌هایی از کلمات یا حتی شخصیت‌ها باشند.

۲-۲-۲- بارگذاری مدل و تنظیم وظیفه تشخیص تناقض

در مرحله بارگذاری، مدل پیش آموزش‌دیده RWKV از مخزن مدل‌ها بارگذاری می‌شود. شبکه عصبی RWKV یک معماری جدید از مدل‌های زبانی بزرگ است که شبکه‌های بازگشتی و شبکه‌های ترانسفورمر را ترکیب می‌کند. این مدل به طور خاص برای پردازش دنباله‌های طولانی با حفظ اطلاعات کلیدی طراحی‌شده است.

شبکه عصبی RWKV از یک ساختار شبیه به شبکه‌های بازگشتی برای حفظ و انتقال اطلاعات از دنباله‌های قبلی به دنباله‌های بعدی استفاده می‌کند. برخلاف ترانسفورمرها که تمام

[1] Tokenizer

ورودی‌های دنباله را به‌صورت موازی پردازش می‌کنند، شبکه عصبی RWKV با استفاده از سازوکارهای بازگشتی مشابه شبکه عصبی بازگشتی، هر توکن را به‌صورت ترتیبی پردازش می‌کند.

- پیاده‌سازی کلاس RwkvForSequenceClassification

این کلاس به طور خاص برای انجام طبقه‌بندی دنباله‌ای طراحی‌شده است. مدل‌های زبانی بزرگ مانند RWKV معمولاً برای تولید متن یا پردازش زبان طبیعی عمومی به کار می‌روند، اما برای وظایف خاصی مانند تشخیص تناقض در جملات، نیاز به تنظیمات خاص دارند. در این مرحله، مدل RWKV برای طبقه‌بندی دنباله‌های ورودی به دسته‌های مختلف (مانند تناقض، مطابقت، یا خنثی بودن جملات) تنظیم می‌شود.

این در مدل RWKV برای انجام وظایف طبقه‌بندی دنباله طراحی‌شده است. این کلاس بر اساس معماری اصلی مدل RWKV است و بهینه‌سازی شده تا بتواند برای مسائل طبقه‌بندی، مانند تشخیص تناقض، به کار گرفته شود. در این بخش، به طور کامل اجزای این کلاس و چگونگی استفاده از آن را توضیح می‌دهم.

ورودی کلاس، یک دنباله از توکن‌های ورودی است که از طریق توکنایزر به‌صورت یک بردار از نمایه‌ها تبدیل‌شده‌اند. هر نمایه نمایانگر یک کلمه یا توکن است. برای یک جمله یا دنباله‌ای از جملات، توکن‌ها با توابعی نظیر padding و truncation (در توکنایزر) تنظیم‌شده‌اند.

- استفاده از PEFT و LoRA

PEFT و LoRA فنون بهینه‌سازی هستند که برای بهبود عملکرد مدل‌ها و کاهش تعداد پارامترهای قابل‌آموزش به کار می‌روند.

PEFT، روشی است که تنها بخشی از پارامترهای مدل برای وظیفه جدید تنظیم می‌شوند. به‌جای تنظیم کامل تمام پارامترهای مدل، تنها چندلایه خاص (در اینجا لایه طبقه‌بندی) به‌صورت مجدد آموزش داده می‌شوند. این روش مصرف منابع محاسباتی را کاهش داده و فرایند تنظیم مدل را سریع‌تر می‌کند.

LoRA، یک فن تنظیم پارامتر است که با کاهش تعداد پارامترهای قابل‌تنظیم در مدل، فرایند آموزش را ساده‌تر و بهینه‌تر می‌کند. به‌جای تنظیم کامل ماتریس‌های بزرگ وزن در لایه‌های مدل، از ساختارهای کم رتبه استفاده می‌شود.

۲-۲-۳- آموزش مدل

برای آموزش این مدل، از مجموعه‌داده FarsTail استفاده می‌کنیم که شامل نمونه‌های متنوعی از جملات و عبارات با برچسب‌های معنایی مختلف است. برای بهینه‌سازی عملکرد مدل و کاهش پیچیدگی محاسباتی، از فن PEFT و به طور خاص از LoRA بهره گرفته می‌شود.

ابتدا داده‌های مجموعه‌داده موردنظر آماده‌سازی می‌شوند. سپس، مدل پیش آموزش‌دیده بارگذاری شده و برای وظیفه طبقه‌بندی ترتیبی تنظیم می‌شود. در مرحله بعد، مدل با داده‌های آموزشی از مجموعه‌داده تمرین داده شده و پس از آن با استفاده از داده‌های آزمایشی ارزیابی می‌شود. پس از دستیابی به مدل نهایی، یک مقاله از دانشنامه فارسی با استفاده از کتابخانه هضم به جفت جملات تبدیل می‌شود. این جفت جملات به مدل نهایی داده شده و جملاتی که باهم در تناقض باشند، توسط مدل تشخیص داده می‌شوند.

این اولین باری است که از این مدل برای تشخیص مقالات خود متناقض استفاده‌شده و نوآوری این پژوهش در استفاده از RWKV در این حوزه است.

راه‌حل پیشنهادی برای تشخیص مقالات خود متناقض که معماری آن در شکل (۱) نشان‌داده‌شده است، مبتنی بر استفاده از مدل شبکه عصبی RWKV است. این مدل، ترکیبی از موازی‌سازی ترنسفورمرها و کارایی شبکه‌های عصبی بازگشتی است که نقاط ضعف هر دو را پوشش می‌دهد. زمانی و حافظه‌ای خطی، مقیاس‌پذیری بهتری دارد و دقت بیشتری در عملکرد ارائه می‌دهد.

برای آموزش این مدل، از مجموعه‌داده FarsTail [۱۷] استفاده می‌شود که شامل جملات و عبارات با برچسب‌های معنایی مختلف است. برای بهینه‌سازی عملکرد و کاهش پیچیدگی محاسباتی، از فن تنظیم کارآمد جزئی[1] [۱۸] به‌ویژه از تطبیق کم رتبه[2] [۱۹] بهره گرفته می‌شود.

در ابتدا، داده‌های موردنظر آماده‌سازی می‌شوند. سپس، مدل پیش آموزش‌دیده بارگذاری شده و برای وظیفه طبقه‌بندی ترتیبی تنظیم می‌شود. مدل با داده‌های آموزشی تمرین داده می‌شود و پس از آن با استفاده از داده‌های آزمایشی ارزیابی می‌گردد. پس از دستیابی به مدل نهایی، یک مقاله از دانشنامه فارسی به جفت جملات تبدیل می‌شود و جملاتی که باهم در تناقض‌اند، توسط مدل تشخیص داده می‌شوند.

در بخش آماده‌سازی داده‌ها، اولین گام تبدیل برچسب‌های متنی به اعداد صحیح است. این برچسب‌ها که نوع رابطه بین دو جمله پیش‌فرض و فرضیه را نشان می‌دهند، به‌صورت دسته‌بندی‌های «استنتاج» (e)، «تناقض» (c) و «خنثی» (n) تعریف‌شده‌اند. برای اینکه مدل بتواند با این داده‌ها کار کند، برچسب‌ها به ترتیب به اعداد ۰، ۱، و ۲ تبدیل می‌شوند. این تبدیل به این دلیل است که شبکه‌های عصبی با داده‌های متنی به‌صورت مستقیم کار نمی‌کنند و همه ورودی‌ها باید به قالب

[1] Parameter-Efficient Fine-Tuning
[2] Low-Rank Adaption

عددی تبدیل شوند. سپس از توکن‌ساز[1] مدل پیش آموزش‌دیده شده RWKV برای پردازش داده‌های متنی استفاده می‌کنیم.

در مرحله بارگذاری، مدل پیش آموزش‌دیده RWKV از مخزن مدل‌ها بارگذاری می‌شود. RWKV یک معماری جدید از مدل‌های زبانی بزرگ است که شبکه‌های بازگشتی و شبکه‌های ترنسفورمر را ترکیب می‌کند. این مدل به طور خاص برای پردازش دنباله‌های طولانی با حفظ اطلاعات کلیدی طراحی‌شده است.

مدل RWKV از یک ساختار شبیه به شبکه‌های بازگشتی برای حفظ و انتقال اطلاعات از دنباله‌های قبلی به دنباله‌های بعدی استفاده می‌کند. برخلاف ترنسفورمرها که تمام ورودی‌های دنباله را به‌صورت موازی پردازش می‌کنند، RWKV با استفاده از سازوکارهای بازگشتی مشابه شبکه عصبی بازگشتی، هر توکن را به‌صورت ترتیبی پردازش می‌کند.

محاسبات بازگشتی مدل به شرح زیر است:

- فرض کنید $X = [x_1, x_2, \dots, x_n]$ دنباله‌ای از ورودی‌ها باشد.
- در هر گام زمانی t وضعیت پنهان جدید h_t به‌صورت زیر محاسبه می‌شود:

که در آن f تابع غیرخطی است که توکن فعلی x_t را با وضعیت پنهان قبلی h_{t-1} ترکیب می‌کند.

- خروجی مدل در هر زمان t به‌صورت زیر خواهد بود:

$$g(h_t) = y_t$$

که g تابعی است که وضعیت پنهان h_t را به خروجی نگاشت می‌دهد.

این ساختار به مدل اجازه می‌دهد تا اطلاعات طولانی‌مدت را حفظ کرده و به طور مؤثری روابط معنایی در دنباله‌های طولانی‌تر را درک کند.

در مرحله بعد یک کلاس برای انجام‌وظیفه طبقه‌بندی طراحی‌شده است. مدل‌های زبانی بزرگ مانند RWKV معمولاً برای تولید متن یا پردازش زبان طبیعی عمومی به کار می‌روند، اما برای وظایف خاصی مانند تشخیص تناقض در جملات، نیاز به تنظیمات خاص دارند. در این مرحله، مدل RWKV برای طبقه‌بندی دنباله‌های ورودی به دسته‌های مختلف (مانند تناقض، مطابقت، یا خنثی بودن جملات) تنظیم می‌شود؛ براین‌اساس و برای آماده‌سازی شرایط استفاده از RWKV جهت تشخیص تناقض در متون فارسی، نیاز است در مدل RWKV، کلاسی برای طبقه‌بندی دنباله‌های ورودی به دسته‌های مختلف طراحی شود. این کلاس بر اساس معماری اصلی مدل RWKV بهینه‌سازی شده تا بتواند برای مسائل طبقه‌بندی، مانند تشخیص تناقض، به کار گرفته شود. در ادامه اجزای این کلاس و چگونگی استفاده از آن به طور کامل توضیح داده می‌شود.

این کلاس شامل چندلایه اصلی است که هدف آن طبقه‌بندی دنباله‌ای از توکن‌هاست. لایه تعبیه اولین لایه این مدل است و وظیفه دارد توکن‌های ورودی را به بردارهای تعبیه‌شده تبدیل کند. این توکن‌ها در ورودی توسط ماتریس تعبیه به بردارهایی با ابعاد ثابت تبدیل می‌شوند. به طور خاص، ماتریس تعبیه ابعاد $d \times V$ دارد که در آن V تعداد واژگان و d ابعاد بردار تعبیه است.

لایه‌های اصلی مدل RWKV شامل دو بخش کلیدی هستند: لایه کلید - مقدار و لایه بازگشتی. برخلاف مدل‌های ترنسفورمر که معمولاً از سازوکار توجه استفاده می‌کنند، RWKV باتکیه‌بر سازوکار بازگشتی توکن‌ها را به‌صورت ترتیبی پردازش می‌کند. لایه کلید - مقدار وزن‌هایی را بر اساس موقعیت هر توکن در دنباله اعمال می‌کند، درحالی‌که لایه بازگشتی اطلاعات توکن‌های قبلی را ذخیره کرده و امکان یادآوری اطلاعات گذشته را فراهم می‌کند. این لایه، حالت‌های مخفی را به‌صورت ترتیبی پردازش کرده و بردارهای خروجی را تولید می‌کند.

در انتها، خروجی لایه بازگشتی مربوط به اولین توکن (معمولاً [CLS]) به لایه طبقه‌بندی منتقل می‌شود. این لایه، با استفاده از ماتریس وزن و بایاس، احتمال‌های رده‌های مختلف را از طریق تابع softmax محاسبه می‌کند. در فرایند آموزش، مدل با استفاده از تابع زیان تلاش می‌کند اختلاف بین توزیع احتمالات پیش‌بینی‌شده و توزیع واقعی برچسب‌ها را به حداقل برساند. برچسب‌های ورودی معمولاً رده‌های مرتبط با یک وظیفه طبقه‌بندی، مانند تشخیص در این کار از روش‌های بهینه‌سازی تنظیم کارآمد جزئی و تطبیق هدف اصلی این روش‌ها کاهش مصرف منابع محاسباتی و بهینه‌سازی فرایند تنظیم مدل‌های بزرگ است. این فن‌ها با تنظیم دقیق و سریع‌تر پارامترها به استفاده بهینه‌تر از منابع سخت‌افزاری کمک می‌کنند.

تنظیم کارآمد جزئی روشی است که در آن تنها بخشی از پارامترهای مدل برای وظیفه جدید به‌روزرسانی می‌شوند. به‌جای تنظیم همه پارامترهای مدل، تنها چندلایه خاص، مانند لایه طبقه‌بندی، آموزش داده می‌شوند. این رویکرد علاوه بر کاهش مصرف منابع محاسباتی، فرایند تنظیم مدل را سریع‌تر می‌کند و به مدل اجازه می‌دهد که برای وظایف مختلف بدون نیاز به تغییرات گسترده، مجدداً استفاده شود.

تنظیم کم رتبه با کاهش تعداد پارامترهای قابل‌تنظیم، کارایی مدل را افزایش داده و فرایند آموزش را ساده‌تر می‌کند. در این روش، به‌جای تنظیم مستقیم ماتریس‌های بزرگ وزن، از ساختارهای کم رتبه استفاده می‌شود. به‌این‌ترتیب، وزن هر لایه به‌صورت جمع یک ماتریس ثابت و یک ماتریس کم رتبه تغییرات

[1] Tokenizer

تعریف می‌شود که از حاصل‌ضرب دو ماتریس کوچک‌تر به دست می‌آید. این ساختار باعث می‌شود تنها تغییرات ضروری در پارامترها به‌روزرسانی شده و بقیه ثابت بمانند که به بهبود کارایی و کاهش نیاز به منابع محاسباتی منجر می‌شود و فرایند تنظیم مدل‌های بزرگ را بهینه‌تر می‌کند.

پس از بارگذاری مدل، آموزش مدل RWKV با داده‌های آموزشی آغاز می‌شود. برای آموزش مدل لازم است آرگومان‌های آموزشی مقداردهی شوند، مقادیر هر آرگومان در جدول (۲) آورده شده است.

جدول (۲). آرگومان‌های آموزشی مدل

آرگومان‌های آموزش	مقدار
evaluation_strategy	epoch
learning_rate	2e-5
Train batch size	32
Eval batch size	32
Number of epochs	5
Weight Decay	0.03
FP16 Precision	true

پس از مشخص‌کردن این آرگومان‌ها آموزش مدل آغاز می‌شود. مدل با استفاده از داده‌های آماده‌شده و پارامترهای آموزشی مشخص‌شده به مدت ۵ ای پاک آموزش داده می‌شود. پس از هر ای پاک، مدل بر روی مجموعه‌ای از داده‌های اعتبارسنجی ارزیابی می‌شود و پارامترهای آن به‌روز می‌شوند. درنهایت، مدل آموزش‌دیده ذخیره می‌شود و برای استفاده در کاربردهای واقعی (تشخیص تناقضات در متون فارسی) آماده می‌شود.

در این مرحله، هدف تبدیل جملات مقاله به ورودی‌هایی برای مدل است تا بتواند تناقض‌ها را پیش‌بینی کند. فرایند پیاده‌سازی شامل چند مرحله اصلی است. ابتدا مقاله بارگذاری و به جملات جداگانه تقسیم می‌شود. این کار با استفاده از کتابخانه‌ی «هضم» انجام می‌گیرد که امکان خواندن محتوای فایل و سپس تقسیم آن به جملات مجزا را فراهم می‌کند تا بتوان هر جمله را به طور جداگانه پردازش کرد.

پس از آماده‌سازی جملات، مدل و توکن‌ساز مناسب بارگذاری می‌شوند. در این مرحله، بررسی می‌شود که آیا GPU در دسترس است تا در صورت موجود بودن، مدل به GPU منتقل شود و سرعت پردازش افزایش یابد. سپس مدل به حالت ارزیابی قرار می‌گیرد تا از بروز واریانس‌های تصادفی حین پیش‌بینی جلوگیری شود. مرحله بعدی، ایجاد جفت جملات از طریق حلقه‌های تودرتو است؛ در این روش، هر جمله با جملات بعدی ترکیب می‌شود و از پردازش جفت‌های تکراری با کمک یک مجموعه جلوگیری می‌شود.

در گام بعد، جفت‌های جملات توسط توکن‌ساز به توکن‌های قابل‌فهم برای مدل تبدیل می‌شوند. سپس این توکن‌ها در صورت وجود GPU به آن منتقل می‌شوند. مدل پیش‌بینی لازم را روی این توکن‌ها انجام داده و آمار خام به دست می‌آید. با تابعی در کتابخانه‌ی تورچ[1]، این آمار به رده‌های پیش‌بینی‌شده تبدیل می‌شوند که نشان‌دهنده روابط بین جملات است. درنهایت، اگر کلاس پیش‌بینی‌شده برابر با تناقض باشد، جملات متناقض به‌عنوان خروجی نمایش داده می‌شوند.

٤-۲-۲- تبدیل مقاله به جفت جملات جهت ورودی به مدل

در این مرحله، هدف تبدیل جملات مقاله به ورودی‌هایی برای مدل است تا پیش‌بینی‌های لازم برای تشخیص تناقض‌ها انجام شود. کار پیاده‌سازی شده در این مرحله شامل چندین مرحله اصلی است که هرکدام به‌دقت مورد بررسی قرار می‌گیرد.

- **بارگذاری مقاله و تبدیل آن به آرایه‌ای از جملات**

این مرحله شامل بارگذاری یک فایل متنی و تقسیم آن به جملات است. برای این کار از کتابخانه هضم استفاده می‌شود. فرایند به شرح زیر است:

  - خواندن فایل متنی: محتوای فایل متنی بارگذاری می‌شود.
  - تقسیم متن به جملات: با استفاده از تابع sent_tokenize کتابخانه هضم، متن به جملات جداگانه تقسیم می‌شود. این کار به ما امکان می‌دهد هر جمله را به طور جداگانه پردازش کنیم.

- **بارگذاری مدل و ایجاد توکن**

پس از بارگذاری جملات، مدل و توکن‌های ایجادشده بارگذاری می‌شوند. در این مرحله بررسی می‌شود که آیا GPU در دسترس است یا خیر. اگر در دسترس باشد، مدل به GPU منتقل می‌شود. در اینجا، از مدل آموزش‌داده‌شده خود و توکنایزر آن را که برای تبدیل جملات مقاله به توکن است بارگذاری می‌کنیم. پس از بارگذاری مدل، حالت آن را به حالت ارزیابی تغییر می‌دهیم تا از واریانس‌های تصادفی در حین پیش‌بینی جلوگیری شود.

- **ایجاد جفت جملات**

در این مرحله، هدف ایجاد جفت‌های مختلف از جملات است. این کار شامل مراحل زیر می‌شود:

[1] Torch

- استفاده از دو حلقه تودرتو: از یک حلقه برای انتخاب جمله اصلی و از حلقه دیگر برای انتخاب فرضیه استفاده می‌شود. برای هر جمله، جملات بعدی به‌عنوان جفت‌های ممکن در نظر گرفته می‌شوند.
- کنترل تکراری بودن جفت‌ها: با استفاده از مجموعه‌ای به نام processed_pairs، اطمینان حاصل می‌شود که جفت‌های تکراری پردازش نشوند. به‌عبارت‌دیگر، هر جفت جمله فقط یک‌بار مورد پردازش قرار می‌گیرد.

- **توکن‌سازی جفت جملات**

پس از ایجاد جفت‌های جملات، توکنایزر مدل برای تبدیل این جفت‌ها به توکن‌ها استفاده می‌شود. این مرحله شامل موارد زیر است:

- توکنایز کردن جملات: از توکنایزر برای تبدیل جملات اصلی و جملات فرضیه به توکن‌های قابل‌فهم برای مدل استفاده می‌شود. این فرایند شامل پارامترهایی مانند پدینگ و برش‌دادن است.
- انتقال ورودی‌ها به GPU: داده‌های ورودی در صورت وجود GPU به GPU منتقل می‌شوند تا سرعت پردازش افزایش یابد.

- **پیش‌بینی برچسب‌ها**

مدل با استفاده از توکن‌های ورودی پیش‌بینی‌های لازم را انجام می‌دهد. این مرحله شامل موارد زیر است:

- پیش‌بینی با استفاده از مدل: با استفاده از model(**inputs) که input همان جملات توکنایز شده است پیش‌بینی انجام می‌شود و logits (آمار خام) به‌دقت می‌آید.
- تبدیل logits به رده‌های پیش‌بینی‌شده: با استفاده از تابع argmax (تابعی در کتابخانه تورچ)، رده‌های پیش‌بینی‌شده از logits استخراج می‌شوند. این کلاس‌ها نشان‌دهنده روابط بین جملات هستند.

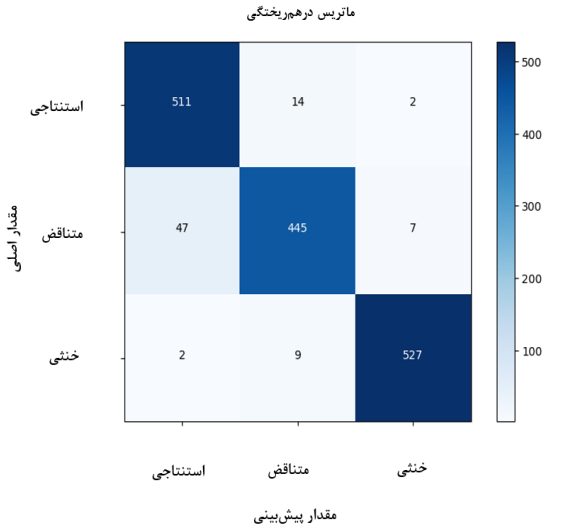
درنهایت نتایج پیش‌بینی‌ها برای نمایش جملات متناقض مورد بررسی قرار می‌گیرند. با استفاده از یک حلقه اگر پیش‌بینی مدل برابر باکلاس c یا همان تناقض باشد. جملات مرتبط به‌عنوان خروجی نمایش داده می‌شوند. در ادامه کد ایجاد جفت جملات، توکنایز کردن جملات و پیش‌بینی برچسب‌ها آورده شده است.

۳. ارزیابی مدل پیشنهادی

ارزیابی مدل یکی از مهم‌ترین مراحل در فرایند توسعه یک مدل یادگیری ماشین است. در این مرحله، هدف این است که عملکرد مدل بر روی داده‌های جدید و نادیده گرفته‌شده (که در مرحله آموزش مورداستفاده نبوده‌اند) سنجیده شود تا اطمینان حاصل شود که مدل به‌درستی تعمیم‌یافته است. ارزیابی معماری پیشنهادی به شرح زیر است:

در ارزیابی مدل تشخیص مقالات خود متناقض، ابتدا داده‌های مجموعه‌داده FarsTail بارگذاری و پیش‌پردازش می‌شوند. این مجموعه شامل جملات فارسی با سه دسته معنایی Entailment (نتیجه‌گیری)،Contradiction (تناقض) و Neutral (خنثی) است. برچسب‌های متنی به اعداد تبدیل می‌شوند و مدل پیش آموزش‌دیده برای تشخیص تناقض بارگذاری می‌شود.

مدل با داده‌های آموزشی آموزش‌دیده و عملکرد آن با استفاده از داده‌های آزمایشی ارزیابی می‌شود. یکی از ابزارهای اصلی برای ارزیابی، ماتریس درهم‌ریختگی است که در شکل (۲) نشان‌داده‌شده است و شامل True Positives (TP)، False Positives (FP)، True Negatives (TN) و False Negatives (FN) است.



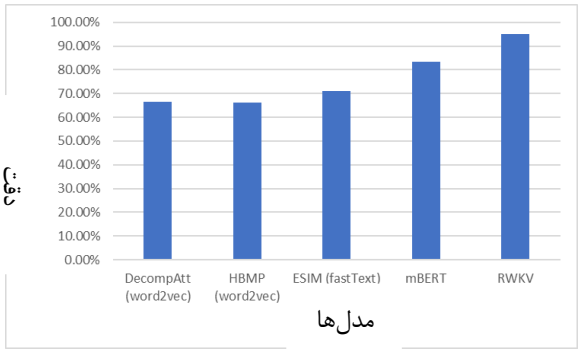
شکل (۲). ماتریس درهم‌ریختگی

در این ارزیابی، مقادیر ماتریس مدل برای کلاس Contradiction در جدول (۲) آورده شده است.

جدول (۱). مقادیر TP، FP، TN و FN برای کلاس متناقض

نام	مقدار
True Positives	۴۴۵
False Positives	۲۳
True Negatives	۱۰۳۸
False Negatives	۵۴

مدل RWKV در مقایسه با سایر مدل‌ها دقت بالاتری دارد که در شکل (۳) نشان‌داده‌شده است.

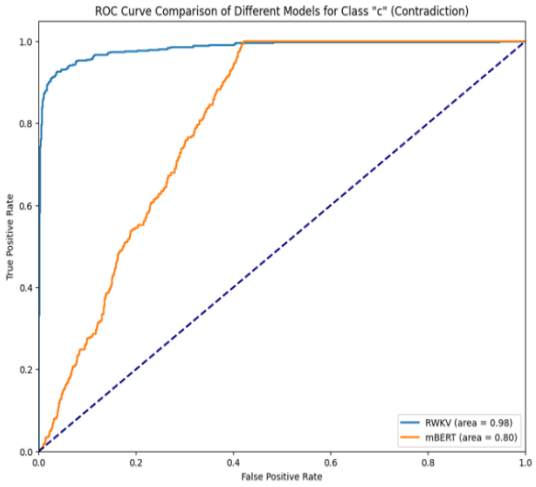


شکل (۳). مقایسه دقت مدل RWKV با سایر مدل‌ها

در شکل (۴) نمودار ROC که به‌طورکلی برای ارزیابی مدل‌های طبقه‌بندی استفاده می‌شود، نشان می‌دهد که چگونه مدل در تفکیک رده‌های مثبت و منفی عمل می‌کند. در این نمودار محور عمودی نشان‌دهنده نرخ مثبت واقعی و محور افقی نشان‌دهنده نرخ مثبت کاذب است.

در نمودار ROC مقایسه بین مدل RWKV و mBERT (که دقت بالاتری را نسبت به سایر مدل‌های دیگر دارد) انجام‌گرفته است و نتایج مقایسه چنین است:

- مدل RWKV (خط آبی)، این مدل عملکرد بسیار خوبی دارد، همان‌طور که از منحنی ROC آن مشخص است. نمودار به نزدیکی گوشه بالا - چپ نزدیک شده و مساحت زیر منحنی = AUC ۰.۹۸ بسیار بالا است. این نشان می‌دهد که مدل RWKV قدرت بسیار خوبی در تشخیص کلاس تناقض دارد و تقریباً تمام نمونه‌های این کلاس را با خطای بسیار کم شناسایی می‌کند. این یعنی نرخ خطای مثبت کاذب پایین و نرخ مثبت واقعی بسیار بالاست.



شکل (۴). نمودار ROC برای مدل RWKV و mBERT

- مدل mBERT (خط نارنجی)، نسبت به RWKV ضعیف‌تر است. نمودار ROC آن با شیب کمتر و مساحت زیر منحنی = AUC ۰.۸۰ نشان می‌دهد که این مدل در مقایسه با RWKV قدرت کمتری در تفکیک کلاس تناقض دارد. این مدل نرخ مثبت کاذب بالاتری دارد و در تفکیک موارد درست و نادرست دقت کمتری از خود نشان داده است.

- **تحلیل نتایج ارزیابی اجرای مدل**

نتایج ارزیابی اجرای مدل نشان می‌دهد که در ارزیابی نهایی روی یک متن واقعی، مدل پیشنهادی، از ۱۰ تناقض موجود در متن ورودی، ۸ مورد به‌درستی تشخیص داده شد، یک کورد از جملات غیر متناقض به‌اشتباه به‌عنوان متناقض تشخیص داده شد و یک مورد تناقض شناسایی نشد. باتوجه‌به نقش هر یک اجزای مدل پیشنهادی در تشخیص تناقض، نتایج ارزیابی اجرای مدل در دو بعد (فرایند جفت‌سازی و میزان داده‌های آموزشی مدل) به شرح زیر قابل‌تحلیل است:

- **فرایند جفت‌سازی**

فرایند جفت‌سازی، جملات نقش کلیدی در عملکرد مدل‌دارند زیرا مدل تنها بر اساس جفت‌هایی که به آن ارائه می‌شود، تناقض‌ها را تشخیص می‌دهد. عدم دقت یا تنوع کافی در این مرحله می‌تواند منجر به کاهش توانایی مدل در شناسایی تناقض‌های پیچیده شود. در مدل پیشنهادی که مبنای تشخیص تناقض، تشکیل و استفاده از جفت جملات است، مشکل اصلی در مواردی که تشخیص تناقض، موفق نبوده به فرایند ساخت جفت جملات برمی‌گردد. اساس پیدایش این مشکل می‌تواند چنین باشد:

- **پوشش ناکامل جملات متناقض در فرایند جفت‌سازی جملات:** زمانی که جملات ورودی به‌درستی به جفت‌های مرتبط و متناقض تقسیم نشوند، ممکن است ارتباط معنایی یا تناقض میان جملات نادیده گرفته شود و این امر موجب شود مدل فرصت تحلیل تناقض در برخی از موارد را از دست بدهد.
- **ساخت نادرست جفت‌ها برای جملات طولانی یا پیچیده:** جملات پیچیده یا دارای چند گزاره ممکن است به‌صورت ناقص به جفت‌های مناسب شکسته شود؛ به همین دلیل، ممکن است تنها بخشی ازجمله را تحلیل کرده و بخش متناقض آن جمله را نادیده بگیرد و این وضعیت منجر به عدم تشخیص تناقض‌ها شود.
- **همپوشانی نامناسب جملات در جفت‌سازی:** اگر یک جمله متناقض با بیش از یک جمله دیگر در متن مرتبط باشد و جفت‌سازی به‌گونه‌ای انجام شود که این ارتباطات چندگانه

در نظر گرفته نشوند، تناقض‌های وابسته به چند جمله نیز شناسایی نمی‌شوند.

- **میزان داده‌های آموزشی مدل**

اگر در فرایند آموزش مدل، موفقیت کامل حاصل شود، نتایج بهتری در تشخیص تناقض‌ها به دست می‌آید؛ زیرا یادگیری علاوه بر این‌که به پارامترهای مختلفی وابسته است، به میزان داده‌های آموزشی نیز وابسته است؛ بنابراین یکی دیگر از مواردی که تحلیل نتایج ارزیابی مدل به دست می‌دهد این است که توانایی مدل در یادگیری نیاز به تقویت دارد. یعنی استفاده از مجموعه‌های داده بزرگ‌تر و متنوع‌تر یا استفاده از فنون تقویت داده می‌تواند به بهبود توانایی مدل در یادگیری و تعمیم به نمونه‌های جدید کمک کند.

- **تحلیل اعتبار نتایج ارزیابی**

نتایج ارزیابی مدل در ارزیابی عملکردی مدل پیشنهادی باتوجه‌به موارد زیر قابل‌اعتماد است:

۱. مجموعه‌داده مورداستفاده مشتمل بر جملات فارسی با سه دسته معنایی استنتاجی، متناقض و خنثی است که هدف تحقیق یعنی تشخیص تناقض در متن‌های فارسی را کاملاً در برمی‌گیرد؛
۲. مجموعه‌داده مورداستفاده ورودی مناسب، شامل جملات متنوع فارسی است و
۳. از معیارهای مهم ارزیابی، دقت، امتیاز F1، و ROC در ارزیابی استفاده‌شده که نتایج آن را شفاف کرده و به واقع‌گرایی مدل کمک می‌کند.

۴. نتیجه‌گیری

در این پژوهش، روشی جدید برای تشخیص تناقض در متن‌های فارسی با استفاده از شبکه عصبی RWKV ارائه و بررسی شد. باتوجه‌به اهمیت شناسایی تناقض در متون، به‌ویژه در منابعی مانند دانشنامه‌های آنلاین و مقالات، رویکردی مبتنی بر مدل‌های یادگیری عمیق انتخاب شد که بتواند زبان طبیعی فارسی را به‌خوبی تحلیل کند.

پس از بررسی روش‌های موجود، مدل RWKV به‌عنوان روش پیشنهادی انتخاب شد. این مدل باتوجه‌به توانایی‌اش در مدل‌سازی دنباله‌های طولانی و بهره‌وری بیشتر ازنظر حافظه و زمان محاسباتی، بر مدل‌های سنتی مانند ترنسفورمرها برتری دارد. برای ارزیابی عملکرد این مدل، از مجموعه‌داده FarsTail که شامل جملات فارسی در سه کلاس معنایی «نتیجه‌گیری»، «تناقض»، و «خنثی» است، استفاده شد. داده‌ها بادقت پیش‌پردازش شده و مدل بر روی آن‌ها آموزش و ارزیابی شد.

نتایج ارزیابی نشان‌دهنده عملکرد بالای مدل RWKV در تشخیص تناقض بود. در مقایسه با مدل‌های دیگری مانند mBERT، مدل RWKV دقت و کیفیت بهتری از خود نشان داد. معیارهایی مانند دقت کلی، امتیاز F1، و مقایسه منحنی ROC حاکی از برتری این مدل بودند. به‌ویژه، مساحت زیر منحنی ROC (AUC) برابر ۰.۹۸ بود که نشان‌دهنده توانایی بالای مدل در تشخیص دقیق رده‌های مثبت و منفی است.

این نتایج نشان می‌دهد که مدل RWKV قابلیت عملی بالایی دارد. این پژوهش گامی مؤثر در جهت توسعه ابزارهای هوش مصنوعی برای تحلیل و پردازش زبان فارسی و تشخیص تناقضات در متون است. مدل RWKV به‌عنوان یک ابزار قدرتمند در این زمینه عملکرد، هرچند هنوز جا برای بهبود و افزایش دقت در تشخیص تناقضات پیچیده وجود دارد.

٤-١. دستاوردهای تحقیق

- بهبود عملکرد در تشخیص تناقض: مدل RWKV بادقت ۹۵/۰۶٪ و امتیاز F1 برابر با ۹۲/۰۳٪ در شناسایی تناقضات موفق عملکرد.
- پیشرفت در مدل‌سازی دنباله‌های طولانی: این مدل توانایی حفظ ارتباطات طولانی‌مدت در جملات را دارد که در تحلیل زبان‌های پیچیده؛ مانند فارسی بسیار مفید است.
- قابلیت کاربرد در شرایط واقعی: در ارزیابی‌های عملی، این مدل توانست ۸۰٪ از تناقضات را به‌درستی شناسایی کند.

٤-٢. نوآوری

نتایج بررسی‌های انجام‌گرفته در این تحقیق نشان داد که تاکنون در داخل کشور برای تشخیص تناقض در متون فارسی، کار تحقیقاتی محدودی انجام‌گرفته است که این محدود کارها از دقت کافی برخوردار نیستند.

در این تحقیق نمونه‌سازی آزمایشگاهی پس از بررسی‌های لازم بر آن شدیم که از شبکه عصبی مناسب RWKV به‌عنوان یک مدل یادگیری عمیق برای تشخیص تناقض در متون فارسی استفاده کنیم. این شبکه عصبی دارای مزایایی ازجمله کاهش هزینه‌های حافظه‌ای و پیچیدگی زمانی است. این مدل توانایی حفظ ارتباطات طولانی‌مدت در جملات فارسی را دارد که در تحلیل زبان‌های پیچیده؛ مانند فارسی بسیار مفید است و از مزایای اصلی مدل محسوب می‌شود؛ این مزیت در کارهای قبلی که در متون فارسی ارائه‌شده، دیده نمی‌شود. همچنین تاکنون برای تشخیص تناقض در متون فارسی از RWKV استفاده‌نشده است. شناسایی، انتخاب و کارسازی استفاده از شبکه عصبی

RWKV به‌عنوان یک مدل یادگیری عمیق در تشخیص تناقض‌های متون فارسی، نوآوری تحقیق محسوب می‌شود.

٤-۳. کارهای آینده

باتوجه‌به نتایج ارزیابی مدل و تحلیل می‌توان گفت:

- فرایند جفت‌سازی جملات نقش کلیدی در عملکرد مدل دارد؛ زیرا مدل تنها بر اساس جفت‌هایی که به آن ارائه می‌شود، تناقض‌ها را تشخیص می‌دهد. عدم دقت یا تنوع کافی در این مرحله می‌تواند منجر به کاهش توانایی مدل در شناسایی تناقض‌های پیچیده شود.
- جمع‌آوری و استفاده از مجموعه‌های داده بزرگ‌تر و متنوع‌تر می‌تواند به بهبود توانایی مدل در یادگیری و تعمیم به نمونه‌های جدید کمک کند.
- بنابراین، پیشنهاد می‌شود برای بهبودبخشی مدل پیشنهادی، طرحی ارائه شود که در آن دو موضوع زیر کاملاً رعایت شده باشد:

❖از الگوریتم‌های پیشرفته‌ای جهت ایجاد جفت‌های معنی‌دار و جامع استفاده شود و از مجموعه‌های داده بزرگ‌تر و متنوع‌تر یا از فنون تقویت داده که می‌تواند به بهبود توانایی مدل در یادگیری و تعمیم به نمونه‌های جدید کمک کند؛ استفاده شود.

۵. مراجع

[1] Z. Rahimi and M. ShamsFard, "Contradiction Detection in Persian Text," arvix, 2021.

[2] C. Hsu, C.-T. Liy, D. Saez-Trumper and Y.-Z. Hsu, "WikiContradiction: Detecting Self-Contradiction Articles on Wikipedia," in IEE انواع E International Conference on Big Data (Big Data), 2021.

[3] B. Peng, E. Alcaide, Q. Anthony, A. Albalak and S. Arcadinho, "RWKV: Reinventing RNNs for the Transformer Era," arvix, 2023.

[4] D. S. Vargas and V. Moreira, "Identifying Sentiment-based Contradictions," Journal of Information and Data Management, vol. 8, no. 3, pp. 242-254, 2017.

[5] N. S. Tawfik and M. R. Spruit, "Automated Contradiction Detection in Biomedical Literature," in International Conference on Machine Learning and Data Mining in Pattern Recognition, 2018.

[6] Sepúlveda-Torres, "Automatic Contradiction Detection in Spanish," in Department of Software and Computing Systems, University of Alicante, Baeza, Spain, 2021.

[7] L. Li, B. Qin and T. Liu, "Contradiction Detection with Contradiction-Specific Word Embedding," algorithms, vol. 10, no. 59, 2017.

[8] V. Lingam, S. Bhuria, M. Nair, D. Gurpreetsingh, A. Goyal and A. Sureka, "Deep learning for conflicting statements detection in text," PeerJ Preprints, 2018.

[9] H. Khodadadi, S. Rahati Ghochani and A. Estejami, "Identifying the conflict relationship in Persian discourse with the help of supervised learning methods," Signal and data processing, 2015.

[10] T. Deußer, M. Pielka, L. Pucknat, B. Jacob, T. Dilmaghani, M. Nourimand, B. Kliem, R. Loitz, C. Bauckhage and R. Sifa, "Contradiction Detection in Financial Reports," Proceedings of the Northern Lights Deep Learning Workshop, vol. 4, 2023.

[11] P. Lendvai and U. Reichel, "Contradiction Detection for Rumorous Claims," in Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM), Osaka, Japan, 2016.

[12] R. Sifa, M. Pielka, R. Ramamurthy, A. Ladi, L. Hillebrand and C. Bauckhage, "Towards Contradiction Detection in German: a Translation-Driven Approach," in IEEE, Xiamen, China, 2019.

[13] A. E. Kharrat, L. Hlaoua and L. B. Romdhane, "Contradiction Detection Approach Based on Semantic Relations and Evidence of Uncertainty," in International Conference on Computational Collective Intelligence, 2022.

[14] J. Dentan and A. Rakotonirainy, "Multi-lingual contradiction detection," 2022.

[15] I. Badache, S. Fournier and A.-G. Chifu, "Predicting Contradiction Intensity: Low, Strong or Very Strong?," in SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, New York, 2018.

[16] A. Alamri and M. Stevensony, "Automatic identification of potentially contradictory claims to support systematic reviews," in IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Washington, DC, USA, 2015.

[17] H. Amirkhani, M. AzariJafari, S. Faridan-Jahromi, Z. Kouhkan, Z. Pourjafari and A. Amirak, "FarsTail: a Persian natural language inference dataset," Springer link, 2023.

[18] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao and F. L. Wang, "Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment," Nature Machine Intelligence, vol. 5, pp. 220-235, 2023.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," arxiv, 2021.