Tabriz Journal of Electrical Engineering (TJEE), vol. 54, no. 1, Spring 2024 Serial no. 107
DOI: 10.22034/tjee.2023.16964

Proposing a model to predict relatedness between knowledge units in programming question-answering websites using deep learning techniques: a case study of Stack Overflow

Hossein Abbasimehr[1*], Mohammad KhodizadehNahari[1]

[1]Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran.
E-mails: abbasimehr@azaruniv.ac.ir; m.khodizadeh@azaruniv.ac.ir

Short Abstract
The Stack Overflow website is one of the most popular communities used by millions of programmers. If we consider a question and its corresponding answers as a knowledge unit on the Stack Overflow website, then there are different semantic relationships between two knowledge units, which include duplicate, direct, and indirect relationships with the proposed question. Recognizing different categories of semantic relationship between knowledge units in Stack Overflow can significantly improve the effectiveness and efficiency of information search. In this study, a hybrid approach based on deep learning methods and traditional similarity criteria is presented to detect the relationship between questions. In particular, two deep network architectures are presented, the first architecture consists of a long short-term memory network as well as a cosine similarity calculation layer. The second architecture is an extension of the first architecture by adding an attention mechanism. The proposed approach was evaluated on a dataset of Java programming language contining 40000 questions. The obtained results show that in terms of F1, Recall and Precision, the proposed model performs better than the existing models. Specifically, the model proposed in this article has a 17.3% improvement in terms of F1 measure compared to the best current model. Also, the results of the experiments show that using the pre-trained word embedding model significantly improves the performance of the presented models.
Keywords
Relatedness prediction, Multiclass classification, BiLSTM method, Attention mechanism, Text similarity measures

1- Short Introduction (4-5 lines)
If we consider a question and its corresponding answers as a knowledge unit on the Stack Overflow website, then there are different semantic relationships between two knowledge units, which include duplicate, direct, and indirect relationships with the proposed question. This research aims to provide a model based on deep learning methods and text similarity criteria to improve the prediction of semantic relationships between knowledge units on the Stack Overflow website. In the existing studies, there are two general approaches to detecting the relationship between knowledge units; the first approach involves representing the text by building features related to similarity measures and then using a machine-learning technique to build a model. In the second approach, word embedding representation is used along with deep learning techniques to create a model.

2- Proposed Work and Methodology (including comprision, simulation/experimental results and discusion)
Although deep learning methods such as long short-term memory and convolutional neural networks have shown successful results in this field, the performance of deep learning methods can be improved by including features related to text similarity and using the attention mechanism. In this regard, this study presents a hybrid approach based on deep learning methods and traditional similarity measures to detect the relationship between questions. In particular, two deep network architectures are presented; the first architecture consists of a long short-term memory network and a cosine similarity calculation layer. The second architecture extends the first architecture by adding an attention mechanism. Also, in this study, pre-trained word embedding models in general texts and domain-specific texts are used to represent words. The dataset of Java programming language questions related to the Stack Overflow site, which contains 40 thousand pairs of knowledge units, was used to test the presented models. The presented methods were implemented using Python programming language and Keras deep learning library. The performance results of the models were measured using the Precision, Recall, and F1-measure, and the results show that the combined BilSTM-Att-Cosine_Fe model with the Glove pre-trained word embedding has better results than other implemented models in terms of all performance criteria. The obtained model performs better than the models presented in (Xu et al., 2018), which indicates the superiority of the proposed method in finding the semantic relationship between knowledge units. Also, the results of the experiments show that the use of the Glove pre-trained word embedding model significantly improves the performance of all utilized models. The models obtained using the Glove representation perform better than the pre-trained representation related to specialized field texts.

3- Conclusion (4-5 lines)
The results of this study show that the use of deep learning methods along with traditional similarity features significantly increases the classification power of the model. The deep learning methods used in this research have a strong ability to represent knowledge units and recognize the relationship between them. The results of the experiments show the effectiveness of using pre-trained representations as a transfer learning approach.

4- References (2-3 references)
- B. Xu, A. Shirani, D. Lo, and M. A. Alipour, "Prediction of relatedness in stack overflow: deep learning vs. svm: a reproducibility study," Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2018.
- N. Ansari and R. Sharma, "Identifying semantically duplicate questions using data science approach: A quora case study," arXiv preprint arXiv:2004.11694, 2020.

ارائه مدلی جهت پیش‌بینی ارتباط بین واحدهای دانشی در وب سایت‌های پرسش و پاسخ برنامه‌نویسی با استفاده از تکنیک‌های یادگیری عمیق: مطالعه موردی Stack Overflow

حسین عباسی مهر
استادیار، دانشکده فناوری اطلاعات و مهندسی کامپیوتر، دانشگاه شهید مدنی آذربایجان، تبریز، ایران

محمد خودی‌زاده نهاری
استادیار، دانشکده فناوری اطلاعات و مهندسی کامپیوتر، دانشگاه شهید مدنی آذربایجان، تبریز، ایران

چکیده
وب‌سایت Stack Overflow یکی از محبوب‌ترین جوامعی است که میلیون‌ها برنامه‌نویس از آن استفاده می‌کنند. اگر یک سؤال و پاسخ‌های متناظر با آن را یک واحد دانشی در نظر بگیریم، آنگاه بین واحدهای دانشی ارتباط مختلف معنایی وجود دارد که این ارتباط شامل ارتباط تکراری، ارتباط مستقیم، ارتباط غیرمستقیم با سؤال طرح‌شده است. تشخیص دسته‌های مختلف ارتباط معنایی بین واحدهای دانشی در Stack Overflow می‌تواند اثربخشی و کارایی جستجوی اطلاعات را به‌طور چشمگیری بهبود بخشد. در این مطالعه، یک رویکرد ترکیبی مبتنی بر روش‌های یادگیری عمیق و معیارهای تشابه سنتی جهت تشخیص ارتباط بین سؤالات ارائه می‌شود. به‌طور خاص دو معماری شبکه عمیق ارائه می‌شود که معماری اول از شبکه حافظه کوتاه‌مدت طولانی دوطرفه و همچنین لایه محاسبه کننده شباهت کسینوسی تشکیل شده است. معماری دوم گسترش‌یافته‌ی معماری اول با اضافه کردن مکانیزم توجه است. رویکرد پیشنهادی روی یک مجموعه داده سؤالات زبان برنامه‌نویسی جاوا شامل ۴۰۰۰۰ مورد ارزیابی قرار گرفت. نتایج به‌دست‌آمده نشان می‌دهد که در معیارهای F1، Recall و Precision مدل پیشنهادی عملکرد بهتری نسبت به مدل‌های موجود از خود نشان می‌دهد. به‌طور خاص مدل پیشنهادی در این مقاله در معیار F1 بهبود ۱۷٫۳ درصدی نسبت به برترین مدل فعلی دارد. همچنین نتایج آزمایش‌ها نشان می‌دهد که استفاده از مدل تعبیه کلمات از پیش آموزش‌دیده به‌طور قابل‌ملاحظه‌ای عملکرد مدل‌های ارائه‌شده را بهبود می‌بخشد.

کلمات کلیدی
تشخیص ارتباط، دسته‌بندی چند کلاسه، روش BiLSTM، مکانیزم توجه، معیارهای شباهت متن.

نام نویسنده مسئول: دکتر حسین عباسی مهر
ایمیل نویسنده مسئول: abbasimehr@azaruniv.ac.ir

تاریخ ارسال مقاله: ۱۴۰۲/۰۴/۲۶
تاریخ(های) اصلاح مقاله: ۱۴۰۲/۰۶/۱۰
تاریخ پذیرش مقاله: ۱۴۰۲/۰۷/۲۲

۱- مقدمه

وب‌سایت‌های پرسش و پاسخ برنامه‌نویسی به‌عنوان یک بستر مناسب طرح سؤالات و خطاها و سایر مشکلات مربوط به برنامه‌نویسی زبان‌های مختلف، مورد استقبال کاربران واقع‌شده‌اند. وب‌سایت Stack Overflow [۱] یکی از محبوب‌ترین وب‌سایت‌های پرسش و پاسخ در حوزه برنامه‌نویسی است که در آن کاربران می‌توانند سؤالات مربوط به مشکلات برنامه را بپرسند و پاسخ دهند و دانش کسب کنند [۲-۷]. اگر یک سؤال و پاسخ‌های متناظر با آن را یک واحد دانشی در نظر بگیریم، آنگاه بین واحدهای دانشی ارتباط مختلف معنایی وجود دارد که این ارتباط شامل ارتباط تکراری، ارتباط مستقیم، ارتباط غیرمستقیم با سؤال طرح‌شده است [۸]. تشخیص دسته‌های مختلف ارتباط معنایی بین واحدهای دانشی در Stack Overflow می‌تواند اثربخشی و کارایی جستجوی اطلاعات را به‌طور چشمگیری بهبود بخشد.

بررسی واحدهای دانشی مطرح‌شده توسط صاحبان این وب‌سایت نشان می دهد که بسیاری از آن‌ها به لحاظ محتوایی تکراری هستند و ارتباط معنایی بین آن‌ها وجود دارد. وجود سؤالات تکراری باعث اتلاف وقت و انرژی کاربران پاسخ‌دهنده می‌شود و همچنین کاربرانی که سؤالات تکراری می‌پرسند مدت زیادی باید منتظر بمانند تا به سؤال آن‌ها پاسخ داده شود درحالی‌که پاسخ‌های آماده از قبل موجود است [۹]. بنابراین، شناسایی واحدهای مرتبط با سؤالات جدید می‌تواند دوباره‌کاری‌ها در این زمینه را کاهش دهد.

مطالعاتی که در زمینه تشخیص محتوای تکراری در Stack Overflow انجام گرفته است به‌صورت کلی به دو حوزه تشخیص ارتباط بین واحدهای دانشی و یا تشخیص ارتباط بین سؤالات تقسیم می‌شوند. بسیاری از تحقیقات، غالباً تشخیص تکراری بودن سؤالات را به‌عنوان مسئله خود در نظر گرفته‌اند؛ یعنی روی ارتباط یا عدم ارتباط بین سؤالات تمرکز کرده و پیش‌بینی ارتباط را به‌عنوان یک مسئله دسته‌بندی دو کلاسه در نظر گرفته‌اند (برای مثال [۱۰-۱۲]).

دسته دیگری از مطالعات در این زمینه پیش‌بینی ارتباط بین واحدهای دانشی را به‌عنوان یک مسئله چند کلاسه فرموله کرده و با استفاده از تکنیک‌های یادگیری ماشین و یادگیری عمیق کار پیش‌بینی را انجام داده‌اند (برای مثال [۸، ۱۳، ۱۴]). تفاوت این پژوهش‌ها در روش‌های بازنمایی متن، مدل‌های دسته‌بندی و مجموعه داده مورد استفاده است. مدل‌های دسته‌بندی مورداستفاده در تحقیقات گذشته این حوزه شامل مدل‌های سنتی نظیر ماشین

بردار پشتیبان[۱] و مدل‌های یادگیری عمیق نظیر شبکه عصبی پیچشی و شبکه حافظه کوتاه‌مدت طولانی[۲] (LSTM) است. همچنین بازنمایی مورداستفاده در آن‌ها شامل بازنمایی سنتی TF-IDF [۱۵، ۱۶] و بازنمایی مبتنی بر تعبیه کلمات[۳] [۱۷] است. یک دسته دیگری از روش‌های سنتی شامل استفاده از بازنمایی‌های به‌دست‌آمده از متن و ساخت ویژگی‌هایی بر اساس معیار شباهت نظیر معیار شباهت کسینوسی است (برای مثال مدل‌های مبتنی بر ماشین بردار پشتیبان ارائه‌شده در [۸، ۱۳]). به‌عبارت‌دیگر ویژگی‌هایی به‌صورت دستی ساخته‌شده و به‌عنوان ورودی دسته‌بند استفاده می‌شود.

روش‌های شبکه عصبی عمیق بازنمایی حاصل از لایه تعبیه کلمات را دریافت کرده و به‌طور خودکار در طول لایه‌های شبکه ویژگی‌هایی را جهت دسته‌بندی می‌سازند. اگرچه در مطالعات قبلی روش‌های یادگیری عمیق عملکرد نسبتاً موفقی از خود نشان داده‌اند، با گنجاندن معیارهای مرتبط با شباهت متن و همچنین استفاده از مکانیزم توجه می‌توان عملکرد روش‌های یادگیری عمیق را بهبود داد. در این مطالعه هدف ما بهره‌گیری از ویژگی‌های مرتبط با معیار شباهت کسینوسی جهت بهبود دقت مدل‌های یادگیری عمیق است. در همین راستا دو معماری مبتنی بر یادگیری عمیق ارائه می‌گردد. معماری اول که به‌طورکلی شامل یک لایه ورودی، لایه تعبیه کلمات، شبکه LSTM دوطرفه، لایه حداکثرگیری کلی[۴]، یک لایه محاسبه شباهت کسینوسی و یک لایه تماماً متصل و یک لایه خروجی کار پیش‌بینی را انجام می‌دهد. نتایج آزمایش‌ها روی مجموعه داده Stack Overflow نشان می‌دهد که مدل پیشنهادی نسبت به مدل‌های موجود عملکرد بهتری دارد.

همچنین معماری شبکه عصبی عمیق دیگری با اضافه کردن مکانیزم توجه [۱۸] به معماری اول ارائه گردید. به‌طور خاص تفاوت اصلی بین این دو معماری این است که خروجی لایه تعبیه کلمات مربوط به جفت عناصر ابتدا وارد لایه توجه شده و جفت عناصر با یکدیگر مقایسه می‌شوند و کلماتی که دارای بیشترین شباهت در جفت عناصر هستند پیدا می‌شوند. بر اساس این مقایسه، یک بازنمایی جدیدی که نشان‌دهنده رابطه بین دو جفت عنصر است حاصل می‌شود.

در این مطالعه بر اساس معماری‌های ارائه‌شده و با در نظر گرفتن مدل‌های تعبیه کلمات مختلف، چندین مدل پیاده‌سازی و مورد ارزیابی قرار گرفتند. نتایج عملکرد مدل‌ها روی مجموعه داده Stack Overflow با ۴۰۰۰۰ رکورد با استفاده از معیارهای Precision، Recall و F1-measure سنجیده شد و بررسی نتایج نشان می‌دهد که مدل ترکیبی BilSTM-Att-Cosine_Fe با مدل بازنمایی تعبیه کلمات از پیش آموزش‌دیده Glove نتایج بهتری نسبت به سایر مدل‌های پیاده‌سازی شده از نظر تمامی معیارهای عملکردی دارد. مدل به‌دست‌آمده نسبت به مدل‌های ارائه‌شده در [۱۳] دارای عملکرد بهتری است که نشان‌دهنده برتری روش ارائه در یافتن ارتباط معنایی بین واحدهای دانشی است. همچنین نتایج آزمایش‌ها نشان می‌دهد که استفاده از مدل تعبیه کلمات از پیش آموزش‌دیده Glove که با استفاده از متون عمومی انگلیسی به‌دست‌آمده است، به‌طور قابل‌ملاحظه‌ای عملکرد تمامی مدل‌های ارائه‌شده را بهبود می‌بخشد. مدل‌های به‌دست‌آمده با به‌کارگیری بازنمایی Glove بهتر از بازنمایی از پیش آموزش‌دیده مربوط به متون حوزه تخصصی عمل می‌کند.

به‌طور خلاصه نوآوری اصلی این مقاله ترکیب بازنمایی‌های ایجاد شده توسط روش‌های یادگیری عمیق با معیارهای شباهت متن است. همچنین ترکیب روش LSTM دو طرفه با مکانیزم توجه به‌منظور یافتن ارتباط معنایی دو واحد دانشی یکی دیگر از نوآوری‌های این مطالعه است.

ساختار ادامه این مقاله بدین گونه است. بخش ۲ مرور ادبیاتی در زمینه پیش‌بینی سؤالات تکراری و پیش‌بینی ارتباط بین واحدهای دانشی در Stack Overflow ارائه می‌دهد. در بخش ۳، ابتدا به بیان مسئله، توصیف مدل‌های مورداستفاده و درنهایت تشریح رویکرد پیشنهادی می‌پردازیم. بخش ۴ به توصیف داده‌ها و آزمایش‌ها پرداخته و نتایج مدل‌ها را مورد مقایسه و تحلیل قرار می‌دهد. در بخش ۵ نتیجه‌گیری و ارائه پیشنهاد کارهای آتی می‌پردازیم.

۲- مرور ادبیات

مقاله [۷] یک مرور ادبیاتی جامع در مورد تحقیقاتی که از تکنیک‌های یادگیری ماشین و یادگیری عمیق جهت تحلیل محتواهای مربوط به سایت‌های پرسش و پاسخ برنامه‌نویسی استفاده کرده‌اند، انجام می‌دهد. مطابق این تحقیق، تشخیص سؤالات تکراری یکی از موضوعات مهم مورد بررسی در تحقیقات قبلی بوده است. همچنین روش‌های یادگیری ماشین سنتی نسبت به روش‌های یادگیری عمیق بیشتر مورد استفاده قرار گرفته‌اند.

همان‌طور که در بخش قبلی بیان شد، مطالعات انجام‌شده در تشخیص ارتباط در سایت‌های پرسش و پاسخ برنامه‌نویسی به‌طورکلی به دو دسته تشخیص تکراری بودن سؤالات و همچنین تشخیص ارتباط بین واحدهای دانشی تقسیم می‌شوند. که در این بخش به مرور برخی از تحقیقات ارائه‌شده در هر دسته می‌پردازیم.

۲-۱- پیش‌بینی تکراری بودن

ژانگ و همکاران [۱۰] مدل DupPredictor را برای تشخیص تکراری بودن یک سؤال ارائه کرده‌اند. DupPredictor، یک سؤال را به‌عنوان ورودی دریافت می‌کند و سؤالات مشابه آن را با شناسایی موضوعات پنهان آن سؤال و محاسبه امتیازات شباهت تشخیص می‌کند. احسان الزمان همکاران [۱۹] روش Dupe را پیشنهاد کرده‌اند که ویژگی‌هایی را از پیکره سؤال استخراج کرده و به‌عنوان ورودی یک دسته‌بندی کننده باینری تشخیص تکراری بودن استفاده می‌کند. ونگ و همکاران[۲۰] از تکنیک‌های یادگیری عمیق نظیر حافظه کوتاه‌مدت طولانی (LSTM) برای شناسایی سؤالات تکراری در Stack Overflow استفاده کرده است. علاوه بر این، در این مقاله از Word2Vec برای به دست آوردن نمایش برداری کلمات استفاده شده است. نتایج این تحقیق نشان می‌دهد که رویکرد ارائه‌شده در مقایسه با چهار رویکرد پایه شامل مدل DupPredictor [۱۰]، مدل Dupe [۱۹] و روش DupeRep [۲۱] عملکرد بهتری از نظر معیار Recall دارد. علاوه بر این رویکرد ارائه‌شده بهتر از چهار رویکرد یادگیری ماشین شامل ماشین بردار پشتیبان، رگرسیون لجستیک، روش جنگل تصادفی XGBoost عمل می‌کند.

در ساخت مدل برای تشخیص سؤالات مشابه و تکراری، کدهای برنامه‌نویسی همراه سؤالات نیز در چندین کار ازجمله ژانگ و همکاران [۲۲] و گائو و همکاران [۱۲] مورد توجه قرارگرفته است. با توجه به اینکه در سایت‌های پرسش و پاسخ برنامه‌نویسی هدف نهایی کد برنامه است شباهت کد بین دو سؤال بار معنایی بیشتری دارد. گائو و همکاران [۱۲] از تعبیه کلمه و شبکه‌های عصبی پیچشی برای استخراج ویژگی‌های متنی از سؤالات برای غلبه بر مسئله شکاف واژگانی استفاده کرده‌اند. همچنین از شبکه‌های عصبی پیچشی مبتنی بر درخت برای استخراج ویژگی‌های ساختاری و معنایی از کد منبع بهره برده‌اند. همچنین برای تقویت نتایج، از پیش‌بینی برچسب‌ها برای سؤالات استفاده

[۱] Support Vector Machine(SVM)

[۲] Long short-term memory (LSTM)

[۳] Word embedding

[۴] Global max pooling

کرده‌اند. مجموعه داده مورداستفاده آن‌ها Stack Overflow هست.

ژانگ و همکاران [۱۱] مسئله تشخیص سؤالات مشابه و تکراری را به‌عنوان یک مسئله دومرحله‌ای «رتبه‌بندی-دسته‌بندی» روی جفت سؤال‌ها مدل کرده‌اند. در مرحله اول سؤالات پیشین در سایت را با توجه به شباهت آن‌ها با سؤال جدید، رتبه‌بندی کرده و رتبه‌های برتر را به‌عنوان کاندید انتخاب می‌کنند تا فضای جستجو کاهش یابد. در مرحله دوم، ویژگی‌های جدیدی را توسعه می دهند که هم شباهت متنی و هم معنایی نهفته را در جفت سؤال‌ها، با به‌کارگیری تکنیک‌های یادگیری عمیق و بازیابی اطلاعات نشان دهد. آزمایش‌ها روی سؤالات دنیای واقعی در مورد چندین زبان برنامه‌نویسی نشان داده که روش پیشنهادی بسیار خوب کار می‌کند.

همچنین در مطالعه [۲۳] با در نظر گرفتن مجموعه داده سؤالات سایت Quora، عملکرد هفت روش یادگیری ماشین سنتی، شبکه LSTM و همچنین شبکه عصبی سیامی[۵] در تشخیص سؤالات تکراری موردبررسی قرار گرفت. نتایج این تحقیق نشان می‌دهد که مدل XGBoost بهتر از سایر روش‌های سنتی و روش‌های یادگیری عمیق پیاده‌سازی شده در این مطالعه عمل می‌کند.

تشخیص سؤالات تکراری و مشابه در حوزه برنامه‌نویسی قابلیت ریزتر شدن موضوع را دارد همان‌گونه که کامینسکی و همکاران [۲۴] این کار را در خصوص توسعه‌دهندگان بازی‌های رایانه‌ای انجام داده‌اند. توسعه بازی در حال حاضر بزرگ‌ترین صنعت در بخش سرگرمی است و تقاضای زیادی برای توسعه‌دهندگان ماهر بازی وجود دارد که بتوانند بازی‌های باکیفیت بالا تولید کنند. برای تأمین این تقاضا، توسعه‌دهندگان بازی به منابعی نیاز دارند که بتواند دانش موردنیاز برای یادگیری و بهبود مهارت‌های خود را در اختیار آن‌ها بگذارد. در این کار تحقیقی روی مجموعه داده‌هایی از Game Development Stack Exchange و Stack Overflow کارشده است و تکنیک‌های از پیش آموزش‌دیده شده و بدون نظارت برای تشخیص مباحث مشابه در توسعه بازی‌های رایانه‌ای بکار گرفته‌شده است.

۲-۲- پیش‌بینی ارتباط بین واحدهای دانشی

تشخیص ارتباط بین واحدهای دانشی مربوط به حوزه زبان‌های برنامه‌نویسی به افزایش کارایی پلتفرم‌های پرسش و پاسخ نظیر Stack Overflow منجر می‌شود. رویکردهای متفاوتی برای تشخیص ارتباط به‌طورکلی و تشخیص تکراری بودن واحدهای دانشی زبان‌های برنامه‌نویسی وجود دارد. به‌طورکلی این روش‌ها را می‌توان بر اساس نوع روش بازنمایی سؤالات و مدل دسته‌بندی، تقسیم‌بندی کرد.

از نظر مدل دسته‌بندی، روش‌های یادگیری ماشین سنتی و یادگیری عمیق در تحقیقات گذشته مورداستفاده قرارگرفته‌اند. روش‌های یادگیری عمیق نتایج موفقیت‌آمیزی را نسبت به دسته‌بندهای سنتی روی مجموعه داده‌هایی با تعداد سؤالات بیشتر به دست آورده‌اند. از نظر بازنمایی نیز بازنمایی تعبیه کلمات، بازنمایی به روش TF-IDF و همچنین بازنمایی از طریق استخراج ویژگی‌های دستی از متن شامل معیارهای مورداستفاده جهت اندازه‌گیری شباهت متن، نظیر شباهت کسینوسی در تحقیقات قبلی استفاده‌شده‌اند.

مدل‌های یادگیری عمیق داده ورودی خام را گرفته و در طول لایه‌های خود، کار ساخت ویژگی را به‌صورت خودکار انجام می‌دهند. این رویکردها مخصوصاً زمانی که داده‌های زیادی را در اختیار داریم مناسب هستند. در حوزه پیش‌بینی ارتباط در تحقیقات گذشته مدل‌های مختلفی با ترکیب شیوه‌های مختلف بازنمایی متن و روش‌های دسته‌بندی ارائه‌شده است که در ادامه به بیان آن می‌پردازیم.

در مقاله [۸] روش شبکه عصبی پیچشی[۶] و روش ماشین بردار پشتیبان روی دو مجموعه داده شامل سؤالات Stack Overflow اعمال شدند که نتایج این مقاله حاکی از برتری مدل شبکه عصبی پیچشی و بازنمایی مبتنی بر تعبیه کلمات نسبت به روش‌های پایه نظیر ماشین بردار پشتیبان و بازنمایی TF-IDF است. در این مقاله یافته‌های نویسندگان نشان‌دهنده تأثیر بهتر روش بازنمایی تعبیه کلمات نسبت به بازنمایی به روش TF-IDF و ساخت ویژگی‌هایی دستی از متن در عملکرد بهتر روش‌های شبکه عصبی پیچشی و ماشین بردار پشتیبان می‌باشد.

در مقاله [۱۳] رویکرد SimBow ارائه‌شده در [۲۵] مورداستفاده قرارگرفته است. به‌طور خاص در این مقاله ابتدا ویژگی‌های مختلفی بر اساس معیار شباهت کسینوسی و معیار شباهت کسینوسی نرم استخراج‌شده و ویژگی‌های به‌دست‌آمده به‌عنوان ورودی مدل ماشین بردار پشتیبان جهت پیش‌بینی ارتباط به کار گرفته‌شده است. مشکل شباهت کسینوسی سنتی این است که وقتی هیچ کلمه مشترکی بین دو متن ورودی وجود نداشته باشد، شباهت کسینوسی صفر می‌شود. درحالی‌که دو متن می‌توانند با داشتن کلمات متفاوت، معنای مشابهی داشته باشند. این مشکل به‌طور مکرر در زمینه تشخیص ارتباط بین سؤالات رخ می‌دهد زمانی که دو سؤال مشابه از نظر معنایی به کلمات متفاوتی بیان می‌شوند. همچنین در مقاله فوق‌الذکر یک روش مبتنی بر شبکه عصبی پیچشی نیز ارائه گردید و نتایج آزمایش‌ها روی دو تا مجموعه داده با تعداد متفاوت رکوردها حاکی از آن بود که در مجموعه داده با تعداد رکورد کمتر روش CNN بهتر عمل می‌کند درحالی‌که روی مجموعه داده بزرگ‌تر روش مبتنی بر رویکرد SimBow و مدل ماشین بردار پشتیبان بهتر از روش مبتنی بر شبکه عصبی پیچشی عمل می‌کند.

در مقاله [۱۴] نویسندگان پیش‌بینی ارتباط بین واحدهای دانشی از دو روش مبتنی بر شبکه LSTM دوطرفه و روش ماشین بردار پشتیبان استفاده کردند. نتایج این مقاله نشان داد که روش مبتنی بر LSTM بهتر از روش ماشین بردار پشتیبان با در نظر گرفتن تمامی معیارهای عملکردی، عمل می‌کند. پی و همکاران [۲۶] از یک مدل مبتنی بر توجه روی جفت جملات با عنوان ASIM برای پیش‌بینی ارتباط بین سؤالات در Stack Overflow استفاده می‌کند. این مکانیزم توجه، برای گرفتن اطلاعات تعامل معنایی بین سؤالات اتخاذ شده است. علاوه بر این، تعبیه کلمه مخصوص حوزه مهندسی نرم‌افزار نیز برای این کار از قبل آموزش داده و استفاده شده است. نتایج آزمایش‌ها نشان می‌دهد که ASIM نسبت به رویکردهای پایه در معیارهای ارزیابی Precision، Recall و Micro-F1 پیشرفت قابل‌توجهی داشته است. مدل پیشنهاد شده همچنین در تشخیص سؤالات تکراری AskUbuntu که یک کار مشابه اما متفاوت است، به‌خوبی عمل کرده است و تعمیم‌پذیری و استحکام روش را ثابت کرده است.

۳- روش ارائه‌شده

در این بخش ابتدا به بیان مسئله تحقیق پرداخته سپس مدل‌های پیشنهادی را توصیف می‌کنیم. تمرکز این پژوهش بر روی سایت‌های پرس و پاسخ در حوزه برنامه‌نویسی است که در آن سؤالاتی مطرح می‌شوند که به‌طور مشخص روی یک موضوع خاص متمرکز باشند[۷]. به همین منظور به‌عنوان مطالعه موردی از مجموعه داده مربوط به Stack Overflow استفاده خواهد شد. در Stack Overflow معمولاً سؤالات معطوف به یک زبان برنامه‌نویسی خاص هستند و هدف رفع مشکل برنامه‌نویسان در پیاده‌سازی یک الگوریتم است نه

[۵] Siamese neural network

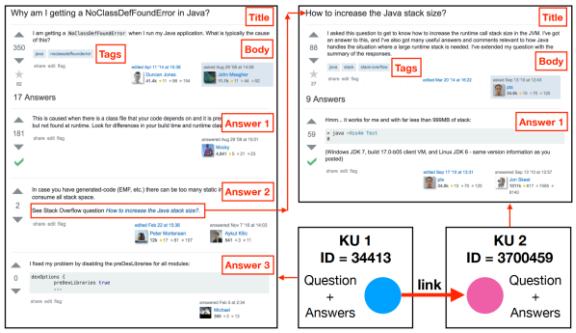
[۶] Convolutional neural network

[۷] Tightly focused on a specific problem

توسعه یک روش مهندسی نرم‌افزار. این نوع سایت‌ها متفاوت از سایت‌هایی هستند که پرسش‌هایی که محدوده وسیع‌تری[۸] را پوشش می‌دهند [۲۷] (به‌عنوان‌مثال Software Engineering Stack Exchange). بنابراین پیش‌پردازش‌های انجام شده در این پژوهش با فرض مواجه بودن با یک زبان برنامه‌نویسی طراحی و اجرا شده است و اگر قرار باشد نوع دیگری از متن به‌عنوان ورودی به الگوریتم‌های طراحی شده ارائه گردد باید پیش‌پردازش‌های متناسب آن‌ها طراحی و پیاده‌سازی شود.

۳-۱- مسئله پیش‌بینی ارتباط بین واحدهای دانشی Stack Overflow

در این بخش به توصیف مسئله مدنظر در این تحقیق می‌پردازیم. همان‌طور که قبلاً اشاره شد زو و همکاران [۸] یک سؤال شامل عنوان، متن مجموعه پاسخ‌های آن را به‌عنوان یک واحد دانش در نظر می‌گیرد. در شکل ۱ مثال واقعی از دو واحد دانشی نشان داده‌شده است. مسئله موردنظر، پیش‌بینی ارتباط بین دو واحد دانشی بر اساس اطلاعات متنی است. در زو و همکاران [۸] روابط بین دو واحد دانش به چهار دسته تقسیم‌شده است که در جدول ۱ توصیف داده شده است. همچنین مثال‌هایی از انواع ارتباط بین دو واحد دانشی در جدول ۲ آورده شده است. همان‌طور که مشخص است سطر اول نشان‌دهنده یک سؤال برنامه‌نویسی بوده و سطرهای بعدی سؤالات مرتبط با آن را بر اساس نوع ارتباط مشخص کرده است. همان‌طور که در جدول ۲ آمده است، سؤال اصلی در مورد «مقایسه رشته‌ها در زبان برنامه‌نویسی جاوا» است. حال سؤال تکراری برای این سؤال، «فرق بین == و equals» است که دقیقاً جوابی همانند سؤال اصلی دارد. در سؤال بعدی می‌توانیم جواب یک بخشی از سؤال را در این سؤال پیدا کنیم. در آخرین سؤال درست است که جواب سؤال در این سؤال نمی‌تواند باشد اما می‌تواند کمکی برای حل جواب سؤال اصلی باشد.



شکل ۱- مثالی از یک واحد دانشی در Sack Overflow [۸]

جدول ۱- توصیف انواع ارتباط بین دو واحد دانشی

نوع ارتباط	توصیف
تکراری	دو واحد دانش درباره یک موضوع به با کلمات و عبارات متفاوت بیان‌شده‌اند و می‌توانیم یک پاسخ برای هر دو واحد دانش بدهیم
ارتباط مستقیم	یک واحد دانش می‌تواند به حل مسئله در واحد دانش دیگر کمک کند. برای مثال، با توضیح مفاهیم خاص، ارائه مثال یا پوشش یک مرحله حل یک مسئله پیچیده.
ارتباط غیرمستقیم	یک واحد دانش اطلاعات مرتبط را ارائه می‌دهد، اما مستقیماً به سؤال در واحد دانش دیگر پاسخ نمی‌دهد.
عدم ارتباط	این دو واحد دانش از نظر معنایی به هم مرتبط نیستند.

[۸] Broader nature

[۹] Tokenization

۳-۲- روش پیشنهادی

در این بخش رویکرد پیشنهادی مبتنی بر روش‌های یادگیری عمیق جهت پیش‌بینی ارتباط بین واحدهای دانش در Stack Overflow را تشریح می‌کنیم. رویکرد کلی شامل بخش پیش‌پردازش و بخش مدل‌سازی و ارزیابی مدل‌ها است. بخش پیش‌پردازش کار جداسازی کلمات، حذف کلمات توقف و همچنین حذف کاراکترهای زائد انجام می‌شود. با توجه به اینکه متون واحدهای دانش Stack Overflow متفاوت از متون عمومی زبان انگلیسی هستند، بنابراین نیاز است تا برخی پیش‌پردازش‌هایی به‌منظور پاک‌سازی متون ورودی و جداسازی کلمه‌ها صورت گیرد. همچنین متناظر با بخش مدل‌سازی، در ادامه دو مدل پیشنهادی مبتنی بر یادگیری عمیق توصیف می‌شود. برای انجام آزمایش‌ها ما از مجموعه داده‌های Stack Overflow استفاده می‌کنیم، همه داده‌ها شامل جفت واحد دانشی هستند که برچسب آن‌ها معین شده است. هر واحد دانشی شامل عنوان سؤال، متن آن، متن پاسخ‌های آن است. این مجموعه داده در مطالعه [۱۳] تهیه شده است.

عملیات پیش پردازش

عملیات پیش‌پردازش نقش خیلی مهمی در روند پردازش داده دارد. عملیات پیش‌پردازش مورداستفاده در این تحقیق عبارت‌اند از:

- **جداسازی**[۹]: واحدهای دانش ورودی که در آن متن ورودی به مجموعه‌ای از کلمات شکسته می‌شود.
- **حذف کاراکترهای خاص** نظیر *%$#
- **حذف کلمات توقف**[۱۰]: مرحله‌ی بعدی حذف کلمات توقف است. کلمات توقف شامل کلماتی می‌شوند که به‌طور متداول در همه متن‌ها وجود دارند به همین دلیل ارزش اطلاعاتی کمتری دارند. به‌عنوان‌مثال کلماتی مانند «از، به، و، را» در زبان فارسی جز کلمات توقف حساب می‌شوند. کلماتی مانند the, >> جز کلمات توقف زبان انگلیسی حساب می‌شوند؛ اما مسئله‌ای که ما روی آن کار می‌کنیم کمی با دیگر متن‌های عمومی متفاوت‌تر است. ما با متون حوزه برنامه‌نویسی سروکار داریم، در چنین متن‌هایی یک سری کلمات جز دستورات برنامه‌نویسی محسوب می‌شوند درحالی‌که همان کلمات در یک متن عمومی جز کلمات توقف حساب می‌شوند. به همین دلیل ما یک لیست توقف اختصاصی برای پژوهش خودمان ایجاد کردیم
- **جایگزین کردن یک متن نمونه برای تمامی اعداد موجود در متن**: برای مثال python3 تبدیل به کلمه pythonCC می‌شود. دلیل این امر این است که معمولاً واژگان تخصصی کامپیوتر شامل شماره نسخه متفاوتی می باشند درحالی‌که هر کدام به یک مفهوم واحد اشاره می‌کند. برای مثال در یک جمله شامل python 2 یا Python 3 هر کدام به زبان برنامه‌نویسی پایتون اشاره می‌کنند. اگر پیش‌پردازش صورت نگیرد آن‌گاه تنوع کلمات به وجود می‌آید و در نتیجه، بردار تعبیه آن کلمه به‌درستی توسط تکنیک‌های تعبیه کلمات، یادگیری نخواهد شد.
- **استخراج بردار واژگان با استفاده از تعبیه کلمات**

همان‌طور که می‌دانیم، روش‌های یادگیری عمیق روی داده‌هایی عددی آموزش می‌بینند؛ بنابراین نیاز است تا تبدیل متن خام به داده‌های عددی انجام

[۱۰] Stop words

شود. برای این منظور روش‌های مختلفی وجود دارد اما روشی که سال‌های اخیر خیلی محبوب شده است روش تعبیه کلمات است. در این روش، متناظر با هر کلمه یک بردار شامل اعداد حقیقی حاصل می‌شود. بردارهای تعبیه کلمات بر اساس کلمات مجاور یک کلمه در یک پیکره متنی بزرگ ساخته می‌شود. برای به دست آوردن تعبیه کلمات دو روش کلی وجود دارد، یکی آموزش خود تعبیه کلمات از متن موجود است و دیگری استفاده از مدل‌های از قبل آموزش دیده است. در این تحقیق، از مدل لایه‌ی تعبیه کلمات کتابخانه یادگیری عمیق Keras و همچنین از مدل تعبیه کلمات از پیش آموزش‌دیده استفاده می‌کنیم. مدل‌های از پیش آموزش‌دیده روی پیکره‌های متنی بسیار بزرگ (برای مثال تمامی مقالات ویکی‌پدیا) آموزش داده می‌شوند و آموزش چنین مدل‌هایی نیاز به منابع پردازشی زیادی دارد. به جهت استفاده از پیکره‌های بزرگ متن، بردارهای تعبیه به‌دست‌آمده به‌خوبی منعکس‌کننده معنای کلمات است. در این پژوهش ما از دو مدل تعبیه کلمه از پیش آموزش‌دیده استفاده می‌کنیم که عبارت‌اند از یک مدل آموزش‌دیده Glove [۲۸] منتشرشده توسط دانشگاه استنفورد[۱۱] و دیگری مدلی است که روی داده‌های Stack Overflow آموزش‌دیده است[۱۲].

۳-۳- مدل‌های یادگیری عمیق مورداستفاده در این پژوهش

پس از ارائه عملیات پیش‌پردازش روی متون در بخش قبلی، در این بخش به توصیف مدل‌های مبتنی بر یادگیری عمیق می‌پردازیم. در این پژوهش دو مدل مبتنی بر یادگیری عمیق مورداستفاده قرارگرفته است. در ادامه به توصیف مدل‌های ارائه‌شده می‌پردازیم.

مدل BiLSTM-Cosine_Fe

همان‌طور که در شکل ۲ توصیف شده است، در این مدل ابتدا سه عنصر واحدهای دانشی شامل عنوان سؤال، متن سؤال و پاسخ‌های آن، به‌طور جداگانه وارد لایه تعبیه کلمات می‌شوند. در این لایه به ازای هر کلمه، یک بردار تعبیه به دست می‌آید. سپس تمامی این بردارها وارد لایه LSTM [۲۹] دوطرفه مشترک می‌شوند. به ازای هر کدام از عناصر ورودی، یک بازنمایی سطح بالای آن توسط BiLSTM به دست می‌آید. روی بردارهای خروجی BiLSTM، یک لایه حداکثرگیری کلی استفاده شده و سپس شباهت کسینوسی روی جفت عناصر محاسبه می‌شود. به‌عبارت‌دیگر این مدل یک مدل ترکیبی یادگیری عمیق و معیارهای شباهت کسینوسی است. در این مدل استفاده از لایه حداکثرگیری باعث انتخاب مؤلفه مهم از بردارها شده و بقیه مؤلفه‌ها را حذف کرده و منجر به کاهش ابعاد داده‌ها می‌شود. همان‌طور که در شکل قابل‌مشاهده است، بازنمایی به‌دست‌آمده توسط Bi-LSTM وارد لایه حداکثرگیری شده و درنهایت به همراه خروجی به‌دست‌آمده از محاسبه معیار تشابه کسینوسی وارد لایه مسطح سازی[۱۳] و سپس لایه ادغام[۱۴] شده و درنهایت وارد یک لایه کاملاً متصل می‌شوند. پس از آن برای جلوگیری از بیش برازش مدل یک لایه حذف تصادفی[۱۵] قرار داده‌شده است که خروجی این لایه به لایه آخر که پیش‌بینی ارتباط را انجام می‌دهد می‌رسد.

مدل BiLSTM-Att-Cosine_Fe

هدف از طراحی این مدل گسترش مدل قبلی و بررسی تأثیر اضافه کردن لایه مکانیزم توجه در بهبود تشخیص ارتباط بین واحدهای دانشی است. در حوزه پردازش زبان طبیعی مکانیزم توجه برای اولین بار در مدل‌های ترجمه ماشینی مورداستفاده قرارگرفته‌اند. همان‌طور که در شکل ۳ مشخص است، بازنمایی‌های به‌دست‌آمده توسط لایه Bi-LSTM وارد لایه توجه شده که در این لایه ارتباط بین جفت عنوان، جفت سؤال و جفت پاسخ به دست می‌آید. در مدل، زیر لایه توجه بین جفت عنوان، جفت سؤال و جفت پاسخ قرار داده‌شده است. در حقیقت با این مکانیزم جفت عناصر با یکدیگر مقایسه می‌شوند و کلماتی که دارای بیشترین شباهت در جفت عناصر هستند پیدا می‌شوند. بر اساس این مقایسه، یک بازنمایی جدیدی که نشان‌دهنده ارتباط معنایی بین دو جفت عنصر است حاصل می‌شود که این موضوع به پیش‌بینی دقیق‌تر ارتباط دو واحد دانشی کمک می‌کند.

در مدل ارائه‌شده ابتدا ارتباط بین دو عنصر ورودی پیدا شده و سپس از لایه حداکثرگیری کلی جهت استخراج مهم‌ترین مؤلفه استفاده می‌شود. همچنین در این مدل شباهت کسینوسی بین جفت عناصر محاسبه می‌شود. درنهایت تمامی بازنمایی‌ها بعد از گذر از لایه مسطح سازی ادغام شده و وارد لایه کاملاً مسطح می‌شوند و بعد از آن وارد لایه حذف تصادفی و درنهایت لایه خروجی می‌شوند.

۴- آزمایش‌ها و نتایج

در این مطالعه برای پیاده‌سازی مدل‌های پیشنهادی از زبان برنامه‌نویسی پایتون و کتابخانه یادگیری عمیق Keras [۳۰] استفاده می‌کنیم.

۴-۱- مجموعه داده‌های پژوهش

برای انجام آزمایش‌ها و مقایسه کارایی مدل‌های ارائه‌شده در این تحقیق از مجموعه داده Stack Overflow جمع‌آوری‌شده توسط زو و همکاران [۱۳] که شامل ۴۰۰۰۰ رکورد است، استفاده می‌کنیم. هر رکورد در این مجموعه داده، حاوی یک جفت واحد دانشی و برچسب مربوطه است. این مجموعه داده، از وب‌سایت Stack Overflow جمع‌آوری‌شده است و همه سؤالات مربوط به زبان برنامه‌نویسی Java است. مطابق [۱۳] تعداد ۳۲۰۰۰ رکورد به‌عنوان مجموعه داده آموزشی و تعداد ۸۰۰۰ رکورد به‌عنوان مجموعه داده تست جهت انجام و ارزیابی آزمایش‌ها مورداستفاده قرار گرفتند. جدول ۳ توصیفی از مجموعه‌داده بکارگرفته شده در این مطالعه را ارائه می‌کند.

جدول ۳- توصیف مجموعه داده استفاده شده

تعداد کل نمونه‌ها	تعداد نمونه‌های آموزش	تعداد نمونه‌های تست	تعداد کلاس‌ها
۴۰۰۰۰	۳۲۰۰۰	۸۰۰۰	۴

۴-۲- اندازه‌گیری عملکرد مدل‌ها

مسئله پیش‌بینی ارتباط بین واحدهای دانشی، یک مسئله دسته‌بندی چند کلاسه است که در آن ارتباط بین واحدهای دانشی (سؤالات و پاسخ‌ها) به ۴ کلاس تقسیم می‌شود. برای این منظور معیارهای F1، Recall و Precision در این حوزه مورد استفاده قرار می‌گیرند. این معیارها بر اساس ماتریس اغتشاش نمایش داده شده در شکل ۴ محاسبه می‌شوند.

		Predicted as			
		C1	C2	C3	C4
Actual Label	C1	C_{11}	C_{12}	C_{13}	C_{14}
	C2	C_{21}	C_{22}	C_{23}	C_{24}
	C3	C_{31}	C_{32}	C_{33}	C_{34}
	C4	C_{41}	C_{42}	C_{43}	C_{44}

شکل ۴- ماتریس اغتشاش برای مسئله دسته‌بندی چهار کلاسه

Recall کلاس C_i برابر است با نسبت رکوردهای درست دسته‌بندی‌شده

[۱۱] http://nlp.stanford.edu/data/glove.840B.300d.zip

[۱۲] https://zenodo.org/record/4641569

[۱۳] Flatten

[۱۴] Concatenation

[۱۵] Dropout

به‌عنوان کلاس C_i به کل رکوردهایی که کلاس واقعی آن‌ها C_i می‌باشد که از طریق رابطه زیر محاسبه می‌شود.

$$Recall_i = \frac{C_{ii}}{\sum_{1\le j\le K} C_{ij}} \quad (۱)$$

Precision کلاس C_i برابر است با نسبت رکوردهای درست دسته‌بندی‌شده به‌عنوان کلاس C_i به کل رکوردهای دسته‌بندی‌شده به‌عنوان کلاس C_i توسط مدل که به‌صورت زیر محاسبه می‌شود.

$$Precision_i = \frac{C_{ii}}{\sum_{1\le j\le K} C_{ji}} \quad (۲)$$

یکی از محبوب‌ترین معیارهای اندازه‌گیری دقت دسته‌بندی معیار F1-measure است. این معیار یک نوع میانگین هارمونیک بین معیار Precision و معیار Recall است که برای کلاس C_i با استفاده از رابطه زیر محاسبه می‌شود.

$$F1-measure_i = 2\cdot\frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (۳)$$

۴-۳- تنظیم پارامترها

در یادگیری عمیق، مبحث تنظیم ابرپارامترها نقش مهمی در عملکرد یادگیری مدل‌های یادگیری عمیق دارد. ابرپارامترهای تنظیم شده در جدول ۴ نمایش داده شده است.

جدول ۴ - ابرپارامترهای مورداستفاده

ابرپارامتر	مقدار
ورودی	۲ جفت واحد دانشی به طول ۲۷۰
خروجی	پیش‌بینی ارتباط
ابعاد لایه تعبیه	۳۰۰
لایه اصلی Bi-LSTM	تعداد نورون‌ها: ۳۰۰
نرخ لایه حذف تصادفی[۱۶]	۰,۲
تابع فعال‌ساز[۱۷]	RELU
تابع بهینه‌ساز	آدام[۱۸] [۳۱]
تابع هزینه[۱۹]	MSE
تعداد تکرار[۲۰]	۱۰

۴-۴- نتایج آزمایش‌ها

جدول ۵ نتایج آزمایش‌ها را بر مبنای معیارهای Recall, Precision, F1-measure نشان می‌دهد. همان‌طور که قبلاً اشاره شد در این مطالعه در لایه بازنمایی کلمات از دو مدل تعبیه کلمات از پیش آموزش‌دیده شامل Stack Overflow و Glove در آموزش مدل‌ها استفاده شده است. همچنین لایه تعبیه کلمات در کتابخانه Keras نیز برای آموزش هر دو مدل استفاده شده است.

نحوه به‌کارگیری مدل‌های آموزش‌دیده تعبیه کلمات به این صورت است که ابتدا بردار عددی هر کلمه از مدل استخراج شده و ماتریس نگاشت کلمه به بردار تشکیل می‌شود. ماتریس به‌دست‌آمده به‌عنوان اوزان لایه تعبیه کلمه در مدل بارگذاری می‌شود. سپس لایه تعبیه کلمه به دو حالت قابل‌آموزش (Trainable) و غیرقابل‌آموزش (Non-trainable) در مدل مورداستفاده قرار می‌گیرد. تفاوت این دو به این صورت است که در حالت قابل‌آموزش، اوزان لایه تعبیه کلمه در طول اجرای مدل آپدیت می‌شوند درحالی‌که در حالت غیرقابل‌آموزش، اوزان لایه تعبیه کلمه فریز شده و در طول اجرا مدل تغییر نمی کنند. استفاده از مدل تعبیه کلمات از پیش آموزش‌دیده یک نوع یادگیری انتقالی محسوب می‌شود چراکه دانش یادگیری شده در یک مسئله قبلی در مسئله فعلی مورداستفاده قرار می‌گیرد.

همان‌طور که در جدول ۵ مشاهده می‌شود مدل ترکیبی حاصل از معماری دوم (BiLSTM-Att-Cosine_Fe) بهترین عملکرد را در بین تمامی مدل‌ها در معیار F1 به دست می‌آورد (F1=0.61). همچنین هر دو مدل یادگیری عمیق ارائه‌شده در این تحقیق، زمانی که از مدل تعبیه کلمات پیش آموزش‌دیده در آن‌ها مورداستفاده قرار می‌گیرد نسبت به زمانی که از لایه تعبیه کلمات کتابخانه کراس استفاده می‌کنند نتایج عملکردی بسیار بالاتری در معیارهای Recall, Precision, F1-measure دارند که این موضوع نشان‌دهنده این است که مدل‌های از پیش آموزش‌دیده، بازنمایی بهتری از کلمات را دارند. همچنین بررسی دقیق‌تر نتایج نشان می‌دهد که در آزمایش‌های ما، دو مدل یادگیری عمیق به‌طورکلی نتایج بهتری هنگام استفاده از مدل از پیش آموزش‌دیده در حالت غیرقابل‌آموزش نسبت به زمانی که اوزان لایه تعبیه در طول آموزش شبکه تغییر می‌کند، از خود نشان می‌دهند. این موضوع برای هر دو مدل Stack Overflow و Glove صادق است. همچنین مطابق نتایج جدول ۵، مدل تعبیه Glove که روی متون عمومی آموزش‌دیده است نسبت به مدل تعبیه به‌دست‌آمده از داده‌های متون برنامه‌نویسی، روی مدل‌های مورداستفاده در این تحقیق تأثیر بهتری داشته است. یک دلیل برای این موضوع می‌تواند اندازه پیکره متنی که مدل‌های تعبیه کلمات با آن آموزش‌دیده شده‌اند باشد.

بررسی نتایج نشان می‌دهد که به‌طورکلی مدل ترکیبی حاصل از معماری دوم (BiLSTM-Att-Cosine_Fe) نتایج بهتری نسبت به مدل به‌دست‌آمده از معماری اول (BiLSTM-Cosine_Fe) در تمامی حالت‌ها دارد که این خود نشان‌دهنده تأثیر استفاده از معیارهای تشابه در کنار لایه‌های شبکه یادگیری عمیق و استفاده از مکانیزم توجه که موجب می‌شود که ارتباط معنایی بین کلمات در نظر گرفته شود.

۴-۵- مقایسه با روش پایه و مدل‌های ارائه‌شده در کارهای فعلی

برای تحلیل بهتر مدل به‌دست‌آمده، نتایج آن را با مدل‌های فعلی ارائه‌شده مقایسه می‌کنیم. برای مقایسه مدل‌ها از نتایج مقاله [۱۳] استفاده می‌کنیم چراکه ما از مجموعه داده استفاده‌شده در آن مقاله برای انجام آزمایش‌های این مطالعه استفاده کرده‌ایم. نتایج در جدول ۶ و شکل ۵ نمایش داده شده است.

جدول۵- نتایج مدل‌های مورداستفاده – در این جدول SO نشان‌دهنده استفاده از مدل بازنمایی کلمات از پیش آموزش‌دیده روی داده‌های Stack Overflow است.

مدل	Precision	Recall	F1-measure
Bilstm-Cosine_Fe +SO (Non-Trainable)	۵۹/۰	۵۸/۰	۵۸/۰
Bilstm-Cosine_Fe +SO (Trainable)	۵۸/۰	۵۸/۰	۵۸/۰
Bilstm-Att-Cosine_Fe+ SO (Non-Trainable)	۵۹/۰	۵۹/۰	۵۹/۰
Bilstm-Att-Cosine_Fe + SO (Trainable)	۵۸/۰	۵۸/۰	۵۸/۰
Bilstm-Cosine_Fe + Glove (Non-Trainable)	۵۹/۰	۵۹/۰	۵۹/۰
Bilstm-Cosine_Fe + Glove (Trainable)	۵۹/۰	۵۸/۰	۵۸/۰
Bilstm-Att-Cosine_Fe+ Glove (Non-Trainable)	**۶۱/۰**	**۶۱/۰**	**۶۱/۰**
Bilstm-Att-Cosine_Fe + Glove (Trainable)	۵۸/۰	۵۸/۰	۵۸/۰
Bilstm-Cosine_Fe + Keras	۵۰/۰	۴۹/۰	۴۹/۰
Bilstm-Att-Cosine_Fe + Keras	۵۱/۰	۵۰/۰	۵۰/۰

[۱۶] Dropout
[۱۷] Activation function
[۱۸] Adam
[۱۹] Loss
[۲۰] Epoch

جدول ۶- مقایسه با مدل‌های فعلی

مدل	Precision	Recall	F1
مدل Soft SVM [۱۳]	۵۳/۰	۵۴/۰	۵۲/۰
مدل Tuning SVM [۱۳]	۴۹/۰	۵۱/۰	۵۰/۰
مدل CNN [۱۳]	۵۰/۰	۴۱/۰	۴۱/۰
Bilstm-Att-Cosine_Fe+ Glove (Non-Trainable)	**۶۱/۰**	**۶۱/۰**	**۶۱/۰**

همان‌طور که مشاهده می‌کنید روش ما از منظر F1 نسبت به مدل Soft SVM در [۱۳] ۹ درصد بهبود داشته است. همچنین نسبت به مدل Tuning SVM بهترین مدل به‌دست‌آمده دارای اختلاف عملکردی ۱۱ درصدی می‌باشد؛ و بیشترین اختلاف نیز نسبت به مدل مبتنی بر شبکه عصبی پیچشی (CNN) وجود دارد که برابر با ۲۰ درصد می‌باشد. همچنین برای مقایسه بیشتر، عملکرد مدل‌ها در پیش‌بینی هر کلاس در جدول ۷ آورده شده است همان‌طور که مشخص است در تمامی برچسب‌ها مدل ارائه‌شده در این تحقیق نتایج بهتری از نقطه‌نظر F1-measure دارد

جدول ۷- نتایج مقایسه عملکردی BiLSTM-Att-Cosine_Fe+ Glove (Non-trainable) با بهترین مدل فعلی

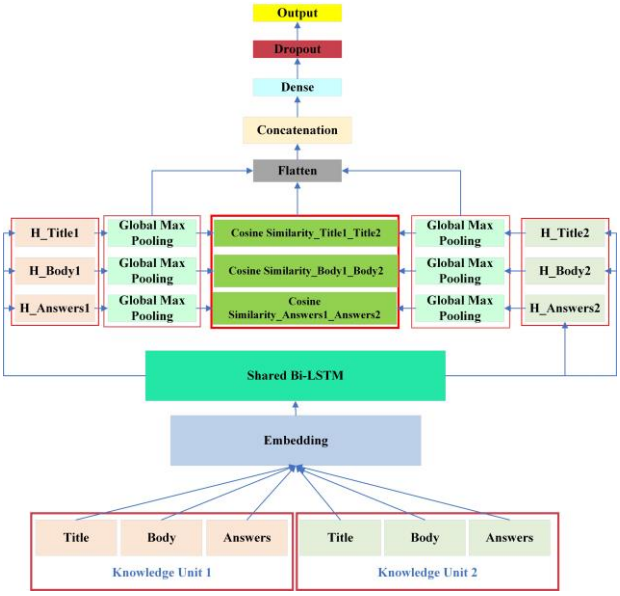
برچسب	مدل	Precision	Recall	F1
Direct Link	BiLSTM-Att-Cosine_Fe+ Glove (non-trainable)	۰,۴۶	۰,۴۶	**۰,۴۶**
	مدل Soft SVM [۱۳]	۰,۴۵	۰,۲۱	۰,۲۹
Duplicate	BiLSTM-Att-Cosine_Fe+ Glove (non-trainable)	۰,۶۳	۰,۶۳	**۰,۶۳**
	مدل Soft SVM [۱۳]	۰,۵۱	۰,۴۸	۰,۵۰
Indirect Link	BiLSTM-Att-Cosine_Fe+ Glove (non-trainable)	۰,۴۹	۰,۵۰	**۰,۵۰**
	مدل Soft SVM [۱۳]	۰,۴۲	۰,۵۸	۰,۴۹
Isolated	BiLSTM-Att-Cosine_Fe+ Glove (non-trainable)	۰,۸۶	۰,۸۴	**۰,۸۵**
	مدل Soft SVM [۱۳]	۰,۷۵	۰,۹۰	۰,۸۲

۵- نتیجه‌گیری و بیان کارهای آتی

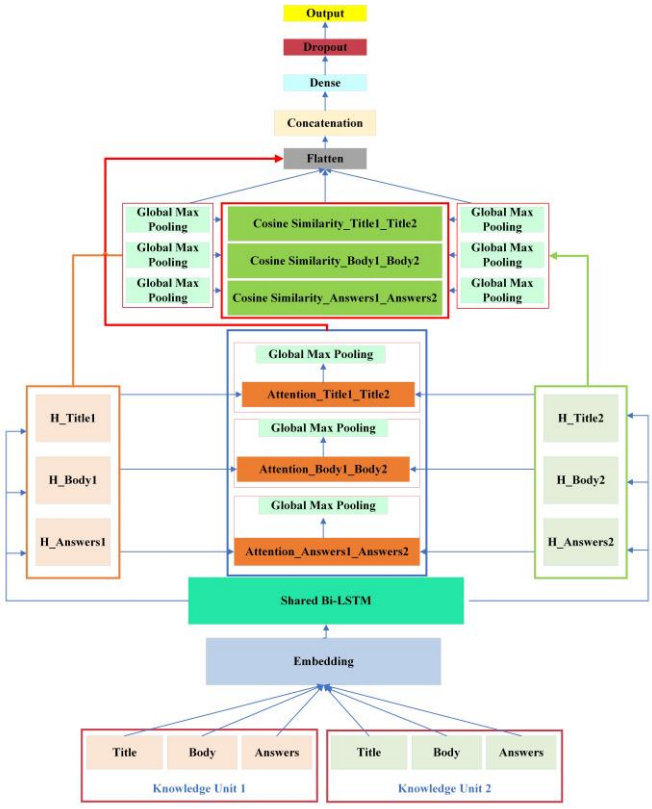
وب‌سایت‌های پرسش‌وپاسخ مانند Stack Overflow، بستری را فراهم کرده‌اند که در آن برنامه‌نویسان می‌توانند سؤالات خود را پرسیده و به پاسخ موردنظر خود برسند. ممکن است سؤالی که برنامه‌نویسان مطرح می‌کنند قبلاً مطرح شده باشد و پاسخ صحیح آن داده شده باشد ازاین‌رو مکانیزمی باید وجود داشته باشد که سؤالات مرتبط را تشخیص بدهد. اگر یک سؤال و مجموعه پاسخ‌های آن را به‌عنوان یک واحد دانشی در نظر بگیریم، آنگاه ارتباطات معنایی بین واحدهای دانشی می‌توان مشخص کرد. در این مطالعه، مسئله تشخیص ارتباط بین واحدهای دانشی به‌صورت یک مسئله چهار کلاسه فرموله شده و دو معماری یادگیری عمیق جهت پیش‌بینی ارتباط ارائه گردید. معماری اول به‌طورکلی شامل لایه ورودی، لایه تعبیه کلمات، لایه BiLSTM، لایه حداکثرگیری، لایه محاسبه شباهت کسینوسی، یک لایه تمام متصل و لایه خروجی است که دو واحد دانشی را به‌عنوان ورودی دریافت کرده و کار پیش‌بینی را انجام می‌دهد. مهم‌ترین ویژگی معماری پیشنهادی گنجاندن لایه شباهت کسینوسی در مدل یادگیری عمیق است که باعث بهبود عملکرد آن می‌شود. همچنین مدل دیگری که گسترش‌یافته معماری اول است ارائه گردید که در این مدل از مکانیزم توجه استفاده‌شده است. به کمک این لایه یک بازنمایی جدیدی که نشان‌دهنده ارتباط معنایی بین دو جفت عنصر است حاصل می‌شود که این موضوع به پیش‌بینی دقیق‌تر ارتباط دو واحد دانشی کمک می‌کند. در این مطالعه از دو مدل تعبیه کلمات از پیش آموزش‌دیده در مدل‌ها استفاده شد. نتایج آزمایش‌ها روی مجموعه داده Stack Overflow نشان می دهد که استفاده از لایه تعبیه از پیش آموزش‌دیده به افزایش دقت مدل کمک می‌کند. همچنین هر دو مدل ارائه‌شده نسبت به مدل‌های فعلی عملکرد بهتری از نظر معیارهای F1-measure، Precision و Recall دارند. مدل BiLSTM-Att-Cosine_Fe با لایه تعبیه از پیش آموزش‌دیده Glove بهترین عملکرد را در بین مدل‌های پیاده‌سازی شده دارد. به‌طورکلی نتایج به‌دست‌آمده در مدل‌های کنونی جای بهبود بسیار دارد که مدل پیشنهادی در این مطالعه ۱۷,۳ درصد این بهبود را محقق ساخته است. به‌کارگیری روش‌های بازنمایی متن نظیر تعبیه کاراکترها به‌جای تعبیه کلمات و همچنین استفاده از روش بازنمایی BERT و در نظر گرفتن وب‌سایت‌های پرسش و پاسخی که سؤالات کلی‌تری در آن‌ها بحث می‌شود جزو کارهای آتی ما خواهد بود.

جدول ۲- مثالی از واحد دانش

واحد دانش	متن	عنوان
سؤال	I've been using the == operator in my program to compare all my strings so far. However, I ran into a bug, changed one of them into .equals() instead, and it fixed the bug. Is == bad? When should it and should it not be used? What's the difference?	How do I compare strings in Java?
سؤال تکراری	String s1 = "andrei"; String s2 = "andrei"; String s3 = s2.toString(); System.out.println((s1==s2) + "␣" + (s2==s3)); Giving the following code why is the second comparison s2 == s3 true ? What is actually s2.toString() returning ? Where is actually located (s2.toString()) ?	Strings in Java : equals vs ==
سؤال دارای ارتباط مستقیم	String a = "devender"; String b = "devender"; String c = "dev"; String d = "dev" + "ender"; String e = c + "ender"; System.out.println(a == b); //case 1: o/p true System.out.println(a == d); //case 2: o/p true System.out.println(a == e); //case 3: o/p false a & b both are pointing to the same String Literal in string constant pool. So true in case 1 String d = "dev" + "ender"; should be internally using something like - String d = new StringBuilder().append("dev").append("ender").toString(); How a & d are pointing to the same reference & not a & e ?	"==" in case of String concatenation in Java
سؤال دارای ارتباط غیرمستقیم	I have a long string that doesn't fit the width of the screen. For eg. String longString = "This␣string␣is␣very␣long..."; To make it easier to read, I thought of writing it this way - String longString = "This␣string␣is␣very␣long..." + "This␣string␣is␣very␣long..." + ...; However, I realized that the second way uses string concatenation and will create 5 new strings in memory and this might lead to a performance hit. Is this the case? Or would the compiler be smart enough to figure out that all I need is really a single string? How could I avoid doing this?	Does concatenating strings in Java always lead to new strings being created in memory?



شکل ۲- BiLSTM-Cosine_Fe



شکل ۳- مدل BiLSTM-Att-Cosine_Fe

مراجع

[۱] مهدی دهقان، احمدعلی آبین، « بازیابی و رتبه‌بندی افراد خبره با استفاده از مدل ترجمه مبتنی بر خوشه‌بندی»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۹، شماره ۳، صفحات ۱۱۰۶-۱۰۹۵، ۱۳۹۸.

[2] P. Chakraborty, R. Shahriyar, A. Iqbal, and G. Uddin, "How do developers discuss and support new programming languages in technical Q&A site? An empirical study of Go, Swift, and Rust in Stack Overflow", Information and Software Technology, vol. 137, pp. 106603, 2021.

[3] H. Shu, P. Gao, Z. Yang, C. Li, and M. Wu, "Exploring the Feasibility of Transformer Based Models on Question Relatedness", Proceedings of 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), pp. 831-838 , 2022.

[4] J. He, Z. Xin, B. Xu, T. Zhang, K. Kim, Z. Yang, et al., "Representation Learning for Stack Overflow Posts: How Far are We?", arXiv preprint arXiv:2303.06853, 2023.

[5] A. Kumar, D. Ghadiyali, S. Chimalakonda, and A. S. M. Venigalla, "SOCluster-Towards Answering Unanswered Questions on Stack Overflow via Answered Questions", Proceedings of the 16th Innovations in Software Engineering Conference, pp . 1-5,2023.
[6] S.-K. Guo, S.-W. Wang, H. Li, Y.-L. Fan, Y.-Q. Liu, and B. Zhang, "Multi-Feature Fusion Based Structural Deep Neural Network for Predicting Answer Time on Stack Overflow", Journal of Computer Science and Technology, vol. 38, no. 3, pp. 582-599, 2023.
[7] P. K. Roy, S. Saumya, J. P. Singh, S. Banerjee, and A. Gutub, "Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art review", CAAI Transactions on Intelligence Technology, vol. 8, no. 1, pp. 95-117, 2023.
[8] B. Xu, D. Ye, Z. Xing, X. Xia, G. Chen, and S. Li, "Predicting semantically linkable knowledge in developer online forums via convolutional neural network", Proceedings of 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 51-62, 2016.
[9] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, "Prompter: Turning the IDE into a self-confident programming assistant", Empirical Software Engineering, vol. 21, pp. 2190-2231, 2016.
[10] Y. Zhang, D. Lo, X. Xia, and J.-L. Sun, "Multi-factor duplicate question detection in stack overflow", Journal of Computer Science and Technology, vol. 30, pp. 981-997, 2015.
[11] W. E. Zhang, Q. Z. Sheng, J. H. Lau, E. Abebe, and W. Ruan, "Duplicate detection in programming question answering communities", ACM Transactions on Internet Technology (TOIT), vol. 18, no. 3, pp. 1-21, 2018.
[12] W. Gao, J. Wu, and G. Xu, "Detecting duplicate questions in stack overflow via source code modeling", Int J Software Eng Knowl Eng, vol. 32, no. 02, pp. 227-255, 2022.
[13] B. Xu, A. Shirani, D. Lo, and M. A. Alipour, "Prediction of relatedness in stack overflow: deep learning vs. svm: a reproducibility study", Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 1-10, 2018.
[14] A. Shirani, B. Xu, D. Lo, T. Solorio, and A. Alipour, "Question relatedness on stack overflow: the task, dataset, and corpus-inspired models", arXiv preprint arXiv:1905.01966, 2019.
[۱۵] رضا خدایی، محمدعلی بالافر، سیدناصر رضوی،« اثربخشی بسط پرس‌وجو مبتنی بر خوشه‌بندی اسناد شبه‌بازخورد با الگوریتم»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۱، صفحات ۱۴۳-۱۵۱، ۱۳۹۵.
[۱۶] مرضیه رحیمی، عرفان جلیلی جلال، حسین رحیمی، « تولید کلمات کلیدی متون فارسی با استفاده از یادگیری انتقالی»، مجله مهندسی برق دانشگاه تبریز، جلد ۵۲، شماره ۲، صفحات ۱۱۵-۱۲۳، ۱۴۰۱ .
[17] E. Zafarani-Moattar, M. R. Kangavari, and A. M. Rahmani, "Topic Detection on COVID-19 Tweets: A Comparative Study on Clustering and Transfer Learning Models", TABRIZ JOURNAL OF ELECTRICAL ENGINEERING, vol. 52, no. 4, pp. 281-291, 2022.
[18] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation", arXiv preprint arXiv:1508.04025, 2015.
[19] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider, "Mining duplicate questions in stack overflow", Proceedings of the 13th International Conference on Mining Software Repositories, pp. 402-412, 2016.
[20] L. Wang, L. Zhang, and J. Jiang, "Duplicate Question Detection With Deep Learning in Stack Overflow", IEEE Access, vol. 8, pp. 25964-25975, 2020.
[21] R. F. G. Silva, K. Paixão, and M. d. A. Maia, "Duplicate question detection in stack overflow: A reproducibility study", Proceedings of 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 572-581, 2018.
[22] W. E. Zhang, Q. Z. Sheng, Y. Shu, and V. K. Nguyen, "Feature analysis for duplicate detection in programming QA communities", Proceedings of Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, pp. 623–638, 2017.
[23] N. Ansari and R. Sharma, "Identifying semantically duplicate questions using data science approach: A quora case study", arXiv preprint arXiv:2004.11694, 2020.
[24] A. Kamienski, A. Hindle, and C.-P. Bezemer, "Analyzing Techniques for Duplicate Question Detection on Q&A Websites for Game Developers", Empirical Software Engineering, vol. 28, no. 1, pp. 17, 2023.
[25] D. Charlet and G. Damnati, "Simbow at semeval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering," Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 315–319, 2017.
[26] J. Pei, Y. Wu, Z. Qin, Y. Cong, and J. Guan, "Attention-based model for predicting question relatedness on Stack Overflow", Proceedings of. 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), pp. 97-107, 2021.
[27] "Stack Overflow", avaiable online at: https://en.wikipedia.org/wiki/Stack_Overflow
[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation", Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543, 2014.
[29] A. Darvish and S. Shamekhi, "A hybrid multi-scale CNN-LSTM deep learning model for the identification of protein-coding regions in DNA sequences", TABRIZ JOURNAL OF ELECTRICAL ENGINEERING, vol. 52, no. 2, pp. 137-146, 2022.
[30] F. Chollet, "Keras: The python deep learning library", Astrophysics source code library, 2018.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980, 2014.