مجله محاسبات نرم، جلد 14، شماره 2 (پاییز و زمستان 1404)، صفحه 2_19
https://doi.org/10.22052/scj.2024.254464.1224

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

University of Kashan

مرور روش‌های جاسازی گراف‌های دانش✧

سید مهدی وحیدی‌پور[1*]، استادیار، داود دانشمند[1]، دانشجوی دکتری، محمد علی ظریف[1]، دانشجوی دکتری

[1] دانشکده برق و کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت 5 اسفند ماه 1402
پذیرش 14 مرداد ماه 1403

کلمات کلیدی:
گراف دانش
مدل‌های فاصله‌ای
مدل‌سازی دوخطی
شبکه‌های عصبی
نمونه‌گرافهای OGB

چکیده

گراف‌های دانش به عنوان ابزارهای حیاتی و بسیار پرکاربرد در زمینه‌های داده‌کاوی، یادگیری ماشین، شبکه‌های پیچیده و حوزه‌های مرتبط، توجه بسیاری را به خود جلب کرده‌اند. در این مقاله، ما سعی داریم تا به بررسی گسترده‌ای از این گراف‌ها و روش‌های مرتبط با آنها بپردازیم. ما روش‌های شبکه‌های RGNN، TDM و BLM را جزئی‌تر بررسی و ارزیابی می‌کنیم و نتایج به دست آمده از هر کدام را با دقت مورد بررسی قرار می‌دهیم. این مطالعه یک تجزیه و تحلیل جامع از عملکرد این روش‌ها را ارائه می‌دهد و به دنبال ارزیابی دقیق‌تر نقاط قوت و ضعف هر یک از آنها در مواجهه با چالش‌ها و کاربردهای واقعی است. امیدواریم که این بررسی گامی مفید در جهت بهبود درک ما از این فناوری مهم باشد و در توسعه آینده آن موثر باشد.

1. مقدمه

گراف دانش[1] (KG) نوع خاصی از ساختار گراف با موجودیت‌ها به عنوان گره و روابط به عنوان یال، هم در داده‌کاوی و هم در یادگیری ماشین کاربرد دارد [1]. به عنوان مثال در یافتن هدف در تعامل کاربر و آیتم [2]، استدلال [3]، سامانه‌های پاسخ به سوال[2] [4]، سامانه‌های توصیه‌گر[3] [5] و پردازش زبان‌های طبیعی کاربرد [6] دارد.

یکی از مسائل اساسی یادگیری ماشین در گراف دانش، پیش‌بینی سه‌گانه جدید است؛ محاسبه احتمال وجود یک سه‌گانه. به عبارت دیگر، با استفاده از سه‌گانه‌های موجود در گراف یادگیری انجام شده و سه‌گانه‌های جدیدی پیش‌بینی شوند. استفاده از جاسازی گراف دانش[4] (KGE)، به عنوان یک روش یادگیری ماشین در این مساله توسعه یافته است [7]. KGE برای هر موجودیت و هر رابطه یک نمایش برداری کم‌بُعد (جاسازی[5]) ایجاد می‌کند. حال بر اساس مجموعه‌ای از سه‌گانه‌های موجود در گراف دانش (داده‌های آموزشی)، جاسازی‌های ایجاد شده ارزیابی می‌شوند؛ از تابع امتیازدهی SF[6] برای ارزیابی استفاده می‌شود. با استفاده از این ارزیابی KGE تلاش می‌کند تا

✧ نوع مقاله: مروری
* نویسنده مسئول
پست(های) الکترونیک: vahidipour@kashanu.ac.ir (وحیدی‌پور)
daneshmand@grad.kashanu.ac.ir (دانشمند)
mazarif@grad.kashanu.ac.ir (ظریف)

[1] Knowledge Graph
[2] Question-Answering systems
[3] Recommender system
[4] Knowledge Graph Embedding
[5] Embedding
[6] Score Function

نحوه ارجاع به مقاله: سید مهدی وحیدی‌پور، داود دانشمند، محمد علی ظریف، «مرور روش‌های جاسازی گراف‌های دانش»، مجله محاسبات نرم، جلد 14، شماره 2، ص2_19، پاییز و زمستان 1404.

جاسازی‌های بهتری را برای موجودیت‌ها و روابط ایجاد کند (یاد بگیرد) که امتیاز بالاتری در SF دریافت کنند. این پروسه تکراری تا زمانی که عملکرد روی داده آموزشی بهبود می‌یابد ادامه خواهد داشت. همان‌طور که مشخص است یادگیری بهترین جاسازی در گراف دانش، نیازمند تابع امتیازدهی SF مناسب است. در همین راستا، روش‌های مبتنی بر شبکه پتری فازی نیز برای مدل‌سازی پیچیدگی‌های سیستم‌های دانش‌بنیان معرفی شده‌اند [8]، که این روش‌ها توانسته‌اند با استفاده از مدل‌های سلسله مراتبی، میزان پیچیدگی قوانین و استنتاج رفتار آنها را کاهش دهند.

در شبکه‌های اجتماعی نیز، پیش‌بینی پیوند به دلیل ماهیت دینامیک شبکه‌ها و افزایش مستمر اعضا و پیوندها از اهمیت بالایی برخوردار است [9]. این پیوندها ممکن است به دلایل مختلف از دست بروند و روش‌هایی برای بازیابی و پیشگویی آنها پیشنهاد شده‌اند. در زمینه پیش‌بینی پیوند در شبکه‌های اجتماعی و گراف‌های دانش، استفاده از روش‌های مبتنی بر شبکه‌های عصبی گرافی (GNN) نیز گسترش یافته است. به عنوان مثال، در مقاله [10]، روشی برای پیش‌بینی پیوند با استفاده از یک چارچوب جدید به نام SGAE ارائه شده است که بر اساس مفهوم زیرگراف‌ها بهبود یافته است. این روش با استفاده از شبکه‌های عصبی گرافی به منظور استخراج بردار ویژگی جفت گره‌ها و با هدف بهبود دقت در پیش‌بینی پیوندها، عملکرد بالاتری را نسبت به روش‌های پایه نشان داده است. مقایسه‌های انجام شده در این تحقیق نشان می‌دهد که چارچوب SGAE توانسته است بهبود قابل‌توجهی در معیارهایی مانند دقت، F1-Score و مساحت زیر نمودار صحت-فراخوانی نسبت به روش‌های پایه ایجاد کند که این موضوع می‌تواند الهام‌بخش توسعه روش‌های جدید در گراف‌های دانش نیز باشد.

هدف این مقاله، بررسی و توضیح روش‌های مختلف جاسازی گراف دانش امتیاز محور[1] می‌باشد که خود دارای سه دسته کلی می‌باشد: (1) روش‌های مبتنی بر مدل‌های فاصله‌ای[2]، (2) روش‌های مبتنی بر مدل‌سازی دو خطی[3] و (3) روش‌های مبتنی بر شبکه‌های عصبی[4]. پس از آن مقایسه‌ای میان روش‌ها و دسته‌های مذکور انجام می‌دهیم.

ساختار مقاله در ادامه به صورت زیر است. در قسمت دوم به بررسی مفاهیم اولیه پرداخته می‌شود. در بخش سوم روش‌های جاسازی امتیاز محور به طور کامل مورد بحث و بررسی قرار می‌گیرند. در بخش چهارم روش‌های بحث شده در بخش سوم را بر اساس معیارها و مجموعه داده‌های مختلف مورد مقایسه قرار می‌دهیم.

2. مفاهیم اولیه

این بخش به بحث و بررسی مفاهیم اولیه در حوزه گراف دانش می‌پردازد.

1.2. گراف دانش

یک گراف دانش با چهارتایی $G = (V, E, R, T)$ تعریف می‌شود که در آن V مجموعه گره‌ها، E مجموعه یال‌ها، R مجموعه رابطه‌ها و T مجموعه انواع گره‌ها است. در گراف دانش هر یال به صورت سه‌گانه[5] نشان داده می‌شود: (سرآیند، رابطه ، تالی) یا (Head، Relation، Tail). این سه‌گانه به صورت خلاصه به ترتیب با h، r و t نشان داده می‌شوند. Head و Tail هر کدام می‌توانند یک گره یا موجودیت باشند و Relation نیز رابطه بین این دو موجودیت است؛ این ترکیب سه‌گانه تشکیل یک حقیقت را می‌دهد.

گراف دانش، بر اساس تعداد زیادی سه‌گانه ساخته می‌شود و هر سه‌گانه بیانگر یک حقیقت در دنیای دانش است. به عنوان مثال جمله «علی پدر حسن، راننده است» را در نظر بگیرید. واضح است که این جمله ساده دارای دو حقیقت است: «نام پدر حسن، علی است» و «علی یک راننده است». این دو حقیقت، به سه‌گانه‌های (علی، شغل، راننده)، (علی، پدر، حسن) و (حسن، پسر، علی) تبدیل می‌شوند. با این سه‌گانه‌ها گراف دانش زیر به

[1] Score function based
[2] Translational distance models
[3] Bilinear models
[4] Neural network based models
[5] Triple

دست خواهد آمد:

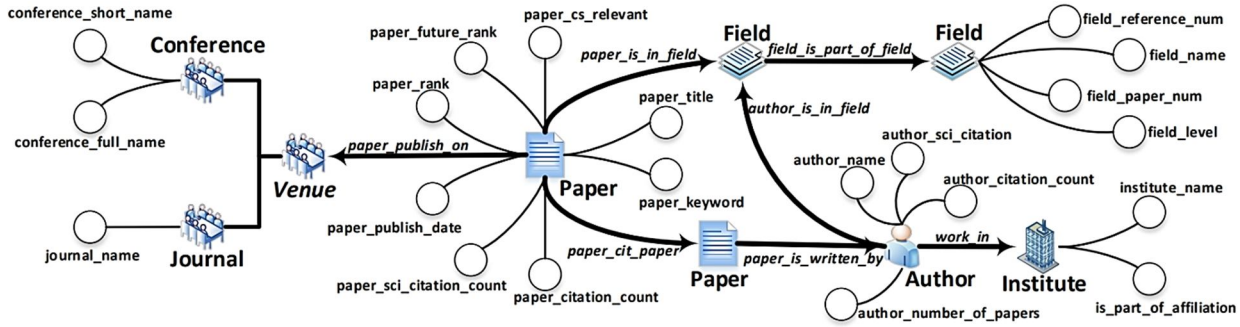
V = {"راننده", "حسن", "علی"}

E = {("راننده" و "علی")، ("حسن" و "علی")، ("علی" و "حسن")}

R = {"پسر", "پدر", "شغل"}

T = {"شغل", "انسان"}

در شکل (1)، یک مثال از گراف دانش در حوزه دانشگاه ها و تعاملات بین اعضای هیئت علمی و مقالات ارائه شده آمده است [11]. تمام اشیا (مانند مقالات، موسسات، نویسندگان) به عنوان موجودیت‌ها در AceKG نمایش داده می‌شوند. دو موجودیت می‌توانند با هم، یک رابطه داشته باشند. ویژگی‌های معمول هر موجودیت، شامل اعداد، تاریخ‌ها، رشته‌ها و دیگر متغیرهای ساده، نیز نمایش داده می‌شوند. موجودیت‌های مشابه در کلاس‌هایی گروه‌بندی می‌شوند. در مجموع، AceKG پنج کلاس از موجودیت‌های آکادمیک را تعریف می‌کند: مقالات، نویسندگان، مکان‌ها، حوزه‌های مطالعاتی و موسسات. واقعیت‌ها، شامل ویژگی‌های پرکاربرد هر موجودیت و روابط بین موجودیت‌ها، به صورت سه‌گانه‌ها در نمودار دانش توصیف می‌شوند.



شکل (1): یک مثال تصویری از یک گراف دانش ناهمگون

2.2. کاربردهای گراف دانش

گراف‌های دانش دارای کاربردهای فراوانی هستند که در ادامه به دو کاربرد آن به‌صورت نمونه اشاره می‌شود [12]:

الف) ارائه اطلاعات: اگر گراف دانشی وجود داشته باشد که در آن نام فیلم‌ها و کارگردان‌های آنها و همچنین بازیگران فیلم‌ها آمده باشد، این گراف می‌تواند برای پاسخ‌دهی به انواع کوئری‌ها قابل استفاده باشد، به عنوان مثال نام تمام فیلم‌های کارگردان فیلم «تایتانیک». این گونه اطلاعات از طریق ساخت گراف دانش و تحلیل آنها قابل ارائه هستند.

ب) پاسخ به سوالات: به طور سنتی، پاسخ به سوالات با استفاده از محتواهای بدون ساختار مانند پاراگراف، متن و یا صفحه ویکی پدیا انجام می‌شود. قابلیت‌های اکتشافی چنین سیستم‌هایی محدود به محتوای داده شده است، به عنوان مثال، با داشتن یک صفحه ویکی‌پدیا در مورد آلبرت انیشتین یک کاربر می‌تواند با پرسیدن سوال «آلبرت انیشتین متولد کجاست؟» محل تولد انیشتین را به دست آورد. اما پرسیدن سوالات بیشتر در مورد این شهر (به عنوان مثال، «چند نفر در یوالِام زندگی می‌کنند؟») نتایج مناسبی را برنمی‌گردانند. در این حالت، ما به پیشینه ساختاری احتیاج داریم که گراف‌های دانش برای ما به همراه می‌آورند.

3.2. گراف‌های دانش در دسترس عموم

شرکت‌های بزرگ فناوری اطلاعات، گراف‌های دانش متنوعی را ارائه کرده‌اند. همچنین دانشگاه استنفورد نیز به تازگی مجموعه داده‌های متنوعی را برای آزمون و ارزیابی الگوریتم‌ها ارائه کرده است[1]. در ادامه به چند گراف دانش اشاره می‌شود که به صورت عمومی ارائه شده است:

الف) فری‌بیس[2]: این گراف دانش توسط شرکت گوگل و بر اساس داده‌های موجود در ویکی‌پدیا تولید شده است. اولین بار

[1] شامل دو گراف دانش بزرگ مرتبط به ویکی‌پدیا و داده‌های بیولوژیکی هستند که در https://ogb.stanford.edu/docs/dataset_overview قابل دسترسی است.

[2] FreeBase

این گراف در سال 2007 ایجاد شده و توسعه آن در سال 2016 متوقف شده است. این گراف شامل بیش از 50 میلیون گره، 38 هزار نوع رابطه و بیش از 3 میلیارد سه‌گانه است. با این حال بخش زیادی از رابطه‌ها در این گراف کامل نیستند. به عنوان مثال 78 درصد از تمام اشخاصی که در این گراف ثبت شده‌اند، فاقد ویژگی «ملیت» هستند و 93 درصد آنها فاقد «تاریخ تولد» هستند. به همین دلیل لزوم توسعه ابزارهای تکمیل گراف، ضروری به نظر می‌رسد.

ب) ویکی‌دیتا: یک گراف دانش بسیار بزرگ و زنده که توسط مجموعه ویکی‌پدیا ارائه شده و همچنان در حال گسترش است [13].

ج) گراف‌ها BioKG[1] و WikiKG2[2]: این دو گراف دانش زیرمجموعه تمام مجموعه داده‌هایی هستند که در فریم‌ورک OGB ارائه شده‌اند [14]. برای اینکه یادگیری رابطه‌ای روی داده‌های زیست‌پزشکی، استانداردتر و قابل تکرارتر شود، یک گراف دانش بیولوژیکی جدید توسط دانشگاه استنفورد ارائه شده است که مجموعه‌ای از داده‌های رابطه‌ای انتخاب شده از پایگاه‌های بیولوژیکی باز را در قالبی یکپارچه با شناسه‌های مشترک و مرتبط به هم ارائه می‌دهد. مجموعه داده ogbl-biokg یک گراف دانش است که آقای لسکووک به همراه همکاران، با استفاده از داده‌های تعداد زیادی از مخازن داده‌های زیست پزشکی ایجاد کرده‌اند. این گراف، شامل 5 نوع موجودیت است: بیماری ها (10687 گره)، پروتئین‌ها (17499)، داروها (10533 گره)، عوارض جانبی (9969 گره)، و عملکردهای پروتئین (45085 گره). 51 نوع رابطه مستقیم وجود دارد که دو نوع موجودیت را به هم متصل می‌کند، از جمله 39 نوع تداخل دارو-دارو، 8 نوع تعامل پروتئین-پروتئین، و همچنین دارو-پروتئین، اثر جانبی دارو، دارو- پروتئین و روابط عملکرد-عملکرد. همه رابطه‌ها به عنوان لبه‌های جهت دار مدل می‌شوند، که در میان آنها روابطی که انواع موجودات یکسان را به هم متصل می‌کنند (به عنوان مثال، پروتئین-پروتئین، دارو-دارو، عملکرد-عملکرد) همیشه متقارن هستند، به عنوان مثال، لبه‌ها دو جهته هستند. این مجموعه داده مربوط به هر دو تحقیقات زیست پزشکی و بنیادی ML است. در سمت اساسی ML، مجموعه داده، چالش‌هایی را در مدیریت یک KG ناقص با مشاهدات متناقض احتمالی ارائه می‌کند. این به این دلیل است که مجموعه داده ogbl-biokg شامل برهمکنش‌های ناهمگنی است که از مقیاس مولکولی (به عنوان مثال، برهمکنش‌های پروتئین-پروتئین در یک سلول) تا کل جمعیت (به عنوان مثال گزارش‌هایی از عوارض جانبی ناخواسته تجربه شده توسط بیماران در یک کشور خاص) را شامل می‌شود. علاوه بر این، سه‌گانه‌ها در KG از منابعی با سطوح مختلف اطمینان، از جمله بازخوانی‌های تجربی، حاشیه‌نویسی‌های تنظیم شده توسط انسان، و ابرداده‌های استخراج‌شده به طور خودکار به دست می‌آیند.

د) گراف دانش AceKG[3]: یک گراف دانش تخصصی در حوزه علمی و تحقیقاتی است که اطلاعات گسترده‌ای از مقالات، نویسندگان، مؤسسات و موضوعات تحقیقاتی را شامل می‌شود. این گراف که اولین بار در سال 2019 به عنوان یک پروژه تحقیقاتی معرفی شد، شامل بیش از 200 میلیون مقاله، 10 میلیون نویسنده و 20 هزار موسسه است و روابط نویسنده-مقاله و نویسنده-موسسه را پوشش می‌دهد. اهدف این پروژه فراهم کردن یک گراف دانش بزرگ و جامع برای تحلیل و کاوش داده‌های علمی است. توسعه‌دهندگان این گراف از تکنیک‌های پیشرفته پردازش زبان طبیعی و یادگیری ماشین استفاده کردند تا داده‌های علمی را از منابع مختلف جمع‌آوری و ساختاردهی کنند. بعضی از مهم‌ترین روابط در AceKG از قرار زیر هستند:

نویسنده-مقاله: این رابطه نویسندگان را به مقالاتی که نوشته‌اند متصل می‌کند و به تحلیل شبکه‌های همکاری علمی کمک می‌کند.

مقاله-کنفرانس/مجله: این رابطه نشان می‌دهد که یک مقاله در کدام کنفرانس یا مجله منتشر شده است و به بررسی تاثیرگذاری مقالات و کیفیت منابع انتشار کمک می‌کند.

نویسنده-موسسه: این رابطه، ارتباط بین نویسندگان و موسساتی

[1] BIOKG
[2] WIKIKG2
[3] Academic Community Exploration Knowledge Graph

که در آنها فعالیت می‌کنند را نمایش می‌دهد و به تحلیل پراکندگی جغرافیایی و سازمانی تحقیقات علمی می‌پردازد.

مقاله-موضوع تحقیقاتی: این رابطه مقالات را به موضوعات تحقیقاتی مرتبط متصل می‌کند و به شناسایی روندها و موضوعات داغ در تحقیقات علمی کمک می‌کند.

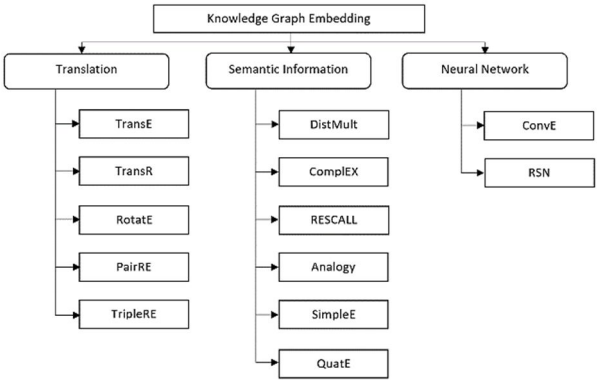
AceKG با این روابط متنوع به محققان و دانشجویان امکان می‌دهد تا به راحتی به اطلاعات علمی دسترسی پیدا کنند و شبکه‌های علمی را تحلیل کنند.

3. تکمیل گراف با روش‌های جاسازی گراف دانش

همان‌طور که گفته شد، گراف دانش، کامل نیست و داده‌های بسیار زیادی هستند که لبه‌های مرتبط با آنها در گراف دانش وجود ندارد [15].بعضی از این لبه‌ها بر اساس لبه‌های دیگر قابل حدس هستند. به عنوان مثال اگر لبه (حسن، به دنیا آمده، تهران) در گراف دانش وجود داشته باشد و برای حسن رابطه ملیت وجود نداشته باشد، آنگاه سه‌گانه (حسن، ملیت، ایرانی) را می‌توان حدس زد و با اضافه کردن لبه متناظر با این سه‌گانه گراف دانش را تکمیل کرد.

جاسازی یک بردار ویژگی به ابعاد d است. استفاده از جاسازی مناسب گره‌ها و روابط، به تکمیل گراف از طریق پیش‌بینی لبه کمک می‌کند. در مثال قبل با داشتن بردار جاسازی گره «حسن» و بردار جاسازی رابطه «ملیت» به برداری می‌رسیم که در میان بردارهای جاسازی تمام ملیت‌های مختلف، کوتاهترین فاصله را با بردار جاسازی «ایران» دارد. بنابراین مساله اصلی آن است که برای گره‌ها و روابط جاسازی‌هایی پیدا شود که در فضای برداری آنها چنین معادلاتی برقرار باشد. روش‌های KGE به دنبال یافتن (یا یاد گرفتن) چنین جاسازی‌هایی هستند.

به طور کلی، مدل‌های گراف دانش جاسازی (KGE) را می‌توان به سه دسته روش‌های مبتنی بر مدل‌های فاصله‌ای، روش‌های مبتنی بر مدل‌سازی دو خطی و روش‌های مبتنی بر شبکه‌های عصبی تقسیم کرد که در ادامه مقاله، هر دسته به صورت جدا مورد بررسی قرار می‌گیرد [16]. شکل (2) این تقسیم‌بندی را به تصویر کشیده است.



شکل (2): نمودار درختی الگوریتم‌های جاسازی بررسی شده در مقاله

4. TDM: روش‌های مبتنی بر مدل‌های فاصله‌ای[1]

این دسته از الگوریتم‌ها با الهام از قابلیت تغییرناپذیری نسبت به انتقال[2] که بار اول در روش word2vec ارائه شد، به وجود آمده‌اند. یک مدل انتقالی، متکی بر این اصل است که بردارهای جاسازی موجودیت‌ها پس از اعمال یک انتقال رابطه‌ای مناسب در فضای هندسی که در آن تعریف شده‌اند، به یکدیگر نزدیک خواهند بود. به عبارتی دیگر، زمانی که یک حقیقت را داریم با اضافه کردن جاسازی Head به جاسازی رابطه، نتیجه حاصله باید جاسازی Tail باشد. الگوریتم‌های متنوعی در این دسته قرار می‌گیرند که در ادامه 5 مورد از آنها بررسی می‌شوند.

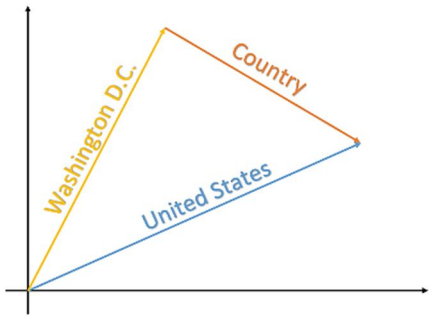
1.4. TransE

TransE تلاش می‌کند تا تمام گره‌ها را تبدیل به نقاطی در یک فضای d بعدی کرده و تمام رابطه‌ها را تبدیل به بردارهای انتقالی در این فضای حالت کند. به نحوی که اگر یک سرآیند h با رابطه r (که یک بردار انتقال است) جمع شود به نقطه‌ای برود که کمترین فاصله را با گره تالی t و همچنین بیشترین فاصله را با گره‌های دیگر داشته باشد [17]. سه‌گانه به صورت (h, r, t) نشان داده می‌شود. پس TransE به دنبال معادله $h + r \approx t$ است و لذا می‌توان تابع امتیاز آن را در رابطه (1) تعریف کرد.

$$F_r(h.t) = -||h + r - t|| \qquad (1)$$

[1] TDM - Translational Distance Models
[2] Translation invariance

هر چه حاصل جمع سرآیند و رابطه به تالی نزدیکتر باشد، مقدار این رابطه، بیشتر بوده و به سمت صفر میل خواهد کرد. شکل (3)، نمایانگر یک مثال از TransE است [18]. در این مثال گره «واشنگتن دی سی» با رابطه «کشور» به گره «ایالات متحده» می‌رسد. TransE سعی می‌کند تا تمام گره‌ها و لبه‌ها را طوری آموزش داده و تبدیل به بردار کند که رابطه «ملیت»، تمام افراد را به ملیت حقیقی آنها در گراف دانش متصل کند.



شکل (3): نمود بصری از یک رابطه و دو موجودیت در TransE

در زمان یادگیری، تعدادی لبه منفی را با تغییر در سرآیند و یا تالی، به عنوان لبه‌های آموزشی ایجاد می‌شوند. لبه منفی یا نمونه برداری منفی به لبه‌هایی گفته می‌شود که در حالت عادی در گراف دانش وجود ندارند ولی از آنجایی که در فرآیند آموزش نیاز به داده غلط یا منفی داریم، این لبه‌ها را ایجاد می‌کنیم. نحوه ایجاد آنها نیز همان‌طور که پیش‌تر گفته شد این است که به عنوان مثال یک لبه مثبت که در گراف دانش وجود دارد را انتخاب کرده و گره تالی آن را تغییر می‌دهیم. بدین ترتیب یک لبه منفی برای آموزش ایجاد می‌گردد. سپس تابع ضرر نوشته شده در رابطه (2) بازنویسی شده و با استفاده از یک بهینه‌ساز به سمت کمینه کردن آن حرکت می‌شود.

$$L = \sum_{(h.l.t)\in S} \sum_{(h'.l'.t')\in S'} [\gamma + d(h + l.t) - d(h' + l.t')]_+ \quad (2)$$

$$S'_{(h.l.t)} = \{(h'.l.t|h' \in E\} \cup \{(h.l.t')t' \in E\} \quad (3)$$

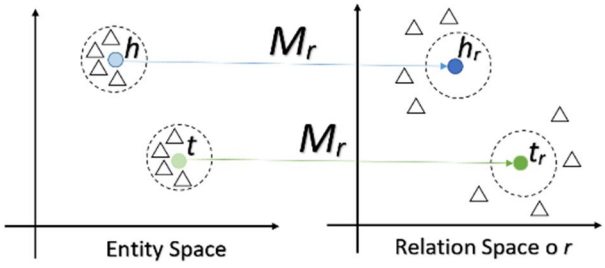
یکی از مشکلات روش TransE هنگام مواجهه با روابط غیر «یک به یک» می‌باشد که برای حل این مشکل روش‌های متعددی ارائه شده است. یکی از این روش‌ها TransH می‌باشد [19]. این روش برای حل مشکل اختصاص دادن بردارهای روابط مختلف هنگام مواجهه با این نوع روابط، برای هر رابطه r یک ابرصفحه تعریف می‌کند و با استفاده از بردار translation و بردار نرمال w مربوط به رابطه، برای بازنمایی آن استفاده می‌کند. در مرجع [20] اثبات شده است که روش TransH عملکرد بهتری نسبت به TransE دارد. ویژگی‌های مربوط به روش جاسازی TransE با توضیحات کامل در پیوست (الف) آمده است.

2.4. TransR

به دلیل بعضی مشکلات TransE، مدل TransR معرفی گردید [20]. ایده و تفاوت اصلی نسبت به TransE، آن است که به ازای هر رابطه، یک فضای حالت مجزا در نظر گرفته می‌شود. فرض کنید که گره سرآیند جاسازی h و گره تالی جاسازی t را دارد. در زمان خوانش سه‌گانه (h, r, t)، بردار سرآیند و تالی به یک فضای دیگر نگاشت می‌شوند؛ رابطه (4) که در آن h_r و t_r بردارهای جاسازی جدید گره سرآیند و گره تالی را در فضای جاسازی جدید نشان می‌دهد. در آن فضای جدید بردار انتقالی رابطه r میان آنها وجود دارد. روابط حاکم در این دو فضای جاسازی و انتقال بین آنها در رابطه (5) و شکل (4) دیده می‌شود [21] که در آن به ازای هر رابطه یک ماتریس M وجود دارد.

$$h_r = hM_r \quad t_r = tM_r \quad (4)$$

$$F_r(h.t) = ||h_r + r - t_r|| \quad (5)$$



شکل (4): نمودار بصری از نحوه تغییر فضا در الگوریتم TransR

بدین ترتیب می‌توان تابع خطا را در رابطه (6) بازنویسی کرد.

برای یادگیری جاسازی‌ها و ماتریس‌های M از روش SGD[1] استفاده شده است. ویژگی‌های این روش نیز به همراه توضیحات آن در پیوست (الف) آمده است.

$$L = \sum_{(h.r.t)\in S} \sum_{(h'.r.t')\in S'} max(0. f_r(h.t) + \gamma - f_r(h'.t')) \quad (6)$$

3.4. RotatE

تابع امتیاز در RotateE در رابطه (7) نشان داده شده است که عملگر o در آن «ضرب هادامارد[2]» است. این عملگر هر یک از درایه‌های یکسان دو ماتریس و یا بردار هم‌اندازه را در هم ضرب می‌کند. اگر ماتریس رابطه به درستی تنظیم شده باشد، می‌تواند باعث شود تا یک نقطه در فضا چرخش کرده و به یک نقطه دیگر برود. ویژگی‌های این روش نیز به همراه توضیحات آن در پیوست (الف) آمده است [21].

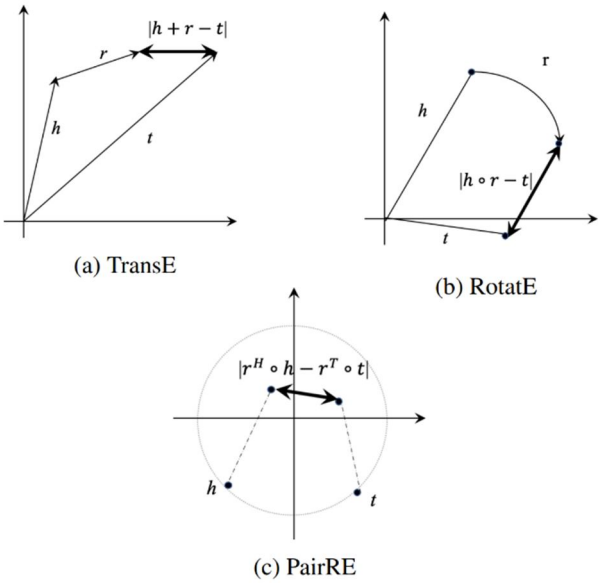
$$F_r(h.t) = -||h\, o\, r - t|| \quad (7)$$

4.4. PairRE

این روش هم مشابه روش RotateE از ضرب هادامارد در تابع امتیاز خود استفاده می‌کند [22]. تفاوت این روش با روش‌های دیگر در آن است که در این روش برای هر رابطه دو بردار تعریف می‌شود که یکی از آنها رابطه مربوط به h (سرآیند) و دیگری رابطه مربوط به t (تالی) می‌باشد. در نتیجه این طور تصور می‌شود که اگر رابطه r بین h و t وجود داشته باشد، باید نقطه‌ای که از ضرب h در بردار r^H به آن می‌رسیم از نقطه‌ای که از ضرب t در بردار r^T به آن می‌رسیم، کمترین فاصله را با هم داشته باشند که این نکته را در تابع امتیاز این روش در رابطه (8) می‌توان مشاهده نمود.

$$F_r(h.t) = -||h \circ r^H - t \circ r^T|| \quad (8)$$

در رابطه (8)، بردار r^H نشان‌دهنده رابطه سرآیند و بردار r^T نشان‌دهنده رابطه تالی می‌باشد. شکل (5)، روش‌های TransE، RotatE و PairRE را با هم مقایسه می‌کند. در این شکل اختلاف بین نقاط به دست آمده مشاهده می‌شود. جاسازی باید این اختلاف را برای تمام نمونه‌ها به حداقل برساند. از آنجایی که تابع امتیاز روش TransR بسیار شبیه به TansE می‌باشد و تنها تفاوت این است که بردارهای سرآیند و تالی به فضای جدیدی برده می‌شوند، تصویر آن در شکل (5) نیامده است و همانند تصویر TransE خواهد بود.



شکل (5): مقایسه سه الگورتیم TDM [23]

5.4. TripleRE

این الگوریتم بر مبنای الگوریتم PairRE بنا شده است و نسبت به آن برتری‌هایی دارد. تابع امتیاز این روش در رابطه (9) آمده است [24].

$$F_r(h.t) = -||h \circ r^H - t \circ r^T + r^m|| \quad (9)$$

همان‌طور که در رابطه بالا مشخص است، این روش تنها در اضافه شدن بردار سوم r^m با روش PairRE متفاوت است [23]. هرچند که نسخه‌های دیگری نیز برای این الگورتیم با توابع امتیاز دیگر معرفی شده‌اند. بردار سوم، می‌تواند سبب شود که فاصله نهایی بین دو نقطه‌ای که از h و t با رابطه‌های r^H و r^T به آنها رسیده‌ایم، کمتر شود. چرا که در نهایت هرچقدر این فاصله زیاد

[1] SGD-Stochastic Gradient Descent
[2] Hadamard Product

باشد می‌تواند با یک بردار دیگر جمع زده شده و تابع امتیاز مقدار بیشتری داشته باشد.

روش‌ها و الگوریتم‌های متنوع دیگری مانند TransH در این حوزه تعریف و پیاده‌سازی شده‌اند. اثبات شده است که این دسته به طور کامل توصیف‌گر[3] نیستند و عملکرد تجربی آنها به نسبت دیگر مدل‌ها کمتر می‌باشد [19].

5. BLM: روش‌های مبتنی بر مدل‌سازی دو خطی[4]

این دسته از روش‌ها از یک معادله خطی برای جایگذاری[5] ارتباط بین موجودیت‌ها از طریق یک رابطه استفاده می‌کنند. در واقع بازنمایی رابطه جایگذاری شده، یک ماتریس دو بعدی می‌باشد. این مدل‌ها در طی فرآیند جاسازی فقط از یک حقیقت[6] برای محاسبه بازنمایی استفاده می‌کنند و ارتباط با دیگر موجودیت‌ها و یا روابط را نادیده می‌گیرند. BLM های متفاوت از قیود مختلفی برای تنظیم‌سازی ماتریس R استفاده کرده‌اند تا بهتر با مجموعه داده‌های مختلف خود را انطباق دهند. در اینجا R همان ماتریس مربعی مربوط به جاسازی رابطه می‌باشد. در ادامه سه روش از این دسته مورد بررسی قرار می‌گیرند.

1.5. DistMult

تابع امتیازدهی مربوط به این روش در رابطه (10) ذکر شده است [25].

$$F_r(h.t) = < h.r.t > = \sum_i h_i.r_i.t_i \qquad (10)$$

در رابطه (10)، h، r و t به ترتیب نشان‌دهنده تالی، رابطه و سرآیند می‌باشند و «نقطه» بین آنها نیز ضرب داخلی می‌باشد. بر اساس رابطه (10)، در تابع امتیازدهی DistMult برای محاسبه امتیاز وجود یک رابطه بین دو گره تالی و سرآیند، از حاصلضرب مقادیر بردار ویژگی گره‌های تالی و سرآیند در بردار ویژگی رابطه استفاده می‌شود. عمل ضرب، قابلیت جابجایی دارد، لذا مسلما هر رابطه‌ای که بین سرآیند و تالی وجود داشته باشد، بین تالی و سرآیند هم وجود خواهد داشت. ویژگی‌های این جاسازی در پیوست (ب) مورد بررسی قرار گرفته است.

2.5. روش CP

روش Canonical Tensor Decomposition (CP) [26] همانند روش DistMult یک مدل تجزیه تنسور می‌باشد که چند تفاوت دارد. روش CP به طور کلی یک روش ریاضیاتی برای تجزیه هر گونه تنسور می‌باشد که از آن به طرق مختلف در مساله تکمیل گراف دانش استفاده شده. در مقاله مذکور برای بهبود عملکرد آن در گراف دانش نسبت به مقالات پیشین، ابتدا یک تنظیم‌ساز جدید بر پایه p-نرم‌های اتمی تنسوری ارائه شده است. سپس یک بازنمایی از مساله ارائه می‌گردد که آن را نسبت به انتخاب دلخواه روابط تک جهته یا دو طرفه مقاوم می‌سازد. ترکیب این دو، روش ارائه شده در مقاله را نتیجه می‌دهد. یکی از تفاوت‌های این روش با DistMult این است که CP یک مدل تجزیه تنسور عمومی می‌باشد در حالی که DistMult منحصرا برای عمل تکمیل گراف دانش طراحی شده است. به عبارتی روش CP به طور کلی توانایی تجزیه هر تنسوری را دارد ولی منحصرا برای تجزیه تنسورهای گراف دانش طراحی شده است که این باعث می‌شود که DistMult بتواند از ساختار خاص مربوط به گراف دانش استفاده کند. تفاوت دیگر دو روش این است که روش CP از نظر محاسباتی پر هزینه‌تر از DistMult می‌باشد. این امر بدین دلیل است که CP باید تنسور بزرگتری را تجزیه کند. اما از طرفی این اتفاق می‌تواند باعث گویاتر شدن این روش نسبت به DistMult شود و در نتیجه برخی از اوقات عملکرد بهتری نسبت به DistMult پیدا می‌کند.

3.5. ComplEx

در این روش، مقادیر بردارها می‌توانند عدد مختلط باشند. همین مساله قدرت جاسازی را افزایش می‌دهد [27]. لذا تنها تفاوت بین تابع امتیاز ComplEx با DistMult این است که اعداد شرکت کننده در عملیات ضرب، از نوع مختلط هستند و در نهایت تنها

[3] expressive
[4] BLM-Bilinear Models
[5] Embed
[6] Fact

قسمت حقیقی عدد مختلط به دست آمده به عنوان مقدار امتیاز تشابه برای آن سه‌گانه در نظر گرفته می‌شود. تابع امتیاز این روش در رابطه (11) آمده است.

$$f_r(h.t) =< h.r.t >= Re(\sum_i h_i.r_i.t_i) \qquad (11)$$

بدیهی است اگر مقدار موهومی در تمام بردارها را صفر در نظر بگیریم، همان روش DistMult را خواهیم داشت. ویژگی‌های مربوط به این روش در پیوست (ج) مورد بررسی قرار گرفته است.

4.5. QuatE

از دیگر روش‌های یادگیری جاسازی موجودیت‌ها در گراف دانش که سعی بر به دست آوردن وابستگی‌های ارتباطی بلند مدت هستند، می‌توان به روش QuatE اشاره کرد [28]. این روش بر پایه چهارگانها[7] می‌باشد که نوعی عدد هستند که از آنها می‌توان برای بازنمایی دوران در فضا استفاده کرد. در این روش هر موجودیت در گراف دانش با یک چهارگان نمایش داده می‌شود. سپس ارتباط بین موجودیت‌ها با ضرب چهارگانها بازنمایی می‌گردند. روش QuatE چندین مزیت نسبت به دیگر روش‌های یادگیری جاسازی در گراف دانش دارد. مزیت اول این است که یافتن وابستگی‌های بلند مدت را دارد. دلیل آن این است که از ضرب چهارگانها برای بازنمایی دوران در فضا می‌توان استفاده نمود و دوران‌ها را می‌توان به یکدیگر متصل کرد تا دنباله‌های طولانی از روابط را بازنمایی نمود. دوم اینکه روش QuatE گویاتر[8] از دیگر روش‌ها می‌باشد. دلیل آن نیز این است چهارگانها دارای درجه آزادی بیشتری نسبت به اعداد حقیقی و یا اعداد مرکب هستند. این باعث می‌شود که این روش بتواند گستره وسیع‌تری از روابط بین موجودیت‌ها را بازنمایی کند. سوم اینکه QuatE بازدهی بهتری نسبت به دیگر روش‌ها دارد. دلیل آن نیز این است که ضرب چهارگانها از نظر محاسباتی یک عمل بهینه می‌باشد.

[7] Quaternions
[8] Expressive

روش‌های مشابه دیگری همچون RESCALL، Analogy[9] [29] و SimpleE [30] را نیز می‌توان در این دسته، بررسی کرد. باید دقت شود که هر یک از روش‌های بالا و روش‌های دیگر، مزایا و معایب خود را دارد اما این مزایا و معایب کاملا بستگی به گراف دانش دارد و باید بررسی کرد که به طور مثال آیا ویژگی تقارنی در گراف دانش وجود دارد یا خیر، اگر وجود نداشته باشد، مدل TransE، مدلی مناسب جهت جاسازی آن گراف دانش خواهد بود [31].

6. NNM: روش‌های مبتنی بر شبکه‌های عصبی

این دسته از روش‌ها خود دارای دسته‌های مختلفی مانند روش‌های یادگیری عمیق[10] (DNN)، روش‌های شبکه‌های عصبی کانولوشنی[11] (CNN) و روش‌های شبکه‌های بازگشتی[12] (RNN) می‌باشند. هدف مدل‌های عصبی خروجی دادن احتمال سه‌گانه‌ها براساس شبکه‌هایی است که جاسازی‌های روابط و موجودیت‌ها را به عنوان ورودی دریافت می‌کنند. در روش‌های یادگیری عمیق از شبکه‌های عصبی عمیق برای یادگیری الگوها از گراف دانش که همان ورودی می‌باشد، استفاده می‌کنند. با وجود پردازش سنگین در قسمت آموزش، استفاده از شبکه‌های عمیق در جاسازی گراف دانش عملکرد خوبی از خود نشان داده است. روش MLP ارائه شده در مرجع [32]، یکی از جدیدترین تلاش‌ها در استفاده از مدل‌های عصبی می‌باشد. هر دو روش از تعداد زیادی پارامتر برای ترکیب جاسازی‌های موجودیت‌ها و روابط استفاده می‌کنند. روش ConvE از شبکه عصبی کانولوشنی برای افزایش تعامل بین ابعاد مختلف جاسازی‌ها بهره می‌برد [33]. یکی از روش‌های بر پایه شبکه‌های بازگشتی، RSN می‌باشد که در بخش 1.6 مورد بررسی قرار می‌گیرد.

با توجه به موفقیت شبکه‌های عصبی عمیق در زمینه‌های مختلف، بعضی از این مدل‌های شبکه‌های عصبی به عنوان تابع امتیاز مورد بررسی قرار گرفته‌اند. با وجود اینکه شبکه‌های عصبی توانایی

[9] Analogy
[10] Deep Neural Networks methods
[11] Convolutional Neural Networks methods
[12] Recurrent Neural Networks methods

توصیفی بالایی دارند، ولی مدل‌های شبکه‌های عصبی در گراف‌های دانش به دلیل عدم تنظیم‌سازی مناسب، عملکرد خوبی ندارند. در بین توابع امتیاز موجود، توابع بر پایه بی‌ال‌ام، بر اساس نتایج عملی و تضمین نظری برای توصیف‌گر بودن، قوی‌ترین هستند.

1.6. روش RSN

این روش یکی از روش‌های شبکه‌های بازگشتی می‌باشد. مزیت این دسته از روش‌ها به خاطر سپردن دنباله‌ای از حقیقت‌ها می‌باشد و مانند مابقی روش‌ها تنها به یک پدیده دقت نمی‌کنند [34]. این روش سعی بر حل مشکل روش‌های فعلی یادگیری جاسازی گراف‌های دانش دارد که به طور معمول بر یادگیری از سه‌گانه‌ها تمرکز دارند و باعث می‌شود که نتوانند وابستگی‌های دراز مدت بین موجودیت‌ها را به دست آورند. دلیل آن این است که سه‌گانه‌ها تنها یک قدم از مسیر رابطه‌ای را بازنمایی می‌کنند در حالی که معنی یک موجودیت می‌تواند متاثر از روابطش با موجودیت‌هایی باشد که چند قدم فاصله دارند. برای حل این مشکل، نویسندگان روشی به نام شبکه‌های بازگشتی پرشی (RSNs) ارائه کردند. این شبکه‌ها نوعی شبکه عصبی بازگشتی هستند که قابلیت یادگیری از وابستگی‌های ارتباطی بلند مدت را دارند. آنها ابتدا با ساخت دنباله‌ای مسیرهای ارتباطی که موجودیت‌ها را در گراف دانش به یکدیگر متصل می‌کنند، کار خود را شروع می‌نمایند. سپس RSN با توجه و در نظر گرفتن موجودیت‌های قبلی و ارتباط بین آنها، پیش‌بینی موجودیت بعدی در دنباله را یاد می‌گیرد.

7. مقایسه روش‌ها

مدل‌های انتخاب شده برای این مطالعه از دسته‌های مختلفی هستند. برای مقایسه آنها، چند مجموعه داده را انتخاب کرده و تمامی روش‌ها را بر روی آنها، اجرا می‌کنیم. سپس جاسازی حاصله از هر روش را با معیارهایی سنجیده و نتایج را با هم مقایسه کرده و در نهایت تفسیری از نتایج به دست آمده ارائه می‌دهیم.

1.7. معیارهای ارزیابی

روش‌های مختلف جاسازی گراف‌های دانش، اکثرا با استفاده از تسک‌های پیش‌بینی لبه ارزیابی می‌شوند. معیارهای بسیاری برای ارزیابی جاسازی گراف دانش موجود است؛ با این حال، به طور معمول معیارهای میانگین رتبه (MR)، میانگین رتبه تنظیم شده (AMR)، میانگین نرخ معکوس (MMR) و موفقیت در (Hits@K) در بررسی روش‌ها، استفاده می‌شوند. در ادامه دو معیار MMR و Hits@K را بررسی می‌کنیم.

1.1.7. معیار MRR

معیار میانگین رتبه‌بندی متقابل[13] یک معیار آماری برای ارزیابی هر پردازشی است که لیستی از جواب‌های محتمل، ناشی از نمونه‌ای از کوئری‌ها را به ما خروجی می‌دهد که این لیست بر اساس احتمال درست بودن مرتب شده است. رتبه دو طرفه یک جواب کوئری، وارون ضربی اولین پاسخ صحیح می‌باشد. برای رتبه اول $\frac{1}{2}$ برای رتبه دوم $\frac{1}{3}$ برای رتبه سوم و الی آخر. میانگین رتبه‌بندی متقابل یا دو طرفه، میانگین رتبه‌های دو طرفه نتایج مربوط به نمونه‌ای از کوئری‌های Q می‌باشد که در رابطه (12) قابل مشاهده می‌باشد.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \qquad (12)$$

که در این رابطه $rank_i$ مربوط به رتبه اولین سند[14] مرتبط برای کوئری i ام می‌باشد.

2.1.7. معیار H@10

معیار Hits@K و یا به طور مختصر H@K، یک معیار ارزیابی برای احتمال پیدا کردن پیش‌بینی صحیح در K پیش‌بینی اول مدل می‌باشد. به طور معمول مقدار K برابر 10 قرار می‌گیرد. Hits@K نشان‌دهنده دقت یک مدل جایگذاری در پیش‌بینی صحیح رابطه

[13] Mean Reciprocal Rank
[14] Document

بین دو سه‌گانه است. در رابطه (13) رابطه مربوط به این معیار قابل مشاهده می‌باشد.

$$Hits@K = \frac{|\{q \in Q : q < k\}|}{|Q|} \in [0.1] \qquad (13)$$

مقادیر بیشتر Hits@K نشان‌دهنده قدرت پیش‌بینی‌کنندگی بهتر است.

2.7. آزمایشات

برخی از مهم‌ترین روش‌هایی که در این مقاله بررسی شده‌اند، بر روی مجموعه داده‌ها و سخت‌افزارهای مختلف ارزیابی شده‌اند. در ادامه نتایج این آزمایشات شرح داده می‌شوند.

1.2.7. مجموعه داده‌ها

OGB، مجموعه‌ای از مجموعه داده‌های بنچمارک با مقیاس بزرگ، واقعی و گسترده می‌باشد که برای استفاده از یادگیری ماشین بر روی گراف‌ها بکار گرفته می‌شوند. مجموعه داده‌های OGB به صورت خودکار بارگیری، پردازش و توسط OGB Data Loader تقسیم‌بندی می‌گردند. دو مجموعه داده BioKG و WIKIKG2 در این چارچوب موجودند که برای ارزیابی بهتر، روش‌های مختلف بر روی این دو مجموعه داده اجرا شدند تا بتوان نتایج را باهم مقایسه کرد. جدول (1) مشخصات مجموعه داده‌ها را نشان می‌دهد.

جدول (1): مشخصات مجموعه داده‌ها

مجموعه داده	تعداد موجودیت‌ها	تعداد انواع روابط	تعداد روابط
WN18	40,943	18	141,442
FB15k	14,951	1,345	484,142
WN18RR	40,943	11	86,835
FB15k237	14,541	237	272,115
YAGO3-10	123,188	37	1,079,040
BioKG	93,773	5	5,088,434
WIKIKG2	2,500,604	535	17,13,181

همچنین در جداول (3) تا (5) [35]، نیز روش‌های مختلف از جمله روش‌های بحث شده از نظر معیارهای MRR، H@1 و H@10 در مجموعه داده‌های WN18، FB15K و WN18RR مورد مقایسه قرار گرفته‌اند. نتایج بهتر پررنگ شده‌اند.

2.2.7. آزمایش اول: بررسی اجرای الگوریتم‌ها بر روی یک مجموعه داده یکسان

در این آزمایش، نتایج روش‌های مختلف بر روی مجموعه داده BioKG بررسی می‌شوند. نتایج در جدول (2) قابل مشاهده هستند [36]. مجموعه داده BioKG شامل 5 نوع رابطه متفاوت مانند (پروتئین-پروتئین، دارو-دارو، غیره) است. همان‌طور که مشخص است در مجموعه داده، داده‌هایی موجود هستند که متقارن هستند. به عنوان مثال پروتئین «الف» بر روی پروتئین «ب» اثر دارد و بالعکس. از آنجایی که ماهیت تابع امتیازدهی و روش TransE نمی‌تواند ویژگی «تقارن» را مدل کند، بالطبع داده‌های آموزشی هستند که این روش به هیچ وجه، قادر به یادگیری آنها نیست و به همین سبب کمترین امتیاز در بین تمام روش‌ها به TransE تعلق دارد. RotatE این نقص را نداشته و به همین دلیل نسبت به TransE عملکر بهتری دارد.روش‌های ComplEx و DistMult در رده‌های بعدی قرار دارند که متعلق به دسته BLM هستند. از آنجایی که ComplEx از سیستم اعداد مختلط استفاده می‌کند، برتری نسبی، نسبت به روش DistMult دارد. به طور کلی، عملکرد روش‌های BLM نسبت به روش‌های TDM بهتر است اما روش PairER به دلیل تابع امتیازدهی بهبود یافته و دوطرفه خود، نتایج بهتری کسب کرده است.

روش‌های AutoBLM [37] و AutoSF [35]، روش‌های مبتنی بر جستجو هستند. از آنجایی که داده‌ها و ارتباط بین داده‌ها در مجموعه داده‌های مختلف با هم متفاوت است، لذا می‌توان انواع روش‌های مختلف را به صورت خودکار بر روی مجموعه داده بررسی کرده و بهترین روش را به عنوان پاسخ ارائه داد و سپس جاسازی‌ها را با این پاسخ تولید کرد. به همین دلیل امتیازات بهتری نسبت به بقیه روش‌ها در این روش‌ها به دست می‌آید.

روش TripleRE، مبتنی بر روش PairRE است که بهبودهایی بر روی تابع امتیازدهی آن انجام شده است. به دلیل تابع امتیاز

پیچیده در این روش، پارامترهای آموزشی این روش، از بقیه روش‌ها بیشتر است و فرآیند آموزش کندتر جلو می‌رود.

جدول (2): نتایج روش‌های مختلف بر روی مجموعه داده BIOKG

روش	MRR آزمون	MRR اعتبارسنجی	تعداد پارامترها
AutoBLM-KGBench	0/8536	0/8548	192/047/104
ComplEx-RP	0/8492	0/8497	187/750/000
TripleRE	0/8348	0/8360	496/630/002
AutoSF	0/8309	0/8317	93/824/000
PairRE	0/8164	0/8172	187/750/000
ComplEx	0/8095	0/8105	187/648/000
DistMult	0/8043	0/8055	187/648/000
RotatE	0/7989	0/7997	187/597/000
TransE	0/7452	0/7456	187/648/000

3.2.7. آزمایش دوم: بررسی اجرای الگوریتم‌ها بر روی سه مجموعه داده مختلف

روش‌های مختلف بر روی سه مجموعه داده WN18RR، FB15K و WN18 اجرا شده و نتایج در جداول (3) تا (5) ارائه شده است. تفسیر نتایج مختلف الگوریتم‌های امبدینگ در مجموعه داده‌های مختلف نیاز به درک تفاوت‌های ذاتی بین این مجموعه داده‌ها و خصوصیات هر یک از الگوریتم‌ها دارد. در ادامه به بررسی این تفاوت‌ها و تاثیرات آنها بر نتایج می‌پردازیم.

WN18: این مجموعه داده شامل سه‌گانه‌هایی از یک گراف معنایی بزرگتر است. ساختار آن شامل تعداد زیادی روابط مختلف است که برای الگوریتم‌های مبتنی بر شبکه‌های عصبی و روش‌های ترجمه‌ای مناسب است. این تنوع باعث می‌شود مدل‌های مختلف عملکرد خوبی داشته باشند، به ویژه مدل‌هایی که می‌توانند پیچیدگی‌های شبکه را در نظر بگیرند.

FB15K: این مجموعه داده شامل داده‌های روابط در پایگاه داده بزرگ Freebase است و شامل تعداد زیادی از موجودیت‌ها و روابط پیچیده است. این پیچیدگی و تنوع در روابط، به الگوریتم‌هایی که می‌توانند بهتر از روابط پیچیده استفاده کنند، مانند مدل‌های خودکار و مدل‌های مبتنی بر ترجمه، کمک می‌کند تا عملکرد بهتری داشته باشند.

جدول (3): نتایج روش‌های مختلف بر روی مجموعه داده WN18

دسته	مدل	MRR	H@1	H@10
TDM	TransH	0/521	-	94/5
	RotateE	0/949	94/4	95/9
NNM	ConvE	0/942	93/5	95/5
	RSN	0/94	92/2	95/3
BLM	DistMult	0/821	71/7	95/2
	SimplE/CP	0/95	94/5	95/9
	HolE/ComplEx	0/951	94/5	95/7
	Analogy	0/95	94/6	95/7
	QuatE	0/95	94/5	95/9
Auto	AutoBLM	**0/952**	**94/7**	**96/1**
	AutoBLM+	**0/952**	**94/7**	**96/1**

جدول (4): نتایج روش‌های مختلف بر روی مجموعه داده FB15K

دسته	مدل	MRR	H@1	H@10
TDM	TransH	0/452	-	76/6
	RotateE	0/797	74/6	88/4
	PairE	0/811	76/5	89/6
NNM	ConvE	0/745	67	87/3
BLM	DistMult	0/775	71/4	87/2
	SimplE/CP	0/826	79/4	90/1
	HolE/ComplEx	0/831	79/6	90/5
	Analogy	0/816	78	89/8
	QuatE	0/782	71/1	90
Auto	AutoBLM	0/853	82/1	91
	AutoBLM+	0/861	83/2	91/3

WN18RR: این مجموعه داده نسخه کاهش یافته و بهبود یافته WN18 است که روابط معکوس تکراری را حذف کرده است تا از یادگیری بر مبنای میانبرهای ساده جلوگیری کند. این ویژگی

باعث می‌شود که این مجموعه داده به مراتب چالش‌برانگیزتر باشد و نیاز به مدل‌هایی داشته باشد که بتوانند روابط پیچیده‌تری را یاد بگیرند و تعمیم‌دهی بهتری داشته باشند.

جدول (5): نتایج روش‌های مختلف بر روی مجموعه داده WN18RR

دسته	مدل	MRR	H@1	H@10
TDM	TransH	0/186	-	45/1
	RotateE	0/476	42/8	57/1
NNM	ConvE	0/46	39	48
BLM	DistMult	0/443	40/4	50/7
	SimplE/CP	0/462	42/4	55/1
	HolE/ComplEx	0/471	43	55/1
	Analogy	0/467	42/9	55/4
	QuatE	0/488	43/8	58/2
Auto	AutoBLM	0/49	45/1	56/7
	AutoBLM+	0/492	45/2	56/7

مدل‌های مبتنی بر ترجمه (TDM): این مدل‌ها مانند TransH و RotateE در مجموعه داده‌هایی با روابط پیچیده و متنوع مانند FB15K و WN18 عملکرد خوبی دارند. اما در مجموعه داده WN18RR که چالش‌های بیشتری دارد، نتایج ضعیف‌تری دارد چون نمی‌توانند به همان اندازه روابط پیچیده را مدل‌سازی کنند.

مدل‌های شبکه عصبی (NNM): مدل‌هایی مانند ConvE برای یادگیری ویژگی‌های پیچیده و روابط غیرخطی طراحی شده‌اند. این مدل‌ها در مجموعه داده‌هایی که تنوع زیادی در روابط دارند مانند FB15K و WN18 می‌توانند عملکرد خوبی داشته باشند، اما همان‌طور که در نتایج جدول مشخص شده است، در مجموعه داده چالش‌برانگیز WN18RR که نیاز به تعمیم‌دهی بهتری دارد، کمتر موثر است.

مدل‌های باینری و لاجیک (BLM): مدل‌هایی مانند DistMult و ComplExقادر به مدل‌سازی روابط پیچیده هستند. این مدل‌ها در همه مجموعه داده‌ها عملکرد خوبی دارند به ویژه در مجموعه داده‌هایی که دارای ساختار پیچیده و چندلایه هستند.

مدل‌های خودکار (Auto): مدل‌های AutoBLM و AutoBLM+ به دلیل استفاده از روش‌های خودکار و پیشرفته، توانایی بهینه‌سازی پارامترها و تعمیم‌دهی بهتری دارند. این مدل‌ها در تمامی مجموعه داده‌ها عملکرد برتری داشته‌اند، زیرا قادر به انطباق با انواع مختلف داده‌ها و روابط پیچیده هستند.

تفاوت در عملکرد الگوریتم‌های امبدینگ در مجموعه داده‌های مختلف عمدتا به دلیل تفاوت در پیچیدگی و نوع روابط در هر مجموعه داده است. مجموعه داده‌های پیچیده‌تر و چالش‌برانگیزتر نیاز به الگوریتم‌های قوی‌تری دارند که بتوانند روابط پیچیده را مدل‌سازی و تعمیم دهند. مدل‌های خودکار به دلیل انعطاف‌پذیری و توانایی بهینه‌سازی پارامترها، در تمامی موارد عملکرد برتری داشته‌اند.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] A. Singhal, “Introducing the Knowledge Graph: Things, not strings,” Official Google Blog, 2012, [Online]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not/

[2] X. Wang et al., “Learning intents behind interactions with knowledge graph for recommendation,” in Proc. Assoc. Comput. Mach. (ACM), New York, NY, USA, 2021, doi: 10.1145/3442381.3450133.

[3] X. Chen, S. Jia, and Y. Xiang, “A review: Knowledge reasoning over knowledge graph,” Expert Syst. Appl., vol. 141, 2020, doi: 10.1016/j.eswa.2019.112948.

[4] X. Li et al., “Entity-relation extraction as multi-turn question answering,” in Proc. 57th Annu.

Meeting Assoc. Comput. Linguistics (ACL), Florence, Italy, 2019, doi: 10.18653/v1/P19-1129.

[5] F. Zhang, N.J. Yuan, D. Lian, X. Xie, and W.Y. Ma, "Collaborative knowledge base embedding for recommender systems," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA, 2016, doi: 10.1145/2939672.2939673.

[6] T. Shen et al., "Exploiting structured knowledge in text via graph-guided representation learning," in Proc. 2020 Conf. Empirical Methods Nat. Lang. Process. (EMNLP), Abu Dhabi, UAE, 2020, doi: 10.48550/arXiv.2004.14224.

[7] X. Zou, "A survey on application of knowledge graph," in Proc. 4th Int. Conf. Control Eng. Artif. Intell., Singapore, 2020, doi: 10.1088/1742-6596/1487/1/012016.

[8] M. Chavoshi and S.M. Babamir, "Extracting Rules from Specifications and Their Modeling using Colored Fuzzy Petri-Nets," Soft Comput. J., vol. 11, no. 1, pp. 22-47, 2022, doi: 10.22052/scj.2022.246562.1078 [In Persian].

[9] A. Keypour, "Link Prediction in Social Networks through classifiers combination," Soft Comput. J., vol. 4, no. 2, pp. 2-17, 2016, dor: 20.1001.1.23223707.1394.4.2.54.4 [In Persian].

[10] S.M. Vahidipour and R. Karami, "Link prediction framework based on graph neural networks using subgraphs," Soft Comput. J., vol. 14, no. 1, pp. 18-35, 2025, doi: 10.22052/scj.2024.253458.1179 [In Persian].

[11] X. Wang and L. Fu, "AceMap: Knowledge discovery through academic graph," arXiv preprint arXiv:2403.02576, 2024, doi: 10.48550/arXiv.2403.02576.

[12] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P.S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 2, 2022, doi: 10.1109/TNNLS.2021.3070843.

[13] Welcome to Wikidata, Jan. 22, 2023. [Online]. Available: https://www.wikidata.org/wiki/Wikidata:Main_Page.

[14] W. Hu, M. Fey, M. Zitnik, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS), Vancouver, BC, Canada, 2020, doi: 10.48550/arXiv.2005.00687.

[15] Z. Chen et al., "Knowledge graph completion: A review," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3030076.

[16] Y. Dai, S. Wan, N.N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," Electronics, vol. 9, no. 5, 2020, doi: 10.3390/electronics9050750.

[17] A. Bordes, N. Usunier, and A. Garcia-Duran, "Translating embeddings for modeling multi-relational data," in Proc. Neural Inf. Process. Syst. (NIPS), 2013, pp. 2787–2795.

[18] F. Bianchi, G. Rossiello, L. Costabello, M. Palmonari, and P. Minervini, "Knowledge graph embeddings and explainable AI," arXiv preprint arXiv:2004.14843, 2020, doi: 10.48550/arXiv.2004.14843.

[19] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in Proc. 28th AAAI Conf. Artif. Intell., 2014, doi: 10.1609/aaai.v28i1.8870.

[20] X. Wang, W. Li, and H Tong, "A comparative study of TransE and TransH algorithms in spatial address representation learning," Biomed. J. Sci. Tech. Res., vol. 26, no. 5, 2020, doi: 10.26717/BJSTR.2020.26.004427.

[21] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proc. 29th AAAI Conf. Artif. Intell., 2015, doi: 10.1609/aaai.v29i1.9491.

[22] Z. Sun, Z.H. Deng, J.Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in Proc. Int. Conf. Learn. Representations (ICLR), 2019, doi: 10.48550/arXiv.1902.10197.

[23] L. Chao, J. He, T. Wang, and W. Chu, "PairRE:

Knowledge graph embeddings via paired relation vectors," arXiv preprint arXiv:2011.03798, 2021, doi: 10.48550/arXiv.2011.03798.

[24] L. Yu et al., "TripleRE: Knowledge graph embeddings via triple relation vectors," arXiv preprint arXiv:2209.08271, 2022, doi: 10.48550/arXiv.2209.08271.

[25] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in Proc. Int. Conf. Learn. Representations (ICLR), San Diego, CA, USA, 2015, doi: 10.48550/arXiv.1412.6575.

[26] F. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," J. Math. Phys., vol. 6, pp. 164-189, 1927.

[27] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in Proc. Int. Conf. Mach. Learn. (ICML), New York, NY, USA, 2016, doi: 10.48550/arXiv.1606.06357.

[28] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," in Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS), Vancouver, BC, Canada, 2019, doi: 10.48550/arXiv.1904.10281.

[29] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in Proc. 34th Int. Conf. Mach. Learn. (ICML), 2017, doi: 10.48550/arXiv.1705.02426.

[30] S.M. Kazemi and D. Poole, "SimplE embedding for link prediction in knowledge," in Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS), Montreal, QC, Canada, 2018, doi: 10.48550/arXiv.1802.04868.

[31] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," ACM Trans. Knowl. Discovery Data, vol. 15, no. 2, 2021, doi: 10.1145/3424672.

[32] Q. Xu et al., "MlpE: Knowledge graph embedding with multilayer perceptron networks," in IEEE SmartWorld Ubiquit. Intell. Comput. Scalable Comput. Commun. Digit. Twin Priv. Comput. Metaverse Auton. Trusted Veh. (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp /Meta), Haikou, China, 2022, pp. 856-863, doi: 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00130..

[33] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolution 2D knowledge graph embeddings," in Proc. 32nd AAAI Conf. Artif. Intell., 2018, doi: 10.48550/arXiv.1707.01476.

[34] Y. Zhang, Q. Yao, and L. Chen, "Interstellar: Searching recurrent architecture for knowledge graph embedding," Adv. Neural Inf. Process. Syst., vol. 33, 2020, doi: 10.48550/arXiv.1911.07132.

[35] Y. Zhang, W. Dai, Q. Yao, and L. Chen, "AutoSF: Searching scoring functions for knowledge graph embedding," in Proc. Int. Conf. Data Eng. (ICDE), 2020, doi: 10.48550/arXiv.1904.11682.

[36] "Leaderboards for link property prediction," 2024, [Online]. Available: https://ogb.stanford.edu/docs/leader_linkprop/#ogbl-biokg

[37] Y. Zhang, Q. Yao, and J.T.Y. Kwok, "Bilinear scoring function search for knowledge graph learning," arXiv preprint arXiv:2107.00184, 2021, doi: 10.48550/arXiv.2107.00184.

[38] F. Hutter, L. Kotthoff, and J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges. Springer Nature, 2019.

[39] L. Guo, Z. Sun, and W. Hu, "Learning to exploit long-term relational dependencies in knowledge graphs," in Proc. 36th Int. Conf. Mach. Learn. (ICML), 2019.

[40] X.L. Dong et al., "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2014, doi: 10.48550/arXiv.1905.04914.

[41] M. Nickel, V. Tresp, and H.P. Kriegel, "A three-way model for collective learning on multi-relational data," in Proc. 28th Int. Conf. Mach. Learn. (ICML), 2011, doi:

10.1145/2623330.2623623.
[42] T. Shen, F. Zhang, and J. Cheng, “A comprehensive overview of knowledge graph completion,” Knowl.-Based Syst., vol. 255, 2022, doi: 10.1016/j.knosys.2022.109597.
[43] R. Socher, D. Chen, C. Manning, and A. Ng, “Reasoning with neural tensor networks for knowledge base completion,” Adv. Neural Inf. Process. Syst., vol. 26, 2013.
[44] M. Wang, L. Qiu, and X. Wang, “A survey on knowledge graph embeddings for link prediction,” Symmetry, vol. 13, no. 3, 2021, doi: 10.3390/sym13030485.
[45] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 12, 2017, doi: 10.1109/TKDE.2017.2754499.

پیوست (الف)

ویژگی‌های روش TransE:

TransE با فرض اینکه جاسازی در فضای دوبعدی انجام شده باشد و نقطه الف (1و1) نقطه ب (3و3) و رابطه ج (2و2) امبد شده باشد، ویژگی‌های زیر را خواهد داشت:

1. تقارن: اگر فرض کنیم که نقطه «الف» با نقطه «ب» رابطه «ج» را دارد، مسلما اگر بخواهیم از نقطه «ب» نیز با بردار «ج» حرکت کنیم بـه نقطـه‌ای بسـیار دورتـر از «الـف» خواهیم رسید.

صحیح است $(1,1) + (2,2) = (3,3)$

$(5,5) = (3,3) + (2,2)$ بـا مختصـات «الـف» منطبـق نیست، لذا این جاسازی، **تقارن** را پشتیبانی **نمی‌کند**. به عنوان مثال فرض شود رابطه «ج»، رابطه «همکاری» باشد، TransE نمی‌تواند جاسازی‌ها را طوری بسازد که کارمند اول با کارمند دوم رابطه همکاری داشته باشد و همزمان، کارمند دوم هم با کارمند اول، رابطه همکاری داشته باشد.

2. همان‌طور که پیش از این گفته شد، هیچ وقت با جمـع بردار انتقال رابطه و تالی، به نقطه سرآیند نمی‌رسیم، لذا این جاسازی **پادتقارن** را پشتیبانی **می‌کند**.

3. آینه‌ای: اگر رابطه «ج» بین «الف» و «ب» وجود دارد، آیا می‌توانیم یک رابطه «د» را بـین «ب» و «ج» تعریـف و مدل کرد؟ همان‌طور که از مثال عددی بالا مشخص است می‌توانیم بردار انتقال «د» را مساوی $(-2,-2)$ تعریف کنیم که با این تعریف می‌توانیم رابطه زیر را بازنویسی کنیم:

$$(3,3) + (-2,-2) = (1,1)$$

لذا این جاسازی، **آینه‌ای** را پشتیبانی **می‌کند**.

4. برای بررسی **تراگذاری**، کافی است تصور کنیم که بـه ازای هر رابطه یک بردار انتقال داریم که ما را از نقطـه سرآیند به نقطه تالی می‌رساند، لذا اگر تعـدادی رابطـه پی‌درپی داشته باشیم در نهایت ما را بـه یـک نقطـه در فضای حالت می‌رساند که مسلما با یـک بـردار انتقـال می‌توانیم از نقطه سرآیند ابتدایی به طور مستقیم به این نقطه تالی برسیم، لذا این جاسازی **تراگذاری** را پشتیبانی **می‌کند**.

5. شرط پشتیبانی از ویژگی «یک به همه» این است که اگر «*A*» رابطه «*R*» را با «*B*» داشته باشد، بتواند همین رابطه با گره دیگری به نام «*C*» داشته باشد. فرض کنیم که جاسازی زیر وجود داشته باشد:

$$A(1,1) + R(2,2) = B(3,3)$$

حال اگر بخواهیم همین رابطه را با یک نقطه دیگر داشته باشیم خواهیم داشت:

$$A(1,1) + R(2,2) = C(3,3)$$

که در این صورت دو نقطه «*B*» و «*C*»، کاملا یکسان هستند و در عمل می‌بایست یک نقطه باشند که این در گراف دانش صحیح نیست، لذا این جاسازی **رابطه یک به چند** را پشتیبانی **نمی‌کند**.

ویژگی‌های روش TransR:

تقارن: می‌توان M را طوری تعیین کرد که دو نقطه «الف» و «ب» با ضرب در M، به یک نقطه یکسان در فضای جدید تبدیل شوند، در نتیجه در فضای جدید هر دو نقطه رابطه یکسان با یکدیگر دارند و در نهایت می‌تواند ویژگی **تقارن** را پشتیبانی **کند**.

1. ویژگی‌های پادتقارن و آینه‌ای به مانند روش TransE در این روش قابل پشتیبانی هستند.
2. یک به چند: همان‌طور که در بند اول گفته شد، می‌توان M را طوری تنظیم نمود که دو یا بیشتر نقاط در فضای مساله با ضرب در M به فضایی بروند که همه آنها در فضای جدید در یک مختصات قرار بگیرند. به همین دلیل **می‌توان** رابطه «**یک به چند**» را با روش TransR پیاده‌سازی نمود.
3. تراگذاری: به دلیل ماهیت و روش پیاده‌سازی TransR، این ویژگی **نمی‌تواند** در جاسازی وجود داشته باشد، چرا که اصولا هر رابطه در یک فضای مجزا محاسبه می‌شود و نمی‌توان دو رابطه را در یک فضای یکسان محاسبه نمود.

پیوست (ب)

با توجه به توضیحات بخش 5 می‌توان نتیجه گرفت که قطعا ویژگی **تقارن** در این جاسازی قابل پیاده‌سازی بوده و همین‌طور ویژگی **پادتقارن** در این جاسازی قابل پیاده‌سازی نخواهد بود. در ادامه، سایر ویژگی‌ها را بررسی می‌کنیم:

1. رابطه یک به چند: اگر بخواهیم یک ویژگی، رابطه یک به چند را پشتیبانی کند باید بتوانیم داشته باشیم:

$$< h.r.t_1 >=< h.r.t_2 >$$

تساوی بالا، با تعیین درست مقادیر بردار ویژگی تمام پارامترها، برقرار خواهد بود. لذا رابطه **«یک به چند»** در این جاسازی قابل استفاده **است**.

2. معکوس‌پذیری و تراگذاری: به دلیل ماهیت این جاسازی، واضح است که این ویژگی‌ها، توسط این جاسازی قابل پیاده‌سازی نیست. چون هر رابطه‌ای که سرآیند با تالی داشته باشد، همان رابطه نیز بین تالی و سرآیند برقرار خواهد بود.

پیوست (ج)

1) رسیدگی به جاسازی‌های مختلط: یک بردار مختلط $v \in C^d$ با $v = v_{re} + iv_{im}$، از یک قسمت حقیقی $v_{re} \in \mathrm{Z}^d$ و یک قسمتی موهومی $v_{im} \in \mathrm{Z}^d$ تشکیل شده است. برای رسیدگی به جاسازی‌های مختلط، می‌توانیم از بردار حقیقی $2d$ بُعدی $[v_{re}.v_{im}]$ برای نمایش بردار مختلط d بُعدی v استفاده کنیم. فرض کنید جاسازی مختلط $h = h_{re} + ih_{im}$، که $h_{re}.h_{im} \in \mathrm{Z}^d$ (همچنین برای t و r) سپس ComplEx را می‌توان به‌صورت زیر توصیف کرد:

$$Re \ll h.r.conj(t) \gg =< h_{re}.r_{re}.t_{re} > + < h_{im}.t_{re}.t_{im} > +< h_{re}.r_{im}.t_{im} > +< h_{im}.r_{im}.t_{re} >$$

به‌طور مشابه، جاسازی‌های دو بُعدی را نیز می‌توان با $[v_{re} + v_{im}]$ نشان داد و به صورت دو قسمتی نمایش داد:

$$< h.r.t >=< h_{re}.r_{re}.t_{re} > +< h_{im}.r_{im}.t_{im} >$$

2) رسیدگی به تقسیم‌بندی مختلف: برای اینکه پارامترهای آموزش را پایدار کنیم، از جاسازی‌های با مقدار حقیقی دوبعدی برای نمایش Analogy و SimplE استفاده می‌کنیم. در Analogy جاسازی‌ها به یک قسمت حقیقی $\hat{h} \in \mathrm{Z}^d$ و یک قسمتی مختلط $\check{h} \in \mathrm{Z}^d$ تقسیم‌بندی می‌شوند که می‌توان آن را به‌صورت بردار حقیقی به‌هم‌پیوسته $[\widetilde{h_{re}}.\widetilde{h_{im}}] \in \mathrm{Z}^d$ شبیه به ComplEx نشان داد. تابع امتیاز نیز به صورت زیر تقسیم می‌شود:

$$< \hat{h}.\hat{r}.\hat{t} >= Re(< \check{h}.\check{t}.conj(\check{t}) >)$$

در SimplE، دو بردار جاسازی مستقل $\hat{h} \in \mathrm{Z}^d$ و $\check{h} \in \mathrm{Z}^d$ برای نمایش هر موجودیت و رابطه استفاده شده است. تابع امتیاز حاصله به صورت زیر است:

$$< \hat{h}.\hat{r}.\hat{t} > + < \check{h}.\check{r}.\check{t} >$$