ارائهٔ روشی جدید برای تعبیهٔ اسناد جهت دسته‌بندی متون خبری

زهرا رحیمی [۱] و محمد مهدی همایون‌پور [۲*]

[۱ و ۲] دانشکده مهندسی رایانه و فناوری اطلاعات - دانشگاه صنعتی امیرکبیر، تهران، ایران

چکیده

یکی از کاربردهای مهم در پردازش زبان طبیعی، دسته‌بندی متون است. برای دسته‌بندی متون خبری باید ابتدا آنها را به شیوه مناسبی بازنمایی کرد. روش‌های مختلفی برای بازنمایی متن وجود دارد ولی بیشتر آنها روش‌هایی همه منظوره هستند و فقط از اطلاعات هم‌رخدادی محلی و مرتبه اول کلمات برای بازنمایی استفاده می‌کند. در این مقاله روشی بی‌ناظر برای بازنمایی متون خبری ارائه شده‌است که از اطلاعات هم‌رخدادی سراسری و اطلاعات موضوعی برای بازنمایی اسناد استفاده می‌کند. اطلاعات موضوعی علاوه‌بر اینکه بازنمایی انتزاعی‌تری از متن ارائه می‌دهد، حاوی اطلاعات هم‌رخدادی‌های مراتب بالاتر نیز هست. اطلاعات هم‌رخدادی سراسری و موضوعی مکمل یکدیگرند؛ بنابراین در این مقاله به‌منظور تولید بازنمایی غنی‌تری برای دسته‌بندی متن، هر دو به‌کارگرفته شده‌اند. روش پیشنهادی بر روی پیکره‌های R8 و 20-Newsgruops که از پیکره‌های شناخته‌شده برای دسته‌بندی متون هستند، آزمایش و با روش‌های مختلفی مقایسه شد. در مقایسه با روش پیشنهادی با سایر روش‌ها افزایش دقتی به میزان افزایش ۳٪ مشاهده شد.

واژگان کلیدی: بازنمایی سند، تعبیه سند، تعبیه کلمه، هم‌رخدادی کلمات، اطلاعات موضوعی، دسته‌بندی متن

A New Document Embedding Method for News Classification

Zahra Rahimi[1] And Mohammad Mahdi Homayounpour[2*]

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Abstract

Text classification is one of the main tasks of natural language processing (NLP). In this task, documents are classified into pre-defined categories. There is lots of news spreading on the web. A text classifier can categorize news automatically and this facilitates and accelerates access to the news. The first step in text classification is to represent documents in a suitable way that can be distinguishable by a classifier. There is an abundance of methods in the literature for document representation which can be divided into a bag of words model, graph-based methods, word embedding pooling, neural network-based, and topic modeling based methods. Most of these methods only use local word co-occurrences to generate document embeddings. Local word co-occurrences miss the overall view of a document and topical information which can be very useful for classifying news articles.

In this paper, we propose a method that utilizes term-document and document-topic matrix to generate richer representations for documents. Term-document matrix represents a document in a specific way where each word plays a role in representing a document. The generalization power of this type of representation for text classification and information retrieval is not very well. This matrix is created based on global co-occurrences (in document-level). These types of co-occurrences are more suitable for text classification than local co-occurrences. Document-topic matrix represents a document

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۳۹۹/۵/۱۱ • تاریخ پذیرش: ۱۳۹۹/۱۲/۱۸ • تاریخ انتشار: ۱۴۰۱/۱۲/۲۹ • نوع مطالعه: پژوهشی

in an abstract way and the higher level co-occurrences are used to generate this matrix. So this type of representation has a good generalization power for text classification but it is so high-level and misses the rare words as features which can be very useful for text classification.

The proposed approach is an unsupervised document-embedding model that utilizes the benefit of both document-topic and term-document matrices to generate a richer representation for documents. This method constructs a tensor with the help of these two matrices and applied tensor factorization to reveal the hidden aspects of data. The proposed method is evaluated on the task of text classification on 20-Newsgroups and R8 datasets which are benchmark datasets in the news classification area. The results show the superiority of the proposed model with respect to baseline methods. The accuracy of text classification is improved by 3%.

Keywords- Text classification, Document representation, Document Embedding, Topic modeling, word co-occurrences

۱- مقدمه

امروزه منابع متنی زیادی مانند متون رایانامه‌ها، تار نماها، اتاق‌های گفتگوی موجود در شبکه‌های اجتماعی و غیره در دسترس است. برای استفاده و درک این متون به‌وسیله رایانه باید بتوان آنها را به‌طریق مناسبی دسته‌بندی کرد. دسته‌بندی متن[۱]، به فرایند تقسیم خودکار متون به دسته‌های از پیش تعیین‌شده گفته‌می‌شود.

قبل از اینکه یک متن بتواند به‌صورت خودکار و به‌وسیله رایانه دسته‌بندی شود باید به شیوه مناسبی بازنمایی شود. روش‌های مختلفی تاکنون برای بازنمایی متن[۲] ارائه شده‌است که می‌توان آنها را به پنج دسته تقسیم کرد: روش‌های مبتنی بر کیسه کلمات[۳]، روش‌های مبتنی بر تعبیه کلمات، روش‌های مبتنی بر مدل موضوعی، روش‌های مبتنی بر شبکه عصبی و روش‌های مبتنی بر گراف. این روش‌ها به‌اختصار در این بخش و با جزئیات بیشتر در بخش کارهای پیشین توضیح داده‌شده‌اند.

روش‌های مبتنی بر کیسه کلمات [1]–[3] به‌طورمعمول وجود یا عدم وجود یک کلمه در سند را نشان می‌دهند. بازنمایی اسناد با استفاده از این روش‌ها، ساده است، ولی آن‌ها نمی‌توانند به‌خوبی معنای سند و ترتیب کلمات را نشان دهند [4]. همچنین ابعاد کیسه کلمات ساخته‌شده برای یک سند به‌طورمعمول به‌اندازه تعداد کلمات موجود در لغت‌نامه[۴] است که اندازه بزرگی است و این بردار بسیار تنک است؛ در نتیجه می‌تواند به نفرین بعد[۵] منجر شود [5].

بعد از ارائه روش‌های تعبیه کلمه ([6], [7]) به‌دلیل ویژگی ذاتی آنها در بازنمایی معنا، نویسندگان بسیاری سعی در استفاده از این بردارها برای تولید بازنمایی اسناد کردند ([8]–[10]). این روش‌ها را می‌توان به دو زیرشاخه روش‌های ادغام[۶] و روش‌های کیسه معنا تقسیم کرد. روش‌های ادغام [11]–[14] از عملیات جبری مثل میانگین‌گیری، محاسبه مجموع، بیشینه یا کمینه‌گیری بر روی بردارهای کلمه استفاده می‌کند. این عملیات جبری مثل میانگین‌گیری نمی‌توانند نشان‌دهنده معنای کلی سند باشند [1]. روش‌های مبتنی بر کیسه مفهوم[۷] [8], [9] به‌طورمعمول از خوشه‌بندی بردارهای کلمه استفاده می‌کنند. این روش‌ها توسط انسان قابل تفسیرند و بازنمایی ارائه‌شده توسط آنها بهتر از روش‌های کیسه کلمات، نشان‌دهنده معنای سند است؛ ولی بسیاری از جنبه‌های موجود در سند مثل اطلاعات موضوعی را نادیده می‌گیرند و از طرفی تعیین تعداد مناسب خوشه‌ها در آنها مشکل است.

روش‌های مبتنی بر شبکه عصبی، از یک شبکه عصبی مثل شبکه پیش‌روی[۸] چندلایه، کانولوشنال، شبکه بازگشتی[۹]، حافظه کوتاه‌مدت ماندگار[۱۰] (LSTM) [15]–[17] برای تولید بردار اسناد استفاده می‌کنند. بسیاری از این روش‌ها با ناظر هستند و مشکل آنها این است که بسیاری از آنها فقط از هم‌رخدادی‌های محلی کلمات برای تولید بردارهای سند استفاده می‌کنند و اطلاعات هم‌رخدادی سراسری[۱۱] (در سطح سند) و اطلاعات موضوعی را نادیده می‌گیرند. زمان آموزش این روش‌ها به‌طورمعمول طولانی است. روش‌های مبتنی بر گراف [18]–[20]، معمولاً از گراف کلمات یا گراف کلمات و اسناد برای تولید بردار اسناد استفاده می‌کنند. زمان آموزش این روش‌ها طولانی است و فقط از اطلاعات هم‌رخدادی استفاده می‌کنند.

[1] Text classification (Text Categorization)
[2] Text representation (document representation)
[3] Bag of Words (BOW)
[4] Vocabulary
[5] Curse of dimensionality
[6] Pooling
[7] Bag of concepts approach
[8] Feedforward
[9] Recursive
[10] Long short-term memory (LSTM)
[11] Global co-occurrences

همان‌طورکه بیان شد، یکی از معایب بیشتر روش‌های پیشین، این است که فقط از اطلاعات هم‌رخدادی محلی برای تولید بردار اسناد استفاده می‌کنند؛ در صورتی که متن، حاوی اطلاعات بیشتری مثل اطلاعات هم‌رخدادی سراسری و اطلاعات موضوعی نیز هست. برتری اطلاعات هم‌رخدادی سراسری در این است که می‌توانند نشان‌دهنده موضوع یا دسته متن باشند و سبب درک و دریافت بهتر آن خواهند شد [21], [22]. اطلاعات موضوعی نیز دید انتزاعی‌تری از متن را در اختیار ما قرار می‌دهد که سبب افزایش قدرت تعمیم دسته‌بند می‌شود [23]. برای غلبه بر این ضعفِ روش‌های پیشین، روشی ارائه شده‌است که می‌تواند هم از اطلاعات هم‌رخدادی سراسری (ماتریس کلمه-سند) و هم از اطلاعات موضوعی (ماتریس سند-موضوع) برای تولید بازنمایی اسناد استفاده کند. ماتریس کلمه-سند حاوی اطلاعاتی هم‌رخدادی اسناد و کلمات است. در این ماتریس که کلمات سطرهای آن و اسناد ستون‌های آن را نشان می‌دهند، هر درایه نشان‌دهنده تعداد بارهای تکرار یک کلمه در یک سند است. این ماتریس به‌نوعی دربردارنده اطلاعات هم‌رخدادی کلمات در سطح سند نیز هست (می‌توان به این اطلاعات با استفاده از ضرب این ماتریس در ترانهاده‌اش دست یافت). اگر فقط از فرکانس کلمات برای تولید بردارهای سند استفاده شود، بیش از اندازه به کلمات توجه می‌شود و می‌توان گفت بازنمایی سند با این روش فقط یک دیدگاه خاص از سند را ایجاد می‌کند و اگر کلمات عوض شوند، ولی سند همان معنا را بدهد، شاید این دو سند به یک رده تعلق نگیرند در نتیجه قدرت تعمیم دسته‌بند تحت تأثیر قرار خواهد گرفت. برای اینکه اسناد بتوانند به‌خوبی از هم جدا شوند، نیاز به دیدگاه انتزاعی‌تر از سند داریم که ماتریس سند-موضوع می‌تواند این دیدگاه را در اختیار ما قرار دهد. ولی ماتریس سند –موضوع به تنهایی یک دیدگاه بسیار کلی از سند را ارائه می‌دهد که کلمات را به‌طور کلی نادیده می‌گیرد؛ درصورتی‌که گاهی‌اوقات کلمات نادر و خاص از یک سند می‌توانند به ما کمک کنند. برای استفاده از مزایای هر دو دیدگاه، روشی ارائه شده است که با استفاده از تنسور این دو دیدگاه را ترکیب کنیم و با استفاده از تجزیه تنسور که قادر است ویژگی‌های پنهان داده‌ها را آشکار کند [24] بازنمایی مناسبی را برای اسناد ایجاد کنیم. شیوه تجزیه تنسور باید به‌گونه‌ای باشد که بتواند به‌خوبی ویژگی‌های پنهان در داده‌ها را استخراج کند. همچنین باید بتواند فاکتورهای هموار ایجاد کند و در روند بهینه‌سازی دچار بیش‌برازش نشود. در روش پیشنهادی از شیوه تجزیه تنسور فاکتورهای موازی منظم‌سازی‌شده استفاده شده‌است که به‌دلیل استفاده از نرم دو در عبارت منظم‌ساز[1] فاکتورها نسبت به شیوه تجزیه به‌کارگرفته‌شده در [25] فاکتورهای هموارتری ایجاد می‌کند و دچار بیش‌برازش نمی‌شود.

۲- کارهای پیشین

در این بخش به معرفی روش‌های مختلف بازنمایی سند که در سال‌های پیش مطرح‌شده‌اند و معایب و مزایای آنها پرداخته شده‌است. همان طور که در بخش مقدمه عنوان شد، روش‌های بازنمایی سند را می‌توان به پنج دسته تقسیم کرد که عبارتند از: روش‌های مبتنی بر کیسه کلمات، روش‌های مبتنی بر مدل موضوعی، روش‌های مبتنی بر تعبیه کلمات، روش‌های مبتنی بر شبکه عصبی و روش‌های مبتنی بر گراف که در ادامه توضیح داده‌شده‌اند.

۲-۱- روش‌های مبتنی بر کیسه کلمات

یکی از روش‌های ساده برای بازنمایی سند روش کیسه کلمات است. در این روش، طول بردار یک سند به‌اندازه طول لغت‌نامه است. اگر ith کلمه از لغت‌نامه m بار در سند ظاهر شده باشد، مقدار ith مؤلفه از بردار کلمات آن سند برابر با m قرار می‌گیرد. Tf-IDF یکی از روش‌های وزنده‌ی به کیسه کلمات است [3]. در این روش، کلماتی که در تعداد زیادی سند تکرار شده‌اند، جریمه می‌شوند. روش‌های دیگری نیز در این زمینه ارائه شده‌است؛ مثل کیسه کلمات فازی [2] که از فاصله کسینوسی بین تعبیه کلمات برای وزن‌دهی به آنها استفاده می‌کند. در [26] از یک شیوه جدید وزن‌دهی با استفاده از منطق فازی استفاده شده‌است.

۳-۱- روش‌های مبتنی بر مدل موضوعی

ابتدایی‌ترین روش در این زمینه تحلیل معنایی پنهان[2] است [27]. در این روش، ماتریس کلمه-سند با استفاده از تجزیه مقدار منفرد (SVD)[3] تجزیه شده و با این تجزیه به فضای موضوع‌ها می‌رود. ماتریس سند-موضوع به‌دست‌آمده از این تجزیه را می‌توان به‌عنوان بازنمایی برای سند در نظر گرفت. این روش، مدلی احتمالاتی را

[1] regularization
[2] Latent Semantic Analysis (LSA)
[3] Singular Value Decomposition (SVD)

برای سند در نظر نمی‌گیرد. بعدها در [28] مدل احتمالاتی برای سند در نظر گرفته شد. در این مدل که یک مدل مولد[1] است، سند به‌صورت یک توزیع احتمالاتی روی موضوع‌ها مدل می‌شود و هر کلمه، موضوع مخصوص به خود را دارد؛ ولی این مدل ممکن است، روی داده‌ها بیش‌برازش[2] شود و همچنین مدل احتمالاتی از کلمات بر روی موضوع‌ها در نظر نمی‌گیرد. به همین دلیل روش تخصیص پنهان دیریکله[3] (LDA) [29] ارائه شد. در این روش، هر سند به‌صورت توزیعی روی موضوع‌ها و هر موضوع به‌صورت توزیعی روی کلمات تعریف می‌شود. خروجی این روش دو ماتریس ϕ و θ هستند که ϕ نشان‌دهنده توزیع احتمال موضوع‌ها روی کلمات است و θ توزیع احتمال موضوع‌ها در اسناد را نشان می‌دهد. بعدها روش‌های دیگری ارائه شدند، مثل بازنمایی متن با استفاده از موضوع‌های پنهان [30] که از توزیع پنهان دیریکله گوسی [31] استفاده می‌کنند. این روش‌ها به‌طورمعمول دیدگاه انتزاعی از متن را ارائه می‌دهند که سبب افزایش قدرت تعمیم در هنگام دسته‌بندی می‌شود، ولی بسیاری از ویژگی‌های در سطح کلمه مثل کلمات خاص را نادیده می‌گیرند[23].

۴-۱- روش‌های مبتنی بر تعبیه کلمات

در این روش‌ها، از تعبیه کلمات برای تولید بردارهای سند استفاده می‌شود. روش‌هایی که اغلب مورد استفاده قرار می‌گیرند، عبارتند از استفاده از میانگین، مجموع یا بیشینه و کمینه‌گیری یا پیوست‌کردن تعبیه کلمات موجود در یک سند [11]–[14]. تنها استفاده از این عملیات جبری به‌خوبی نمی‌تواند نشان‌دهنده معنای متن باشد [1]. روش‌های دیگری نیز در سال‌های اخیر ارائه شدند که از خوشه‌بندی تعبیه کلمات استفاده می‌کنند که به این روش‌ها، روش‌های مبتنی بر مفهوم[4] گفته‌می‌شود [8], [9]. این روش‌ها به‌وسیله انسان قابل‌تفسیر هستند، ولی تعیین تعداد خوشه‌های مناسب در آنها مشکل است [9].

۶-۱-روش‌های مبتنی بر شبکه عصبی

در این روش‌ها از شبکه‌های عصبی برای تولید بردارهایی برای بازنمایی اسناد استفاده می‌شود. یکی از معروف‌ترین روش‌ها در این زمینه، paragraph-vector [15] است. این روش در سال ۲۰۱۴ توسط لی[5] و میکولو[6] ارائه شد. در این روش که بسیار به word2vec شباهت دارد از یک شبکه عصبی پیش‌رو با یک لایه مخفی استفاده می‌شود تا بردار اسناد و کلمات همراه با یکدیگر یاد گرفته شود. این روش، یک روش بی‌ناظر[7] است. در [16] روش دیگری برای بازنمایی جملات با استفاده از شبکه کانولوشنال ارائه شده‌است. این روش یک روش با ناظر است. روش‌های دیگری نیز برای بازنمایی اسناد با استفاده از حافظه کوتاه‌مدت ماندگار (LSTM) [32] و حافظه کوتاه‌مدت ماندگار دوطرفه (Bi-LSTM)[8] [33] ارائه شده‌است. در این روش‌ها، ترتیب کلمات در نظر گرفته‌می‌شود، ولی شبکه بازگشتی[9] به‌طورمعمول زمان آموزش طولانی دارد. در [34] از روش ادغام مبتنی بر توجه[10] برای تولید بازنمایی اسناد استفاده می‌کند.

از مزایای بیشتر روش‌های مبتنی بر شبکه عصبی این است که ترتیب کلمات موجود در سند که می‌تواند روی معنای سند تأثیرگذار باشد را به‌نوعی در نظر می‌گیرند؛ ولی این روش‌ها، زمان آموزش طولانی دارند و فقط هم‌رخدادی‌های مرتبه اول را در نظر می‌گیرند؛ پس هم‌رخدادی‌های مراتب بالاتر نادیده گرفته می‌شوند. در این روش‌ها اسناد به‌صورت یک دنباله از کلمات در نظر گرفته و بسیاری از اطلاعات پنهان در سند مثل اطلاعات موضوعی نادیده گرفته می‌شوند.

۵-۱-روش‌های مبتنی بر گراف

این روش‌ها از گراف برای تولید بردارهای کلمه استفاده می‌کنند. در مقاله [35] روشی مبتنی بر گراف ارائه شده‌است که در آن برای هر سند یک گراف هم‌رخدادی کلمات تولید می‌شود. [19] از شبکه عصبی گراف کانولوشنی[11] استفاده می‌کند. در [19] ابتدا یک گراف از کلمات تولید می‌شود و سپس از روش‌های تعبیه نود برای تولید تعبیه‌های کلمه و سند استفاده می‌کند. این روش‌ها نیز ترتیب کلمات را در نظر نمی‌گیرند، مگر اینکه از گراف جهت‌دار استفاده‌کنند که در این‌صورت پیچیدگی مدل‌بالا می‌رود. همچنین اطلاعات اسناد و اطلاعات هم‌رخدادی مراتب بالاتر را که می‌توانند بسیار مفید باشند، استفاده نمی‌کنند.

[1] Generative
[2] Over-fit
[3] Latent Dirichlet allocation (LDA)
[4] Concept-based
[5] Le
[6] Mikolov
[7] unsupervised
[8] Bi-directional Long short-term memory (Bi-LSTM)
[9] Recurrent neural network
[10] Attentive pooling
[11] Graph Convolutional neural net

۲- پیشزمینهها

این بخش به شرح تعاریف و نمادهای مورد نیاز در مقاله میپردازد.

۲-۱-تعریفها

تعریف ۱: تنسور- تنسور یک آرایه چندبعدی است که برای بازنمایی دادههای با بیش از دو بعد مورد استفاده قرار میگیرد. بهطور رسمیتر یک تنسور N-way یا از مرتبه N[1]، جزئی از ضرب تنسوری N فضای برداری است که هر کدام دارای سیستم مختصات مربوط به خود هستند. در این مقاله تنسورها با حروف خطاطی شده بهشکل $\mathcal{D}$ نشان داده میشوند.

تعریف ۲: اسلایس- اسلایسها یا برشها، بخشهای دوبعدی یک تنسور هستند که از ثابت نگهداشتن همه ابعاد بهجز دو تا از آنها بهدست میآیند. تنسور از مرتبه ۳ دارای سه نوع اسلایس است که عبارتند از اسلایس روبهرویی[2]، اسلایس افقی[3] و اسلایس پهلویی[4] که برای تنسور $\mathcal{D}$، این اسلایسها بهترتیب با نمادهای $\mathcal{D}_{::k}$، $\mathcal{D}_{k::}$ و $\mathcal{D}_{:k:}$ نشان داده میشوند.

تعریف ۳: ضرب خارجی: ضرب خارجی دو بردار با استفاده از نماد ⊙ نشان دادهمیشود.

تعریف ۴: ضرب Hadamard- ضرب ماتریسی مؤلفه به مؤلفه که با نماد ⊛ نشان دادهشدهاست.

تعریف ۵: ضرب Kronecker- ضرب Kronecker دو ماتریس $A \in \mathbb{R}^{I\times J}$ و $B \in \mathbb{R}^{K\times L}$ بهصورت $A\otimes B$ نشان دادهمیشود و حاصل آن یک ماتریس با اندازه $(IK) \times (JL)$ است:

$$A\otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1J}B \\ \vdots & \ddots & \vdots \\ a_{I1}B & \cdots & a_{IJ}B \end{bmatrix} \quad (1)$$

تعریف ۶: ضرب Khatri-Rao- یک ضرب Kronecker ستونی است که با نماد ⊙ نشان داده میشود.

[1] Nth-order
[2] Frontal slice
[3] Horizontal slice
[4] Lateral slice

۲-۲- نمادها

نمادهای به کار رفته در این مقاله بهاختصار در جدول (۱) آورده شدهاست.

(جدول- ۱): فهرست نمادها
(Table-1): Symbols used in the paper

نماد	توضیح
اسکار	حرف کوچک انگلیسی (a)
بردار	حرف کوچک پررنگ انگلیسی (**a**)
مؤلفه i از بردار a	a_i
ماتریس	حرف بزرگ انگلیسی (A)
درایه ij از ماتریس A	a_{ij}
تنسور	حرف اسکریپت انگلیسی ($\mathcal{A}$)
یک درایه تنسور سه بعدی $\mathcal{A}$	a_{ijk}

۳- توضیح روش

اطلاعات زیادی در یک متن، نهفتهاست. از این اطلاعات میتوان برای تولید بازنمایی غنیتری برای متن استفادهکرد؛ برای مثال میتوان از اطلاعات موضوعی و اطلاعات همرخدادی برای تولید بازنمایی متن استفادهکرد. اطلاعات موضوعی مثل ماتریس سند-موضوع حاوی اطلاعات همرخدادی مراتب بالاتر نیز هست و اطلاعات همرخدادی در سطح سند یا همان همرخدادی سراسری، برای تولید بازنمایی مناسب برای دستهبندی اسناد میتواند مفید باشد. در بیشتر روشهای ارائهشده پیشین بهطورمعمول از اطلاعات همرخدادی محلی استفاده میشود که برای دستهبندی سند زیاد مناسب نیست [21] و همچنین اطلاعات موضوعی که دید کلیتری [23] از متن را نسبت به کلمات در اختیار قرار میدهد نادیده گرفتهمیشود. در روش ارائهشده، ابتدا با استفاده از اطلاعات مستخرج از ماتریس سند-موضوع و ماتریس کلمه-سند اسلایسهایی برای تنسور ساخته، سپس تنسور حاصل، تجزیه شده و بازنماییهای اسناد تولید میشوند. هر سطر ماتریس کلمه –سند، Tf-idf یک کلمه در اسناد مختلف را نشانمیدهد که از اطلاعات همرخدادی سراسری استفاده میکند و در فرایند استخراج موضوعها بهوسیله تحلیل پنهان دیریکله نیز از اطلاعات هم رخدادی سراسری استفاده شدهاست. همچنین تحلیل پنهان دیریکله شامل اطلاعات همرخدادی مراتب بالاتر نیز است در نتیجه حاوی اطلاعات بیشتری از همرخدادی کلمات باشد.

۳-۱- ساخت اسلایس روبه‌رویی بـا اســتفاده از اطلاعات حاصل از ماتریس کلمه-سند

ماتریس کلمه-سند، ماتریسی است که سطرهای آن کلمه‌های موجود در لغت‌نامه و ستون‌های آن اسناد هستند. اگر کلمه i در یک سندj ظاهر شده باشد و ماتریس کلمه-سند با M نشان داده شود، m_{ij} برابر با تعداد بارهایی است که کلمه i در سند j ظاهر شده‌است. در اینجا از وزن‌دهی با $\mathrm{TF}-\mathrm{IDF}$[1] به‌جای فرکانس کلمه استفاده شده‌است.

هدف، تولید تنسوری است که اندازه هر یک از اسلایس‌های روبه‌رویی‌اش $d \times d$ باشد که d تعداد اسناد است. برای ساخت چنین اسلایس‌های روبه‌رویی با استفاده از اطلاعات موجود در ماتریس کلمه-سند از رابطه زیر استفاده می‌شود:

$$\mathcal{D} = M^T \times M \qquad (۲)$$

در این رابطه D یک ماتریس سند-سند است که به‌عنوان یکی از اسلایس‌های روبه‌رویی تنسور قرار می‌گیرد. در واقع با انجام این کار رابطه بین اسناد از دیدگاه کلمات آنها مشخص می‌شود.

۳-۲- ساخت اسلایس روبه‌رویی بـا اســتفاده از ماتریس سند-موضوع

ماتریس سند-موضوع ماتریسی است که سطرهای آن نشان‌دهنده سندهای مختلف و ستون‌های آن نشان‌دهنده موضوعات هستند. این ماتریس یکی از خروجی‌های الگوریتم تخصیص پنهان دیریکله[2] است.

همان طور که بیان شد، هر سطر از ماتریس سند-موضوع نشان‌دهنده توزیع احتمال موضوع‌ها در آن سند است؛ بنابراین هر سطر این ماتریس یک بردار از احتمالات موضوعات برای سند مربوطه است و می‌توان گفت یک بازنمایی موضوعی برای آن سند ارائه می‌کند؛ بنابراین می‌توان با به‌کارگیری الگوریتم k-means بر روی سطرهای این ماتریس، اسناد را خوشه‌بندی کرد؛ ولی قبل از خوشه‌بندی، به‌دلیل اینکه الگوریتم k-means در مختصات حقیقی مورد استفاده قرار می‌گیرد، باید با استفاده از تبدیلی داده‌های احتمالاتی که داده‌هایی ترکیبی[3] هستند، به داده‌های حقیقی تبدیل شوند. برای این منظور از نگاشت نرخ لگاریتمی ایزومتریک[4] (ilr) می‌توان استفاده‌کرد. این نگاشت به‌صورت رابطه زیر است:

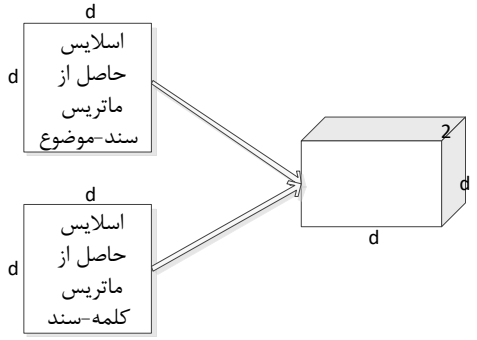
$$ilr: S^D \to \mathbb{R}^{D-1} \qquad (۳)$$

در این رابطه S^D، سیمپلکس Aitchison است. بعد از انجام این تبدیل، الگوریتم k-means بر روی بردارهای سند-موضوع اعمال می‌شود تا خوشه‌بندی انجام شود. بعد از اینکه خوشه‌بندی انجام شد از رابطه زیر برای تولید اسلایس روبرویی مربوطه استفاده می‌گردد.

$$d_{ij} = \begin{cases} 1 & i \wedge j \; are \; members \; of \; the \; same \; cluster \\ 0 & i \wedge j \; are \; not \; members \; of \; the \; same \; cluster \end{cases} \qquad (۴$$

۳-۳- ساخت تنسور با استفاده از اسلایس‌هـای تولید شده

بعد از اینکه اسلایس‌های روبرویی با استفاده از اطلاعات حاصل از ماتریس سند-موضوع و کلمه-سند تولید شدند، این اسلایس‌ها تشکیل یک تنسور می‌دهند. اندازه هر یک از اسلایس‌های روبرویی تنسور حاصل، $d \times d$ است و اندازه آن $d \times d \times 2$ است. این فرایند در شکل (۱) نشان داده‌شده‌است.



(شکل- ۱): ساخت تنسور با استفاده از اسلایس‌های روبه‌رویی
(Figure-1): Constructing tensor with frontal slices

۳-۴- ساخت بردارهای سند

برای ساخت بردارهای سندی که از مزایای ماتریس سند-موضوع و کلمه-سند بهره ببرند، تنسوری که در بخش قبل تولید شد، با استفاده از روش فاکتورهای موازی منظم‌سازی‌شده[5] تجزیه می‌شود تا بازنمایی غنی‌تری حاصل شود. تجزیه تنسور می‌تواند الگوهای پنهان موجود در داده‌ها را آشکار کند [24]. این بخش به شرح تجزیه فاکتورهای موازی[6] و تجزیه فاکتورهای موازی منظم‌سازی‌شده[7] و چگونگی ساخت بردارها به‌وسیله آنها می‌پردازد.

[1] Term frequency (TF)-inverse document frequency (IDF)
[2] Latent Dirichlet Allocation (LDA)
[3] compositional
[4] Isometric log-ratio transformation (ilr)
[5] Regularized Parallel Factors
[6] Parallel Factors
[7] Regularized Parallel Factors

۱-۴-۳-تجزیه فاکتورهای موازی

در این الگوریتم تجزیه، تنسور به‌صورت مجموعی از مؤلفه‌های مرتبه[1] یک تجزیه می‌شود. تجزیه تنسور مرتبه سه تولید شده، با استفاده از روش تجزیه فاکتورهای موازی به‌صورت زیر انجام‌می‌شود:

$$\mathcal{D} \approx \widehat{\mathcal{D}} = \sum_{r=1}^{R} a_r \circ b_r \circ c_r \quad (۵)$$

در این رابطه R یک عدد صحیح مثبت و نشان‌دهنده رنک تنسور است و $a_r \in \mathbb{R}^I$ و $b_r \in \mathbb{R}^J$ و $c_r \in \mathbb{R}^k$ برای $r = 1, \dots R$ هستند. رابطه بهینه‌سازی برای تولید بهترین تجزیه (بهترین $\widehat{\mathcal{D}}$) به‌صورت زیر است:

$$\min_{A,B,C} \frac{1}{2} \| \mathcal{D} - \widehat{\mathcal{D}} \|_F^2 \quad (۶)$$

در این رابطه A و B و C ماتریس‌هایی هستند که i-امین ستون آنها شامل یک بردار a_i، b_i و c_i است؛ یعنی A را می‌توان به‌صورت $A = [a_1, a_2, \dots, a_R]$ نشان داد. به این ماتریس‌ها، ماتریس‌های فاکتور[2] گفته‌می‌شود. برای حل این مسئله بهینه‌سازی که بر روی هر سه فاکتور محدب نیست از شیوه بهینه‌سازی کمینه مربعات تناوبی[3] استفاده می‌شود. در این شیوه هر یک از فاکتورها با درنظرگرفتن ثابت‌بودن دو فاکتور دیگر به‌صورت جداگانه و با شیوه‌ای تکراری بهینه می‌شوند [24].

یکی از معایب این روش این است که فاکتورهای حاصل از ماتریس به‌صورت هموار[4] نیستند و نیاز به یک مرحله نرمال‌سازی بعد از تولید فاکتورهاست و همین‌طور مستعد بیش‌برازش است [36]. به همین دلیل در روش ارائه‌شده در این مقاله از منظم‌سازی نرم ۲ در تابع بهینه‌سازی تجزیه تنسور استفاده شده‌است تا فاکتورهای مناسب‌تر و هموارتری تولید شود که منجر به افزایش دقت و از بیش‌برازش جلوگیری می‌شود؛ ولی حل رابطه بهینه‌سازی در این روش به‌مراتب سخت‌تر از حالت قبل است. در بخش بعدی به‌اختصار روش فاکتورهای موازی منظم‌سازی‌شده توضیح داده‌شده‌است.

۲-۴-۳-تجزیه فاکتورهای موازی منظم‌سازی شده با نرم ۲

در این روش تجزیه تنسور نرم دو برای هر یک از ماتریس‌های فاکتور با استفاده از یک ضریب منظم‌سازی به تابع هدف همانند رابطه ۷ اضافه می‌شود. در این رابطه λ نرخ منظم‌سازی است.

$$\min_{A,B,C} \frac{1}{2} \| \mathcal{D} - \widehat{\mathcal{D}} \|_F^2 + \frac{\lambda}{2} (\| A \|_F^2 + \| B \|_F^2 + \| C \|_F^2) \quad (۷)$$

برای تنسورهای تنک به‌طور بهینه‌سازی فقط بر روی عناصر غیر صفر انجام‌می‌شود[36] و رابطه بالا به‌صورت رابطه زیر در می‌آید:

$$\min_{A,B,C} \frac{1}{2} \sum_{\{i,j,k\} \in \Omega} (\mathcal{D}(i,j,k) - \sum_{r=1}^{R} A(i,r)B(j,r)C(k,r))^2 + \frac{\lambda}{2} A \|_F^2 + \| B \|_F^2 + \| C \|_F^2) \quad (۸$$

در این رابطه Ω مجموعه همه عناصر غیر صفر تنسور است. برای حل این مسئله بهینه‌سازی، دیگر نمی‌توان همانند حالت قبل از الگوریتم کمینه‌مربعات تناوبی[5] استفاده نمود و باید از روش AO-ADMM[6] استفاده‌کرد. در این روش ابتدا مسئله بهینه‌سازی با در نظر گرفتن یک جمله کمکی به یک مسئله بهینه‌سازی با محدودیت تساوی تبدیل می‌شود و بعد به‌صورت تکراری و با ترکیب دو روش AO و ADMM حل می‌شود[36].

۳-۴-۳-تولید بردارهای سند

ماتریس‌های فاکتور A و B دارای اندازه $d \times R$ هستند و هر سطر آنها بردار مربوط به یک سند است. اندازه تعبیه‌های سند به‌اندازه رنک تنسور است. می‌توان از ترکیب بردارهای حاصل از این ماتریس‌های فاکتور برای تولید بردار سند استفاده‌کرد ولی از آزمایش‌ها این نتیجه گرفته شد که A به‌طورمعمول بردار مناسب‌تری را برای دسته‌بندی متن تولید می‌کند.

۵-۳-تولید بردار سند برای سندهای دیده‌نشده

با استفاده از روش ارائه‌شده در بخش قبل برای سندهای موجود می‌توان بردار تولید نمود ولی سؤالی که در اینجا مطرح می‌شود این است که برای اسناد دیده‌نشده بدون تکرارِ این فرایند چطور باید بردار تولید نمود؟ در این بخش، روشی برای تولید بردار برای اسناد دیده‌نشده ارائه شده‌است.

[1] rank
[2] Factor matrices
[3] Alternating Least Square (ALS)
[4] smooth
[5] Alternating Least square (ALS)
[6] Alternating optimization(AO)-Alternating Direction Method of Multipliers (ADMM)

تاکنون ماتریس‌های فاکتور A و B و C در روند آموزش، تولید شده‌اند. برای داده‌های جدید باید بتوان با استفاده از A، B و C قبلی و داده‌های جدید، بدون تکرار تجزیه، A_{new}، B_{new} و C_{new} را تولید کرد. برای تولید آن‌ها از روابط زیر استفاده می‌شود:

$$A_{new}(i,:) = (\mathrm{H}_i^T H_i + \lambda I)^{-1} H_i^T vec(\mathcal{D}(i,:,:)) \quad (۹)$$

$$H_i(l,:) = [B(j,:) \circledast \mathrm{C}(\mathrm{k},:)] \quad (۱۰)$$

که $\circledast$ ضرب Hadamard است. اگر l امین درایه غیر صفر در $\mathcal{D}(i,:,:)$ دارای مودهای (i,j,k) باشد آنگاه l امین سطر H_i به‌صورت رابطه ۱۰ به‌دست می‌آید [37].

۴-آزمایش‌ها

برای آزمایش اینکه این روش تا چه اندازه می‌تواند بردارهای مناسبی برای دسته‌بندی متن ارائه کند از دو پیکره R8 و 20NG استفاده شده‌است. این دو پیکره، جزء پیکره‌های معروف در زمینه دسته‌بندی متون خبری هستند. پیکره R8 شامل ۸ دسته و پیکره 20NG شامل ۲۰ دسته است. آمار این پیکره‌ها در جداول (۲) و (۳) آورده شده‌است.

(جدول- ۲): اطلاعات پیکره R8
(Table-2): R8 corpus statistics

نام رده‌ها	تعداد اسناد در هر رده
acq	۲۲۹۲
crude	۳۷۴
earn	۳۹۲۳
grain	۵۱
interest	۲۷۱
Money-fx	۲۹۳
ship	۱۴۴
trade	۳۲۶

در پیکره R8، ۵۴۸۵ سند برای آموزش و ۲۱۸۹ سند برای آزمایش استفاده شد و در پیکره 20-Newsgroups، ۱۱۳۰۰ سند برای آموزش و ۷۵۱۸ سند برای آزمایش مورد استفاده قرار گرفت.

در پیکره R8 و 20NG ایست‌واژه‌ها، اعداد و علائم حذف شدند. همچنین در پیکره 20NG همه سرنوشت‌ها، پانوشت‌ها که حاوی اطلاعات دسته‌ها هستند نیز حذف شدند.

(جدول- ۳): اطلاعات پیکره 20-Newsgroups
(Table-3): 20-Newsgroups corpus statistics

نام رده‌ها	تعداد اسناد در هر رده
Alt.atheism	۷۹۹
Comp.graphics	۹۷۳
Comp.os.ms-windows.misc	۹۶۶
Comp.sys.ibm.pc.hardware	۹۸۲
Comp.sys.mac.hardware	۹۶۳
Comp.windows.x	۹۸۵
Misc.forsale	۹۷۵
Rec.autos	۹۸۹
Rec.motorcycles	۹۹۶
Talk.sport.baseball	۹۹۶
Rec.sport.hockey	۹۴۴
Sci-crypt	۹۹۹
Sci.electronics	۹۹۱
Sci.med	۹۸۴
Sci.space	۹۹۰
Soc.religion.christian	۹۸۷
Talk.politics.guns	۹۶۶
Talk.politics.mideast	۹۰۹
Talk.politics.misc	۹۴۰
Talk.religion.misc	۷۷۵
	۶۲۸

۴-۱-روش‌های پایه

از روش‌های زیر که در سال‌های اخیر استفاده‌شده‌اند برای مقایسه و سنجش میزان بهبود حاصل از روش ارائه‌شده استفاده شده‌است.

- CNN-random یک شبکه عصبی کانولوشنال را برای تولید بازنمایی اسناد استفاده می‌کند این روش، با ناظر است [16].
- CNN-non-static: این روش مشابه CNN است، فقط تفاوت آن این است که در فرایند بازانتشار خطا فقط از یک کانال ورودی استفاده می‌شود و با این کار بردار کلمات نیز در طی آموزش fine-tune می‌شوند [16].
- Paragraph vector این روش توسط Le و Mikolov ارائه شد. در این روش که بسیار مشابه با word2vec است در طی فرایند تولید بردار کلمات بردار اسناد نیز تولید می‌شوند. در این روش از یک شبکه پیش‌روی سه لایه استفاده شده‌است [15].
- روش TWE[1] در این روش اطلاعات موضوع‌ها نیز به شبکه پیش‌رویی که برای تولید بردار کلمات در word2vec استفاده می‌شد، داده‌می‌شود [38].

[1] Topical word embedding (TWE)

- LTTR[1] در این روش نیز از اطلاعات موضوع‌ها استفاده شده‌است و اسناد به‌صورت یک مخلوط گوسی در نظر گرفته‌شده‌اند [30].
- Gaussian LDA: LDA که از توزیع گوسی استفاده می‌کند[31].
- LSTM[2] در این روش از یک شبکه LSTM برای تولید بردار اسناد استفاده می‌شود.
- BI-LSTM[3] در این روش از یک شبکه LSTM دو جهته برای تولید بردار کلمات استفاده می‌شود.
- Tens-Embedding: در این روش نیز از هر دو اطلاعات سراسری و موضوعی و همین طور تنسور برای تولید بردار کلمات استفاده شده‌است[25].

روش پیشنهادی در جدول‌ها با regularized Tens-Embedding+"kmeans"+"number of clusters" نشان داده‌شده‌است. برای دسته‌بندی متون در همه روش‌ها از دسته‌بند SVM استفاده شده‌است.

۱-۱-۴-نتایج به‌دست‌آمده بر روی پیکره R8

نتایج دسته‌بندی متون توسط دسته‌بند SVM با مقایسه انواع روش‌های بازنمایی اسناد که در بخش قبل به آنها اشاره شد با استفاده از داده‌های پیکره R8 در جدول (۴) نشان داده‌شده‌است. نتایج موجود در این جدول نشان می‌دهند که روش پیشنهادی این مقاله بر روی پیکره R8 نیز بهتر از سایر روش‌ها عمل می‌کند. بر طبق نتایج، روش‌هایی مثل LTTR و TWE که هم از اطلاعات موضوع‌ها و هم از اطلاعا هم‌رخدادی استفاده می‌کنند، بسیار ضعیف‌تر از روش پیشنهادی مبتنی بر تنسور (regularized-Tens-Embedding) ظاهر شده‌اند.

۱-۲-۴- نتایج به‌دست‌آمده بر روی پیکره 20-Newsgroups

نتایج دسته‌بندی متون توسط دسته‌بند SVM با مقایسه انواع روش‌های بازنمایی اسناد که در بخش قبل به آنها اشاره شد با استفاده از داده‌های پیکره 20-Newsgroups در جدول (۵) نشان داده‌شده‌اند.

بر طبق نتایج ارائه‌شده در جدول، دقت به‌دست آمده از روش مبتنی بر تنسور (regularized-Tens-Embedding) بهتر از سایر روش‌هاست و در مقایسه با LTTR و TWE و Tens-Embedding که از اطلاعات تاپیک‌ها استفاده می‌کنند، دقت بالاتری دارد.

[1] Latent topic text representation (LTTR)
[2] Long short-term memory (LSTM)
[3] Bi-directional LSTM (BI-LSTM)

(جدول- ۴): درصد دقت دسته‌بندی متون بر روی داده‌های آزمایشی پیکره R8

(Table-4): Accuracy of text classification on R8

روش	دقت بر روی داده-های آزمایش (درصد)
TF-IDF	92.5
paragraph-vector (CBOW)	85.87
TWE	91.67
CNN-rand	94.0
CNN-Non-static	95.7
Fast text	96.0
Bi-LSTM	96.09
LSTM	96.3
LDA	93.65
Gaussian LDA	93.78
LTTR	93.55
Tens-Embedding	97.07
regularized Tens-Embedding	97.3

(جدول- ۵): درصد دقت دسته‌بندی بر روی داده‌های آزمایشی پیکره 20-Newsgroups

(Table-5): Text classification accuracy on 20-Newsgroups

روش	دقت (%)
TF-IDF	۷۰.۰
(DBOW) paragraph-vector	۷۰.۰۱
TWE	۶۸.۸۷
W2vec averaging	۷۰.۵۶
LDA	۶۹.۸۵
Gaussian LDA	۷۲.۴۳
LTTR	۷۳.۲۲
Bag of Concepts	۵۳.۰۲
LSI	۷۴.۰
Tens-Embedding	۷۶.۷۴
Regularized Tens-Embedding +Kmeans+20 clus	۷۷.۶۰

۱-۳-۴- بحث بر روی نتایج

با توجه به نتایج نشان داده‌شده در جدول (۴) و جدول (۵) روش پیشنهادی regularized-Tens-Embedding برای تعبیه کلمات به‌دقت بالاتری نسبت به سایر روش‌ها رسیده است. در ادامه به مقایسه عمیق‌تری بین روش پیشنهادی و سایر روش‌ها خواهیم پرداخت. یکی از روش‌هایی که به‌روش ارائه‌شده از جهت استفاده از موضوع و هم‌رخدادی کلمات شباهت دارد روش TWE[4] است. در این روش برخلاف روش ارائه‌شده از هم‌رخدادی محلی (هم‌رخدادی کلمات در سطح پنجره اطراف کلمه هدف) کلمات استفاده می‌شود. هم‌رخدادی سراسری (هم‌رخدادی کلمات در سطح سند) برای دسته‌بندی متن می‌تواند اطلاعات بیشتری را در اختیار سامانه قرار دهد. همچنین در این روش برای یافتن بازنمایی اسناد از مجموع وزن‌دار

[4] Topical Word Embedding (TWE)

بردارهای کلمات استفاده شده‌است. استفاده از متوسط بردارهای کلمات برای بازنمایی اسناد نمی‌تواند به‌خوبی نشان‌دهنده خصوصیات معنایی یک متن باشد.

روش دیگری که به سبب استفاده از اطلاعات اسناد به‌روش ارائه‌شده شباهت دارد روش LTTR[1] است. نتایج نشان‌می‌دهد که روش ارائه‌شده نسبت به این روش دارای دقت بالاتری است. در روش LTTR، هر موضوع، عنوان یک مخلوط گوسی از کلمات است و احتمال یک کلمه در یک سند با مجموع وزن‌دار مؤلفه‌های گوسی آن کلمه (یعنی عنوان موضوع‌های آن کلمه) به‌دست می‌آید. هر وزنی سهم یک عنوان موضوع در سند را نشان‌می‌دهد. LTTR نسبت به TWE وزن بالاتری دارد چون موضوع‌های مختلفی را در هر سند در نظر می‌گیرد ولی استفاده از توزیع گوسی برای موضوعات و مخلوط گوسی برای اسناد، شاید ساده‌ترین روش باشد ولی لزوماً فرض درستی نیست و می‌تواند دقت مدل را تحت‌تأثیر خود قرار دهد.

روش Paragraph Vector فقط از هم‌رخدادی‌های محلی برای تولید بردار اسناد استفاده می‌کند؛ بنابراین این روش در دسته‌بندی متن ضعیف‌تر از روش ارائه‌شده عمل کرده است. در سایر روش‌ها مثل روش‌هایی که فقط از اطلاعات موضوعی استفاده می‌کنند، مثل LDA، LSA روش ارائه‌شده بهتر عمل کرده است. در روش‌های شبکه عصبی با وجود اینکه ترتیب بین کلمات را در نظر می‌گیرند؛ ولی از اطلاعات موضوعی استفاده نمی‌کنند؛ بنابراین روش ارائه‌شده از آنها بهتر عمل کرده است و دارای دقت بالاتری است.

یک سری از روش‌ها مثل استفاده از میانگین بردارهای کلمات برای بازنمایی سند نیز نمی‌توانند نشان‌دهنده معنای سند باشند؛ و علاوه‌بر روش ارائه‌شده از روش‌های LSA، LDA، Paragraph-Vector و LTTR نیز دارای دقت پایین‌تری هستند.

۱-۴-۴- بررسی اثر تعداد موضوعات بر روی دقت

در این بخش اثر تعداد موضوع‌ها بر روی دقت روش مورد ارزیابی قرار گرفته است. ارزیابی‌ها بر روی پیکره ارزیابی که ۱۰ درصد از آموزش را تشکیل می‌دهد انجام شده‌است. تعداد موضوع‌ها با توجه به تعداد رده‌های موجود در مجموعه داده انتخاب‌شده‌اند. تعداد موضوع‌ها برای پیکره 20-Newsgroups برابر با تعداد رده‌ها، کمتر از تعداد رده‌ها و بیشتر از تعداد رده‌ها قرار گرفته‌اند، ولی برای پیکره R8 با توجه به اینکه تعداد رده‌ها ۸ است و مقدار کمتر از آن بسیار کوچک می‌شود تعداد موضوع‌ها برابر با تعداد رده‌ها و بیشتر از تعداد رده‌ها قرار داده‌شده‌است. نتایج برای پیکره‌های R8 و 20-Newsgroups در جداول (۶) و (۷) نشان داده‌شده‌اند. علاوه‌بر دقت، روش‌های مختلف به‌وسیلهٔ معیار F1-Measure نیز با یکدیگر مقایسه شده‌اند. این آزمایش به‌دلیل تعیین بهترین مقدار برای تعداد موضوعات الگوریتم LDA انجام شده‌است و همان طور که مشاهده می‌شود، تغییر تعداد موضوعات بر روی دقت تأثیر چندانی ندارد و روش نسبت به تغییر آنها مقاوم است. روش پیشنهادی به‌دلیل استفاده از منظم‌سازی نرم ۲ که فاکتورهای هموارتری را در اختیار ما قرار می‌دهد و همچنین از بیش‌برازش جلوگیری می‌کند توانسته به‌دقت بالاتری نسبت به‌روش Tens-embedding دست یابد.

(جدول- ۶): بررسی اثر تعداد موضوع‌ها بر روی پیکره R8
(Table-6): The effects of the number of topics on R8

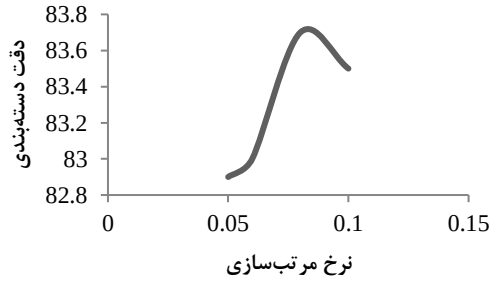
تعداد موضوع‌ها	دقت	F1-score
۸	۹۶.۷۲	۹۷.۰۰
۱۶	۹۵.۹	۹۶.۰۰
۲۴	۹۶.۳	۹۶.۰۰

(جدول- ۷): بررسی اثر تعداد موضوعات بر روی پیکره 20-Newsgroups
(Table-7): The effects of the number of topics on 20-Newsgroups

تعداد موضوع‌ها	دقت	F1-score
۲۰	۸۲.۶	۸۲.۰۰
۵۰	۸۲.۷	۸۲.۰۰
۸۰	۸۲.۷	۸۲.۰۰

۱-۵-۴- عیین نرخ منظم‌سازی

برای تعیین نرخ منظم‌سازی یا همان پارامتر λ از پیکره ارزیابی 20NG استفاده شد و دقت به‌دست آمد. نتایج در شکل(۲) نشان داده‌شده‌است.



(شکل- ۲): اثر نرخ منظم‌سازی بر روی دقت روی داده‌های ارزیابی
(Figure-1): The effect of regularization rate on the accuracy

[1] Latent topic text representation (LTTR)

تقدیر و سپاس

بدین‌وسیله از صندوق حمایت از پژوهش‌گران و فناوران کشور برای حمایت مادی و معنوی از این پژوهش به شماره ۹۷۰۰۹۳۰۸ سپاس‌گزاری به عمل می‌آوریم.

۶- نتیجه‌گیری و کارهای آینده

در این مقاله روشی برای بازنمایی اسناد مطرح شد که هم از اطلاعات هم‌رخدادی کلمات و اسناد و هم از اطلاعات هم‌رخدادی اسناد و موضوعات استفاده می‌کند. برای ترکیب این اطلاعات از تنسور استفاده و برای تولید بردارهای سند از تجزیه تنسور فاکتورهای موازی کمک گرفته شد. نتایج به‌دست آمده نشان داد که روش مطرح‌شده به‌خوبی قادر است برای دسته‌بندی متن عمل کند و بسیار بهتر از روش‌های پایه عمل می‌کند. برای کارهای آینده در نظر داریم که از روش‌های دیگر تجزیه تنسور مثل روش تاکر و اطلاعات دیگری که می‌توان از متن استخراج نمود مثل اطلاعات بار احساسی استفاده‌کرد.

۷- مراجع 7- Refrece

[1] M. Fu, H. Qu, L. Huang, and L. Lu, “Bag of meta-words: A novel method to represent document for the sentiment classification, ” *Expert Syst. Appl.*, vol. 113, pp. 33–43, 2018.

[2] R. Zhao and K. Mao, “Fuzzy Bag-of-Words Model for Document Representation, ” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 794–804, 2018.

[3] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. 1987.

[4] M. A. M. Garcia, R. P. Rodriguez, M. V. Ferro, and L. A. Rifon, “Wikipedia-Based Hybrid Document Representation for Textual News Classification,” *2016 3rd Int. Conf. Soft Comput. Mach. Intell.*, no. November, pp. 148–153, 2016.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information, ” *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2016.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space, ” in *International estimation on learning representations: Workshop Track*, 2013, pp. 1–12.

[7] R. Collobert and J. Weston, “A unified architecture for natural language processing, ” pp. 160–167, 2008.

[8] P. Li, K. Mao, Y. Xu, Q. Li, and J. Zhang, “Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base, ” *Knowledge-Based Syst.*, vol. 193, no. xxxx, 2020.

[9] H. K. Kim, H. Kim, and S. Cho, “Bag-of-concepts: Comprehending document representation through clustering words in distributed representation, ” *Neurocomputing*, vol. 266, pp. 336–352, 2017.

[10] M. Kamkarhaghighi and M. Makrehchi, “Content Tree Word Embedding for document representation, ” *Expert Syst. Appl.*, vol. 90, pp. 241–249, 2017.

[11] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, “Knowledge-enhanced document embeddings for text classification, ” *Knowledge-Based Syst.*, vol. 163, pp. 955–971, 2019.

[12] J. Camacho-Collados and M. T. Pilehvar, “From word to sense embeddings: A survey on vector representations of meaning, ” *J. Artif. Intell. Res.*, vol. 63, pp. 743–788, 2018.

[13] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment Embeddings with Applications to Sentiment Analysis, ” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, 2016.

[14] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, “Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification, ” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2014, vol. 1, pp. 1555–1565.

[15] Q. Le and T. Mikolov, “Distributed representations of sentences and documents, ” in *International conference on machine learning*, 2014, vol. 32, pp. 1188–1196.

[16] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” 2014.

[17] G. Rao, W. Huang, Z. Feng, and Q. Cong, “LSTM with sentence representations for document-level sentiment classification, ” *Neurocomputing*, vol. 308, no. May, pp. 49–57, 2018.

[18] W. Etaiwi and A. Awajan, “Graph-based Arabic text semantic representation, ” *Inf. Process. Manag.*, vol. 57, no. 3, p. 102183, 2020.

[19] L. Yao, C. Mao, and Y. Luo, “graph convolutional networks for text classification,” 2018.

[20] K. Bijari, H. Zare, E. Kebriaei, and H. Veisi, “Leveraging deep graph-based text representation for sentiment polarity applications, ” *Expert Syst. Appl.*, vol. 144, 2020.

[21] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, “Improve Word Representation via Global Context and Multiple Word Prototypes, ” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012, no. July, pp. 873–882.

[22] J. Pennington, R. Socher, and C. Manning, “‘Glove: Global Vectors for Word Representation, ’” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[23] C. Chemudugunta, P. Smyth, and M. Steyvers, “Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model, ” *Adv. Neural Inf. Process. Syst.*, pp. 241–248, 2007.

[24] T. G. Kolda and B. W. Bader, “Tensor Decompositions and Applications, ” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[25] Z. Rahimi and M. M. Homayounpour, “Tens-embedding: A Tensor-based document embedding method, ” *Expert Syst. Appl.*, vol. 162, p. 113770, 2020.
[26] R. Lakshmi and S. Baskar, “Novel term weighting schemes for document representation based on ranking of terms and Fuzzy logic with semantic relationship of terms, ” *Expert Syst. Appl.*, vol. 137, pp. 493–503, 2019.
[27] S. Deerwester, S. T. Dumias, G. W.Furmas, T. K.Lander, and R. Harshman, “Indexing by Latent Semantic Analysis,” *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[28] T. Hofmann, “probabilistic latent semantic analysis, ” in *Hofmann, Thomas. “Probabilistic latent semantic analysis.” Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 289–296.
[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation, ” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[30] B. Jiang, Z. Li, H. Chen, S. Member, and A. G. Cohn, “Latent Topic Text Representation Learning on Statistical Manifolds, ” *IEEE Trans. Neural Networks Learn. Syst. 29*, pp. 5643–5654, 2018.
[31] R. Das, M. Zaheer, and C. Dyer, “Gaussian LDA for Topic Models with Word Embeddings, ” *Proc. 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process.*, pp. 795–804, 2015.
[32] P. Liu, X. Qiu, and X. Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning, ” *Proc. Twenty-Fifth Int. Jt. Conf. Artif. Intelligen*, pp. 2873–2879, 2016.
[33] T. N.Kipf and M. Welling, “Semi-Supervised classification with Graph Convolusional Networks,” *Iclr*, pp. 1–11, 2017.
[34] C. Wu, F. Wu, T. Qi, X. Cui, and Y. Huang, “Attentive Pooling with Learnable Norms for Text Representation, ” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2961–2970.
[35] Í. C. Dourado, R. Galante, M. A. Gonçalves, and R. da Silva Torres, “Bag of textual graphs (BoTG): A general graph-based text representation model, ” *J. Assoc. Inf. Sci. Technol.*, no. April, 2019.
[36] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, “A Flexible and Efficient Algorithmic Framework for Constrained Matrix and Tensor Factorization, ” *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5052–5065, 2016.
[37] S. Smith, J. Park, and G. Karypis, “SPLATT: Efficient and Parallel Sparse Tensor-Matrix Multiplication Sparse Tensor Factorization on Many-Core Processors with High-Bandwidth Memory, ” no. May, 2015.
[38] Y. Liu, Z. Liu, T. Chua, and M. Sun, “Topical Word Embedding, ” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Topical*, 2015, pp. 2418–2424.

زهرا رحیمی در حال حاضر، دانشجوی دکترای مهندسی رایانه در دانشگاه صنعتی امیرکبیر است. ایشان مدرک کارشناسی‌ارشد خود را در گرایش هوش‌مصنوعی در همان دانشگاه در سال ۱۳۹۲ اخذ کرد و تحصیلات کارشناسی خود را در گرایش نرم‌افزار در دانشگاه صنعتی شاهرود در سال ۱۳۸۴ به پایان رسانید. زمینه‌های پژوهشی مورد علاقه ایشان، پردازش زبان طبیعی، یادگیری ماشین و داده کاوی می‌باشد.
نشانی رایانامه ایشان عبارت است از:
zah-ra@aut.ac.ir

محمد مهدی همایون‌پور تحصیلات خود در مقطع کارشناسی را در رشته مهندسی برق (الکترونیک) در دانشگاه صنعتی امیرکبیر (سال ۱۳۶۶)، کارشناسی‌ارشد را در رشته برق (مخابرات)، از دانشگاه خواجه نصیرالدین طوسی (سال ۱۳۶۹)، کارشناسی‌ارشد دوم خود را در زمینه فونتیک (۱۳۷۴) در دانشگاه سوربون جدید در فرانسه و هم‌زمان دوره دکترای خود را در دانشگاه پاریس ۱۱ در زمینه مهندسی برق (۱۳۷۴) به پایان رسانید. ایشان از سال ۱۳۷۴ در سمت عضو هیئت علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر به تدریس و پژوهش مشغول است. زمینه‌های تخصصی مورد علاقه ایشان شامل پردازش سیگنال‌های دیجیتال، پردازش گفتار، پردازش زبان طبیعی، یادگیری ماشین، یادگیری عمیق، اتوماسیون صنعتی و طراحی سخت‌افزار است.
نشانی رایانامه ایشان عبارت است از:
homayoun@aut.ac.ir