



**دانشگاه اصفهان**

**دانشکده مهندسی کامپیوتر**

**گروه هوش مصنوعی و رباتیک**

**پایان نامه کارشناسی ارشد رشته‌‌ی مهندسی کامپیوتر**

**گرایش هوش مصنوعی و رباتیک**

خلاصه‌سازی استخراجی متن با استفاده از رویکرد‌های یادگیری عمیق[[1]](#footnote-1)

**استاد راهنما:**

**دکتر حمیدرضا برادران کاشانی**

**استاد مشاور:**

**دکتر افسانه فاطمی**

**دانشجو:**

**امیررضا صدیقین**

**شهریور 1402**



**تعهدنامه اصالت اثر**

اینجانب **امیررضا صدیقین** دانشجوی مقطع کارشناسی ارشد رشته مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک متعهد مى‌شوم که مطالب مندرج در این پایان نامه و بروندادهاى حاصل از آن، دستاورد پژوهشی اینجانب با اشراف و راهنمايىِ استاد **دکتر حمیدرضا برادران کاشانی** است و آن دسته از مطالب اين پایان نامه که حاصل تحقیقات دیگران بوده نیز طبق شيوه‌نامه‌هاى مصوّبِ ارجاع، مستند شده و در فهرست منابع و مآخذ اين پژوهش آمده است. ضمنا اظهار مى‌دارم که اين پایان نامه پيش‌تر برای دریافت هیچ مدرک تحصیلی ارائه نشده است. بديهى است دانشگاه اصفهان برای خود اين حق را قائل است که در صورت احراز و اثبات هر گونه تخلف در اين باره، مدرک تحصیلی اینجانب را از درجه اعتبار ساقط نماید و ضمن درج موضوع در جراید کثیر الانتشار، کليه امتیازات و حقوقی را که به موجب آن پس از دوران تحصیل، از آنها بهره‌مند گشته‌ام، از اینجانب سلب و به طرف ذى‌نفع مسترد کند.

|  |
| --- |
| ـ برای رساله‌هايى که با حمایت جزيىِ مراکز برون‌دانشگاهى تدوين شده است، عبارت زير تکمیل شود**:**  **اين پایان نامه در دانشگاه اصفهان و با حمایت انجام شده است.** |

|  |
| --- |
| ـ برای رساله‌هايى که طی یک قرارداد مشخص، تحت حمایت سازمان یا نهادی تدوين شده است، عبارت زير تکمیل شود**:**  **اين پایان نامه در دانشگاه اصفهان و با حمایت طی قرارداد شماره انجام شده است.** |

|  |
| --- |
| ـ برای همه رساله‌های تدوین شده در دانشگاه اصفهان عبارت زير درج شود:  **کليه حقوق مادی و معنوی مترتب بر دستاوردهای مطالعات و نوآورى‌های ناشی از پژوهش در اين پایان نامه در چارچوب آيين‌نامه مالکیت فکری و تجاری‌سازىِ دانشگاه تعيين مى‌شود.** |

**نام و نام خانوادگى دانشجو: امضاء**

**نام و نام خانوادگى استاد (/ استادان) راهنما: امضاء**

حوزه معاونت پژوهش و فناوری

**دانشگاه اصفهان**

**دانشکده مهندسی کامپیوتر**

**گروه هوش مصنوعی و رباتیک**

پايان‌نامه آقای

**امیررضا صدیقین**

دانشجوی رشته‌ی مهندسی کامپیوتر گرایش هوش مصنوعی و رباتیک

**عنوان پایان نامه**

به عنوان بخشي از ملزومات درجه کارشناسي ارشد

در تاريخ 01/01/1402 توسط هيات داوران زير بررسي و با درجه ............ به تصويب نهايي رسيد.

1. استاد راهنمای پایان‌نامه دکتر نام استاد راهنما با مرتبه‌ی علمی مرتبه استاد راهنما از دانشگاه/موسسه نام-دانشگاه-یا-موسسه امضا
2. استاد مشاور پایان‌نامه دکتر نام استاد مشاور با مرتبه‌ی علمی مرتبه استاد مشاور از دانشگاه/موسسه نام-دانشگاه-یا-موسسه امضا
3. استاد داور داخل گروه دکتر نام داور داخلی با مرتبه‌ی علمی مرتبه علمی داور داخلی از دانشگاه/موسسه نام-دانشگاه-یا-موسسه امضا
4. استاد داور خارج از گروه دکتر نام داور خارجی با مرتبه‌ی علمی مرتبه علمی داور خارجی از دانشگاه/موسسه نام-دانشگاه-یا-موسسه امضا

مهر و امضای مدیر گروه

**سپاس ‌گزاری**

سپاس از اساتید بزرگوارم که بدون کمک آنها امکان حرکت در این مسیر وجود نداشت.

سپاس از همه افرادی که در این مسیر من را یاری نمودند.

**تقدیم به**

تقدیم به پدر و مادر عزیزم که من را در این مسیر همراهی کرده اند.

**چکیده**

متن چکیده فارسی

**کلیدواژه‌ها:** کلیدواژه‌های فارسی

# 

# فصل اول بیان مسئله‌ی پژوهشی

## مقدمه

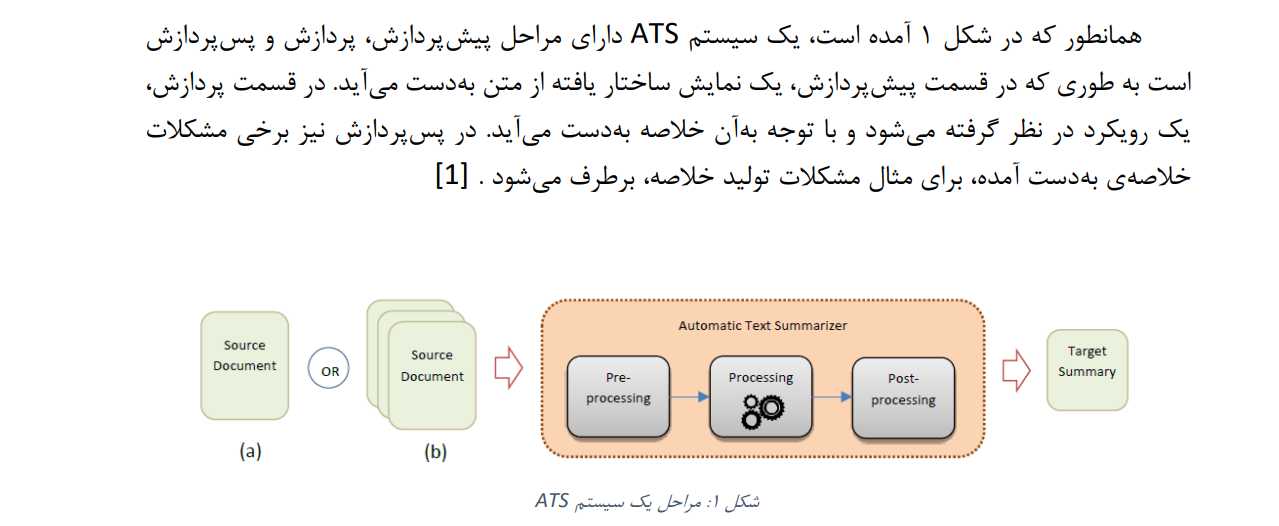
در دهه‌‌های گذشته، نحوه‌‌ی ذخیره و توزیع اطلاعات به‌ صورت چشمگیری تغییر پیدا کرده است. در حالی‌ که کتابخانه‌ها مسئولیت نگهداشت تعداد زیادی کتاب چاپی بوده است، امروزه بخش‌ زیادی از محتوای متنی به‌ صورت الکترونیکی در دسترس‌ هستند [1]. در جهان امروز، روزانه منابع متنی گسترده‌ای با سرعت بالا در حال تولید هستند، این منابع در تنوع و حجم بالا، مخزن غنی از اطلاعات می‌باشند [2]؛ با این حال، این توسعه یک چالش را به همراه دارد؛ مقادیر متنی موجود برای هر موضوع معین، معمولاً آنقدر زیاد است که یک فرد بتواند در یک زمان معقول آن را پردازش کند. به عنوان مثال، نسخه‌ی انگلیسی ویکی‌پدیا در سپتامبر 2018، حاوی 6 میلیون مقاله و سند بود. پروژه‌ی Google Books بیش از 25 میلیون نسخه کتاب را تا سال 2015 دیجیتالی کرده بود. همچنین تعداد کل صفحات شاخص گذاری شده، در سپتامبر 2018 برابر 4/4 میلیارد برآورد شده است [3]. فرآیند خلاصه‌سازی دستی متن بدون شک راهی موثر برای حفظ معنا و مفهوم متن است. با این حال، خلاصه‌سازی دستی متن، یک فعالیت وقت‌گیر و پرهزینه می‌باشد [4]. به همین منظور، مسئله‌ای در حوزه‌ی پردازش زبان‌های طبیعی[[2]](#footnote-2)، به عنوان خلاصه‌سازی خودکار متن[[3]](#footnote-3) (ATS) به وجود آمده است.

خلاصه سازی متن یک فرآیند مهم در پردازش زبان طبیعی است که در آن سعی می‌شود اطلاعات کلیدی و مهم متن اصلی به طور خلاصه و مختصری باقی بماند. این فرآیند به افزایش دسترسی به اطلاعات برای مخاطبین، صرفه‌جویی در زمان و انرژی، و همچنین ایجاد یک نمای کلی از متن‌های بزرگتر کمک می‌کند. یکی از رویکرد های خلاصه سازی خودکار متن، خلاصه سازی استخراجی متن می باشد که در آن جملات مهم متن استخراج می‌شوند. اگر متن دارای ساختار منظم و جملات معینی باشد، این رویکرد می‌تواند بهترین انتخاب باشد.

شبکه‌های عصبی به علت قدرت پردازشی و توانایی فهم و تحلیل داده‌های پیچیده از اهمیت بالایی در خلاصه سازی متن برخوردار هستند. با استفاده از شبکه‌های عصبی می‌توان اطلاعاتی را از متن‌ها استخراج کرد و یا جملات جدیدی را برای تشکیل خلاصه ایجاد کرد. این شبکه‌ها می‌توانند از مدل‌های زبانی پیشرفته مانند ترنسفرمر[[4]](#footnote-4) استفاده کنند که قدرت بسیار بالایی در فهم زبان و تولید محتوا دارند [5].

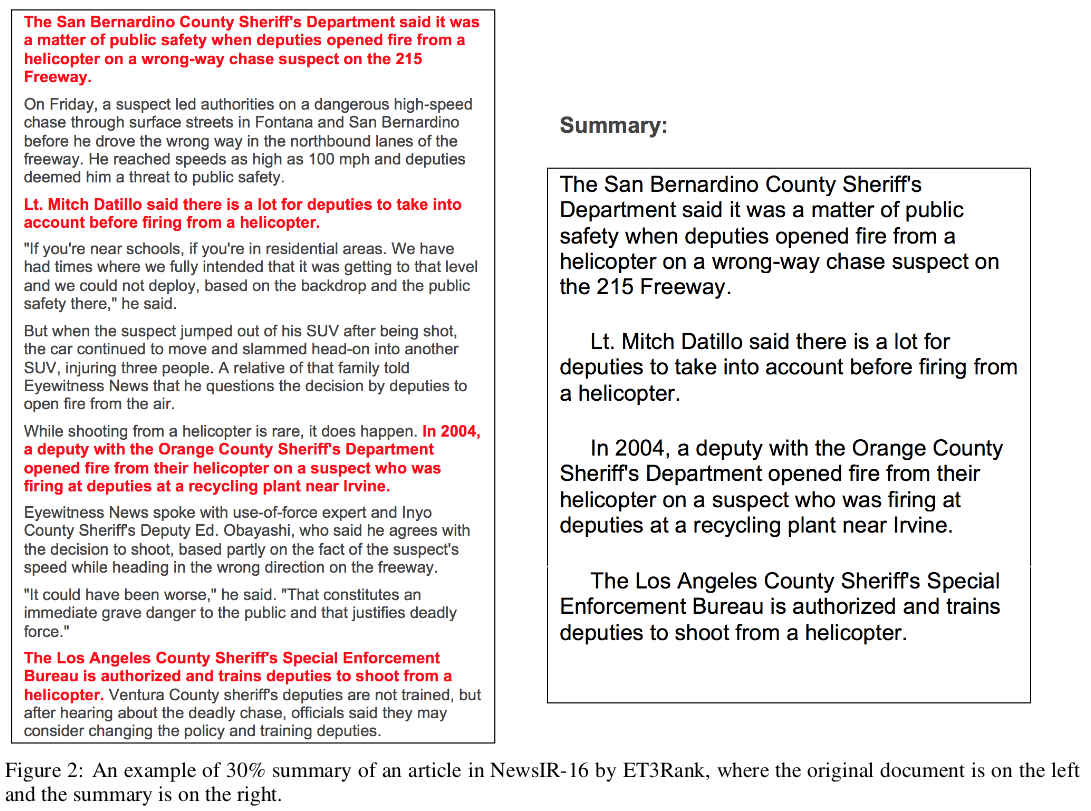
## 2-1- کلیات پژوهش

هدف سیستم‌های ATS، ساخت خلاصه‌هایی حاوی اطلاعات مهم و مرتبط با سند مربوطه، در زمان کوتاه و هزینه‌ی کم می‌باشند [3]. عمده‌ی کاربردهای سیستم‌های ATS در سیستم‌های بازیابی اطلاعات، استخراج اطلاعات و جواب‌دهی به سوالات است. برای مثال در سیستم‌های بازیابی اطلاعات، از ATS برای تقویت موتورهای جستجو استفاده می‌شود. منابع متنی می‌تواند در حوزه‌های مختلفی باشد، برای همین می‌توان کاربردهای متنوعی برای سیستم‌های ATS در نظر گرفت که خلاصه‌سازی اخبار، نظرات، کتاب، داستان، ایمیل، مقالات علمی، توییت‌ها و اطلاعات شبکه‌های اجتماعی، از موارد آن‌ها می‌باشند. همانطور که در شکل 1 آمده است، یک سیستم ATS دارای مراحل پیش‌پردازش[[5]](#footnote-5)، پردازش و پس‌پردازش[[6]](#footnote-6) متن است. در قسمت پیش‌پردازش متن، یک نمایش ساختار یافته از متن به دست می‌آید [2]. از عملیات موجود در این بخش می‌توان به برچسب‌گذاری POS[[7]](#footnote-7)، جداسازی کلمات توقف[[8]](#footnote-8)، هرس کردن[[9]](#footnote-9) و محاسبه‌ی ویژگی‌ها با رویکرد‌های مختلف، اشاره کرد. در قسمت پردازش، یک رویکرد خلاصه‌سازی در نظر گرفته می‌شود و با توجه به آن خلاصه به‌دست می‌آید [4]. در پس‌پردازش نیز برخی مشکلات خلاصه‌ی به‌دست آمده، برای مثال مشکل عدم تشخیص مرجع ضمایر یا مرجع زمان‌های نسبی، بر طرف می‌شود [2]. انتخاب ویژگی‌های مناسب یکی از مهم‌ترین گام‌های موجود در تولید یک مدل ATS می‌باشد. شباهت کلمات یک جمله با عنوان[[10]](#footnote-10)، موقعیت جملات در متن، ویژگی‌های TF-IDF[[11]](#footnote-11)، طول جملات، کلمات موضوعی[[12]](#footnote-12)(کلمات مرتبط با دامنه و دارای میزان تکرار زیاد)، اسامی خاص، شباهت بین جمله‌ای و اعداد، ویژگی‌های مهمی هستند که در یک سیستم ATS مورد توجه قرار می‌گیرند [4].

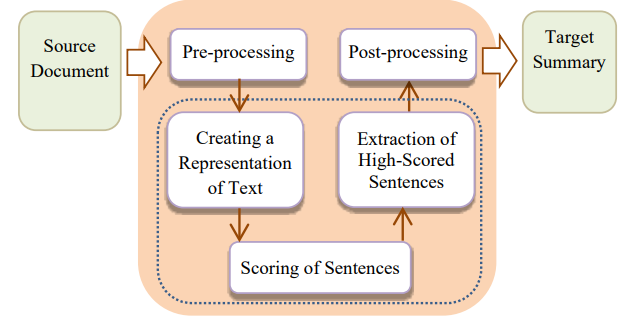


شکل 1: نمای کلی از مراحل یک سیستم ATS [4].

سیستم‌های ATS را از زاویه‌های متنوع دسته‌بندی کرد. از لحاظ تعداد سند، این سیستم‌ها به دو دسته‌ی تک‌سندی و چند‌سندی تقسیم می‌شوند. همچنین برپایه‌ی رویکردهای موجود در این نوع سیستم‌ها، سه رویکرد استخراجی[[13]](#footnote-13)، انتزاعی یا اسنتتاجی[[14]](#footnote-14) و ترکیبی[[15]](#footnote-15) وجود دارد . در رویکرد استخراجی، جملات خلاصه عیناً از جملات متن مرجع استخراج می‌شوند. در رویکرد انتزاعی، سعی در درک متن مرجع و ساخت خلاصه‌ی منسجم نزدیک به خلاصه‌های انسانی است. در رویکرد ترکیبی نیز، ابتدا با رویکرد استخراجی، جملات از متن ورودی استخراج، سپس با استفاده از این جملات، یک خلاصه‌ی منسجم تولید می‌شود. در ادامه به بررسی دقیق‌تر رویکرد‌های حوزه‌ی ATS، پرداخته خواهد شد [2]. شکل ؟؟ نمونه‌ای از خلاصه‌سازی متن با رویکرد استخراجی است. همچنین شکل ؟؟ نمای کلی از مراحل یک سیستم ATS با رویکرد استخراجی را نشان می‌دهد. در این مقاله به رویکرد استخراجی در خلاصه‌سازی متن‌های تک‌سندی پرداخته شده است.



شکل 2: نمونه‌ای از خلاصه‌سازی خودکار متن با رویکرد استخراجی [6].



شکل 3: نمای کلی از معماری سیستم‌های ATS با رویکرد استخراجی [2].

برای ساخت یک سیستم ATS چالش‌های زیادی وجود دارد، شناسایی بخش‌های مهم متن، خلاصه‌سازی سند‌های بزرگ مثل کتاب، خلاصه‌سازی هم‌زمان چند سند در قالب یک سیستم، ارزیابی سیستم و ساخت خلاصه شبیه به خلاصه‌های انسان از جمله این چالش‌ها هستند [2]. در ادامه به برخی از چالش‌های اصلی در رویکرد استخراجی پرداخته خواهد شد.

* **مشکل افزونگی**[[16]](#footnote-16)[3]

افزونگی باعث کاهش کیفیت یک خلاصه می‌شود. وظیفة خلاصه‌ی یک متن علاوه بر دور انداختن مطالب زائد، انتقال محتوای اصلی متن نیز می‌باشد. اگر مطالب تکراری را حذف کنیم، می‌توانیم اطلاعات مهم بیشتری در خلاصه جای دهیم. یکی از روش‌های جلوگیری از مشکل افزونگی، استفاده از اندازه‌گیری‌های شباهت بین جملات است .

* **مشکل بی‌ربط بودن**[[17]](#footnote-17)[3]

هدف اصلی یک سیستم‌ ATS، استخراج مطالب مرتبط با سند در یک نمای سریع و کلی، است. برخی از ویژگی‌های مورداستفاده در خلاصه‌سازی، ممکن است تمایل به ایجاد محتوای فرعی یا نامربوط در خلاصه داشته باشند. ازاین‌رو، بسیار مهم است که بدانیم کدام ویژگی‌ها و با چه میزان نسبت توجه به آن‌ها، برای ایجاد خلاصه باکیفیت مناسب هستند.

* **مشکل پوشش ندادن تمام موضوعات اصلی**[[18]](#footnote-18)[3]

پوشش تمام موضوعات اصلی سند در خلاصه، یک موضوع اساسی در سیستم‌های ATS است. یک خلاصه عمومی خوب باید اطلاعات مربوط به هر جنبه از سند را ذکر کند. بیشتر تکنیک‌های خلاصه‌سازی کنونی، تضمین پوشش تمام موضوعات اصلی را به کاربر نمی‌دهند. این مشکل بیشتر در خلاصه‌سازی چند سندی به دلیل تنوع و پراکندگی بیشتر موضوعات، قابل‌مشاهده هستند.

* **مشکل ناخوانایی و انسجام کم محتوا**[[19]](#footnote-19)[3]

یک خلاصه خوب باید خوانا و منسجم باشد. منظور از خواندنی و منسجم این است که مطالب خلاصه باید از نظر مفهومی با یکدیگر مرتبط و پیوستگی داشته باشند. بیشتر روش‌های استخراجی، فاقد این ویژگی هستند.

* **چالش درنظرگرفتن روابط بین جمله‌ای با فاصله زیاد از هم**[[20]](#footnote-20)[7]

جملات موجود در یک سند، هرچند با فاصله از هم ممکن است روابط و وابستگی‌های زیادی با هم داشته باشند. بیشتر مدل‌های پیشنهادی ارائه شده، توجه زیادی به روابط بین جمله‌ای ندارند. مدل‌های مبتنی بر شبکه‌های عصبی بازگشتی، تا حدی این مورد را در نظر می‌گیرند. با این‌حال مدل‌های مبتنی بر شبکه‌های عصبی بازگشتی، نمی‌توانند به‌خوبی روابط بین جملات با فاصله‌ی زیاد از هم را در نظر بگیرند.

تاثیر شبکه‌های عصبی بر روی خلاصه سازی خودکار متن بسیار قابل توجه است. این شبکه‌ها با بهره‌گیری از معماری‌های عمیق و مدل‌های توجه، قدرتمندترین ابزارها برای انجام خلاصه سازی متن هستند. با پیشرفت تکنولوژی شبکه‌های عصبی و استفاده از مجموعه داده‌های بزرگ، می‌توان به دقت و کیفیت بالاتری در خلاصه‌های تولید شده دست یافت. همچنین، توانایی‌های ترنسفورمر و شبکه‌های بازگشتی[[21]](#footnote-21)(RNN) در فهم ارتباطات طولانی‌تر متن و اجزای آن، موجب بهبود چشم‌گیری در تولید خلاصه‌های معنادار و مرتبط با محتوای متن اصلی شده است. از طرفی، شبکه‌های عصبی می‌توانند با تعمیم‌دهی و یادگیری از داده‌های متنوع و چندزبانه، بهبود مسائل ترجمه و خلاصه‌سازی متن را در زبان‌های مختلف ممکن سازند. به‌طور خلاصه، تحولات در حوزه شبکه‌های عصبی، از اهمیت بسیاری برای ارتقاء و بهبود روش‌های خلاصه سازی خودکار متن برخوردار است [8]. همچنین تاثیر گراف و شبکه‌های عصبی گراف بر روی خلاصه سازی خودکار متن از جمله موضوعات پرطرفدار در حوزه پردازش زبان طبیعی است. در این رویکرد، اطلاعات متنی به صورت گراف ترسیم می‌شوند که اجزا و ارتباطات میان واژه‌ها را نشان می‌دهد. این گراف‌ها می‌توانند با استفاده از روش‌های مبتنی بر شبکه‌های عصبی گراف، مورد استفاده قرار گیرند تا خلاصه‌های متنی با کیفیت و منطبق با محتوای اصلی ایجاد شوند. شبکه‌های عصبی گراف با توجه به ارتباطات بین واژه‌ها و ساختار معنایی متن، می‌توانند بهبود قابل توجهی در فهم و تولید خلاصه‌های معنادار داشته باشند [9], [10].

## 3-1- سوالات پژوهش

در طی پژوهش انجام شده به سوالاتی در زمینه‌ی خلاصه‌سازی خودکار متن با رویکرد استخراجی پاسخ داده شده است. در بخش نتایج و پیشینه پژوهش به میزان تاثیر شبکه‌های عصبی، به خصوص شبکه‌های عصبی گراف و میزان تاثیر ساخت گراف در مسئله‌ی ATS پرداخته شده است. همچنین این بخش‌ها، شامل مباحث دیگر از قبیل ویژگی‌ها و پیش‌پردازش‌های مناسب برای مسئله‌ی ATS می‌شوند.

## 4-1- اهداف پژوهش

با وجود روابط و متغیر‌های گوناگون در یک متن، هدف این پژوهش ارائه‌ی مدلی با رویکرد شبکه‌های عصبی جهت مدل‌سازی مناسب روابط و اطلاعات مسئله‌ی خلاصه‌سازی خودکار متن با رویکرد استخراجی می‌باشد. در این پژوهش به بررسی معماری‌های مختلف شبکه‌ی عصبی، تاثیر ساختار‌های گرافی بر روی نتایج و ویژگی‌های مناسب برای این مسئله پرداخته شده است.

## 5-1- روش ارائه شده

؟؟؟

## 6-1- سازماندهی مطالب

مطالبی که در فصول آینده آمده است به قرار زیر است:

* فصل دوم به ادبیات موضوع و توضیح مفاهیم پایه برای استفاده در این پژوهش پرداخته است. در این بخش مواردی همچون مفاهیم شبکه‌ی عصبی، سنجه‌ها و معیار‌های مناسب در مسئله‌ی ATS و همچنین مباحث ریاضی و گرافی در این زمینه وجود دارند.
* در فصل سوم کار‌های پیشین در این حوزه و چالش‌های آن‌ها بررسی شده است. برخی تحقیقات و مدل‌های مبتنی بر شبکه‌عصبی که در حوزه‌ی ATS موجود است، در این بخش ذکر شده است.
* فصل چهارم شامل توضیح روش ارائه شده و بررسی جوانب مختلف آن می‌باشد. در این بخش برخی روش‌های آزمایش شده به همراه بررسی آن‌ها وجود دارند.
* در فصل پنجم پنجم نتایج آزمایشهای روش ‌های ارائه شده بر روی مجموعه داده مختلف بررسی شده و نتایج مورد تحلیل و بررسی قرار گرفته است.
* در نهایت در فصل ششم نتیجه‌گیری نهایی و جمع‌بندی پژوهش ارائه و پیشنهاداتی برای ادامه‌ی پژوهش داده شده است.

# فصل دوم ادبیات موضوع

# 2-1- مقدمه

در حل مسائل خلاصه‌سازی متن، از ساختارها و روش‌های مختلفی استفاده می‌شود. در اکثر روش‌های نوین، یادگیری عمیق و ساختارهای آن به عنوان چهارچوب اصلی مورد توجه قرار می‌گیرند. شبکه‌های عصبی پیچشی (CNN)[[22]](#footnote-22)و شبکه‌های عصبی بازگشتی (RNN)[[23]](#footnote-23) از جمله ابزارهای مهم در زمینه‌های پردازش زبان طبیعی[[24]](#footnote-24) و متن‌کاوی[[25]](#footnote-25) به شمار می‌روند. همچنین، با ظهور شبکه‌های عصبی گراف[[26]](#footnote-26)، تحولات چشمگیری در حوزه یادگیری عمیق[[27]](#footnote-27) رخ داده است.

در این فصل، به بررسی برخی از ساختارهای داده‌ای اساسی، مفاهیم یادگیری ماشین، و نیز شبکه‌های عصبی و برخی تکنیک‌های آن ر پردازش زبان طبیعی پرداخته شده است. همچنین، معیارهای ارزیابی مدل‌ها در زمینه خلاصه‌سازی متن مورد بحث قرار گرفته‌اند. این تشریحات به ما کمک خواهد کرد تا در فهم بهتری از تکنیک‌ها و مفاهیم مرتبط با حوزه پژوهشمان به دست آید.

# 2-2- یادگیری ماشین[[28]](#footnote-28)

یادگیری ماشین به صورت عام به دسته‌ای از الگوریتم‌ها و مدل‌های ریاضی اطلاق می‌شود که از داده‌ها به صورت خودکار الگوها و ارتباطات را استخراج کرده و از آنها برای اتخاذ تصمیمات در آینده استفاده می‌کنند. این زمینه را می‌توان به سه دسته‌ی یادگیری نظارت‌شده، یادگیری بدون نظارت و یادگیری نیمه نظارتی تقسیم کرد. در یادگیری نظارت شده، مدل‌ها با استفاده از داده‌های برچسب‌خورده آموزش می‌بینند و سپس برای پیش‌بینی برچسب یا مقدار جدید از این دسته داده‌ها استفاده می‌شود. در یادگیری بدون نظارت، مدل بدون داده‌های برچسب‌خورده آموزش می‌بیند و به دنبال الگوها و ساختارهای نهفته در داده‌ها می‌گردد. همچنین یادگیری نیمه نظارت شده، ترکیبی از دو حالت قبلی است که هم داده‌های برچسب‌خورده و هم بدون برچسب را برای آموزش مدل استفاده می‌کند. معروف‌ترین مسائل در حوزه‌ی یادگیری ماشین نیز، مسائل دسته‌بندی، خوشه‌بندی و رگرسیون می‌باشند. هدف از مسائل دسته‌بندی، پیشبینی برچسب یا دسته برای داده‌ی ورودی و هدف از مسائل خوشه‌بندی، گروه‌بندی داده‌ها بدون دانستن برچسب‌های آن‌ها و بر اساس ویژگی‌های مشترک آن‌ها است. همچنین مسائل رگرسیون، سعی بر پیش‌بینی مقدار یک عدد یا متغیر پیوسته دارند. [11], [12]. در دسته‌بندی با یادگیری ماشین، می‌توان از روش‌های متعددی برای ایجاد مدل طبقه‌بندی استفاده کرد. یکی از روش‌های معروف این زمینه ‌الگوریتم SVM‌[[29]](#footnote-29) است. این روش با تعیین یک فضای جداکننده بین دسته‌های مختلف، سعی در بهینه‌سازی فاصله‌ی حاشیه بین داده‌های مختلف دارد [13]. از الگوریتم‌های مهم در حوزه‌ی دسته‌بندی می‌توان به الگوریتم k نزدیک‌ترین همسایه (KNN)[[30]](#footnote-30)، درخت تصمیم[[31]](#footnote-31) ، جنگل تصادفی[[32]](#footnote-32) و Naïve Bayes نام برد. همچنین می‌توان از روش‌های خوشه‌بندی مثل الگوریتم k-means به منظور پیدا کردن k خوشه بر اساس معیار‌های شباهت بین نقاط، برای خوشه‌بندی و شباهت سنجی بین اسناد استفاده کرد.

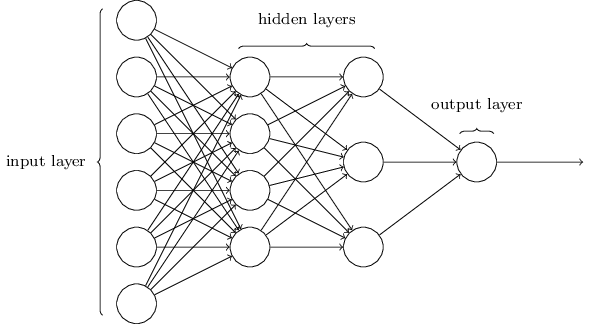
یادگیری تقویتی یکی دیگر از رویکرد‌های یادگیری ماشین می‌باشد. در این رویکرد، مدل‌ها با یک محیط تعاملی در ارتباط هستند و از بازخورد‌ها برای بهبود عملکرد خود استفاده می‌کنند. این روش مختص به مواردی است که عملکرد مدل مستقیماً به دنباله اقداماتی که انجام می‌دهد، وابسته است. برای مثال، در پردازش زبان طبیعی، مسائلی مانند ترجمه ماشینی یا چت‌بات‌ها از یادگیری تقویتی بهره می‌برند. مدل با انجام اقدامات مختلف، مانند تولید جملات یا پاسخ به سوالات، از تجربیات خود یاد می‌گیرد. سپس از بازخورد‌ها و پاداش‌ها برای بهبود کارایی و تصحیح عملکرد استفاده می‌کند [14].

# 3-2- یادگیری عمیق

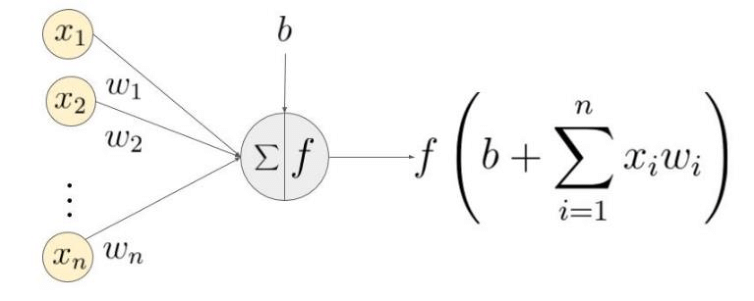
یادگیری عمیق یک حوزه پویا و حیاتی در علم داده[[33]](#footnote-33) و یادگیری ماشین است که با استفاده از شبکه‌های عصبی عمیق سعی دارد نمایش‌های پیچیده داده‌ها به منظور استخراج ویژگی‌های نهان بدست آورد. شبکه‌های عصبی عمیق به عنوان مدل‌های محاسباتی با تعداد لایه‌های بسیار زیاد شناخته می‌شوند. این روش‌ها با الهام از ساختار مغز انسان و اقتباس از عملکرد نورون‌ها، مفاهیمی چون یادگیری نمایش‌ها، تشخیص الگوها و پیش‌بینی‌های دقیق را فراهم می‌کنند. شبکه‌های عصبی عمیق به عنوان ابزاری کارآمد در بسیاری از حوزه‌ها از جمله پردازش تصویر، پردازش زبان‌های طبیعی، پردازش صورت و حتی بازیابی اطلاعات، به کار گرفته می‌شوند [12].

## 1-3-2- شبکه‌ی عصبی پرسپترون چند لایه[[34]](#footnote-34) و مفاهیم اولیه یادگیری عمیق

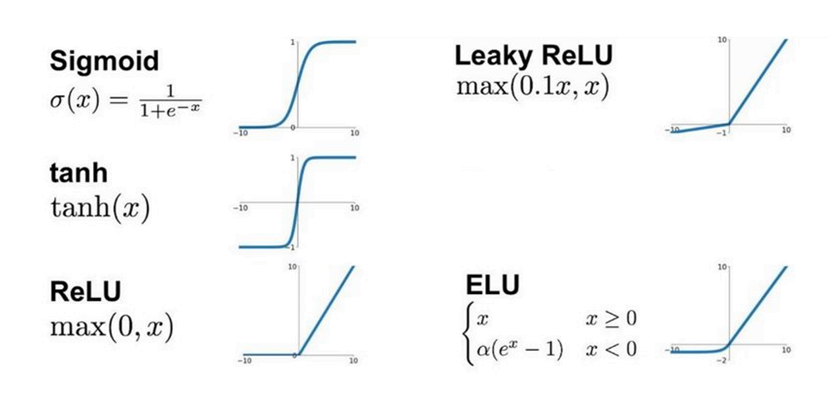
شبکه عصبی چندلایه یکی از مدل‌های اولیه و مهم در زمینه یادگیری عمیق است. این شبکه به عنوان یک نمونه از شبکه‌های عصبی عمیق، از چند لایه از نورون‌ها (واحدهای پردازشگر) تشکیل شده است. همانطور که در شکل ؟؟ آمده است، در این نوع شبکه‌ها، از سه بخش اصلی به نام لایه‌ی ورودی[[35]](#footnote-35)، لایه‌های پنهان[[36]](#footnote-36)، و لایه خروجی[[37]](#footnote-37) تشکیل شده است. در لایه ورودی، ویژگی‌های ورودی به شبکه وارد می‌شوند. لایه‌های پنهان با انجام عملیات‌های خطی و توابع غیرخطی[[38]](#footnote-38) یا فعالسازی[[39]](#footnote-39) بر روی ورودی‌ها، اطلاعات را انتزاع می‌کنند و ویژگی‌های پنهان را استخراج می‌کنند. در نهایت، لایه خروجی نتایج پایانی را تولید می‌کند. این شبکه‌ها از یک معماری کاملاً متصل [[40]](#footnote-40)هستند، به این معنی که هر نورون در یک لایه با تمام نورون‌های لایه قبل و بعد از خود ارتباط دارد. این ویژگی باعث تغییر ابعاد داده و استخراج ویژگی‌های پیچیده‌تر می‌شود. برخی مواقع داده‌ها به صورت خطی[[41]](#footnote-41) تفکیک پذیر نیستند، به همین منظور در برخی از لایه‌ها از توابع غیرخطی یا فعالسازی استفاده می‌کنند. همانطور که در شکل ؟؟ نحوه‌ی محاسبات درون هر نرون به نمایش گذاشته شده، ورودی هر نرون خروجی لایه‌های قبل یا لایه‌ی ورودی شبکهی عصبی می‌باشد. همچنین متناظر با هر مؤلفه ورودی، وزنی در آن ضرب می‌شود که وزن‌ها مقادیر قابل آموزش می‌باشند. در هر نرون یک ورودی با مقدار 1 به همراه وزن آن (بایاس) نیز اضافه می‌شود. در آخر تابع فعالسازی برروی مجموع ضرب ورودی با وزن‌های متناظر اعمال می‌شود. در شکل ؟؟، برخی توابع فعالسازی متداول، معرفی شده‌اند.



شکل 4: ساختار یک شبکه عصبی که دارای سه بخش اصلی لایه‌ی ورودی، لایه مخفی و لایه‌ی خروجی میباشد [15].



شکل 5: نحوه‌ی محاسبه‌ی خروجی هر نرون در شبکه‌ها عصبی. هر نرون از چندین ورودی و وزن به همراه بایاس(مقدار وزن با ورودی 1) تشکیل شده و با استفاده از تابع غیر خطی f ویژگی غیر خطی به خروجح میدهد [16].



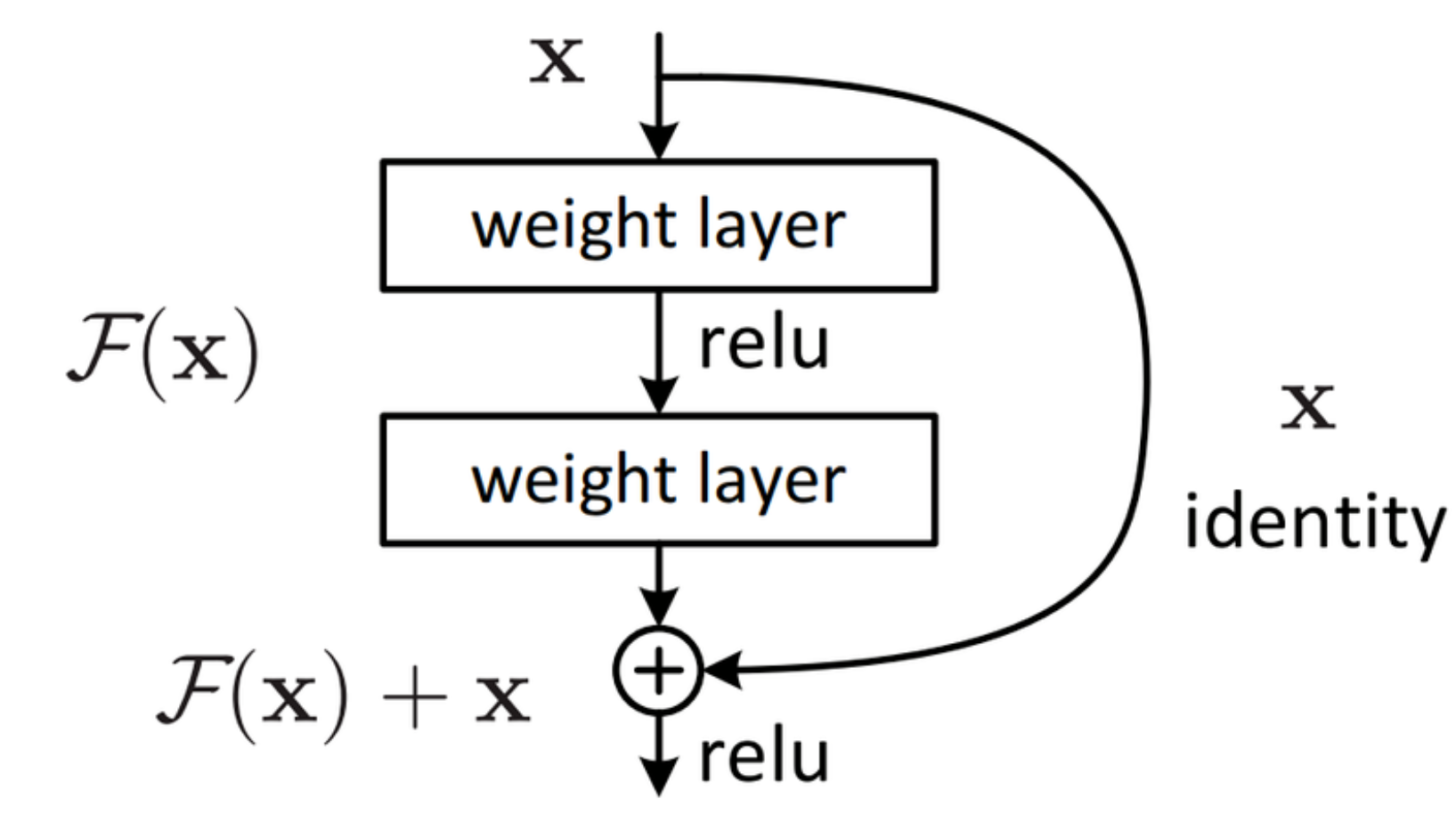
شکل 6: برخی توابع فعالسازی معروف[17]

متناسب با نوع مسئله و الگوریتم‌های بهینه‌سازی، فرایند آموزش شبکه‌ی عصبی شامل دو عنصر مهم تابع خطا[[42]](#footnote-42) و الگوریتم بهینه‌سازی[[43]](#footnote-43) می‌باشد. در هر دور از آموزش، داده‌ها به شبکه‌ی عصبی ارسال شده و متناسب با نوع مسئله، تابع خطا مقادیر خطا را محاسبه می‌کند و با استفاده از الگوریتم‌های بهینه‌سازی که معمولا با محاسبه‌ی مشتق خطا و قابلیت زنجیره‌ای آن همراه است، مقادیر وزن‌ها را بروزرسانی می‌کند و سعی می‌کند مقادیر خطا را به حداقل حالت خود برسند [12]. از الگوریتم‌های بهینه‌سازی معروف می‌توان به الگوریتم‌های SGD، Momentum، Adagrad، Adadelta، RSMProp و Adam اشاره کرد [18]. همچنین جدول ؟؟ برخی توابع خطا را معرفی می‌کند.

|  |  |  |
| --- | --- | --- |
| توضیحات | رابطه‌ی ریاضی | نام تابع |
| میانگین مجموع مجذور تفاضل مقدار واقعی با مقدار پیش‌بینی شده در مسائل رگرسیون. |  | MSE  (mean squared error) |
| میانگین مجموع قدرمطلق تفاضل مقدار واقعی با مقدار پیش‌بینی شده در مسائل رگرسیون. |  | MAE  (mean absolute error) |
| از توابع خطا مرسوم برای مسائل دسته‌بندی دوتایی[[44]](#footnote-44) . |  | Binary Cross Entropy |
| بسط یافته‌ی تابع binary cross entropy است و خطا را برای k کلاس محاسبه می‌کند. |  | Categorical Cross Entropy |

جدول 1: چند تابع خطا معروف در شبکه‌های عصبی[19]

یکی از مشکلات اساسی در آموزش شبکه‌های عصبی، مشکل گم‌شدن گرادیان در شبکه‌های عمیق است. در شبکه‌های عمیق، زمانی که اطلاعات از لایه به لایه برای آموزش منتقل می‌شوند، گرادیان‌ها ممکن است به گونه‌ای کاهش یابند که به سختی یا حتی ناتوانایی در آموزش لایه‌های عمیق‌تر منجر می‌شود. به منظور رفع این مشکل ساختار‌های متنوعی معرفی شده‌اند. یکی از این ساختار‌ها، ساختار Residual است. همانطور که در شکل ؟؟ نشان داده شده است، در ساختار Residual، این مشکل با افزودن یک مسیر مستقیم (یا اتصال مابقی) بین ورودی و خروجی لایه‌ها حل می‌شود. به عبارت دیگر، خروجی لایه جدید با ورودی لایه قبلی جمع می‌شود. این اتصال مستقیم به مدل این امکان را می‌دهد تا اطلاعات را به سرعت از لایه به لایه منتقل کرده و مشکل گم‌شدن گرادیان را کاهش دهد.

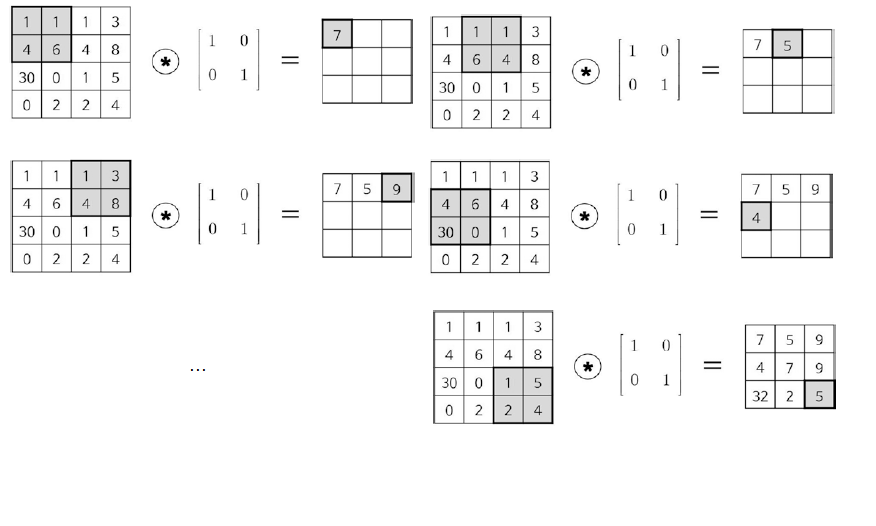


شکل 7: ساختار residual باعث جلوگیری از گم شدن گرادیان می‌کند [20].

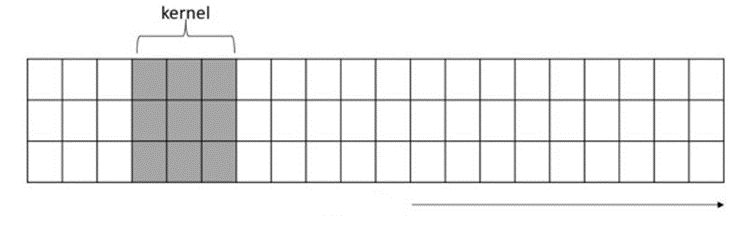
در ادامه به یک سری از شبکه‌های عصبی متداول و روش‌های معروف در این حوزه، پرداخته می‌شود.

## 2-3-2- شبکه‌های عصبی پیچشی[[45]](#footnote-45)

شبکه‌های عصبی پیچشی یا CNNها، نوع دیگری از شبکه‌های عصبی هستند که معمولا برای استخراج ویژگی‌ها و استخراج الگو‌ها استفاده می‌شود. این شبکه‌ها معمولا برای پردازش تصاویر و پردازش داده‌های توالی استفاده می‌شوند. شبکه‌های عصبی پیچشی از لایه‌های پیچشی[[46]](#footnote-46) و لایه‌های ادغام[[47]](#footnote-47) تشکیل شده‌اند. لایه‌های پیچشی به عنوان فیلترها یا کرنل‌ها عمل کرده و با حرکت این فیلترها روی داده‌ها، ویژگی‌های مختلف با محاسبه‌ی عملیات پیچشی[[48]](#footnote-48) از آن استخراج می‌کنند. عملیات پیچشی، حاصل جمع ضرب هر مؤلفه کرنل با مؤلفه نظیرش در داده می‌باشد. لایه‌های ادغام به وسیله‌ی روش‌های تجمیع[[49]](#footnote-49)، با کاهش ابعاد داده و حذف اطلاعات غیرضروری، کمک به کاهش تعداد پارامترها و افزایش سرعت آموزش می‌کنند[14] [15]. لایه‌های پیچشی به دو دسته‌ی یک بعدی و دو بعدی تقسیم‌میشوند. کرنل‌ها‌ی لایه‌ی پیچشی دو بعدی در دو جهت حرکت می‌کنند و بیشتر برای پردازش تصویر کاربرد دارند. شکل ؟؟، عملیات پیچشی دو بعدی را نشان می‌دهد. در مقابل، کرنل‌های لایه‌ی پیچشی یک بعدی تنها در یک بعد حرکت می‌کنند و بیشتر برای پردازش‌ داده‌های توالی، مثل پردازش سیگنال یا پردازش متن، استفاده می‌شوند. شکل ؟؟ نحوه‌ی حرکت کرنل پیچشی یک بعدی را نشان می‌دهد. در فرایند آموزش این شبکه‌ها، مؤلفه‌های شبکه بروزرسانی شده و الگو‌های مناسب از داده‌ها استخراج می‌شود.



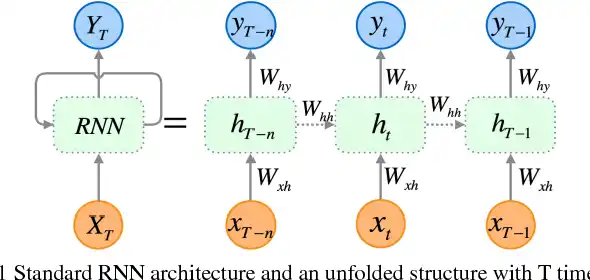
شکل 8 : مراحل بدست آوردن خروجی از اعمال عملیات پیچش یک فیلتر بر روی یک ماتریس ورودی [23] .



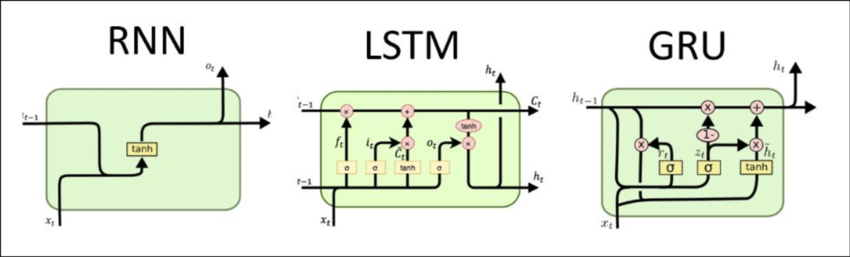
شکل 9: نحوه‌ی حرکت کرنل پیچشی یک بعدی[24]

## 3-3-2- شبکه‌های عصبی بازگشتی [[50]](#footnote-50)

شبکه‌های عصبی بازگشتی یا RNN‌ها یک نوع از معماری‌های شبکه‌های عصبی هستند که برای پردازش داده‌های توالی، مانند داده‌های زمانی یا پردازش کلمات در جمله، استفاده می‌شوند. همانطور که در شکل ؟؟ نمایش داده شده است، این نوع از شبکه‌های عصبی از ویژگی حافظه‌ی بازگشتی[[51]](#footnote-51) برخوردار هستند که این امکان را فراهم می‌سازد تا اطلاعات از گذشته در فرایند آموزش و پیش‌بینی مورد استفاده قرار گیرد. این شبکه‌ها، در مسائل مختلفی مانند ترجمه ماشینی[[52]](#footnote-52)، تشخیص گفتار[[53]](#footnote-53)، پیش‌بینی متن[[54]](#footnote-54)، و حتی تولید متون[[55]](#footnote-55) جدید مؤثر هستند. محو شدن اطلاعات داده‌های اولیه و مشکل حذف گرادیان از مشکلات جدی این نوع معماری می‌باشد، با این حال با تغییرات و اصلاح طراحی‌های موجود در این شبکه‌ها، بهبود‌های چشمگیری در مدیریت این چالش‌ها ایجاد شده است. LSTM‌[[56]](#footnote-56) و GRU‌[[57]](#footnote-57) که در به نمایش درآمده‌اند، از متداول‌ترین شبکه‌های عصبی بازگشتی می‌باشند که در زمینه‌ی این چالش‌ها بهینه شده‌اند [25]. شکل ؟؟، معماری‌های GRU و LSTM و تفاوت آن‌ها با RNNهای اولیه را نمایش می‌دهد.



شکل 10: نحوه‌ی عملکرد شبکه‌های عصبی بازگشتی[26] .

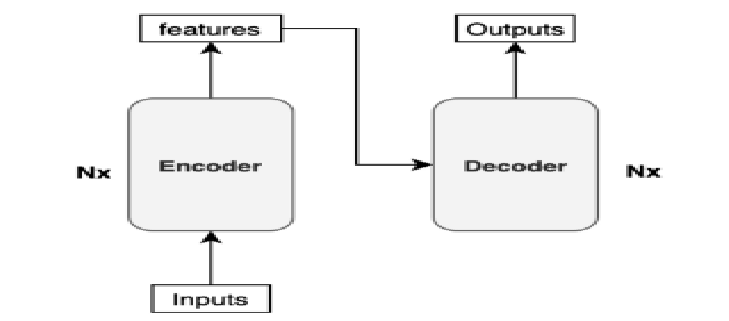


شکل 11: شبکه‌های GRU و LSTM و تفاوت آن‌ها با شبکه‌های RNN اولیه [27].

به منظور مدل‌سازی بهتر داده‌های دنباله‌ای، می‌توان از RNNهای دوطرفه[[58]](#footnote-58) (Bi-RNNها) استفاده نمود. استفاده از Bi-RNN‌ها می‌تواند در بهبود دقت مدل‌ها در وظایف پردازش زبان طبیعی کمک کند، به خصوص زمانی که وابستگی‌ها و اطلاعات دوطرفه مهم هستند. این شبکه‌ها به خوبی درک ارتباطات دوسویه در داده‌ها و بازیابی اطلاعات موثر از هر دو جهت دنباله را فراهم می‌کنند. به عنوان مثال، در تشخیص اسامی یا موجودیت‌ها در یک متن، Bi-RNNs ممکن است نقش مهمی ایفا کنند [28].

## 4-3-2- ساختار کدگذار-کدگشا [[59]](#footnote-59)

ساختار‌های مختلفی برای حل مسائل متفاوت بر بستر یادگیری عمیق ارائه شده‌اند. یکی از این ساختار‌ها، ساختار کدگذار-کدگشا است. ساختار کدگذار-کدگشا یک معماری متداول در حوزه شبکه‌های عصبی است که برای مسائل مختلفی از جمله ترجمه ماشینی، تولید متن، و تولید تصویر[[60]](#footnote-60) مورد استفاده قرار می‌گیرد. کدگذار مسئول تبدیل ورودی به یک فضانی نهان[[61]](#footnote-61) یا بردار ویژگی است. این فضای نهان معمولاً اطلاعات مهم و تجمعی از ویژگی‌های ورودی را نمایندگی می‌کند. سپس کدگشا با استفاده از این نمایش نهان، وظیفه بازسازی و یا تولید خروجی مورد نظر را برعهده دارد [29]. شکل ؟؟، ساختار کلی کدگذار-کدگشا را نشان می‌دهد.

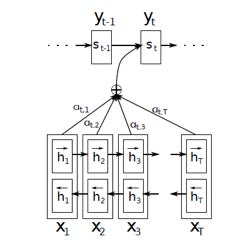


شکل 12: ساختار کدگذار-کدگشا [30]

## 5-3-2- روش‌های مبتنی بر توجه[[62]](#footnote-62) و معماری ترنسفرمر[[63]](#footnote-63)

ترنسفرمر یک معماری مدل عمیق یادگیری ماشین است که به عنوان یکی از پیشرفت‌های قابل توجه در زمینه پردازش زبان طبیعی و وظایف مرتبط با آن شناخته می‌شود و معمولاً به عنوان یک انقلاب در مدل‌های زبانی و توانایی پردازش اطلاعات دنباله‌ای شناخته می‌شود. یکی از ویژگی‌های اصلی ترنسفرمر، استفاده از ساختار توجه است که به مدل این امکان را می‌دهد تا به مرور دنباله ورودی خود، توجه متفاوتی به بخش‌های مختلف از داده داشته باشد. این ساختار توجه، توانمندی مدل را در درک و استخراج اطلاعات مهم از داده‌های ورودی افزایش می‌دهد. ترنسفرمر به دلیل قابلیت پردازش دنباله‌های طولانی، امکان توجه به ارتباطات دور و ابعاد بالا، به خصوص در وظایفی که وابستگی‌های طولانی دارند مانند ترجمه ماشینی و تولید خودکار متن، مورد توجه قرار گرفته است [5].

همانطور که گفته شده، ساختار توجه، اصلی‌ترین ویژگی معماری ترنسفرمر‌ها می‌باشد. همه‌ی قسمت‌های یک دنباله، در خروجی به یک اندازه اهمیت ندارند. روش‎‌های مبتنی بر توجه در مسائلی مثل خلاصه‌سازی متن، فهم متون، دسته‌بندی متون کاربرد فراوانی دارند [31]. مکانیزم توجه یک مفهوم کلیدی در زمینه یادگیری عمیق و پردازش زبان طبیعی است که به شبکه‌های عصبی اجازه می‌دهد تا به نقاط مهم یا ویژگی‌های مشخصی در ورودی خود تمرکز کنند. برای مثال، در پردازش زبان طبیعی، مکانیزم توجه به مدل‌ها امکان تمرکز بیشتر بر روی کلمات مهم یا جملات کلیدی را می‌دهد. این توانایی می‌تواند به بهبود کارایی مدل در ترجمه ماشینی، پرسش و پاسخ و وظایف دیگر کمک کند. این مفهوم ابتدا در سال 2015، در معماری ترنسفرمر معرفی شد و از آن زمان به عنوان یک عنصر اساسی در بسیاری از مدل‌های پیشرفته عصبی مورد استفاده قرار گرفته است [5]. شکل ؟؟ ساختار توجه در فرایند بدست آوردن نمایش بهتر برای دنباله را نشان می‌دهد.



شکل 13: نمایش مکانیزم توجه برای بدست آوردن نمایش برای یک دنباله[32]

در هر سلول توجه[[64]](#footnote-64)، سه مفهوم کلید[[65]](#footnote-65)، مقدار[[66]](#footnote-66) و پرسش[[67]](#footnote-67) وجود دارد. در هر لایه با استفاده از یک معیار شباهت، میزان شباهت بین کلید و پرسش محاسبه می‌شود. خروجی هر عنصر متناظر با ورودی به صورت معادله‌های (؟؟) محاسبه می‌شود [5].

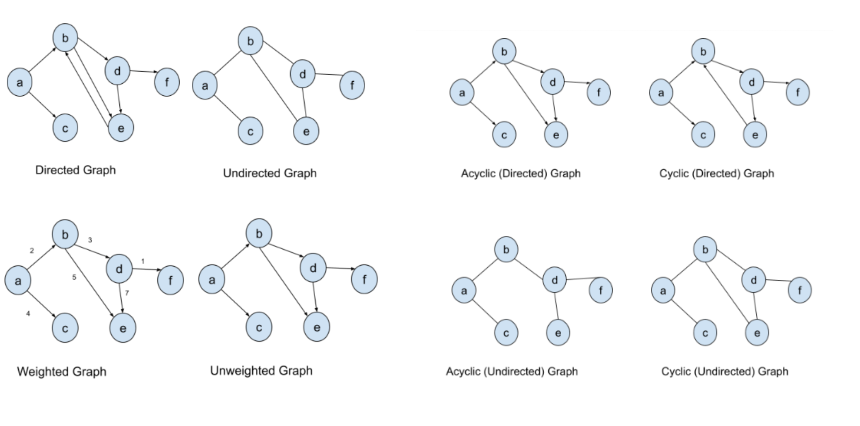
در عبارات ریاضی بالا منظور از q پرسش، k کلید و v مقدار می‌باشد. تابع SIM، یک تابع شباهت می‌باشد. معادله‌ی 2 نیز نحوه‌ی محاسبه‌ی SoftMax است که در آن ضریب اهمیت، محاسبه می‌شود. یکی از روش‌های محاسبه‌ی شباهت می‌تواند ضرب داخلی مقیاس‌بندی شده[[68]](#footnote-68) باشد. معادله (؟؟) نحوه‌ی محاسبه‌‌ی ضرب داخلی مقیاس‌ بندی شده را نشان می‌دهد [5].

در عبارت ریاضی بالا منظور از Q و K دو بردار برای محاسبه‌ی شباهت هستند. d بعد بردار‌های Q و K می‌باشد.

استفاده همزمان و موازی از چند سلول توجه می‌تواند ویژگی‌های بهتر از دیدگاه‌های مختلف تولید کند. به ساختار استفاده چند سلول توجه به صورت همزمان و موازی، توجه چند سره[[69]](#footnote-69) می‌گویند [5]. همچنین اگر در ساختار توجه، کلید و مقدار یکسان باشند و با استفاده از دنباله‌ی ورودی، سعی در ایجاد نمایش از همان دنباله شود، ساختار خود توجه[[70]](#footnote-70) گویند [33].

## 6-3-2- شبکه‌های عصبی گراف[[71]](#footnote-71)

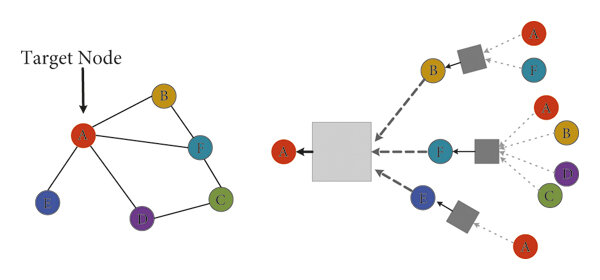
در ریاضیات، گراف را می‌توان به عنوان یک جفت مرتب (V,E) تعریف کرد، که در آن V مجموعه‌ای از گره‌ها[[72]](#footnote-72) (نقاط) و E مجموعه‌ای از یال‌ها[[73]](#footnote-73) (ارتباطات) است. هر یال با دو گره مختلف از V مرتبط می‌شود. این تعریف مشخص می‌کند که گراف چگونه اجزا و ارتباطات بین آنها را نمایش می‌دهد [34]. همچنین گراف‌ها ابزارهای قدرتمندی در زمینه هوش مصنوعی هستند که در تحلیل و نمایش روابط بین اشیاء و اطلاعات مورد استفاده قرار می‌گیرند. گراف‌ها به تدریج به عنوان یک مدل انتزاعی جهت نمایش روابط میان داده‌ها و اجزای سیستم‌ها مورد توجه قرار گرفته‌اند. در حوزه هوش مصنوعی، گراف‌ها به عنوان یک ساختار داده‌ای اساسی مورد استفاده قرار می‌گیرند. این ساختارها قابلیت نمایش روابط پیچیده میان داده‌ها را فراهم می‌کنند و در بسیاری از مسئله‌ها مانند معماری داده[[74]](#footnote-74)، شبکه‌های اجتماعی[[75]](#footnote-75)، پردازش زبان طبیعی، مسائل مسیریابی[[76]](#footnote-76)، و تجزیه و تحلیل شبکه‌های پیچیده[[77]](#footnote-77) کاربرد دارند.[35] همانطور که در شکل ؟؟؟ نشان داده شده است،گراف می‌تواند جهت‌دار[[78]](#footnote-78) یا بی‌جهت[[79]](#footnote-79) ، وزن‌دار[[80]](#footnote-80) یا بی وزن[[81]](#footnote-81)، گراف دوره دار[[82]](#footnote-82) و بدون دور[[83]](#footnote-83) باشد و تبع آن، مفاهیم متنوعی از گراف‌ها ایجاد می‌شود [34]. همچنین اگر گره‌های یک گراف ماهیت یکسان داشته باشند، به آن گراف همگن[[84]](#footnote-84) و اگر ماهیت گره‌ها یا هم یکسان نباشند به آن گراف ناهمگن[[85]](#footnote-85) گفته می‌شود.



شکل 14:گراف‌ها می‌توانند وزن‌دار یا بی وزن، جهت‌دار یا بی‌جهت، دوردار یا بی‌دور باشند[25]

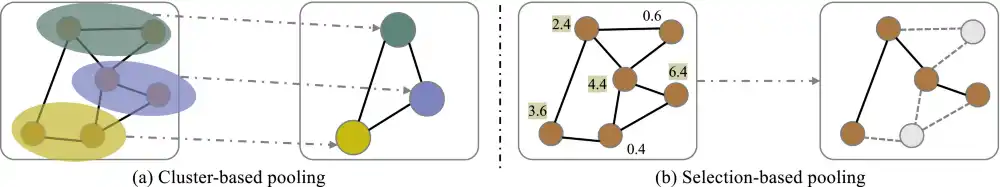
شبکه‌های عصبی گراف یک دسته از مدل‌های یادگیری عمیق هستند که برای مدل‌سازی داده‌های گرافی مانند شبکه‌های اجتماعی، شبکه‌های مولکولی و ساختارهای گرافی مشابه به کار می‌روند. این شبکه‌ها توانمندی بالایی در تفسیر و تحلیل روابط پیچیده و الگوهای شبکه‌های گرافی دارند. به عبارت دیگر، آنها قابلیت مدل‌سازی تعاملات و وابستگی‌های بین گره‌ها و لینک‌های یک گراف را دارا هستند [36].

شبکه‌های عصبی گراف از مکانیزم انتقال پیام[[86]](#footnote-86)، به منظور تبادل اطلاعات بین گره‌ها در یک گراف بهره می‌برند. در این مکانیزم، هر گره اطلاعات مرتبط با خود را با همسایگان خود به اشتراک می‌گذارد و از اطلاعات جمع‌آوری شده برای به‌روزرسانی ویژگی‌های خود استفاده می‌کند. این فرآیند چند مرحله‌ای ادامه می‌یابد تا اطلاعات به صورت تدریجی از یک گره به دیگری منتقل شوند. همچنین به غیر از اطلاعات درون یک گره، میتوان برای یال‌ها نیز اطلاعات و ارتباطاتی[[87]](#footnote-87) نیز در نظر گرفت. مدیریت پیچیدگی مدل‌ها، تعامل با ساختار گراف، انعطاف پذیری در مقابل اندازه‌ی گراف‌ها، آموزش ناهمگن[[88]](#footnote-88) گراف‌ها و مقاومت در برابر نویز از قابلیت‌های شبکه‌های عصبی می‌باشد[36]. یکی از اولین و متداول‌ترین شبکه‌های عصبی گراف، GCN‌ها[[89]](#footnote-89) می‌باشند که در شکل ؟؟، نحوه‌ی عملکرد آن نمایش داده شده است. یکی دیگر از شبکه‌های عصبی گراف معروف، شبکه‌ی عصبی GAT‌[[90]](#footnote-90) می‌باشد که با استفاده از مکانیزم توجه، می‌تواند میزان اهمیت بین گره‌ها در فرایند انتقال پیام را آموزش ببیند.



شکل 15: نحوه‌ی انتقال پیام برای یک گره مشخص در شبکه‌ی GCN [37]

برخی مواقع در شبکه‌های عصبی گراف از لایه‌های ادغامی گراف[[91]](#footnote-91) به منظور به‌دست آوردن گراف کوچک‌تر استفاده می‌شود. همچنین با این روش می‌توان نمایشی برای گراف مورد نظر نیز به‌دست آورد. در شکل ؟؟ یک نمونه از ادغام گره‌های گراف نمایش داده شده است.



شکل 16: ادغام گره‌های گراف و تولید گراف کوچکتر با استفاده از لایه‌ی ادغام گراف [38]

# 4-2- جانشانی کلمه[[92]](#footnote-92)

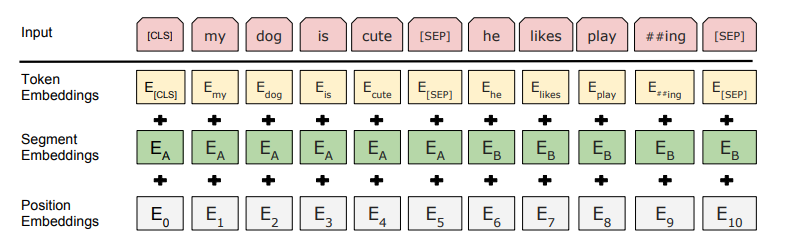
در پردازش زبان طبیعی، برای استفاده از روش‌های یادگیری ماشین، کلمات یا عبارات باید به بردار‌های عددی تبدیل شوند. بنابراین روشی برای تبدیل متن به بردار نیاز است. جانشانی کلمه یک روش در زمینه پردازش زبان طبیعی است که اهمیت زبانی کلمات را در یک فضای چندبعدی عددی نمایش می‌دهد. این تکنیک مفهومی از معانی و ارتباطات بین واژگان را در یک فضای چندبعدی مدل می‌کند. به این ترتیب، واژگان مشابه در این فضا به نزدیکی هم نقل می‌کنند و این اطلاعات معنایی برای الگوریتم‌های پردازش زبان طبیعی بسیار حائز اهمیت است. برای مثال، اگر دو کلمه معنای مشابهی داشته باشند، مطابق با این روش، بردارهای متناظر با این کلمات نیز به هم نزدیک خواهند بود. این امر به الگوریتم‌ها کمک می‌کند تا معانی و ارتباطات میان واژگان را درک و استخراج کنند [39]. از روش‌های معروف در این حوزه می‌توان به word2vec، glove‌[[93]](#footnote-93) و برخی مدل‌های از پیش آموزش دیده شده[[94]](#footnote-94) مثل مدل BERT‌[[95]](#footnote-95) نام برد.

## 1-4-2- جانشانی کلمه Glove

جانشانی کلمه Glove یک روش برای ایجاد بردارهای کلمات در پردازش زبان طبیعی است. این روش توسعه داده شده توسط پژوهشگران گروه گوگل به منظور بهبود نمایش واژگان در فضای چندبعدی می‌باشد. در این روش، بردارهای کلمات بر اساس آمارهای مشاهده شده در متون آموزش دیده می‌شوند. الگوریتم Glove به این صورت عمل می‌کند که بردارهای کلمات را طوری بهینه می‌سازد که معانی و ارتباطات معنایی بین واژگان در فضای چندبعدی به خوبی بازنمایی شوند. این روش تاکید بر توزیع احتمالاتی کلمات در متون و سعی در بازنمایی معانی ضمنی و روابط معنایی میان واژگان را دارد [40]. Glove به عنوان یکی از مدل‌های موفق در زمینه جانشانی کلمه به شناخته شده است و به طور گسترده در پروژه‌های پردازش زبان طبیعی مورد استفاده قرار گرفته است.

## 3-4-2- مدل **BERT**

مدل BERT یکی از مدل‌های برجسته در زمینه پردازش زبان طبیعی است که توسط گوگل منتشر شده است. این مدل بر اساس معماری ترنسفورمر ساخته شده و به دلیل توانایی بازنمایی بردارهای کلمات در متن به صورت دوطرفه[[96]](#footnote-96)، از دقت واژگان بالایی برخوردار است. BERT قابلیت انجام ماموریت‌های مختلف در زمینه پردازش زبان طبیعی، توانایی درک ارتباطات و مفاهیم پیچیده در یک متن را داراست. برای استفاده از BERT، متن ورودی به شکل تکه‌های کوچکتر[[97]](#footnote-97) تقسیم می‌شود و سپس با استفاده از لایه‌های ترنسفورمر، بردارهای بازنمایی متن به دست می‌آیند. این بردارها سپس می‌توانند برای ماموریت‌های متنوعی از جمله تشخیص ماهیت‌های نامدار[[98]](#footnote-98)، استخراج ارتباطات[[99]](#footnote-99)، ترجمه ماشینی، و حتی فهم متن وابسته به سوالات[[100]](#footnote-100) استفاده شوند. همانطور که در شکل ؟؟ آمده است، این مدل از سه جانشانی توکن، بخش و موقعیت، برای بدست آوردن جانشانی کلمه استفاده می‌کند. این معماری و آموزش آن بر روی مقدار زیادی داده باعث شده است تا جانشانی کلمات با استفاده از ویژگی‌های زمینه‌ای[[101]](#footnote-101) استخراج شوند [41].



شکل 17: جانشانی کلمه در مدل BERT با استفاده از سه جانشانی توکی، بخش و موقعیت تولید می‌شود.

# 5-2- معیار‌های شباهت بین بردار‌ها

در یادگیری ماشین، معیار‌های شباهت بین بردار‌ها از اهمیت چشمگیری برخوردارند. این معیارها به تعیین میزان تشابه یا فاصله بین داده‌ها کمک می‌کنند و در مسائل مختلفی از جمله دسته‌بندی، خوشه‌بندی، یادگیری نظارت‌نشده، و بازیابی اطلاعات مورد استفاده قرار می‌گیرند. برای اندازه‌گیری شباهت بین دو بردار، معیارهایی مانند فاصله اقلیدسی[[102]](#footnote-102) (معاله ؟؟) یا شباهت کسینوسی[[103]](#footnote-103) (معادله ؟؟) استفاده می‌شود.این معیارها می‌توانند به مدل‌های یادگیری ماشین کمک کنند تا الگوها و ارتباطات پنهان در داده‌ها را درک کرده و اطلاعات مفیدی را استخراج نمایند.

استفاده از معیار‌های شباهت در یادگیری ماشین به ما این امکان را می‌دهد تا الگوها و ارتباطات پنهان در داده‌ها را بهبود بخشیم و به دقت و کارآیی مدل‌ها افزوده و اطلاعات بیشتری از داده‌ها استخراج کنیم.

(??)

در محاسبه‌ی شباهت کسینوسی از تقسیم ضرب داخلی[[104]](#footnote-104) دو بردار به روی اندازه‌ی هر کدام استفاده می‌شود.

# 5-2- شاخص‌های ارزیابی در مسئله‌ی خلاصه‌سازی متن

یکی از چالش‌های موجود در سیستم‌های خلاصه‌سازی متن، چگونگی تعیین میزان کیفیت یک خلاصه‌ی تولید شده توسط سیستم می‌باشد. به منظور مقایسه‌ی میزان کیفیت و درستی خلاصه‌های تولید شده توسط این سیستم‌‌ها در هر دو رویکرد انتزاعی و استخراجی، پژوهش‌هایی برای پیداکردن نمایش عددی، انجام شده است. معیار‌های ارزیابی یک سیستم خلاصه‌سازی خودکار متن به دو دسته‌ی ذاتی[[105]](#footnote-105) و بیرونی[[106]](#footnote-106) تقسیم می‌شوند. معیار‌های بیرونی میزان پیروی از یک ساختار (بیشتر مواردی مثل مسائل پاسخ‌دهی به سؤالات[[107]](#footnote-107) یا بازیابی اطلاعات[[108]](#footnote-108)) را نمایش می‌دهند. اما معیار‌های ذاتی بر روی انسجام و میزان کیفیت معنایی، تمرکز دارند. معیار‌های ذاتی نیز خود به دو دسته‌ی، معیار‌های ارزیابی کیفیت متن[[109]](#footnote-109) و ارزیابی محتوا[[110]](#footnote-110) تقسیم می‌شوند. در سیستم‌های خلاصه‌سازی متن، معمولاً از معیار‌های ذاتی استفاده می‌شود [42], [43]. در ادامه به برخی از معروف‌ترین و پرکاربردترین معیار‌های ارزیابی در این سیستم‌ها اشاره خواهد شد.

## 1-5-2- معیار صحت یا **Precision**

معیار صحت یا precision تعیین می‌کند که خلاصه‌ی به‌دست‌آمده، به چه میزان مختصر و فاقد کلمات اضافی می‌باشد. میزان این معیار مطابق با رابطه‌ی ؟؟ بدست می‌آید [43].

دراین‌رابطه، نسبت تعداد کلمات مشترک خلاصه‌ی مرجع () و خلاصه‌‌ی حاصل از سیستم ATS () به تعداد کل کلمات خلاصه‌ی بدست آمده، محاسبه می‌شود .

## 2-5-2- معیار **Recall**

معیار Recall میزان پوشش خلاصه‌ی مرجع توسط خلاصه‌ی حاصل شده را تعیین می‌کند. مقدار این معیار با استفاده از رابطه‌ی ؟؟ بدست می‌آید [43].

## 3-5-2- معیار **F-measure**

به منظور ادغام دو معیار Recall و precision، از معیار F-measure استفاده می‌شود. رابطه‌ی ؟؟ مقدار این معیار را به دست می‌آورد [43].

## 3-5-2- معیار**ROUGE‌‌[[111]](#footnote-111)**

معیار ROUGE مرسوم‌ترین معیار ارزیابی در سیستم‌های خلاصه‌سازی خودکار متن است. معیار ROUGE با استفاده از چندتایی‌های واژگان متوالی (N-grams) مشترک بین خلاصه‌ی مرجع و خلاصه‌ی تولید شده، یک مقدار عددی برای کیفیت محتوای خلاصه ارائه می‌دهد. رابطه‌ی ؟؟ میزان عددی ROUGE متناسب با تعداد واژگان متوالی موردنظر را بدست می‌آید [44].

رابطه‌ی فوق مقدار Rouge با رویکرد Recall را نشان می‌دهد که نسبت مجموع n-gramهای مشترک در جملات خلاصه‌ی مرجع و خلاصه‌ی تولید شده () به مجموع n-gramهای مشترک در جملات خلاصه‌ی مرجع () را بدست می‌آورد [43]. همانطور که در رابطه‌ی ؟؟ آمده است، Rouge با رویکرد Perecision محاسبه می‌شود و با توجه به دو معیار Perecision و Recall می‌توان Rouge با معیار F-measure را به صورت معادله‌ی ؟؟ محاسبه کرد.

در مسئله‌ی خلاصه‌سازی خودکار متن 3 معیار R1، R2 و RL مرسوم‌ترین شاخص‌های گزارش شده در مقالات می‌باشند. R1 به معنای محاسبه‌ی ROUGE با n برابر 1 می‌باشد. این شاخص میزان همپوشانی توکن‌های یکتایی[[112]](#footnote-112) بین خلاصه‌ی سیستم و خلاصه‌ی مرجع را نشان می‌دهد. همچنین R2 به معنای ROUGE با n برابر 2 است که نشان‌دهنده‌ی همپوشانی توکن‌های دوتایی[[113]](#footnote-113) بین خلاصه‌ی سیستم و خلاصه‌ی مرجع می‌باشد. RL نیز میزان ROUGE بر اساس طول بزرگترین زیردنباله‌ی مشترک بین دو خلاصه را نشان می‌دهد [43]. برای مثال دو متن "گربه‌ی سیاه روی پنجره نشست." و "کنار پنجره، گربه‌ سیاه نشست." در این مثال کلمات "گربه"، "سیاه"، "پنجره" و "نشست" کلمات مشترک این دو متن می‌باشند. در محاسبه‌ی R1 این چهار کلمه در نظر گرفته می‌شوند. در حالی که تنها عبارت "گربه‌ی سیاه" به عنوان عبارت دوتایی مشترک هستند. در RL نیز توکن‌های یکتا با حفظ ترتیب در نظر گرفته می‌شوند. در این مثال کلمات "گربه"، "سیاه" و "نشست" بزرگ‌ترین زیر دنباله با حفظ ترتیب در دو متن هستند، و برای RL تنها این 3 کلمه در نظر گرفته می‌شوند.

# 9-2- جمع‌بندی

در این فصل به بررسی مفاهیم‌ مورد استفاده در این پژوهش پرداخته شد. ابتدا یادگیری ماشین و یادگیری عمیق و برخی تکنیک‌های آن ها معرفی گردید. یکی از بزرگ‌ترین مزیت‌های یادگیری عمیق استخراج ویژگی‌های نهان داده‌های ورودی می‌باشد. شبکه‌های عصبی معمولا شامل یک لایه‌ی ورودی، لایه‌ی نهان و لایه‌ی خروجی هستند. با استفاده از برخی لایه‌های شبکه‌های عصبی مثل لایه‌های CNN و RNN می‌توان ویژگی‌ها و الگو‌های داده‌های توالی را استخراج کرد. همچنین برخی مسائل را می‌توان با استفاده از گراف مدل کرد و شبکه‌های عصبی گراف باعث می‌شوند تا اطلاعات بین گره‌های مجاور با سیاست‌های مختلف، به اشتراک گذاشته و در حل مسائل استفاده شوند. همچنین یکی از ساختارهای معروف شبکه‌های عصبی عمیق ساختار کدگذار -کدگشا است که یک نمایش میانی توسط کدگذار از دنباله ورودی ایجاد می‌شود و کدگشا این نمایش را به دنباله خروجی تبدیل می‌کند. در ادامه به مکانیزم توجه پرداخته شد. مکانیزم توجه یک مفهوم کلیدی در زمینه یادگیری عمیق و پردازش زبان طبیعی است که به شبکه‌های عصبی اجازه می‌دهد تا به نقاط مهم یا ویژگی‌های مشخصی در ورودی خود تمرکز کنند. همچنین به منظور تبدیل کلمات به بردار‌های عددی از مدل‌های جانشانی کلمات استفاده می‌شود. از روش‌های معروف در این حوزه می‌توان به word2vec، glove‌ و برخی مدل‌های از پیش آموزش دیده شده مثل مدل BERT‌ نام برد. در آخر نیز، شاخص‌های ارزیابی در مسئله‌ی خلاصه‌سازی خودکار متن معرفی گردید. شاخص ROUGE مرسوم‌ترین شاخص ارزیابی این حوزه می‌باشد. ROUGE انواعی دارد که در بین آن‌ها R1، R2 و RL مهم‌ترین آن‌ها می‌باشد و معمولا در پژوهش‌های مربوط به حوزه‌ی خلاصه‌سازی متن گزارش می‌شوند. در فصل آینده به بررسی رویکرد‌ها و پژوهش‌های انجام شده در خلاصه‌سازی خودکار متن پرداخته شده است.

# فصل سوم پیشینه پژوهش

# 1-3- مقدمه

رویکرد استخراجی خلاصه‌سازی متن را می‌توان به‌صورت یک مسئله‌ی دسته‌بندی جملات[[114]](#footnote-114) در نظر گرفت، بدین صورت که باید مدل تشخیص دهد، هر جمله در دسته‌ی خلاصه قرار بگیرد یا خیر[45]. پس هر مدل استخراجی، از یک دسته‌بند[[115]](#footnote-115) برای عملیات دسته‌بندی جملات، استفاده می‌کند. این دسته‌بند‌ها می‌تواند به صورت یک شبکه‌ی عصبی (MLP، LSTM و ...) یا به صورت الگوریتم‌هایی مثل الگوریتم‌های رتبه‌بندی[[116]](#footnote-116) باشد. برای مثال یک الگوریتم رتبه‌بندی می‌تواند از ویژگی‌های آماری، نظیر طول جملات، موقعیت جملات، ویژگی‌های TF-IDF، شباهت‌های کسینوسی و مواردی دیگر، به جملات امتیاز داده و آن‌ها را مرتب سازد [46]. پیش از شبکه‌های عصبی عمیق روش‌هایی با رویکرد یادگیری ماشین یا الگوریتم‌های مبتنی بر منطق فازی وجود داشته است. با رواج پیدا کردن شبکه‌های عصبی، این رویکرد‌ها توانست بهبود‌های بسیاری در زمینه‌ی خلاصه‌سازی متن بدهد. مدل SummRuNNer یکی از اولین مدل‌های مبتنی بر شبکه‌های عصبی در زمینه‌ی خلاصه‌سازی بوده است، که توانسته است پیشرفت قابل توجهی بدست آورد [45], [47]. در سال‌های گذشته، مدل‌های مبتنی بر شبکه‌عصبی زیادی در زمینه‌ی خلاصه‌سازی ارائه شده است. با پیدایش شبکه‌ی عصبی BERT و نمایش مبتنی بر زمینه‌ی متن[[117]](#footnote-117)، بهبود‌های بسیاری در این زمینه رخ داد [45]. یکی از اصلی‌ترین چالش‌های موجود در این حوزه‌، در نظر گرفتن روابط موجودیت‌ها‌ی با فاصله‌ی زیاد است. در این امر، شبکه‌های بازگشتی نتوانسته‌اند موفق ظاهر شوند. به همین منظور، شبکه‌های عصبی گراف و استفاده از گراف در زمینه‌ی خلاصه‌سازی، توانسته‌اند روابط مختلف بین کلمات و جملات و عناصر مختلف متن را در خلاصه‌سازی دخیل کنند. ساخت انواع گراف و تعریف انواع روابط در یک متن به صورت گراف، از دلایل موفقیت شبکه‌های عصبی گراف در این مسائل بوده است [48], [49]. شبکه‌های عصبی گراف به چهار دسته‌ی شبکه‌های عصبی گراف بازگشتی[[118]](#footnote-118)، شبکه‌های پیچشی گراف[[119]](#footnote-119)، خود رمزگذار گراف[[120]](#footnote-120) و شبکه‌های عصبی گراف فضایی-زمانی[[121]](#footnote-121) تقسیم می‌شوند [50]. در گراف، گره‌ها به طور طبیعی توسط همسایگان و اتصالات خود تعریف می‌شوند. وظیفه‌ی شبکه‌ی عصبی گراف این است که با مشاهده‌ی اطلاعات موجود در گره‌های مجاور، بردار ویژگی هر گره را به دست آورد [51].

روش‌های موجود در خلاصه‌سازی استخراجی متن به دو دسته‌ی کلی تقسیم می‌شوند: 1- رویکردهای کلاسیک و 2- رویکردهای مبتنی بر ماشین لرنینگ.

رویکردهای کلاسیک در خلاصه‌سازی استخراجی بر اساس قوانین و الگوریتم‌های ثابت عمل می‌کنند. این روش‌ها به استفاده از معیارهایی مانند فراوانی واژگان، ساختار متن، و خصوصیات دیگر متن برای انتخاب جملات مهم متن می‌پردازند. به عنوان مثال، روش‌هایی مبتنی بر محاسبه TF-IDF از تکرار واژگان و اهمیت آن‌ها در متن استفاده می‌کنند.

روش‌های مبتنی بر ماشین لرنینگ از مدل‌های آموزش دیده بر داده‌های آموزشی برای تشخیص جملات مهم و استخراج خلاصه استفاده می‌کنند. این مدل‌ها می‌توانند از شبکه‌های عصبی بازگشتی (RNN)، شبکه‌های عصبی ترنسفرمری (Transformer)، یا روش‌های دیگر بر اساس ماشین لرنینگ استفاده کنند. به عنوان مثال، مدل‌های مبتنی بر ترنسفر مانند BERT می‌توانند با در نظر گرفتن روابط بلندمدت و اطلاعات متن در تصمیم‌گیری برای استخراج جملات کلیدی بهبود بخشند.

بر اساس ادعاهای موجود در مطالعات، مدل‌های مبتنی بر ماشین لرنینگ اغلب دقت و کارآیی بهتری در خلاصه‌سازی استخراجی نسبت به رویکردهای سنتی[[122]](#footnote-122) ارائه کرده‌اند [47]. این مدل‌ها می‌توانند با توجه به مجموعه‌های داده بزرگ و گسترده، الگوهای پیچیده‌تری را در متون شناسایی کنند و در نتیجه خلاصه‌های دقیق‌تری ایجاد کنند.

در این فصل، به بررسی و مقایسه‌ی معماری مدل‌های استخراجی معروف برپایه‌ی شبکه‌های عصبی در سال‌های اخیر پرداخته می‌شود. در فصل‌های بعد این مدل‌ها به همراه مدل پیشنهادی، با هم مقایسه خواهند شد.

# 2-3- مدل‌های مبتنی بر روش‌های سنتی

در حوزه خلاصه‌سازی استخراجی متن، روش‌های سنتی به استفاده از الگوریتم‌ها و قوانین قابل تعریف برای انتخاب جملات کلیدی از متون بلند می‌پردازند. این روش‌ها بر اساس معیارهایی مانند فراوانی واژگان، ساختار متن، یا خصوصیات دیگر متن، جملات مهم را شناسایی و به عنوان جزوه‌ای از خلاصه نهایی انتخاب می‌کنند.

در بیشتر روش‌های سنتی از روش‌های آماری برای خلاصه‌سازی متن استفاده می‌کنند. در روش‌های آماری ابتدایی برای انتخاب محتوا، اغلب از محاسبه فراوانی کلمات در متن برای شناسایی کلمات مهم مربوط به سند استفاده می‌شود. مکانیزم اصلی این روش‌ها اندازه‌گیری فراوانی هر کلمه در یک سند است و فراوانی کلمات با شواهدی اضافی از جمله فراوانی معکوس کلمه در مجموعه کلی اسناد (مثل ویژگی‌های TF-IDF‌[[123]](#footnote-123)) تنظیم می‌شود. این اقدام به افزایش امتیاز کلمات کم‌فراوان در یک مجموعه کلی و تخصیص امتیاز‌های کمتر به کلمات خیلی فراوان در متن کمک می‌کند. این روش‌ها فرض می‌کنند که اهمیت یک مفهوم خاص در یک متن به تعداد بارهایی که مفهوم در سند ذکر شده است، درصد متن بستگی دارد. این در حالی است که هر کلمه متفاوت به یک مفهوم مختلف نظیر می‌شود. با این حال، شمارش حضور مفاهیم در متن به دلیل وجود مترادف‌ها (مثل "سگ" و "کودک سگ") و بیان‌های مشارکتی (مثل "اوباما" و "رئیس‌جمهور") که به اتصال متون کمک می‌کنند، کار ساده‌ای نیست. بعد از شناسایی کلمات کلیدی در سند، جملات حاوی این کلمات می‌توانند با استفاده از روش‌های مختلف امتیازدهی و رتبه‌بندی جملات (مانند ارتباط جمله با تعداد کلمات کلیدی آن) انتخاب شوند. همچنین جایگاه جملات در متن نیز به عنوان نشانه‌ای از اهمیت جمله در نظر گرفته می‌شود. به عنوان مثال، در داستان‌های خبری، پاراگراف اول یا ابتدایی معمولاً شامل اطلاعات اصلی در مورد رویداد گزارش شده در خبر است، در حالی که بخش‌های دیگر متن جزئیات و همچنین اطلاعات زمینه‌ای در مورد رویداد را ارائه می‌دهد. بنابراین، انتخاب جملات از ابتدای متن می‌تواند یک استراتژی مناسب باشد. با این حال، در متون علمی معمولاً مقدمه اطلاعات زمینه‌ای را ارائه می‌دهد، در حالی که توسعه‌های اصلی در نتیجه‌گیری‌ها گزارش می‌شود، بنابراین استراتژی مکانی باید به نوع متنی که قرار است خلاصه شود، سازگار شود [52]. اهمیت جایگاه جملات در متون رسمی به قدری است که مدلی با نام Lead-3 که یک مدل ساده و تنها انتخاب سه جمله‌ی اول به عنوان خلاصه است، توانسته‌ است نتایج خوبی به‌دست آورد [53].یکی دیگر از روش‌های ساده در رویکر‌د‌های سنتی، بررسی اهمیت جملات از طریق مقایسه آن‌ها با عنوان سند خلاصه شده است. به عنوان مثال، یک جمله که حاوی کلمات عنوان باشد، به عنوان مرتبط در نظر گرفته می‌شود و هر چه کلمات عنوان بیشتر باشد، جمله به میزان بیشتری مرتبط خواهد بود [52]. با وجود این که روش‌های جدید با رویکرد‌های یادگیری ماشین و یادگیری عمیق، پیشرفت‌های زیادی در زمینه‌ی خلاصه‌سازی متن بدست آورده‌اند، ولی همچنان روش‌های سنتی نیز مورد استقبال هستند و از آن‌ها در روش‌های جدیدتر نیز استفاده می‌شوند.

# 3-3- مدل‌های مبتنی بر یادگیری ماشین

با ظهور روش‌های یادگیری ماشین، پیشرفت در زمینه‌های مختلف از جمله رگرسیون، خوشه‌بندی، و دسته‌بندی به شدت افزایش یافت. همانطور که پیش‌تر بیان شد، خلاصه‌سازی متن را می‌توان به عنوان یک مسئله دسته‌بندی تعریف کرد. با توجه به حجم بالای داده‌های موجود امروزی و روش‌های موثر جمع‌آوری مجموعه‌دادگان برای خلاصه‌سازی خودکار متن، روش‌های یادگیری ماشین با موفقیت‌های چشمگیری در این زمینه مواجه شده‌اند [54].Top of Form

کانداسامی و همکاران از تکنیک یادگیری ماشین برای دسته‌بندی هرزنامه[[124]](#footnote-124) در توییتر استفاده کردند. در این پژوهش از SVM و الگوریتم Naive Bayes به منظور دسته‌بندی جملات به دو دسته‌ی هرزنامه و غیر هرزنانه استفاده شده است [55]. سیلوا و همکاران نیز، آزمایش‌هایی با روش Naive Bayes روی مجموعه داده CNN-Corpus انجام داده‌اند و در خلاصه‌های استخراجی متن موفق ظاهر شدند [56].

استفاده از روش KNN نیز به عنوان دسته‌بند در این گونه مسائل جایگاه خود را تثبیت کرده است. برای مثال در مقاله "الگوریتم KNN با استفاده از شباهت بین ویژگی‌ها برای خلاصه‌سازی متن" یک نسخه خاص از الگوریتم KNN ارائه می‌دهد که در آن شباهت بین بردار‌های ویژگی و موقعیت مکانی جملات محاسبه می‌شود [57].

فتاح و همکاران نیز یک رویکرد جدید یادگیری ماشین برای بهبود انتخاب محتوا در خلاصه‌ساز خودکار متون چند سنده ارائه داده است. این روش از یک مدل خلاصه‌ساز آموزش‌پذیر استفاده می‌کند که ویژگی‌های متنوعی از جملات و پاراگراف‌ها را در نظر می‌گیرد، از جمله شباهت کلمات، ساختار متن، عبارات کلیدی، و اطلاعات غیرضروری. این ویژگی‌ها ترکیب شده و با استفاده از مدل‌های آماری مثل مدل آنتروپی حداکثر[[125]](#footnote-125)، دسته‌بند Naïve Bayes و ماشین بردار پشتیبان، مدل‌های خلاصه‌سازی ساخته می‌شوند. نهایتاً، این مدل‌ها به یک مدل ترکیبی تبدیل شده و جملات را بر اساس اهمیت رتبه‌بندی می‌کند [58].Top of Form

همچنین با به وجود آمدن روش‌های جدید‌، جانشانی کلمات قوی‌تر و ترکیب آن‌ها با روش‌های یادگیری ماشین ابتدایی، توانست با پیچیدگی محاسباتی کمتر نسبت به آموزش شبکه‌های عصبی، نتایج خوبی را بدست آورد. در سال 2019، میلر و همکاران، توانستند با ترکیب مدل BERT به عنوان جانشانی کلمه و الگوریتم k-means به عنوان مدل خوشه‌بندی، یک نرم‌افزار به منظور خلاصه‌سازی سخنرانی ها تولید کنند و در صنعت به موفقیت‌های چشمگیری برسند [59].

همچنین می‌توان مسئله‌ی خلاصه‌سازی متن را به صورت یک مسئله‌ی بندیت[[126]](#footnote-126) در حوزه‌ی یادگیری تقویتی در نظر گرفت. در این مسئله یک عامل باید تصمیم‌های توزیع‌شده را انتخاب کند تا بهترین پاداش را برای عملکرد خود بدست آورد. این مسأله به یک شکل ساده‌تر از مسائل یادگیری تقویتی مطرح و به عنوان یک مسأله تصمیم‌گیری توزیعی در نظر گرفته می‌شود. در یک مسئله بندیت، عامل با مواجهه با مجموعه‌ای از گزینه‌ها یا "باندها" که هرکدام دارای یک توزیع احتمال مشخصی از پاداش‌ها هستند، باید تصمیم بگیرد که کدام باند را انتخاب کند. هدف این عامل بهینه کردن جمعیت پاداش کلی از طریق انتخاب بهترین باند‌ها با استفاده از تجربیات خود است. در خلاصه‌سازی با این رویکرد، هر سند یک زمینه است و هر زیرمجموعه مرتب‌شده از جملات یک سند یک عمل متفاوت است. در این تحقیق، یک روش نوآورانه برای آموزش شبکه‌های عصبی به منظور انجام خلاصه‌سازی استخراجی روی اسناد تک‌سنده بدون نیاز به برچسب‌های استخراجی تولید شده به شیوه فوق ابتکاری[[127]](#footnote-127) معرفی می‌شود. در این رویکرد استخراج خلاصه را به عنوان یک مسئله بندیت در نظر می‌گیرد. در این سناریو، مدل یک سند را به عنوان ورودی جهت خلاصه‌سازی دریافت می‌کند و باید یک دنباله از جملات را برای درج در خلاصه انتخاب کند. الگوریتم تقویت یادگیری با رویکرد مسأله بندیت برای آموزش مدل به منظور انتخاب دنباله‌های جملاتی که امتیاز ROUGE را بیشینه می‌کنند، به کار گرفته می‌شود [60].

# 4-3- مدل‌های مبتنی بر یادگیری عمیق

تا قبل از ظهور یادگیری عمیق، روش‌های سنتی بر مبنای مدل‌های زبانی و ویژگی‌های انتخاب شده، به طور محدود به پیشرفت‌های محدودی دست یافتند. با این حال، این روش‌ها همچنان مشکلاتی از قبیل توانایی ضعیف در درک معانی عمیق متون و تولید خلاصه با پوشش بهتر متن، داشتند.

در دهه‌های اخیر، با پیشرفت مداوم نظریه و فناوری شبکه‌های عصبی، یادگیری عمیق به عنوان یکی از روش‌های موثر در زمینه‌ی خلاصه‌سازی متن شناخته‌شده است. تقسیم‌بندی‌های متنوعی برای مدل‌های مبتنی بر یادگیری عمیق می‌توان در نظر گرفت. در ادامه به بررسی چند رویکرد و ایده‌های اصلی در این حوزه پرداخته می‌شود.

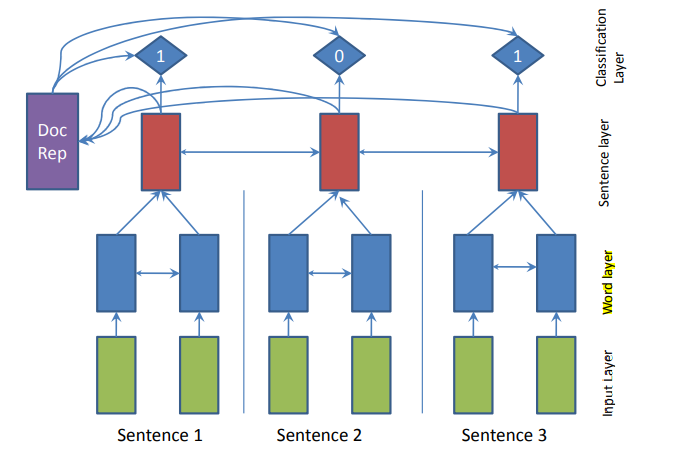
## 1-4-3- مدل‌های مبتنی بر شبکه‌های عصبی بازگشتی

در حوزه خلاصه‌سازی متن، مدل‌های استخراجی مبتنی بر شبکه‌های عصبی بازگشتی (RNN) ابزارهای موثری برای خلاصه‌سازی خودکار و تولید خروجی هستند. این مدل‌ها عمدتاً از معماری‌هایی مانند شبکه‌های LSTMو یا GRU بهره می‌برند. در طراحی این مدل‌ها، معمولاً یک شبکه بازگشتی به عنوان کدگذار برای تجسم ویژگی‌های مهم متون و یک مکانیزم انتخاب برای تمرکز بر نقاط کلیدی متن به‌کار می‌روند. برای مثال، یک مدل RNN می‌تواند با دریافت ویژگی‌های هر جمله به‌عنوان ورودی، توانایی انتخاب جملات مهم و تولید یک خلاصه فشرده از متن را داشته باشد. همچنین، شبکه‌های RNN دوطرفه نیز مورد استفاده قرار می‌گیرند تا به مدل اطلاعات کامل‌تری از متن دسترسی داده شود [61], [62].Top of Form

یکی از اولین مدل‌های مبتنی بر شبکه عصبی در زمینه‌ی خلاصه‌سازی خودکار متن، شبکه‌ی SummaRuNNer است [47]. در این مدل مسئله‌ی خلاصه‌سازی به‌صورت یک مسئله‌ی دسته‌بندی دنباله‌ای[[128]](#footnote-128) در نظر گرفته شده است. این مدل از شبکه‌ی عصبی GRU دوطرفه[[129]](#footnote-129) برای به‌دست‌آوردن اطلاعات زمینه‌ای در متن برای تولید مدل استفاده کرده است. همان‌طور که در شکل?? آمده است، این مدل از سه بخش لایه‌ی کلمات[[130]](#footnote-130)، لایه‌ی جملات[[131]](#footnote-131) و لایه‌ی دسته‌بندی[[132]](#footnote-132) تشکیل شده است. در بخش لایه‌ی کلمات، با استفاده از روش word2vec، کلمات به بردار‌ها تبدیل می‌شوند. بردار جمله از گذاشتن بردار کلمات درون یک جمله از یک شبکه‌ی RNN به‌دست می‌آید. در لایه‌ی جملات نیز یک شبکه‌ی RNN به‌کار رفته است. در این مدل با استفاده از روابط ؟؟ تا ؟؟ بردار متناسب با هر جمله به دست می‌آید.

در روابط فوق، و بردار مربوط به خروجی بردار مربوط به جمله‌ی j ام است. منظور از U گیت بروزرسانی، r گیت باز راه‌اندازی و نشان‌دهنده‌ی ضرب مؤلفه به مؤلفه[[133]](#footnote-133) می‌باشند. بعد از بدست آمدن بردار هر جمله و وارد شدن آن‌ها به یک ترنسفورمر غیرخطی، با استفاده از یک لایه‌ی دسته‌بندی با تابع فعال‌سازی[[134]](#footnote-134) سیگموئیدی[[135]](#footnote-135)، جملات به دو دسته‌ی 0 یا 1 (به معنای حضور یا عدم حضور در خلاصه) تقسیم می‌شوند. همچنین در این مدل، همانطور که در رابطه‌ی ؟؟ آمده است، یک نمایش برای سند، بدست می‌آید. این نمایش سند، در بخش دسته‌بندی به تشخیص برچسب‌ها کمک می‌کند.

دررابطه‌ی بالا، به ترتیب مقدار بردار عقبرو[[136]](#footnote-136) و پیشرو[[137]](#footnote-137)برای جمله‌ی j ام می‌باشد، که با هم ادغام می‌شوند. همچنین تعداد جملات در سند می‌باشد.

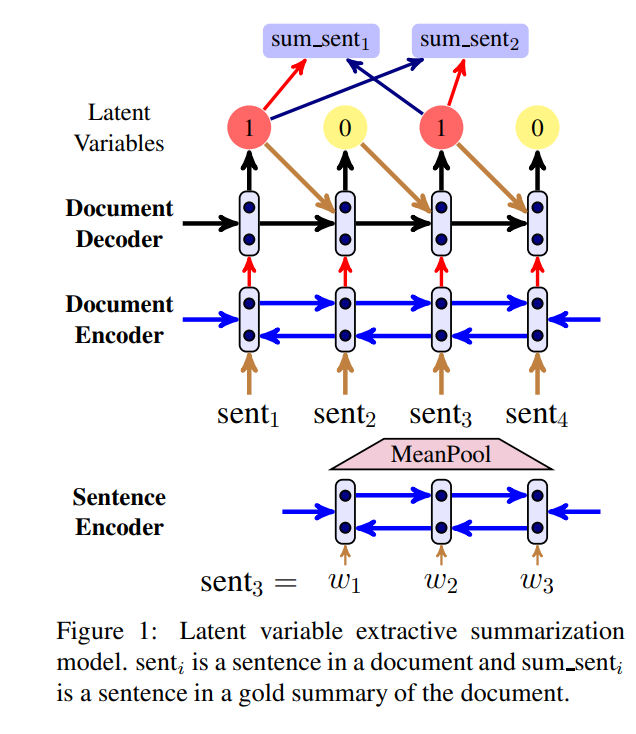


شکل 18: لایه‌های مدل SummaRuNNer [47].

از دیگر مدل‌های معروف در این زمینه می‌توان به مدل LatentExtSum نام برد [63]. در مدل LatentExtSum، از مکانیزم نگاشت جملات انتخابی به جملات خلاصه‌ی موجود در مجموعه‌دادگان و با استفاده از تعریف متغیر نهفته[[138]](#footnote-138)، سعی در آن شده است که در فرایند یادگیری، جملات با مفهوم نزدیک به خلاصه‌ی داده شده، استخراج شوند. همان‌طور که در شکل ؟؟ آمده است، این مدل از سه بخش اصلی رمزگذاری جملات[[139]](#footnote-139)، رمزگذاری سند[[140]](#footnote-140) و رمزگشایی سند[[141]](#footnote-141) تشکیل شده است. در بخش رمزگذاری جملات، هر جمله با استفاده از یک لایه‌ی LSTM دوطرفه و رابطه‌ی ؟؟ به بردار تبدیل می‌شود. در بخش رمز‌گذاری سند نیز از یک لایه‌ی LSTM دوطرفه برای یادگیری نمایش زمینه‌ای جملات و در بخش رمزگشای سند از یک لایه‌ی LSTM برای برچسب‌زنی جملات استفاده شده است. در آخر با استفاده از یک متغیر نهفته که متغیری با مقدار 0 یا 1 (به معنای عدم انتخاب یا انتخاب جمله در خلاصه) و سعی در 1 شدن جملات متن با بیشترین شباهت به خلاصه‌ی انسانی، هستند. به طور ساده این روش با استفاده از روش بیشینه‌سازی شباهت[[142]](#footnote-142)، سعی دارد جملات شبیه به خلاصه‌ی انسانی استخراج کند.

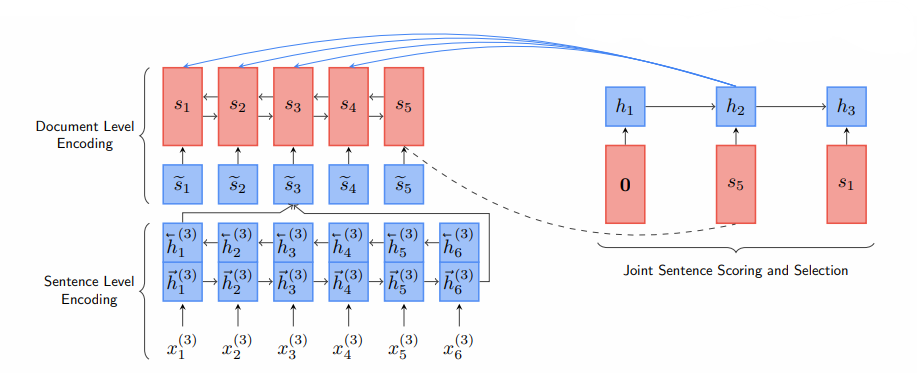
*(؟؟)*

در رابطه‌ی فوق منظور از || اندازه‌ی جمله‌ی i ام می‌باشد و منظور از بردار حالت پنهان[[143]](#footnote-143) j ام LSTM برای جمله‌ی i ام می‌باشد.



شکل 19: معماری کلی LatentExtSum [63].

در مدل NeuSum [53] از دو نوع رمزگذاری در سطح جمله و در سطح سند استفاده می‌کند و چهارچوبی برای چگونگی امتیازدهی و انتخاب جملات ارائه می‌دهد. همانطور که در شکل ؟؟ آمده است، برای جملات یک بار با استفاده از رمزگذاری سطح جمله و بار دیگر با استفاده از رمزگذاری سطح سند، به صورت سلسله مراتبی بردار ویژگی بدست می‌آید. در بخش رمزگذاری سطح جمله، کلمات استخراج شده و با استفاده از روش‌های جانشانی کلمات (برای مثال glove)، بردار ویژگی هر کلمه بدست می‌آید. با استفاده از شبکه‌ی عصبی بازگشتی GRU دو طرف (Bi-GRU)، ویژگی‌های زمینه‌ای در سطح هر جمله استخراج می‌شود. همچنین در سطح سند، دنباله‌ای از این ویژگی‌های استخراج شده در سطح جملات، داخل Bi-GRU دیگر شده و به ازای هر جمله ویژگی‌های زمینه‌ای سطح سند استخراج می‌گردد. در آخر نیز ویژگی‌های سطح جملات و سطح سند با استفاده از یک MLP دو لایه امتیازدهی شده و انتخاب می‌شوند.



شکل 20: استخراج ویژگی‌های جملات به صورت سلسله مراتبی در دو سطح جمله و سند [53].

Top of Form

مدل شبکه‌ی عصبی از پیش آموزش‌داده‌شده‌ی BERT، به منظور جانشانی کلمات با استفاده از یادگیری مدل زبانی برای نمایش ویژگی‌های زمینه‌ای بر روی یک پیکره‌ی متنی مقیاس بزرگ، به وجود آمده است. این شبکه‌، نمایش کلمات و جملات را با استفاده از یک ترنسفورمر بسیار بزرگ با هم ادغام می‌کند و توانسته است در مسائل پردازش متن دست‌آورد‌های بسیار زیادی داشته باشد.

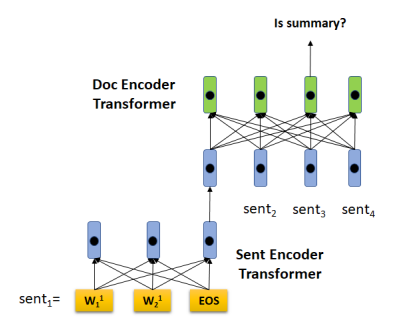
مدل BertSumExt [45] یک مدل خلاصه‌سازی متن با رویکرد استخراجی می‌باشد. این مدل با استفاده از مدل از پیش آموزش‌دیده‌ی BERT و قراردادن چند لایه‌ی ترنسفورمر جمله‌ای در بالای رمزگذار به‌منظور درنظرگرفتن ویژگی‌های سطح سند، پیشرفت قابل‌توجهی در حوزه‌ی خلاصه‌سازی خودکار متن به دست آورده است. شکل ؟؟، معماری مدل BERT اصلی و BERT برای خلاصه‌سازی متن را به نمایش گذاشته شده است. همان‌طور که در شکل ؟؟ آمده است، در معماری BERT اصلی، جملات با توکن‌های [SEP] جدا شده و در ابتدای هر متن یا سند توکن [cls] قرار می‌گیرد. درحالی‌که در معماری BERT برای خلاصه‌سازی ، به‌منظور نمایش مستقلانه‌ی جملات و یادگیری نمایش جملات مخصوص خلاصه‌سازی، توکن [cls] در ابتدای هر جمله می‌آید. در هر دو معماری برای هر کلمه سه جانشانی در نظر گرفته شده است. اولین آن‌ها جانشانی توکن[[144]](#footnote-144) ، به‌منظور نمایش معنای هر کلمه، دومین جانشانی که جانشانی بخش[[145]](#footnote-145) است، به منظور تمایز دادن بین جملات و سومین آن‌ها، جانشانی موقعیت[[146]](#footnote-146) به منظور نمایش موقعیت هر توکن است. این سه جانشانی با هم ادغام شده و به لایه‌های ترنسفورمر داده می‌شود و در آخر برای هر کلمه در جمله بردار مناسبی به دست می‌آید. در مدل پیشنهادی، بردار که نمایش‌دهنده‌ی جمله‌ی iام است، خروجی لایه‌ی آخر شبکه می‌باشد. خروجی هر لایه از شبکه به صورت روابط ؟؟ و ؟؟ است.

*(؟؟)*

در روابط فوق، LN نمایش‌دهنده‌ی لایه‌ی نرمال‌سازی[[147]](#footnote-147)، MHAtt به منظور عملگر توجه چند سره و نمایش‌دهنده‌ی خروجی لایه‌ی است. همچنین مقدار برابر با PosEmb(T) می‌باشد که در آن، T به منظور خروجی BERT برای ATS و PosEmb نیز یک تابع تعبیه کننده‌ی موقعیت سینوسی[[148]](#footnote-148) است. همچنین در بخش دسته‌بند، مطابق با رابطه‌ی ؟؟ از یک لایه‌ی سیگموئیدی استفاده شده است.

در روابط فوق منظور از بردار خروجی جمله‌ی *i* ام با شبکه‌ی *L* لایه (که در آزمایشات تعداد 2 لایه مناسب دیده شده است.) می‌باشد. و مؤلفه‌های قابل آموزش هستند.

پس از پیشرفت‌های مهم مدل BERT در زمینه خلاصه‌سازی متن، مقالات زیادی تلاش کرده‌اند تا بر روی این دستاوردات ادامه دهند. یکی از مدل‌های پیشنهادی که در همین راستا استفاده از مدلHIBERT‌[[149]](#footnote-149) [64]می‌باشد. مدل‌های خودکار خلاصه‌سازی عصبی معمولاً از یک کدگذار سلسله مراتبی برای کدگذاری اسناد استفاده می‌کنند و با استفاده از برچسب‌های سطح جمله آموزش داده می‌شوند. HIBERT یک مدل برای پردازش و خلاصه‌سازی اطلاعات در سطح سند است. این مدل از معماری ترنسفرمر برای نمایش و کدگذاری سند استفاده می‌کند و با ترکیب دو کدگذار، یکی برای جملات و دیگری برای سند، به صورت سلسله مراتبی ساخته شده است. هر سند شامل جملات است که با استفاده از کدگذار جمله به بردارهای مربوط به جملات تبدیل می‌شوند. سپس با استفاده از یک کدگذار سند بردار نهایی سند به دست می‌آید. بیشتر مدل‌های رمزگذار، کلمه را متناسب با کلمات قبل و بعد آن در جمله پیش‌بینی می‌کنن، در حالی که مدل از پیش‌آموزش دیده‌ی HIBERT سعی دارد تا یک جمله را با توجه به جملات قبل و بعد آن پیش‌بینی کند. همانطور که در شکل ؟؟ آمده است، در این مدل از دو رمزگذار استفاده شده است. رمزگذار جمله، ویژگی‌های مربوط به هر جمله و رمزگذار سند، ویژگی‌های مربوط به زمینه‌ی متن را استخراج می‌کند. با استفاده از این مدل رمزگذار‌ها، سند و جملات به صورت سلسه مراتبی، رمزگذاری می‌شوند، به طوری که از یک ترنسفرمر کلمات هر جمله عبور داده شده و ویژگی آخر آن‌ها به عنوان ویژگی جمله در نظر گرفته می‌شود. با استفاده از یک ترنسفرمر دیگر، ویژگی جملات در سطح سند استخراج می‌شوند. در آخر با استفاده از یک لایه‌ی دسته‌بند مشخص می‌شود چه جملاتی در خلاصه حضور پیدا خواهند کرد یا خیر [64].

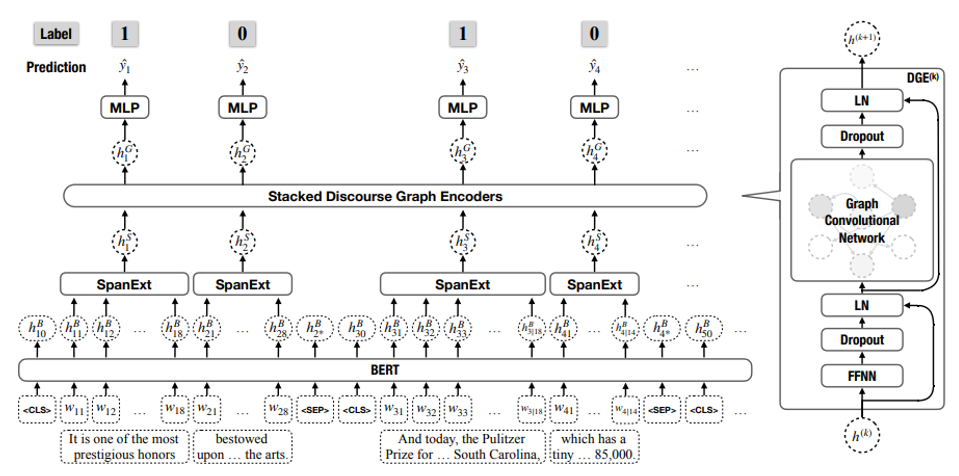


شکل 21: کدگذاری سلسله مراتبی جملات در روش HIBERT [64].

## 2-4-3- مدل‌های مبتنی بر شبکه‌های عصبی گراف

خلاصه‌سازی خودکار متن با استفاده از شبکه‌های عصبی گراف یک حوزه پژوهشی مهم است که در آن، سعی بر این دارد که با بهره‌گیری از قابلیت‌های گرافی مدل‌های عصبی، به‌طور هوشمندانه‌تر و جامع‌تر محتوای یک متن را خلاصه کند. در این رویکرد، معمولاً گره‌های گراف به اجزای مهم متن مثل جملات یا کلمات شکسته می‌شوند و یال‌ها نشان‌دهنده ارتباطات میان اجزا هستند. مدل‌های گراف عصبی می‌توانند با توجه به این ارتباطات و ساختار گراف، خلاصه متنی تولید کنند که اطلاعات کلیدی را حفظ کرده و مفهوم متن اصلی را به‌دست آورند. این رویکرد جدید نسبت به روش‌های سنتی خلاصه‌سازی متن، که معمولاً از روش‌های آماری یا مبتنی بر قوانین استفاده می‌کنند، امکانات بیشتری برای درک متن و تولید خلاصه با کیفیت‌تر فراهم می‌کند. این مدل‌ها معمولاً از معماری‌هایی مانند شبکه‌های توجه بهره‌می‌برند تا به‌طور هوشمندانه به اجزا و ارتباطات مهم توجه کنند. با پیشرفت تکنولوژی گرافی و توانمندی‌های شبکه‌های عصبی، انتظار می‌رود که این رویکرد در آینده بیشتر به‌کار گرفته شود و به بهبود مدل‌های خودکار خلاصه‌سازی متن کمک کند [8], [51].

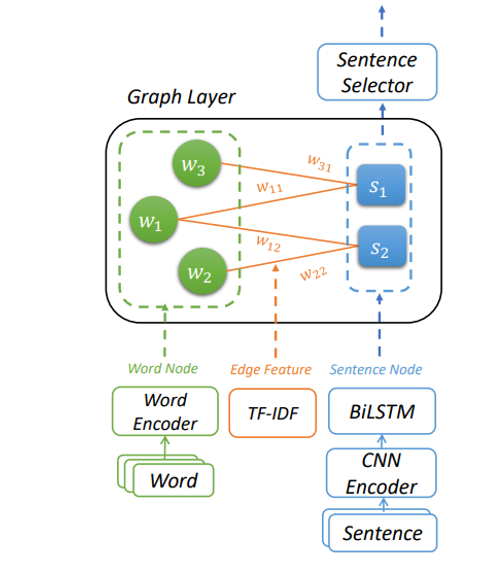
همان‌طور که پیش‌تر ذکر شد، یکی از بزرگ‌ترین چالش‌های موجود در سیستم‌های خلاصه‌سازی، چگونگی درنظرگرفتن ویژگی‌های زمینه‌ای با فاصله از هم می‌باشد. با وجود این که برخی مدل‌ها مثل مدل BERT تا حدی توانست این موضوع را در نظر بگیرد، ولی برخی ضعف‌ها در این زمینه وجود دارد؛ حتی وجود LSTMها و ترنسفورمر‌ها در بالای BERT نیز بهبودی حاصل نمی‌کند. مدل DiscoBERT [65] با استفاده از گراف‌هایی مثل RST[[150]](#footnote-150) و مرجع[[151]](#footnote-151) و با یادگیری شبکه‌های عصبی پیچشی گراف (GCN) توانسته است ویژگی‌های زمینه‌ای طولانی بین جملات را در نظر بگیرد و باعث بهبود فشرده‌سازی و کاهش تکرار شود. در مدل DiscoBERT، ابتدا از متن ورودی یکی از دو گراف RST یا مرجع ساخته می‌شود. گراف RST با استفاده از درخت تجزیه‌ی RST ساخته شده و گراف مرجع با استفاده از همبستگی‌های بین موجودیت‌های متن ورودی[[152]](#footnote-152) تولید می‌شود. همان‌طور که در شکل ؟؟ آمده است، معماری مدل DiscoBERT از BERT به‌عنوان رمزگذاری سند استفاده می‌کند، سپس با استفاده از واحد‌هایی به نام SpanExt واحد‌هایی به نام EDU[[153]](#footnote-153) که کوچک‌ترین واحد‌ خلاصه‌سازی هستند، استخراج می‌شود. بعد از استخراج EDU ها، گراف موردنظر ساخته شده و با استفاده از شبکه‌ی GCN، بردار‌های مخصوص هر EDU استخراج و وارد شبکه‌ی عصبی MLP شده و دسته‌بندی می‌شوند.



شکل 22: معماری کلی از شبکه‌ی DiscoBert [65].

مدل GraphSumTextRank [66] یکی دیگر از مدل‌های مبتنی بر گراف است. در این مقاله یک روش خلاصه‌سازی متن مبتنی بر گراف توضیح داده شده است. در زمان‌های اخیر، به دلیل رشد سریع اینترنت، دسترسی به میزان عظیمی از اطلاعات به یک چالش بزرگ تبدیل شده است. برای مدیریت این حجم زیاد از اطلاعات، نیاز به روش‌ها و ابزارهای کارآمد و موثر مثل روش PageRank وجود دارد. PageRank یک الگوریتم مهم در حوزه موتورهای جستجو و ارتباطات وب است که توسط لری پیج و سرگی برین در دانشگاه استنفورد توسعه یافته است. این الگوریتم برای ارزیابی اهمیت یک صفحه وب استفاده می‌شود و ایده اصلی آن بر اساس ساختار گرافی اینترنتی است. در PageRank، هر صفحه به عنوان یک گره در گراف در نظر گرفته می‌شود و وزن هر گره بر اساس تعداد وزن‌دهی شده یال‌های وارد به آن صفحه محاسبه می‌شود. به عبارت دیگر، یک صفحه ارزش بیشتری دارد اگر صفحات با ارتباط به آن ارتباطات بیشتری داشته باشند. روش معرفی شده در این مقاله از TextRank اصلاح‌شده استفاده می‌کند که بر اساس مفهوم PageRank تعریف شده برای هر صفحه در وب‌سایت‌ها است. این روش یک گراف با جملات به عنوان گره‌ها و شباهت بین دو جمله به عنوان وزن یال بین آن‌ها ایجاد می‌کند. سپس گراف ساخته شده و به خوشه‌های مختلف تقسیم می‌شود با فرض اینکه جملات داخل یک خوشه به یکدیگر شبیه هستند و جملات خوشه‌های مختلف نشان‌دهنده تفاوت آن‌ها هستند. ارزیابی عملکرد این تکنیک خلاصه‌سازی پیشنهادی نشان می‌دهد که این روش کارآمدی دارد.

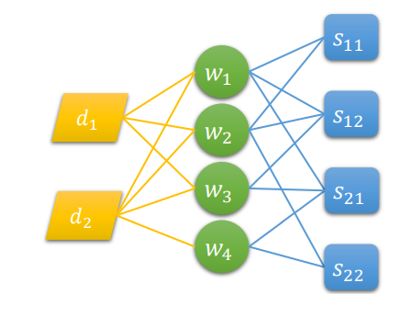
یکی دیگر از مدل‌هایی که سعی کرده است تا وابستگی‌های با فاصله‌ی زیاد در متن را در نظر بگیرد، مدل HSG [48] می‌باشد. در این مدل با طراحی گراف ناهمگن، توانسته است پیشرفت قابل‌توجهی در زمینه‌ی خلاصه‌سازی متن به دست آورد. به منظور ترکیب ویژگی‌های جملات مختلف و همچنین کلمات مختلف، جملات و کلمات به‌عنوان گره‌هایی از یک گراف پیشنهاد شده است. همان‌طور که در شکل ؟؟ دیده می‌شود، در این روش کلمات و جملات به‌عنوان گره‌های گراف و بین هر گره‌ی کلمه با گره‌ی جمله به‌شرط وجود آن کلمه در جمله یالی قرار می‌گیرد. همچنین وزن هر یال بر اساس معیار TF-IDF و میزان تکرار کلمه در جمله محاسبه می‌شود. بردار موجود در گره‌های کلمه به وسیله‌ی یک شبیه‌ساز کلمه و بردار موجود در گره‌های جملات به وسیله‌ی یک رمز‌گذار پیچشی[[154]](#footnote-154) و شبکه‌ی BiLSTM، به دست می‌آیند.



شکل 23: معماری کلی و ساخت گراف در مدل HSG [48].

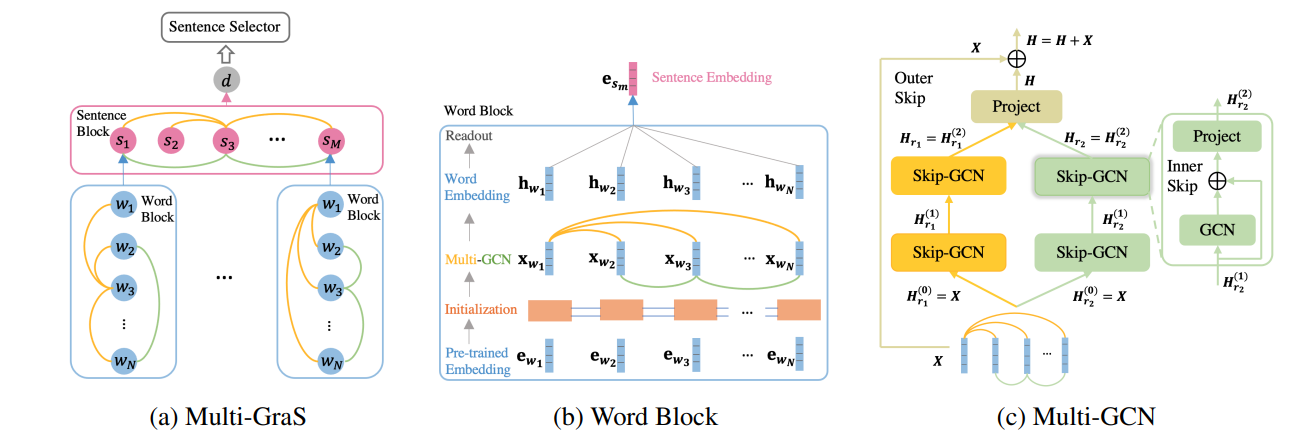
در این روش، برای بخش خلاصه‌سازی چند سندی نیز راه حلی وجود دارد که باعث به‌دست‌آمدن نتایج مفید شده است. همان‌طور که در شکل ?? آمده است، در روش HDSG به مشابه گراف ذکر شده در قسمت قبل، گره‌های جملات و کلمات به دست می‌آید و گره‌هایی برای هر سند در نظر گرفته می‌شود. بردار درون هر گره‌ی سند و نحوه‌ی اتصال یال‌ها مشابه گره‌ی جملات می‌باشد.

در قسمت آموزش گراف‌ از ساختار توجه چند سره و گراف توجه و ساختار اتصال residual استفاده می‌شود. بعد از اعمال الگوریتم پیام‌رسانی و جابه‌جایی اطلاعات بین گره‌ها، در بخش انتخاب جملات، به گره‌های جملات امتیازدهی شده و گره‌های با امتیاز بالا انتخاب می‌شوند.



شکل 24: شمای کلی گراف در مدل HDSG [48].

یکی دیگر از روش‌های گرافی که توانسته است نتایج خوبی را به دست آورد، مدل Multi-GraS‌[67] می‌باشد. همان‌طور که در شکل ?? آمده است، این مدل از سه بخش اصلی بلوک کلمات، بلوک جملات و بخش انتخاب جملات تشکیل شده است. در بخش بلوک کلمات، گرافی شامل گره‌های کلمات یک جمله، تشکیل می‌شود. در این گراف یال‌ها بر اساس بر اساس چندین روابط از جمله روابط نحوی استخراج شده با تجزیه‌گر وابستگی[[155]](#footnote-155)، بین دو گره‌ی کلمه در هر جمله به وجود می‌آیند. در بخش بلوک کلمات، هر کلمه به وسیله‌ی جانشانی کلمات، تبدیل به بردار می‌شود. با استفاده از مکانیزم Multi-GCN که در شکل ؟؟ آمده است، عملیات آموزش و تبادل اطلاعات بر روی گراف صورت می‌گیرد. بردار خروجی هر کلمه با هم ادغام شده و بردار جمله‌ی مربوطه تشکیل می‌شود. در بلوک جملات، هر جمله به‌عنوان یک گره در نظر گرفته شده و یال‌های این گراف بر اساس میزان شباهت بین دو جمله و تعداد کلمات مشترک بین دو جمله به وجود می‌آیند. در این بخش نیز با استفاده از مکانیزم Multi-GCN عملیات آموزش و تبادل اطلاعات بر روی گراف صورت می‌گیرد. در ساختار Multi-GCN از چند شبکه‌ی skip-GCN و ادغام نتایج آن‌ها استفاده شده است. در بخش انتخاب جملات، با استفاده از یک شبکه‌ی عصبی پیشرو جملات امتیازدهی شده و جملات با امتیاز بالاتر استخراج می‌شوند.



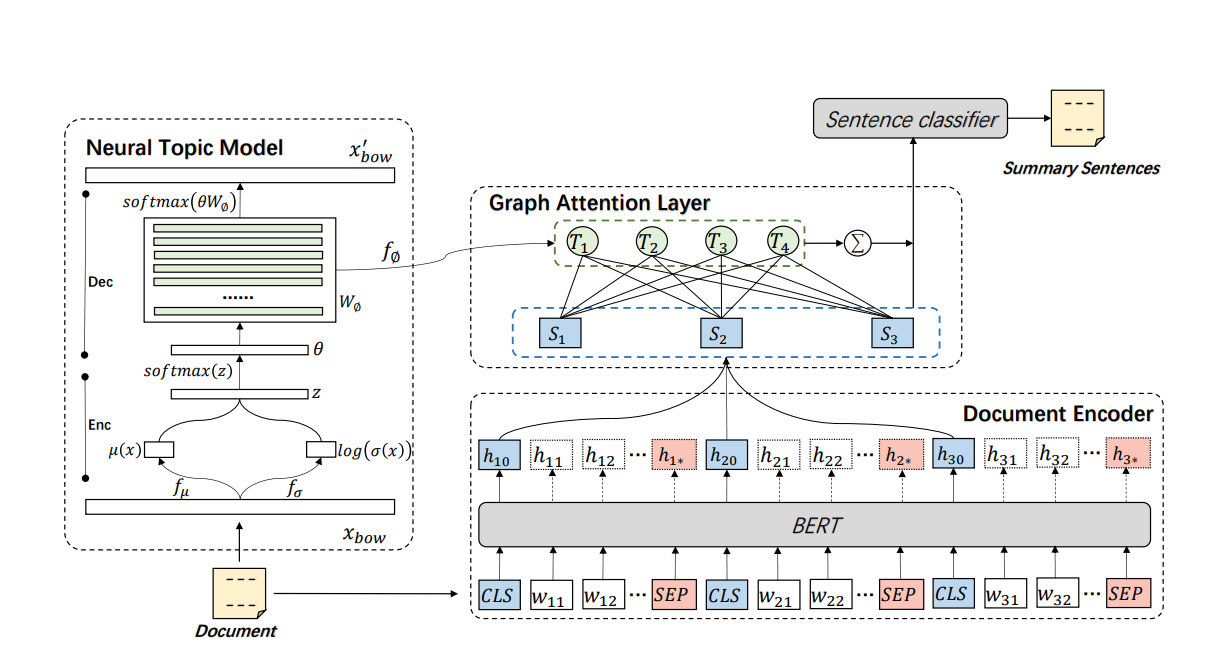
با وجود عبارات تکراری بین جملات استخراج شده، مدل کردن دقیق جملات با روش‌های عمومی نظارت شده مشکل است. مدل HAHSUM‌[[156]](#footnote-156) [68] به خوبی اطلاعات مختلف را مدل می‌کند و با استفاده از شبکه‌های عصبی گراف، وابستگی‌های تکراری بین جملات را برجسته می‌کند.در این روش یک گراف ناهمگن که حاوی سطوح اطلاعات چندگانه برای نمایش یک سند است ساخته می‌شود.در این گراف از سه نوع گره کلمات، جملات و موجودیت‌های نامدار[[157]](#footnote-157) استفاده شده است. موجودیت‌های نامدارو کلمات متوالی داخل یک جمله به هم وصل می‌شوند. کلمات و موجودیت‌های یک جمله به گره‌ی آن جمله به صورت یال یک طرفه وصل می‌شوند. همچنین گره‌های موجودیت‌های نامدار یکسان نیز به‌هم متصل می‌شوند. در آخر جملاتی که دارای trigram یکسان باشند نیز به هم متصل می‌شوند. با استفاده از مدل از پیش آموزش دیده‌ی AlBERT ویژگی‌های گره‌های کلمات و جملات بدست می‌آید. با استفاده از شبکه‌ی عصبی GAT وبه وسیله‌ی کاهش اطلاعات تکراری در فرایند انتقال پیام باعث می‌شود تا جملات با محتوای تکراری امتیاز کمتری بدست آورند و خلاصه از پراکندگی اطلاعات بهتری برخوردار باشد.

استفاده از گراف‌ها موفقیت‌های زیادی در مسئله‌ی خلاصه‌سازی متن داشته است. همچنین بسیاری از روش‌های مبتنی بر عنوان نیز از روش‌های گرافی استفاده می‌کنند. در ادامه به بررسی روش‌های مبتنی بر عنوان نیز بررسی شده است.

## 3-4-3- روش‌های مبتنی بر عنوان

خلاصه‌سازی متن مبتنی بر عنوان یک روش مؤثر در پردازش زبان طبیعی است که بر اساس تشخیص موضوعات اصلی در متن، سعی در تولید خلاصه مطالب می‌کند. در این روش، ابتدا موضوعات کلیدی و عناوین مهم متن شناسایی می‌شوند، سپس با ترکیب و خلاصه‌سازی جملات مرتبط با این موضوعات، یک خلاصه مفید و مختصر از مطلب اصلی تولید می‌شود. استفاده از خلاصه‌سازی مبتنی بر عنوان به تحلیل محتوای متن و شناسایی مفاهیم اصلی می‌پردازد، که این امر می‌تواند در فهم سریع‌تر و بهتر متون به کار گیرد. این روش مخصوصاً در وظایف مانند خلاصه‌سازی اخبار، مقالات و مطالب طولانی کمک بزرگی به کاربران جهت به دست آوردن یک نگاه کلی از محتوا دارد.

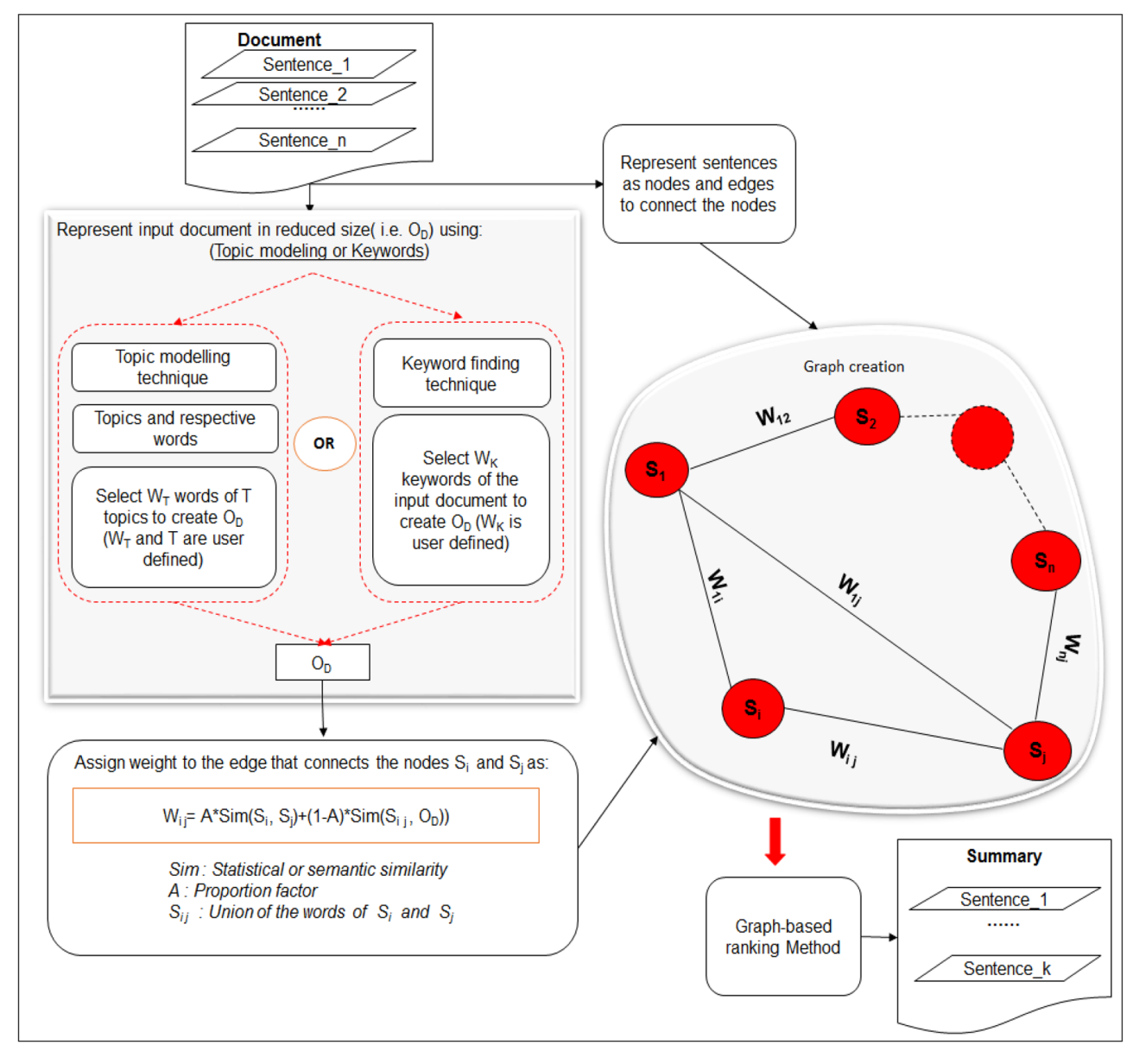
خلاصه‌سازی متون با مقیاس بزرگ یکی از اصلی ترین چالش‌های مسئله‌ی خلاصه‌سازی متن است. مدل TopicGraphSum [69] با درنظرگرفتن روابط بین جملات در متن و عنصر مهم عنوان متن، توانسته است در متون مقیاس بزرگ، موفق ظاهر شود. مدل ارائه شده، حتی نتایج خوبی در متون کوتاه نیز داشته است. در این مدل با استفاده از مدل عصبی مبتنی بر عنوان (NTM[[158]](#footnote-158))، عنوان هر متن استخراج می‌شود. همان‌طور که در شکل ?? آمده است، این مدل شامل بخش رمزگذاری سند می‌باشد. در این بخش سند به وسیله‌ی BERT رمزگذاری شده و کلمات به بردار تبدیل می‌شود.علائم CLS و SEP برای شناسایی جملات در این بخش به کار می‌روند. در بخش NTM، استخراج بردار موضوع از سند ورودی با استفاده از پروسه‌ی رمزگذاری - رمزگشایی[[159]](#footnote-159) صورت می‌گیرد. ورودی این بخش، نمایش کیسه‌ی کلمات از سند است و این بخش آموزش می‌بیند تا بردار ویژگی مناسبی برای عنوان، استخراج شود. در قسمت بعد، یک گراف ناهمگن، شامل گره‌های جملات و عنوان‌ها ساخته شده و همه‌ی جملات یک سند به گره‌ی عنوان وصل می‌شوند. به همین دلیل، همه‌ی جملات با یک مسیر به طول 2 به هم متصل هستند (برای ارتباط بین جملات در یک موضوع و سند). با استفاده از مکانیزم توجه گراف، اطلاعات اشتراک گذشته شده و روابط معنایی بین جملات یاد گرفته و در آخر توسط یک دسته‌بند جملات، گره‌های مناسب برای خلاصه استخراج می‌شود.



شکل 25: معماری مدل TopicGraphSum [69].

در تکنیک‌های خلاصه‌سازی متن مبتنی بر گراف، وزن نسبت داده شده به یال‌های گراف پارامتر حیاتی برای رتبه‌بندی جملات است. این وزن‌ها بر اساس شباهت بین جملات (گره‌ها) تعیین می‌شوند. بیشتر تکنیک‌های مبتنی بر گراف از اندازه‌گیری شباهت کلمات مشترک برای نسبت دادن وزن استفاده می‌کنند. در مدل ارائه شده توسط بلوال و همکاران [70]، یک تکنیک جدید خلاصه‌سازی مبتنی بر گراف پیشنهاد شده است که علاوه بر در نظر گرفتن شباهت بین جملات متن، شباهت بین جملات و متن کلی ورودی را نیز مد نظر قرار می‌دهد. در حین نسبت دادن وزن به یال‌های گراف، دو ویژگی در نظر گرفته می‌شود. ویژگی اول شباهت بین گره‌هایی است که یال‌های گراف را تشکیل می‌دهند. ویژگی دوم وزن داده شده به یک مؤلفه است که نشان‌دهنده این است که چقدر یک یال خاص با موضوعات کلی متن ورودی مشابه است، که در این مدل، موضوع را در آن ادغام می‌کند. به همراه این تغییرات، از اندازه‌گیری معنایی برای یافتن شباهت بین گره‌ها استفاده می‌شود. روند کلی این روش در شکل ؟؟ به نمایش گذاشته شده است. در این روش ابتدا گراف کاملی با گره‌های جملات ساخته می‌شود، همچنین با استفاده از روش‌های متداول استخراج عنوان یا استخراج کلمات کلیدی (روش‌هایی مثل LDA topic modeling)، مجموعه از عبارات اصلی متن استخراج می‌شود. با استفاده از فرمول ؟؟ وزن بین هر گره مشخص می‌شود. به منظور مرتب سازی جملات بر اساس معیار اهمیت از الگوریتم graph ranking استفاده می‌شود. جملات با رتبه‌ی بالاتر به منظور تولید خلاصه انتخاب می‌شوند. Top of Form

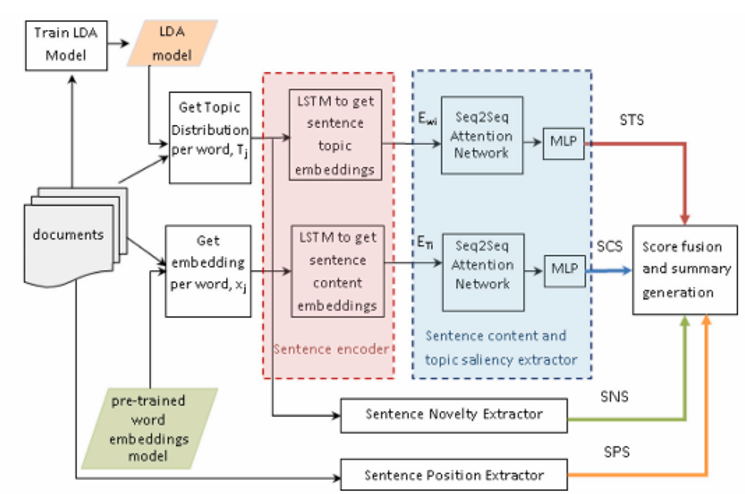
در عبارت ریاضی بالا، A ضریب اهمیت شباهت بین دو جمله می‌باشد. در قسمت دوم شباهت کلمات مشترک دو جمله با مجموعه کلیدواژه‌ها محاسبه می‌شود. توابع شباهت بین دو مجموعه کلمه یا دو جمله با استفاده از روش‌هایی مثل شباهت کسینوسی، شباهت اقلیدسی و شباهت جاکارد محاسبه می‌شود.



شکل 26: نمای کلی از روش ارائه شده توسط بلوال و همکاران و استفاده از گراف و مجموعه کلیدواژه‌ها یا عناوین برای خلاصه‌سازی متن [70].

در مدل DeepSum [71]، روش نوآورانه‌ای برای خلاصه‌سازی متن استخراجی از اسناد تکی ارائه شده است. این روش بر اساس مدل‌های موضوع و جاسازی کلمات برای خلاصه‌سازی متن‌ها استفاده می‌کند. روش‌های اخیر بر اساس شبکه‌های دنباله‌ای ناتوان در درک مفاهیم با طول متن بلند اسناد هستند که در بردارهای موضوع اسناد جاسازی شده‌اند. در این مدل، هدف استفاده از اطلاعات پنهان در اسناد از طریق بردارهای موضوع و شبکه‌های دنباله‌ای است تا کیفیت و دقت متن خلاصه شده را بهبود بخشد.

هر جمله از طریق دو شبکه عصبی بازگشتی مختلف بر اساس توزیع احتمالی موضوع و جاسازی کلمات ساخته شده، سپس یک شبکه RNN به هر کدام ازین جملات اعمال می‌شود. خروجی‌های کدگذار و کدگشا در شبکه دنباله‌ای پس از وزن‌دهی با استفاده از یک مکانیزم توجه ترکیب شده و به وسیله یک شبکه ورودی چندلایه به امتیاز تبدیل می‌شوند. امتیاز حاصل از مدل موضوع[[160]](#footnote-160) (STS) به عنوان امتیاز موضوع جمله و امتیاز حاصل از جاسازی کلمات به عنوان امتیاز محتوای جمله (SCS) [[161]](#footnote-161)اشاره دارد. علاوه بر این، امتیاز جدید نوآوری جمله یا عدم تکراری بودن[[162]](#footnote-162) (SNS)و امتیاز موقعیت جمله (SPS) [[163]](#footnote-163)ارائه شده و انجام ادغام وزن‌دار چهار امتیاز برای هر جمله در سند جهت محاسبه امتیاز نهایی جمله (FSS) انجام می‌شود. برای محاسبه‌ی امتیاز SPS، میزان جایگاه نسبی جمله به کل جملات می‌باشد به طوری که جملات ابتدایی امتیاز بالاتری می‌گیرند. همچنین امتیاز SPS نیز میانگین میزان شباهت کسینوسی جمله مورد نظر با جملات قبل از آن می‌باشد. شکل ؟؟ معماری کلی این مدل را نمایش می‌دهد.



شکل 27: معماری کلی روش DeepSum [71].

Top of Form

Top of Form

1. Extractive text summarization using deep learning approaches [↑](#footnote-ref-1)
2. Natural language processing [↑](#footnote-ref-2)
3. Automatic text summarization [↑](#footnote-ref-3)
4. Transformer [↑](#footnote-ref-4)
5. Preprocessing [↑](#footnote-ref-5)
6. Postprocessing [↑](#footnote-ref-6)
7. Part of speech tagging [↑](#footnote-ref-7)
8. Stop words Filtering [↑](#footnote-ref-8)
9. Stemming [↑](#footnote-ref-9)
10. Title similarity [↑](#footnote-ref-10)
11. Term frequency – inverse term frequency [↑](#footnote-ref-11)
12. Thematic words [↑](#footnote-ref-12)
13. Extractive [↑](#footnote-ref-13)
14. Abstractive [↑](#footnote-ref-14)
15. Hybrid [↑](#footnote-ref-15)
16. Problem of redundancy [↑](#footnote-ref-16)
17. Problem of irrelevancy [↑](#footnote-ref-17)
18. Problem of loss of coverage [↑](#footnote-ref-18)
19. Problem of non-readability and less cohesive content [↑](#footnote-ref-19)
20. Long-distance relationships [↑](#footnote-ref-20)
21. Recurrent Neural Network [↑](#footnote-ref-21)
22. Convolutional neural networks [↑](#footnote-ref-22)
23. Recurrent neural networks [↑](#footnote-ref-23)
24. Natural language processing (NLP) [↑](#footnote-ref-24)
25. Text mining [↑](#footnote-ref-25)
26. Graph neural networks [↑](#footnote-ref-26)
27. Deep learning [↑](#footnote-ref-27)
28. Machine learning [↑](#footnote-ref-28)
29. Support vector machines [↑](#footnote-ref-29)
30. k-nearest neighbors [↑](#footnote-ref-30)
31. Decision tree [↑](#footnote-ref-31)
32. Random forest [↑](#footnote-ref-32)
33. Data science [↑](#footnote-ref-33)
34. Multi-layer perceptron (MLP) [↑](#footnote-ref-34)
35. Input layer [↑](#footnote-ref-35)
36. Hidden layers [↑](#footnote-ref-36)
37. Output layer [↑](#footnote-ref-37)
38. Non-linear [↑](#footnote-ref-38)
39. Activation functions [↑](#footnote-ref-39)
40. Fully connected [↑](#footnote-ref-40)
41. Linear [↑](#footnote-ref-41)
42. Loss function [↑](#footnote-ref-42)
43. Optimizer algorithms [↑](#footnote-ref-43)
44. Binary classification [↑](#footnote-ref-44)
45. Convolutional neural network [↑](#footnote-ref-45)
46. Conolutional layers [↑](#footnote-ref-46)
47. Pooling layers [↑](#footnote-ref-47)
48. Convolution [↑](#footnote-ref-48)
49. Aggregation methods [↑](#footnote-ref-49)
50. Recurrent neural networks [↑](#footnote-ref-50)
51. Recurrent memory [↑](#footnote-ref-51)
52. Machine translation [↑](#footnote-ref-52)
53. Speech recognition [↑](#footnote-ref-53)
54. Text prediction [↑](#footnote-ref-54)
55. Text generation [↑](#footnote-ref-55)
56. Long short-term memory [↑](#footnote-ref-56)
57. Gated recurrent unit [↑](#footnote-ref-57)
58. Bidirectional recurrent neural networks [↑](#footnote-ref-58)
59. Encoder-decoder structure [↑](#footnote-ref-59)
60. Image generation [↑](#footnote-ref-60)
61. Latent space [↑](#footnote-ref-61)
62. Attention-based methods [↑](#footnote-ref-62)
63. Transformer [↑](#footnote-ref-63)
64. Attention cell [↑](#footnote-ref-64)
65. key [↑](#footnote-ref-65)
66. value [↑](#footnote-ref-66)
67. query [↑](#footnote-ref-67)
68. Scaled dot-product [↑](#footnote-ref-68)
69. Multi-head attention [↑](#footnote-ref-69)
70. Self-attention [↑](#footnote-ref-70)
71. Graph neural networks [↑](#footnote-ref-71)
72. Nodes [↑](#footnote-ref-72)
73. Edges [↑](#footnote-ref-73)
74. Data architecture [↑](#footnote-ref-74)
75. Social networks [↑](#footnote-ref-75)
76. Routing problems [↑](#footnote-ref-76)
77. Analysis of complex networks [↑](#footnote-ref-77)
78. Directed graph [↑](#footnote-ref-78)
79. Undirected graph [↑](#footnote-ref-79)
80. Weighted graph [↑](#footnote-ref-80)
81. Unweighted graph [↑](#footnote-ref-81)
82. Cyclic graph [↑](#footnote-ref-82)
83. Acyclic graph [↑](#footnote-ref-83)
84. Homogeneous graph [↑](#footnote-ref-84)
85. Heterogeneous graph [↑](#footnote-ref-85)
86. message passing [↑](#footnote-ref-86)
87. relations [↑](#footnote-ref-87)
88. Heterogeneous Learning [↑](#footnote-ref-88)
89. Convolutional network [↑](#footnote-ref-89)
90. Graph attention networks [↑](#footnote-ref-90)
91. Graph pooling layers [↑](#footnote-ref-91)
92. Word embedding [↑](#footnote-ref-92)
93. Global vectors for word representation [↑](#footnote-ref-93)
94. Pretrained model [↑](#footnote-ref-94)
95. Bidirectional encoder representations from transformers [↑](#footnote-ref-95)
96. bidirectional [↑](#footnote-ref-96)
97. token [↑](#footnote-ref-97)
98. Named entity recognition (NER) [↑](#footnote-ref-98)
99. Relation extraction (RE) [↑](#footnote-ref-99)
100. Question answering (QA) [↑](#footnote-ref-100)
101. Contextual feature [↑](#footnote-ref-101)
102. Euclidean distance [↑](#footnote-ref-102)
103. Cosine similaity [↑](#footnote-ref-103)
104. Dot product [↑](#footnote-ref-104)
105. Intrinsic [↑](#footnote-ref-105)
106. Extrinsic [↑](#footnote-ref-106)
107. Question answering [↑](#footnote-ref-107)
108. Information retrieval [↑](#footnote-ref-108)
109. Text quality evaluation [↑](#footnote-ref-109)
110. Content evaluation [↑](#footnote-ref-110)
111. Recall-Oriented Understudy for Gisting Evaluation [↑](#footnote-ref-111)
112. Unigram [↑](#footnote-ref-112)
113. Bigram [↑](#footnote-ref-113)
114. Sentence classification [↑](#footnote-ref-114)
115. classifier [↑](#footnote-ref-115)
116. Ranking algorithms [↑](#footnote-ref-116)
117. Contextual representations [↑](#footnote-ref-117)
118. Recurrent graph neural networks [↑](#footnote-ref-118)
119. Convolutional graph neural networks [↑](#footnote-ref-119)
120. Graph AutoEncoder [↑](#footnote-ref-120)
121. spatial-temporal graph neural networks [↑](#footnote-ref-121)
122. classical approaches [↑](#footnote-ref-122)
123. Term frequency-inverse document frequency [↑](#footnote-ref-123)
124. Spam detection [↑](#footnote-ref-124)
125. Maximum entropy model [↑](#footnote-ref-125)
126. Bandit Problem [↑](#footnote-ref-126)
127. Heuristic [↑](#footnote-ref-127)
128. Sequence classification [↑](#footnote-ref-128)
129. Bidirectional gated recurrent unit [↑](#footnote-ref-129)
130. Word layer [↑](#footnote-ref-130)
131. Sentence layer [↑](#footnote-ref-131)
132. Classification layer [↑](#footnote-ref-132)
133. Hadamard product [↑](#footnote-ref-133)
134. Activation function [↑](#footnote-ref-134)
135. Sigmoid [↑](#footnote-ref-135)
136. backward [↑](#footnote-ref-136)
137. forward [↑](#footnote-ref-137)
138. Latent variables [↑](#footnote-ref-138)
139. Sentence encoder [↑](#footnote-ref-139)
140. Document encoder [↑](#footnote-ref-140)
141. Document decoder [↑](#footnote-ref-141)
142. Maximum likelihood [↑](#footnote-ref-142)
143. Hidden state [↑](#footnote-ref-143)
144. Token embedding [↑](#footnote-ref-144)
145. Segment embedding [↑](#footnote-ref-145)
146. Position embedding [↑](#footnote-ref-146)
147. Normalization layer [↑](#footnote-ref-147)
148. Sinusoid positional embeddings [↑](#footnote-ref-148)
149. Hierarchical bidirectional encoder representations from transformers [↑](#footnote-ref-149)
150. Rhetorical Structure Theory [↑](#footnote-ref-150)
151. Coreference [↑](#footnote-ref-151)
152. coreference mentions based [↑](#footnote-ref-152)
153. Elementary Discourse Unit [↑](#footnote-ref-153)
154. CNN encoder (convolutional neural network encoder) [↑](#footnote-ref-154)
155. Dependency parser [↑](#footnote-ref-155)
156. Hierarchical Attentive Heterogeneous Graph for Text Summarization [↑](#footnote-ref-156)
157. Named entity [↑](#footnote-ref-157)
158. Neural topic model [↑](#footnote-ref-158)
159. Encoding-decoding process [↑](#footnote-ref-159)
160. Sentence topic score [↑](#footnote-ref-160)
161. Sentence contex score [↑](#footnote-ref-161)
162. Sentence novelty score [↑](#footnote-ref-162)
163. Sentence position score [↑](#footnote-ref-163)