به نام خالق رنگین کمان

ستاره باباجانی – گزارش تمرین دوم

سوال 1: a)

1. One-hot Encoding:

- در One-hot Encoding، هر کلمه در یک واژگان با یک بردار نشان داده می شود که در آن همه عناصر به جز یک عنصر 0 هستند که برای نشان دادن موقعیت آن کلمه در واژگان، روی 1 تنظیم شده است.
- هر کلمه به یک شاخص منحصر به فرد در یک واژگان نگاشت می شود و بردار نشان دهنده آن کلمه دارای مقدار 1 در موقعیت مربوط به شاخص آن و 0 در تمام موقعیت های دیگر است.
- One-hot Encoding منجر به نمایش پراکنده می شود، به این معنی که بردارها ابعاد بالایی دارند و عمدتاً با صفر پر شده اند که می تواند برای واژگان بزرگ ناکارآمد باشد.
- One-hot Encoding هیچ رابطه معنایی بین کلمات را نمی گیرد و هر کلمه را مستقل تلقی می کند.

2. Word Embedding:

- Word Embedding کلمات را به عنوان بردارهای متراکم و کم بعدی در یک فضای برداری پیوسته نشان می دهد که در آن موقعیت هر کلمه بر اساس زمینه و معنای آن آموخته می شود.

- تکنیک‌های Word Embedding، مانند Word2Vec، GloVe و FastText، یاد می‌گیرند که کلمات را به بردارها به گونه‌ای نگاشت کنند که کلمات مشابه در فضای جاسازی به هم نزدیک‌تر باشند و روابط معنایی بین کلمات را به تصویر بکشند.
- Word Embedding از پیکره های متنی بزرگ با استفاده از مدل های شبکه عصبی آموخته می شود و می تواند شباهت های نحوی و معنایی بین کلمات را به تصویر بکشد.
- بر خلاف One-hot Encoding، Word Embedding منجر به نمایش های متراکم می شود که در آن هر کلمه با یک بردار با اندازه ثابت با عناصر با ارزش واقعی نشان داده می شود.
- Word Embedding از نظر محاسباتی کارآمدتر است و اطلاعات بیشتری را در مورد معنایی زیربنایی کلمات در مقایسه با One-hot Encoding به دست می‌آورد.

b) GloVe که مخفف Global Vectors for Word Representation است، یک الگوریتم یادگیری بدون نظارت است که برای ایجاد جاسازی کلمات استفاده می شود. هدف آن به دست آوردن معنای کلمات با یادگیری از آمار همزمانی کلمات در مجموعه بزرگی از متن است.

حال به بررسی نحوه ایجاد Word Embedding توسط این الگوریتم میپردازیم:

1. ساخت ماتریس Co-occurrence:

- GloVe با ساخت یک ماتریس همزمانی X شروع می شود، که در آن هر عنصر Xij تعداد دفعاتی را نشان می دهد که کلمه j در context کلمه i در یک سایز پنجره ثابت ظاهر می شود.
- context یک کلمه را می توان بر حسب کلمات مجاور در یک اندازه پنجره مشخص تعریف کرد. ماتریس Co-occurrence، رابطه آماری بین کلمات را بر اساس هم‌روی آنها در پیکره نشان می‌دهد.

2. راه اندازی اولیه بردارهای کلمه:

- GloVe بردارهای کلمه را برای هر کلمه در واژگان به صورت تصادفی یا با استفاده از تکنیک های دیگر مانند word2vec مقداردهی اولیه می کند.
- این بردارهای کلمه اولیه نشان دهنده Embeddingهایی هستند که هدف GloVe در طول آموزش بهینه سازی آنهاست.

3. تعریف تابع هدف:

- GloVe یک تابع هدف را تعریف می کند که شباهت بین بردارهای کلمه و شمارش همزمان آنها را در پیکره اندازه گیری می کند.
- هدف یادگیری بردارهای کلمه است به گونه ای که حاصل ضرب نقطه آنها لگاریتم احتمال وقوع همزمان کلمات را تقریب می کند.

4. بهینه سازی بردارهای کلمه:

- GloVe بردارهای کلمه را با به حداقل رساندن تفاوت بین حاصلضرب نقطه بردارهای کلمه و لگاریتم شمارش همزمانی بهینه می کند.
- این بهینه‌سازی معمولاً با استفاده از تکنیک‌های گرادیان نزول برای به‌روزرسانی بردارهای کلمه به صورت تکراری انجام می‌شود.

5. تولید Word Embeddings:

- پس از آموزش، GloVe با استخراج بردارهای کلمه آموخته شده از مدل، Word Embedding را تولید می کند.
- این Word Embedding معنی کلمات را در یک فضای برداری پیوسته نشان می‌دهند، جایی که کلمات با معانی مشابه به هم نزدیک‌تر هستند.

c) Word2Vec یک الگوریتم یادگیری بدون نظارت است که برای ایجاد Word Embedding استفاده می شود. هدف آن این است که معانی کلمات را با یادگیری بازنمودهای توزیع شده کلمات بر اساس کاربرد متنی آنها در مجموعه بزرگی از متن به دست آورد. دو معماری اصلی برای Word2Vec وجود دارد: Continuous Bag of Words (CBOW) و Skip-gram. در اینجا به بررسی نحوه ایجاد Word Embedding با استفاده از مدل Skip-gram میپردازیم:

1. تهیه داده های آموزشی:

- Word2Vec به مجموعه بزرگی از متن به عنوان ورودی نیاز دارد.
- برای مدل Skip-gram، داده‌های آموزشی با کشیدن پنجره‌ای با اندازه ثابت روی بدنه تولید می‌شوند. کلمه مرکزی در پنجره کلمه هدف است و کلمات متنی در پنجره به عنوان ورودی استفاده می شود.

2. معماری مدل Skip-gram:

- مدل Skip-gram از یک شبکه عصبی با یک لایه پنهان تشکیل شده است. لایه ورودی بیانگر کلمات زمینه و لایه خروجی نشان دهنده کلمه هدف است.

- لایه پنهان حاوی کلمه embeddings است که پارامترهایی هستند که در طول آموزش باید یاد بگیرند.
- مدل با به حداکثر رساندن احتمال مشاهده کلمه مورد نظر با توجه به context آن، کلمه هدف را بر اساس کلمات context پیش بینی می کند.

3. آموزش Word Embedding:

- در طول آموزش، Word2Vec با تنظیم وزن شبکه عصبی به منظور به حداقل رساندن تفاوت بین کلمات هدف پیش بینی شده و واقعی، واژه های embeddings را یاد می گیرد.
- این فرآیند یادگیری به طور معمول با استفاده از تکنیک هایی مانند نزول گرادیان تصادفی (SGD) انجام می شود.
- تعبیه‌های کلمه به‌طور تکراری بر اساس جفت‌های کلمه متن-هدف مشاهده شده در داده‌های آموزشی به‌روزرسانی می‌شوند.

4. تولید Word Embedding:

- پس از آموزش، کلمه embeddings از لایه پنهان شبکه عصبی استخراج می شود.
- این تعبیه‌های کلمه معانی کلمات را در یک فضای برداری پیوسته نشان می‌دهند، جایی که کلمات مشابه به هم نزدیک‌تر هستند.
- کلمه embeddings را می توان به عنوان ویژگی در کارهای مختلف NLP مانند طبقه بندی متن، تجزیه و تحلیل احساسات و ترجمه ماشینی استفاده کرد.

d) کلمات با معانی چندگانه، که به عنوان کلمات چند معنایی نیز شناخته می‌شوند، چالش‌هایی را برای جاسازی کلمات ایجاد می‌کنند، زیرا ممکن است یک کلمه بسته به context آن معانی متفاوتی داشته باشد. کنترل این معانی چندگانه و گرفتن روابط معنایی ظریف بین آنها یک چالش مداوم در پردازش زبان طبیعی است. در اینجا چند رویکرد مورد استفاده برای مقابله با این چالش آورده شده است:

1. اندازه پنجره context: یکی از راه‌های کنترل چندمعنی، تنظیم اندازه پنجره context استفاده شده در طول آموزش است. یک پنجره بزرگتر ممکن است context های متنوع تری را برای یک کلمه ثبت کند و به مدل اجازه دهد بین حواس مختلف خود تمایز قائل شود.
2. Subword Embeddings: آنها مانند مواردی که توسط تکنیک‌هایی مانند FastText ایجاد می‌شوند، می‌توانند به ثبت تغییرات مورفولوژیکی و اطلاعات زیرکلمه کمک کنند. با نمایش کلمات به عنوان ترکیبی از واحدهای فرعی، این جاسازی‌ها می‌توانند با برداشت معانی مختلف در سطح زیرکلمه، کلمات چند معنایی را بهتر مدیریت کنند.
3. ابهام زدایی حس(sense disambiguation): تکنیک‌های ابهام‌زدایی حسی با هدف ابهام‌زدایی از معانی مختلف یک کلمه بر اساس context آن است. این می تواند شامل استفاده از منابع خارجی مانند فرهنگ لغت یا پایگاه های دانش برای شناسایی معنای صحیح یک کلمه در یک زمینه خاص باشد.
4. Multi-sense Embeddings: برخی از مدل های Word Embedding با یادگیری embedding های جداگانه برای هر حس، به صراحت چندین حس یک کلمه را مدل سازی می کنند. این رویکرد به مدل اجازه می دهد تا معانی

مختلف یک کلمه را به طور جداگانه دریافت کند و نمایش بهتر کلمات چند معنایی را ممکن می سازد.

5. Dynamic Contextual Embeddings: این مدل ها، مانند BERT و GPT، بازنمایی های کلمه را بر اساس کل متن یک جمله تولید می کنند. این مدل‌ها معنای پویای کلمات را بر اساس context اطراف آن‌ها دریافت می‌کنند که می‌تواند به ابهام‌زدایی کلمات چند معنایی کمک کند.

چالش های مرتبط با کلمات چند معنایی عبارتند از:

- ابهام: کلمات چند معنایی ابهام را به وظایف درک زبان طبیعی وارد می کنند و درک دقیق معنای مورد نظر یک کلمه را در یک زمینه خاص برای مدل ها چالش برانگیز می کند.
- پراکندگی داده: گرفتن معانی متعدد از یک کلمه نیاز به داده های آموزشی کافی دارد که زمینه های متنوعی را برای آن کلمه پوشش می دهد. با این حال، جمع آوری و حاشیه نویسی چنین داده هایی می تواند resource-intensive باشد.
- ارزیابی: ارزیابی عملکرد Word Embedding روی کلمات چند معنایی چالش برانگیز است زیرا به مجموعه داده های حاشیه نویسی با حواس چندگانه برای هر کلمه نیاز دارد که ممکن است به راحتی در دسترس نباشد.

e) در اینجا به بررسی روش های ایجاد Word Embedding برای کلمات OOV میپردازیم:

1. Embeddings از قبل آموزش دیده: از Embeddings کلمات از پیش آموزش‌دیده استفاده میکنیم. Word Embedding از پیش

آموزش‌دیده‌شده مانند Word2Vec، GloVe، FastText یا BERT بر روی مجموعه‌های بزرگ آموزش داده می‌شوند و اطلاعات معنایی غنی را به‌دست می‌آورند. این تعبیه‌ها بر اساس واژگان گسترده آموزش داده می‌شوند و اغلب می‌توانند بازنمایی معقولی برای کلمات OOV بر اساس شباهت‌های مورفولوژیکی یا زمینه‌ای آن‌ها با کلماتی که در طول آموزش دیده می‌شوند، ارائه دهند.

2. Subword Embeddings: تکنیک‌هایی مانند FastText نه تنها برای کلمات کامل، بلکه برای واحدهای زیرکلمه (مانند کاراکتر n-gram) جاسازی ایجاد می‌کنند. کلمات OOV را می توان به واحدهای زیر کلمه تجزیه کرد و تعبیه آنها را می توان بر اساس جاسازی این زیرکلمه ها محاسبه کرد. این رویکرد از شباهت‌های مورفولوژیکی برای ایجاد جاسازی برای کلمات OOV استفاده می‌کند.
3. Character-level Embeddings: به جای تکیه بر تعبیه‌های سطح کلمه، کاراکترهای درون کلمات را می‌توان به عنوان Embeddings نشان داد، و Embeddings یک کلمه OOV را می‌توان بر اساس Embeddings کاراکترهای تشکیل‌دهنده آن محاسبه کرد. این رویکرد می‌تواند شباهت‌های مورفولوژیکی و آوایی بین کلمات را به دست آورد.
4. جستجوی نزدیکترین همسایه: با توجه به یک کلمه OOV، نزدیکترین کلمات را از واژگانی که در داده های آموزشی وجود داشت بر اساس برخی معیارهای شباهت (به عنوان مثال، شباهت کسینوس) پیدا میکنیم. سپس، Embedding های این نزدیک‌ترین کلمات را جمع کرده تا یک

Embedding برای کلمه OOV ایجاد شود. این رویکرد بر این فرض تکیه دارد که کلمات مشابه دارای Embedding های مشابهی هستند.

5. مدل های Fine-tuning: اگر به یک مدل زبان از پیش آموزش‌دیده مانند BERT یا GPT دسترسی داریم، می‌توانیم آن را روی داده‌های اضافی حاوی کلمات OOV تنظیم کنیم. تنظیم دقیق به مدل اجازه می دهد تا Embedding های خود را با ویژگی های خاص داده های جدید، از جمله کلمات OOV، تطبیق دهد.
6. Word Synthesis: یک مدل تولیدی، مانند رمزگذار خودکار متغیر (VAE) یا یک شبکه متخاصم مولد (GAN) را آموزش میدهیم تا Embeddings را برای کلمات OOV بر اساس context آنها ایجاد کند. این رویکرد شامل یادگیری یک نقشه برداری از فضای جاسازی های شناخته شده به فضای جاسازی های OOV است.

سوال 2: ماتریس خواسته شده به شرح زیر میباشد: (هر کلمه با دو کلمه بعد و قبل خودش مقایسه میشود)

	I	love	computer	science	and	NLP	even	more
I	0	2	1	1	1	1	0	0
love	2	0	1	1	1	1	1	0
computer	1	1	0	1	1	0	0	0
science	1	1	1	0	1	0	0	0
and	1	1	1	1	0	0	0	0
NLP	1	1	0	0	0	0	1	1
even	0	1	0	0	0	1	0	1
more	0	0	0	0	0	1	1	0