به نام خدا

تکلیف سری دوم NLP

بهاره کاوسی نژاد – 99431217

سوالات تئوری

1. به سوالات زیر پاسخ دهید:
   a. تفاوت بین one-hot encoding و word embedding ها را بیان کنید.

One-hot encoding و Embedding دو تکنیک متفاوت برای نمایش کلمات یا متغیرهای طبقه‌بندی (categorical variables) در NLP هستند.

One-hot encoding یک روش ساده و سرراست برای نمایش متغیرهای طبقه‌بندی است. در NLP، به هر کلمه در یک vocabulary یک index متمایز و یک بردار binary به طول vocabulary داده می‌شود. تمامی مقادیر این بردار به جز مقدار index آن کلمه دارای مقدار صفر هستند و مقدار این index برابر با 1 قرار داده می‌شود. در اصل، هر کلمه به عنوان یک sparse vector نشان داده می‌شود که در آن تنها یک المان hot (یک) است و بقیه المان‌ها cold (صفر) هستند. این رمزگذاری ماهیت گسسته کلمات را حفظ می‌کند و هر کلمه را مستقل از همه کلمات دیگر در نظر می‌گیرد.

به بیان دیگر، One-hot encoding یک روش کلی برای vectorize کردن هر ویژگی دسته‌بندی (categorical feature) است. در این روش، ایجاد و به روز رسانی vectorization ساده و سریع است؛ تنها کافیست که یک entry جدید به بردار با یک ورودی به ازای هر دسته اضافه نمود. با این وجود، سرعت و سادگی با ایجاد یک بعد جدید برای هر دسته منجر به "curse of dimensionality" می‌شود.

از سوی دیگر، word embeddingها بازنمایی dense vectorهایی هستند که روابط معنایی و نحوی بین کلمات را نشان می‌دهند. Word Embeddingها معمولا از داده‌های متنی بزرگ با تکنیک‌هایی مانند Word2Vec، GloVe یا FastText آموخته می‌شوند. این تکنیک‌ها از شبکه‌های عصبی یا روش‌های matrix factorization استفاده می‌کنند تا کلمات را به فضای برداری پیوسته (continuous vector spaces) نگاشت کنند. در این فضاها کلمات مشابه یا بردارهایی نزدیک بهم در فضا نشان داده می‌شوند. Word Embeddingها semantic meaning و contextual information را دریافت می‌کنند و امکان نمایش بهتر شباهت کلمات و مقایسه بین آن‌ها را فراهم می‌کنند.

به عبارت دیگر، Embedding یک روش است که به مقدار زیادی هم در کل حجم داده‌ها و هم در تکرار نمونه‌ها و هم مدت طولانی آموزش نیاز دارد. نتیجه آن یک بردار dense با تعداد ثابت و دلخواه ابعاد است.

این دو روش همچنین در مرحله پیش بینی متفاوت هستند: one-hot encoding چیزی از sentiment نمونه‌ها نمی‌گوید؛ هر بردار یک نمایش متعامد در بعد دیگری است. Embedding مواردی که معمولا همزمان اتفاق می‌افتند را با هم گروه‌بندی می‌کند.

اگر داده‌های آموزش، زمان train و توانایی اعمال الگوریتم train پیچیده‌تر (مانند word2vec و GloVe) وجود داشته باشد، بهتر است از embedding استفاده شود.

تفاوت‌های اصلی بین one-hot encoding و word embeddings:

- ابعاد: One-hot encoding نمایشی با ابعاد بالا و پراکنده ایجاد می‌کند و در این روش ابعاد برابر با اندازه واژگان است. از سوی دیگر word embeddingها معمولا ابعاد کمتری دارند (100، 200 یا 300) و کلمات را به صورت بردارهای متراکم نشان می‌دهند.
- اطلاعات معنایی: one-hot encoding هیچ رابطه معنایی بین کلمات را در نظر نمی‌گیرد زیرا هر کلمه مستقل تلقی می‌شود. از سوی دیگر، word embeddingها، اطلاعات معنایی را در فضای برداری encode می‌کنند و امکان اندازه‌گیری شباهت و گرفتن روابط معنایی بین کلمات را فراهم می‌کنند.
- حافظه و راندمان محاسباتی: one-hot encoding به مقدار زیادی حافظه برای ذخیره بردارهای پراکنده با ابعاد بالا نیاز دارد. در مقابل، word embeddingها از نظر حافظه کارآمدتر هستند زیرا از بردارهای متراکم با ابعاد پایین‌تر استفاده می‌کنند. علاوه بر این، محاسباتی که شامل word embedding است در مقایسه با one-hot encoding به دلیل ماهیت متراکم بردارها سریعتر است.

b. به طور خلاصه توضیح دهید که الگوریتم GloVe چگونه word embedding ها را تولید می‌کند.

الگوریتم GloVe یا Global Vectors با استفاده از دو ایده اصلی word embeddingها را ایجاد می‌کند:

- Local Context Window: مشابه سایر تکنیک ها، GloVe مجموعه ای از متن را تجزیه و تحلیل می‌کند و پنجره ای از کلمات را در اطراف یک کلمه مرکزی در نظر می‌گیرد.
- Global Co-occurrence Matrix: GloVe یک ماتریس بسیار بزرگ می‌سازد که در آن سطرها و ستون‌ها کلمات را نشان می‌دهند. هر سلول در ماتریس تعداد دفعات تکرار دو کلمه را در پنجره ثبت می‌کند و معیاری آماری از رابطه آنها ارائه می دهد.

دو مورد بیان شده اینگونه کنار هم قرار می‌گیرند:

- Capturing Relationships: ماتریس همزمانی (co-occurrence matrix) روابط بین کلمات را رمزگذاری (encode) می‌کند. کلماتی که باهم ظاهر می‌شوند اغلب معانی مشابهی دارند.
- Training on Co-occurrence Ratios: به‌جای شمارش عادی، GloVe بر نسبت همزمانی (co-occurrence) بین یک کلمه و دو کلمه متنی متفاوت تمرکز می‌کند. این کمک می کند تا شباهت معنایی به طور مؤثرتری دریافت شود.
- Optimization and Vector Creation: از طریق یک فرآیند بهینه‌سازی، GloVe نمایش‌های برداری برای هر کلمه ایجاد می‌کند. این بردارها به گونه ای قرار گرفته‌اند که نسبت‌های همزمانی

مشاهده شده در ماتریس را منعکس می‌کند. کلماتی که الگوهای همزمانی مشابهی دارند، در این فضا بردارهایی نزدیک‌تر به هم خواهند داشت.

در اصل، GloVe آمار وقوع همزمان (co-occurrence statistics) را به نمایش‌های برداری معنی‌دار ترجمه می‌کند، جایی که کلمات مشابه در فضای برداری به یکدیگر نزدیک‌ترند.

c. به طور خلاصه توضیح دهید که الگوریتم Word2Vec چگونه word embedding ها را تولید می‌کند.

Word2Vec، بر خلافGloVe ، شامل دو تکنیک اصلی برای ایجاد word embeddingها است:

1. Continuous Bag-of-Words یا CBOW: در این رویکرد، CBOW کلمات context اطراف را بر اساس یک کلمه هدف مرکزی پیش بینی می‌کند. تصور کنید جمله ای با یک جای خالی در وسط آن وجود دارد. CBOW کلمات اطراف را تجزیه و تحلیل می‌کند و سعی می‌کند پیش بینی کند که چه کلمه‌ای باید جای خالی را پر کند. مدل با تجزیه و تحلیل اینکه چگونه کلمات اطراف را به خوبی پیش بینی می‌کند، روابط معنایی بین کلمه هدف و زمینه آن را می‌آموزد.
2. Skip-gram: این روش مفهوم CBOW را تغییر می‌دهد. در اینجا، مدل کلمات اطراف را بر اساس یک کلمه هدف پیش‌بینی می‌کند. بنابراین به جای پیش بینی context از یک کلمه هدف، کلمات همسایه را با یک کلمه خاص پیش بینی می‌کند. این روش به طور کلی موثرتر در نظر گرفته می‌شود، به خصوص برای مجموعه داده‌های بزرگتر.

هر دو روش CBOW و Skip-gram از شبکه های عصبی کم عمق برای انجام این پیش بینی‌ها استفاده می‌کنند. با آموزش بر روی یک corpus بزرگ، شبکه vector representation کلمات را برای پیش بینی بهتر زمینه‌های اطراف اصلاح می‌کند. کلماتی با زمینه‌های مشابه در نهایت دارای نمایش‌های برداری مشابه هستند. به عبارت ساده تر، Word2Vec از رویکرد "predict the neighbors" برای درک معنای یک کلمه بر اساس context آن در داده‌های آموزشی استفاده می‌کند. این به قرار دادن کلمات با معانی مشابه در مجاورت فضای برداری کمک می‌کند.

d. واژ ه‌های دارای چند معنی چگونه در word embedding ها کنترل می‌شوند و چه چالش‌هایی را تولید می‌کنند؟

چالش‌ها:

1. Averaging Effect: هنگامی که یک مدل در زمینه‌های مختلف با یک واژه polysemous (واژگان چند معنی) روبرو می‌شود، embedding ممکن است به میانگین هردو معنا تبدیل شود. این موضوع باعث ایجاد برداری می‌شود که هیچ کدام از معناها را به درسی دریافت نمی‌کند.
2. Context Dependence: در حالت ایده آل، embedding باید به context اطراف برای تعیین معنای مورد نظر بستگی داشته باشد. با این حال، یک single embedding نمی‌تواند هر دو زمینه را برای هر کلمه چندمعنی به‌طور کامل به تصویر بکشد.

3. Disambiguation Needed: از آنجایی که خود embeddingهای کلمه ذاتاً معانی متعددی را مدیریت نمی‌کنند، تکنیک‌های بیشتری برای ابهام‌زدایی از معنای یک کلمه بر اساس context مورد نیاز است. این اغلب شامل استفاده از مدل‌ها یا تکنیک‌های جداگانه متمرکز بر ابهام‌زدایی از معنای کلمه است.

رویکردهای به کار گرفته شده برای حل مشکل polysemy:

- Multi-Sense Embeddings: این رویکرد برای هر معنای متمایز یک کلمه، embeddingهای جداگانه ایجاد می‌کند. این نیاز به اطلاعات یا تکنیک‌های اضافی برای شناسایی معنای مورد استفاده کلمه دارد.
- Contextualized Embeddings: این embeddingها در هنگام ایجاد یک بردار کلمه، context اطراف را به طور واضح‌تری در نظر می‌گیرند. این می‌تواند به embedding در جهت معنای مورد نظر بر اساس زمینه کمک کند.

در این مقاله موضوع کلمات چند معنایی بیشتر بررسی شده است.

e. word embeddingها نیاز دارند که تمام کلمات در مجموعه آموزش حضور داشته باشند. چگونه با کلمات خارج از واژگان (out of vocabulary) برخورد می‌کنید؟ یک روش برای تولید word embedding برای کلماتی که در داده آموزش حاضر نبوده‌اند پیشنهاد دهید.

روش‌های مدیریت کلمات خارج از واژگان:

1. Special Unknown Token یا UNK: یک رویکرد معمول اختصاص یک token ویژه، مانند "<UNK>" به همه کلمات خارج از واژگان است. این توکن embedding vector خود را دارد، اما ممکن است چندان informative نباشد.
2. Subword Embeddings: این روش کلمات را به واحدهای معنی‌دار کوچکتر مانند پیشوندها، پسوندها یا کاراکترها تقسیم می‌کند. این واحدهای subword با embeddings آموزش داده می‌شوند و سپس embedding یک کلمه خارج از واژگان با ترکیب embeddingهای subwordهای آن ساخته می‌شود. این کار به مدل اجازه می‌دهد تا با ترکیب و اشتراک subwordها با کلمات شناخته شده، کلمات دیده نشده را مدیریت کند.
3. Morphological Analysis (در صورت وجود): برای زبان‌هایی با ریخت‌شناسی غنی (rich morphology) ، که کلمات با افزودن پیشوندها و پسوندها تشکیل می‌شوند، می توان ساختار کلمه خارج از واژگان را تجزیه و تحلیل کرد. با بررسی کردن embeddingهای تکواژهای شناخته شده (morphemes یا meaningful word parts)، می‌توان یک embedding معقول برای کل کلمه دیده نشده ایجاد کرد.

4. Context-based Embedding Prediction: تکنیک‌های جدیدتر از context اطراف کلمه خارج از واژگان برای پیش‌بینی embedding آن استفاده می‌کنند. این کار از جاسازی embedding کلمات نزدیک برای استنتاج معنای کلمه دیده نشده استفاده می‌کند.

باید توجه داشت که:

- اثربخشی این روش‌ها به task خاص و حجم واژگان آموزشی بستگی دارد.
- برای کلمات بسیار نادر، حتی این روش‌ها ممکن است embeddingهای بسیار دقیقی را ارائه ندهند.
- به روزرسانی مداوم واژگان با کلمات جدید می تواند توانایی مدل را در مدیریت کلمات دیده نشده در طول زمان بهبود بخشد.

روش‌های بیشتری در این لینک موجود است.

2. ماتریس co-occurrence متن زیر را بنویسید. اندازه پنجره را 2 در نظر بگیرید.

I love computer science and I love NLP even more.

word	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	0	0	0	0
science	1	1	0	0	1	0	0	0
and	1	1	1	1	0	0	0	0
NLP	1	1	0	0	0	0	1	1
even	1	1	1	0	0	1	0	1
more	0	0	0	0	0	1	1	0

منابع:

- https://datascience.stackexchange.com/questions/29851/one-hot-encoding-vs-word-embedding-when-to-choose-one-or-another
- https://www.nature.com/articles/s41598-023-40062-3
- https://www.linkedin.com/advice/0/how-can-you-handle-out-of-vocabulary-words-nlp