به نام خدا

تكليف سري دوم NLP بهاره كاوسي نژاد – 99431217

سوالات تئوری

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

One-hot encoding و Embedding دو تكنيك متفاوت براي نمايش كلمات يا متغيرهاي طبقه‌بندي (categorical variables) در NLP هستند.

One-hot encoding يك روش ساده و سرراست براي نمايش متغيرهاي طبقه‌بندي است. در NLP، به هر كلمه در يك vocabulary يك index متمايز و يك بردار binary به طول vocabulary داده مي‌شود. تمامي مقادير اين بردار به جز مقدار index آن كلمه داراي مقدار صفر هستند و مقدار اين index برابر با 1 قرار داده مي‌شود. در اصل،‌ هر كلمه به عنوان يك sparse vector نشان داده مي‌شود كه در آن تنها يك المان hot (يك) است و بقيه المان‌ها cold (صفر) هستند. اين رمزگذاري ماهيت گسسته كلمات را حفظ مي‌كند و هر كلمه را مستقل از همه كلمات ديگر در نظر مي‌گيرد.

به بيان ديگر، One-hot encoding يك روش كلي براي vectorize كردن هر ويژگي دسته‌بندي (categorical feature) است. در اين روش، ايجاد و به روز رساني vectorization ساده و سريع است؛ تنها كافيست كه يك entry جديد به بردار با يك ورودي به ازاي هر دسته اضافه نمود. با اين وجود،‌ سرعت و سادگي با ايجاد يك بعد جديد براي هر دسته منجر به "curse of dimensionality" مي‌شود.

از سوي ديگر، word embeddingها بازنمايي dense vectorهايي هستند كه روابط معنايي و نحوي بين كلمات را نشان مي‌دهند. Word Embeddingها معمولا از داده‌هاي متني بزرگ با تكنيك‌هايي مانند Word2Vec، GloVe يا FastText آموخته مي‌شوند. اين تكنيك‌ها از شبكه‌هاي عصبي يا روش‌هاي matrix factorization استفاده مي‌كنند تا كلمات را به فضاي برداري پيوسته (continuous vector spaces) نگاشت كنند. در اين فضاها كلمات مشابه يا بردارهايي نزديك بهم در فضا نشان داده مي‌شوند. Word Embeddingها semantic meaning و contextual information را دريافت مي‌كنند و امكان نمايش بهتر شباهت كلمات و مقايسه بين آن‌ها را فراهم مي‌كنند.

به عبارت ديگر، Embedding يك روش است كه به مقدار زيادي هم در كل حجم داده‌ها و هم در تكرار نمونه‌ها و هم مدت طولاني آموزش نياز دارد. نتيجه آن يك بردار dense با تعداد ثابت و دلخواه ابعاد است.

اين دو روش همچنين در مرحله پيش بيني متفاوت هستند: one-hot encoding چيزي از sentiment نمونه‌ها نمي‌گويد؛ هر بردار يك نمايش متعامد در بعد ديگري است. Embedding مواردي كه معمولا همزمان اتفاق مي‌افتند را با هم گروه‌بندي مي‌كند.

اگر داده‌هاي آموزش، زمان train و توانايي اعمال الگوريتم train پيچيده‌تر (مانند word2vec و GloVe) وجود داشته باشد، بهتر است از embedding استفاده شود.

تفاوت‌هاي اصلي بين one-hot encoding و word embeddings:

* ابعاد: One-hot encoding نمايشي با ابعاد بالا و پراكنده ايجاد مي‌كند و در اين روش ابعاد برابر با اندازه واژگان است. از سوي ديگر word embeddingها معمولا ابعاد كمتري دارند (100، 200 يا 300) و كلمات را به صورت بردارهاي متراكم نشان مي‌دهند.
* 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 شاملword embedding است در مقایسه با one-hot encoding به دلیل ماهیت متراکم بردارها سریعتر است.

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

الگوريتم GloVe يا Global Vectors با استفاده از دو ايده اصلي word embeddingها را ايجاد مي‌كند:

* 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 بسيار 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 بيان شده اينگونه كنار هم قرار مي‌گيرند:

* Capturing Relationships: ماتريس همزماني (co-occurrence matrix) روابط بين كلمات را رمزگذاري (encode) مي‌كند. كلماتي كه باهم ظاهر مي‌شوند اغلب معاني مشابهي دارند.
* Training on Co-occurrence Ratios: به‌جای شمارش عادي، GloVe بر نسبت همزماني (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 همزماني 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 يكديگر نزديك‌ترند.

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

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

1. Continuous Bag-of-Words يا 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.

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

چالش‌ها:

1. Averaging Effect: هنگامي كه يك مدل در زمينه‌هاي مختلف با يك واژه polysemous (واژگان چند معني) روبرو مي‌شود، embedding ممكن است به ميانگين هردو معنا تبديل شود. اين موضوع باعث ايجاد برداري مي‌شود كه هيچ كدام از معناها را به درسي دريافت نمي‌كند.
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 embeddingهاي 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.

رويكردهاي به كار گرفته شده براي حل مشكل polysemy:

* Multi-Sense Embeddings: اين رويكرد براي هر معنای متمایز یک کلمه، embeddingهاي 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 [اين مقاله](https://www.nature.com/articles/s41598-023-40062-3) موضوع كلمات چند معنايي بيشتر بررسي شده است.

1. word embedding ها 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.

روش‌هاي مديريت كلمات خارج از واژگان:

1. Special Unknown Token يا 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 كار به مدل اجازه می‌دهد تا با تركيب و اشتراك subwordها با کلمات شناخته شده، کلمات ديده 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 بررسي کردن embedding‌های تکواژهای شناخته شده (morphemes يا meaningful word parts)، می‌توان یک embedding معقول برای کل کلمه ديده 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 ديده نشده استفاده می‌کند.

بايد توجه داشت كه:

* 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 ديده نشده در طول زمان بهبود بخشد.

روش‌هاي بيشتري در [اين لينك](https://www.linkedin.com/advice/0/how-can-you-handle-out-of-vocabulary-words-nlp) موجود است.

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

I love computer science and I love NLP even more.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| more | even | NLP | and | science | computer | love | I | word |
| 0 | 0 | 1 | 1 | 1 | 1 | 2 | 0 | I |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 2 | love |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | computer |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | science |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | and |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | NLP |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | even |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | more |

**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