به نام خدا

تمرین سری دوم

درس مبانی پردازش زبان و گفتار

استاد درس: دکتر مرضیه داودآبادی

سینا علی‌نژاد

۹۹۵۲۱۴۶۹

سوال ۱-

a) هر دو، روشی برای نشان دادن دیتای متنی به صورت وکتور عددی هستند تا بتوان آن را به عنوان ورودی به الگوریتم یادگیری ماشین داد. از لحاظ پیچیده بودن، بهینه بودن، در نظر گرفتن یا نگرفتن رابطه بین کلمات و نیازمند آموزش بودن با هم تفاوت دارند. Word embedding بهینه تر است مخصوصا وقتی دایره لغات ما بزرگ است، روش one-hot encoding برای نشان دادن هر کلمه، به تعداد بیت معادل سایز دایره لغات نیازمند است. روش word embedding ارتباط میان کلمات را در نظر میگیرد، برای مثال کلمه dog و cat به دلیل حیوان بودنشان، اگر آموزش خوب انجام شده باشد، نزدیک به یکدیگر خواهند بود، در حالیکه در روش one-hot encoding فاصله همه کلمات برابر است با رادیکال ۲. همچنین این روش وابسته به داده است و برای موثر بودن، نیازمند داده بیشتر خواهد بود اما در one-hot چنین نیست. روش one-hot ساده تر و سریعتر از لحاظ پیاده‌سازی است.

b) این روش از یک text corpus بزرگ برای ساخت co-occurrence matrix استفاده میکند. این ماتریس نشان می‌دهد کلمات چند بار در یک پنجره با سایز مشخص، همزمان وجود داشته‌اند. الگوریتم با استفاده از این ماتریس، شباهت کلمات را محاسبه میکند. این الگوریتم با استفاده از یک تابع ریاضی مخصوص، یک رابطه بین ضرب داخلی وکتور کلمات و مقادیر ماتریس co-occurrence ایجاد می‌کند و در یک فرایند iterative مقادیر word vector ها را اصلاح می‌کند به گونه‌ای که مقدار پیش‌بینی شده و مقدار واقعی در co-occurrence matrix به یکدیگر نزدیک شوند.

c) روش word2vec، برخلاف GloVe از شبکه عصبی برای یادگیری word embeddings استفاده می‌کند. این روش یک متن بزرگ را گرفته و کلمه به کلمه وارد یک شبکه عصبی کم عمق میکند و این کار را با دو معماری اصلی انجام میدهد. اولی CBOW است که در آن هدف، پیش‌بینی کلمه مرکز با استفاده از کلمات اطراف است. با بررسی اینکه مدل چقدر در پیش‌بینی کلمه درست موفق است، وکتورهای کلمات ورودی اصلاح میشود. دومین مدل، حالت Skip-gram است که در آن هدف، پیش‌بینی کلمات اطراف یک کلمه در یک پنجره با سایز مشخص می‌باشد. در نهایت word embedding هایی خواهیم داشت که کلماتی که در محتواهای یکسان آمده بودند، شباهت بیشتری با یکدیگر خواهند داشت.

d) با وجود مزایای زیاد، word embedding ها در در نظر گرفتن چند معنایی بودن کلمات در نمایش وکتوری آنها مشکل دارد. برای مثال کلمه bank اگر در اطراف کلمه money بیاد، معنای مالی این کلمه تقویت میشود و اگر در متن دیگر در کنار river بیاید، معنای دیگر آن یعنی riverside تقویت

شده و از معنای مالی فاصله میگیرد. در نتیجه اگر این کلمه بیشتر در معنای مالی در متون آموزش آمده باشد، بیشتر همین معنا را خواهد داشت و معنای دیگر آن کم‌رنگ خواهد بود. این مسئله به صورت کامل حل نشده است و در بعضی موارد برای کلماتی که یک معنای غالب دارند، همین معنی بیشترین تاثیر را در نمایش وکتوری آن کلمه دارد.

e) از چندین روش میتوان این مشکل را تا حدی رفع کرد.

- روش اول اینکه از مدل زبانی براساس کاراکتر استفاده کنیم. در این صورت کلا ۳۲ حرف داریم و دیگر به مشکل oov نمی‌خوریم.
- روش دوم استفاده از subword tokenization است، برای مثال BPE. در این صورت، حالات مختلف کلمات نیازی به حضور در Vocab ندارند. برای مثال اگر کلمه lier در vocab نباشد، در subword tokenization به صورت lie + er خواهد بود و اینها دارای معنا هستند.
- روش سوم این است که از یک دیکشنری برای استخراج کلمات هم معنی با کلمه جستجو استفاده کنیم و در صورتی که این کلمات در vocab وجود داشتند، از وکتور مربوط به آنها استفاده کنیم.

سوال ۲-

	I	love	Computer	Science	and	NLP	even	more	<total>
I	0	2	0	0	1	0	0	0	3
Love	2	0	1	0	0	1	0	0	4
Computer	0	1	0	1	0	0	0	0	2
Science	0	0	1	0	1	0	0	0	2
and	1	0	0	1	0	0	0	0	2
NLP	0	1	0	0	0	0	1	0	2
even	0	0	0	0	0	1	0	1	2
more	0	0	0	0	0	0	1	0	1
<total>	3	4	2	2	2	2	2	1	