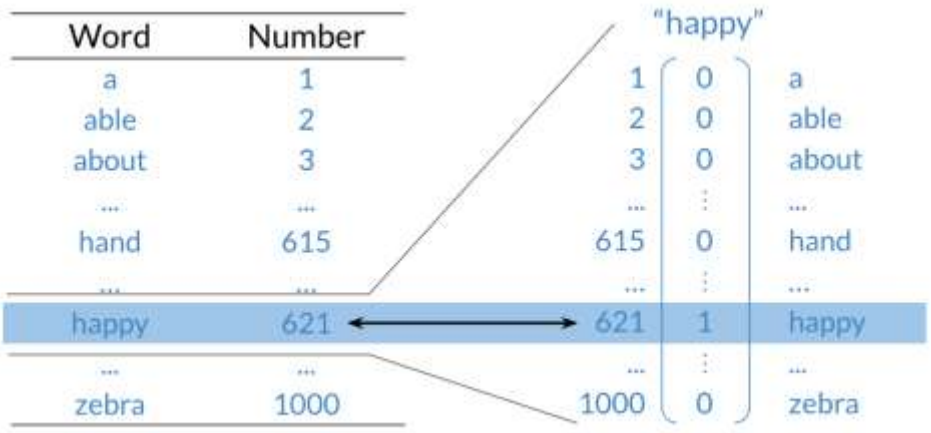
به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

تمرین ۲ – تئوری

۱)

a) نمایش one-hot vector(encode) به این صورت است که فقط از 0,1 استفاده میکنیم.برای هر کلمه یک vector در نظر میگیریم و شماره آنرا در vocab پیدا میکنیم و آنرا در وکتور مربوط به آن کلمه برابر ۱ قرار میدهیم و مابقی ایندکس ها را برابر 0 قرار میدهیم.



برای مثال تصویر بالا را مدنظر قرار میدهیم؛ کلمه happy شماره ۶۲۱ در دیکشنری یا vocabulary ما میباشد ، مقدار آنرا در وکتوری که برای این کلمه در نظر گرفتیم برابر ۱ قرار میدهیم و مابقی کلمات همانطور که مشخصند برابر صفر میشوند.

در word embedding اما به نحوه دیگری عمل میکنیم.در این روش کلماتی که از لحاظ معنایی به هم نزدیکند شباهت بیشتری در وکتور خود دارند.در این روش برای

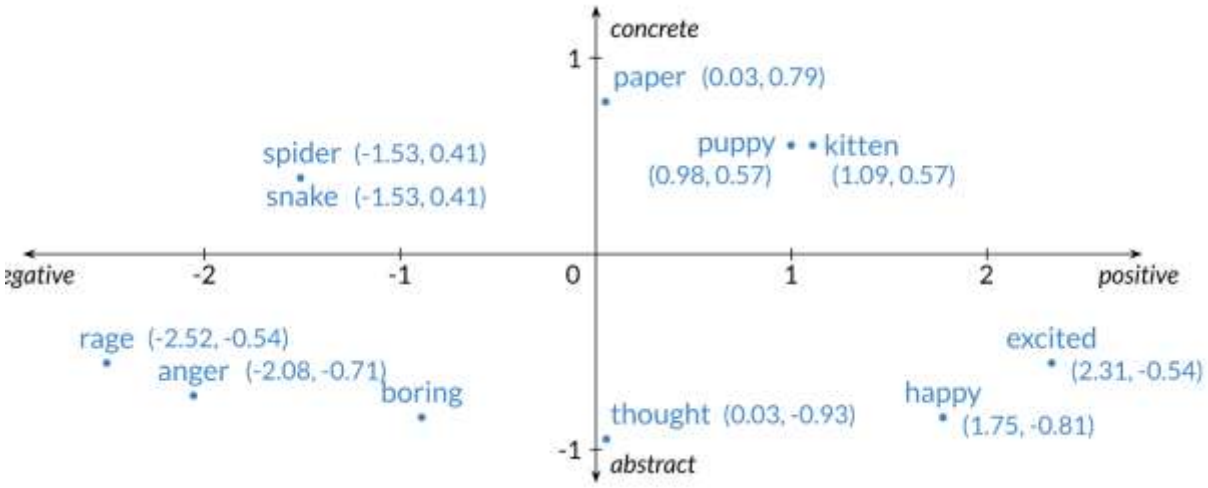
هر وکتور که تشکیل میدهیم ابعاد زیادی دارد. (مثلا بین ۱۰۰ تا ۱۰۰۰) که کلمات را براساس معیار هایی میسنجیم و هر معیار را برابر یک بعد از آن وکتور برای کلمه در نظر میگیریم.

از معایب روش one-hot encode این بود که کلمات از لحاظ معنایی و شباهت هایشان نمیتوانیم بسنجیم.

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97

Activate Windows
Go to Settings to activate

به طور مثال در تصویر بالا مشاهده میکنیم؛ برای مثلا معیار یا فیچر Royal کلمات king , queen دارای مقایر بالایی هستند و میتوانند از این حیث مربوط به هم باشند.



این شکل نیز نمایی را نشان میدهد از وکتور مربوط به هر کلمه که در دو بعد و بر اساس دو معیار ساخته شده است.

از دیگر برتری های روش word embedding نسبت به one hot encode این است که تعداد ابعاد مربوط به هر کلمه در این روش کمتر است. در روش one hot همانطور که پیش تر ذکر شد اندازه وکتور مربوط به هر کلمه به اندازه تعداد کلمات vocab میباشد اما گفتیم در روش word embedding معمولا بین ۱۰۰ تا ۱۰۰۰ میباشد.

b) در این الگوریتم ابتدا ماتریس همزمانی(co-occurnce) کلمات را با هم تشکیل میدهیم. به این صورت که یک ماتریس به اندازه vocabSize * vocabSize در نظر میگیریم و بررسی رخداد کلمه i,j را با هم در نظر میگیریم.

	I	love	Learnerea	DataScience
I	0	2	0	0
love	2	0	1	1
Learnerea	0	1	0	0
DataScience	0	1	0	0

i

j

I love Learnerea

I love DataScience

در این مثال این کار را انجام داده ایم.

در این الگوریتم برای بررسی occurrence بین کلمات باید یک اندازه برای window در نظر بگیریم و تعداد occurrence بین کلمات را در آن شعاع در نظر بگیریم.

در مرحله بعد باید matrix factorization انجام دهیم؛ که باعث کاهش فضای ابعاد در وکتور مربوط به هر کلمه میشود.

در نهایت نیز وکتور مربوط به هر کلمه برای ما ساخته میشود و کلمات با معانی مشابه دارای وکتور های مشابه نیز هستند.

c) به طور کلی در روش word2vec کلمات را بر اساس متن و کلماتی که در اطراف یک کلمه هستند پیش بینی میکنیم.این الگوریتم به گونه ایست که بر اساس وکتور با ابعاد زیاد میتواند فاصله کلمات از هم(از لحاظ معنایی) را بسنجد و با توجه به کلمات اطراف یک کلمه را پیش بینی میکند.
از ویژگی ها جالب این الگوریتم این است که میتوانیم بر روی وکتور مربوط به کلمات عملیات های ریاضی انجام دهیم.
این الگوریتم به دو زیر بخش تقسیم میشود که کاربردهای متفاوتی دارند:
۱) الگوریتم CBOW.بر اساس کلمات اطراف یا context words ها کلمه هدف را پیش بینی میکنیم.

۲) الگوریتم skip-gram : بر عکس حالت قبلی میباشد و بر اساس کلمه هدف یا target word میتواند context words را پیش بینی کند.

d) در word embedding ها به هر کلمه یک وکتور اختصاص داده میشود.حال اگر یک کلمه دارای دو یا چند معنی متفاوت باشد منطقا وکتور مربوط به آن نیز باید متفاوت باشد اما این چنین نمیشود و میتواند معضلاتی را به همراه داشته باشد و باعث شود که کلمات نامناسب باتوجه به context اصلی انتخاب شوند و متن را مبهم کنند که این موارد از چالش های این اتفاق هستند.

از روش هایی که برای این اتفاق پیشنهاد میشود :

الف) contextual embeddings : این روش در Bert نیز استفاده میشود بدین صورت است که معنی یک کلمه را براساس کل جمله در نظر میگیرد که با این اتفاق کلمات دارای چند معنی بر اساس جمله ای که در آن حضور دارند انتخاب و درک میشوند و وکتور مناسب انتخاب میشود.

ب)sense disambiguation : بدین منظور است که برای کلمات با چند معنی ابهام زدایی میکنیم و مشخص میکنیم یک کلمه با چند معنی ، با کدام معنی با توجه به context باید در جمله قرار بگیرد. این کار را میتوانیم با قرار دادن چند وکتور برای یک کلمه که معانی متفاوتی دارد انجام دهیم.

ت) روش دیگری که در الگوریتم fastText استفاده میشود subword representation میباشد که برای این چالش ها نیز میتواند مفید باشد.

e) برای کلماتی که در هنگام training مدل با آنها مواجه نشده روش های متفاوتی میتوان پیشنهاد داد:

الف) اولین روش مربوط به الگوریتم fastText میباشد که subword های کلمه را میسازد.

```
kingdom = ['k','in','kin','king','kingd','kingdo','kingdom',...]
```

همانطور که مشخص است tokenization در این روش به صورت مابقی روش ها که معمولا بر اساس کلمات است صورت نمیگیرد و حتی یک کلمه را به اجزای کوچکتری

میشکند. به همین دلیل هنگامی که یک کلمه جدید ببینیم وکتور مربوط به آن کلمه بر اساس این subword ها ساخته میشود که میتواند OOV ها به این صورت هندل کند.

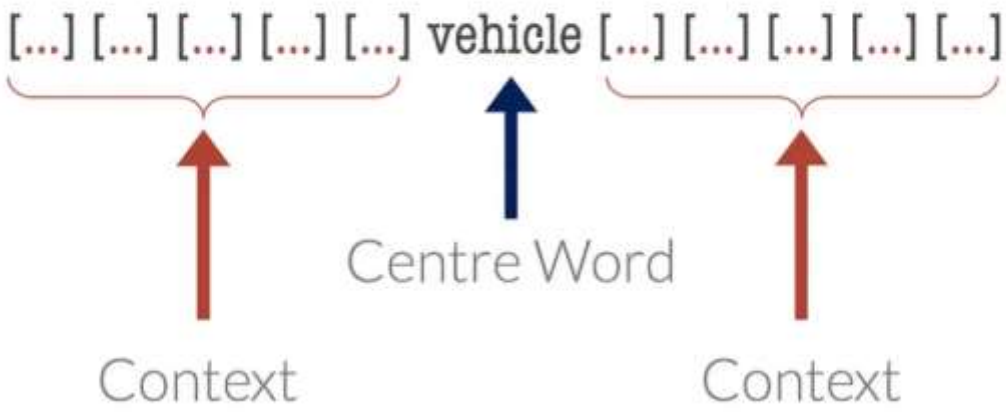
ب) روش دیگری که پیشنهاد میشود این است که در دیکشنری خود کلمه ای مثلا به نام <unk> قرار دهیم و هنگامی که کلمه جدیدی در متن دیدیم به جای آن unk ببینیم.

پ) روش دیگر این است که هنگامی که کلمه جدیدی دیدیم به سراغ نزدیک ترین کلمه مشابه آن برویم.البته این روش دقیق نیست و ممکن است خطا محسوسی داشته باشد اما در مواردی نیز میتواند کمک حال ما باشد.

Sentence : I love computer science and I love nlp even more.

Window size = 2

	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



منابع :

https://medium.com/intelligentmachines/word-embedding-and-one-hot-encoding-ad17b4bbe111

https://www.elastic.co/what-is/word-embedding

https://aman.ai/primers/ai/word-vectors/#global-vectors-for-word-representation-glove

https://medium.com/swlh/co-occurrence-matrix-9cacc5dd396e

coursera/NLP

youtube