تمرین سری سوم
NLP

یاسمین مدنی
۹۷۵۳۲۲۶۵

فهرست

سوال ۱

مرحله اول

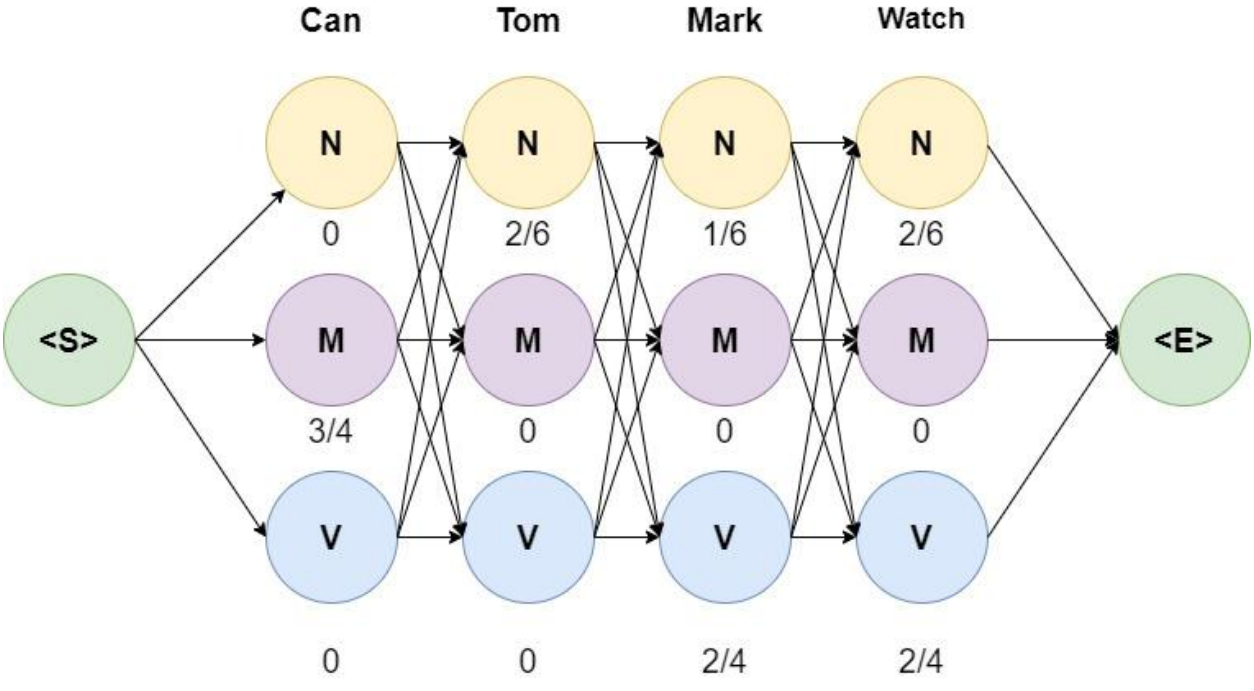
word	Noun	Modal	Verb
Mark	1/6	0	2/4
Tom	2/6	0	0
Can	0	3/4	0
Watch	2/6	0	2/4
Will	1/6	1/4	0

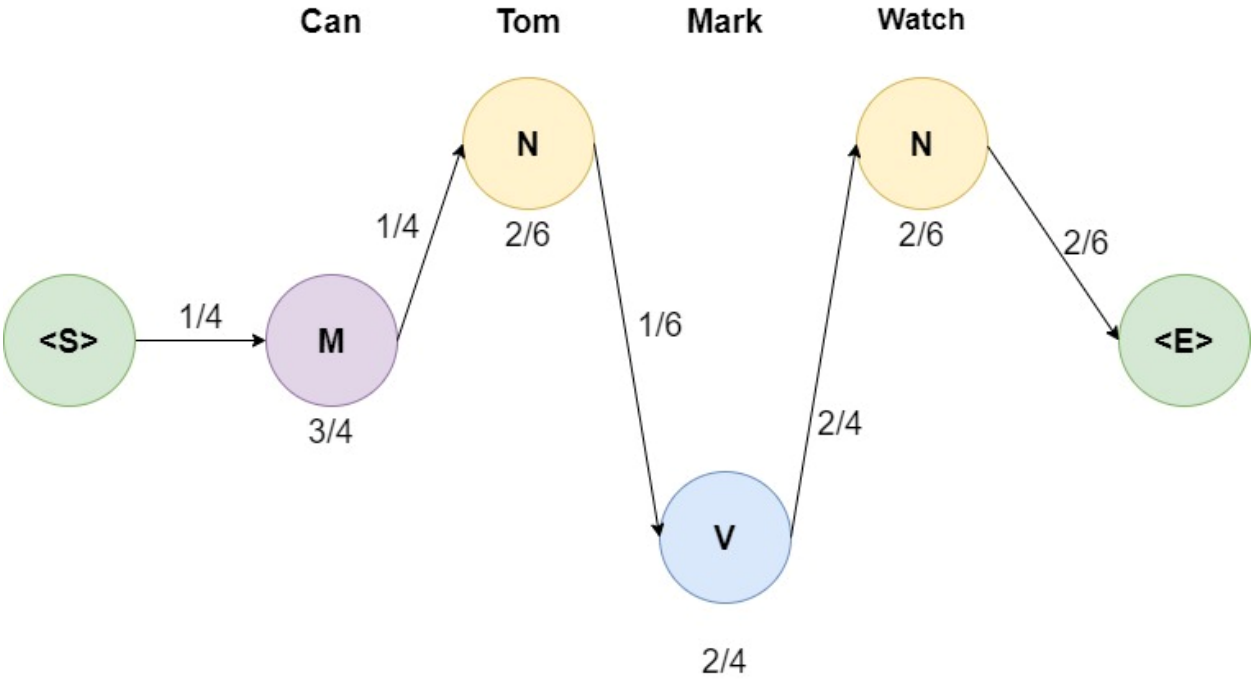
مرحله دوم

- <S> Mark(N) can(M) watch(V) <E>
- <S> Will(N) can(M) mark(V) watch(N)<E>
- <S> Can(M) Tom(N) watch(V) <E>
- <S> Tom(N) will(M) mark(V) watch(N)<E>

	Noun	Modal	Verb	<E>
<S>	3/4	1/4	0	0
Noun	0	3/6	1/6	2/6
Modal	1/4	0	3/4	0
Verb	2/4	0	0	2/4

مرحله سوم





P= 1/4 * 3/4 * 1/4 * 2/6 * 1/6 * 2/4 * 2/4 * 2/6 * 2/6 = 0.000072337

چون احتمال بزرگتر از ۰ است pos tagمناسب به صورت جدول زیر است.

Can	Modal
Tom	Noun
Mark	Verb
Watch	Noun

سوال ۲

نخست گرامر را به فرم نرمال چامسکی تبدیل می کنیم.از این رو بعضی از قواعد به صورت زیر باید تغییر کنند.

S → S S-Conj 0.2 S-Conj → Conj S 1.0

NP → John 0.04 | Jack 0.06 | pie 0.02 | cream 0.06 | cake 0.02 0.2

NP → NP Conj-NP 0.2 Conj-NP → Conj NP 1.0

VP → eat 0.08 | eats 0.12 | drinks 0.2 0.4

VP → Verb NP-NP 0.1 NP-NP → NP NP 1.0

John	eats	pie	with	cream	
Noun (0.2) /NP(0.04)	S (0.00384)	P (0.6)	-	S (0.00000041472)	John
	Verb (0.3) / VP (0.12)	VP (0.0018)	-	VP (0.00001296)	eats
		Noun (0.1) / NP (0.02)	-	NP (0.000144)	pie
			P (0.6)	PP (0.036)	with
				Noun (0.3) / NP (0.06)	cream

حالت های ممکن:

- S → NP VP → Noun VP → John VP → John Verb NP → John eats NP → John eats NP PP → John eats Noun PP → John eats pie PP → John eats pie P NP → John eats pie with NP → John eats pie with Noun → John eats pie with cream
- S → NP VP → Noun VP → John VP → John VP PP → John Verb NP PP → John eats NP PP → John eats Noun PP → John eats pie PP → John eats pie P NP → John eats pie with NP → John eats pie with Noun → John eats pie with cream

سوال ۳

یکی از مشکلات این روش این است که احتمالات را برای همه ی اجزا یکسان درنظر می گیرد و تفاوتی ندارد که کلمات در کجا ظاهر می شوند.کلمات خاص در زیرمجموعه های مختلف منجر به احتمالات متفاوت می شوند.

- John saw the man with the hat.

  VP => VBD NP, NP => NP PP

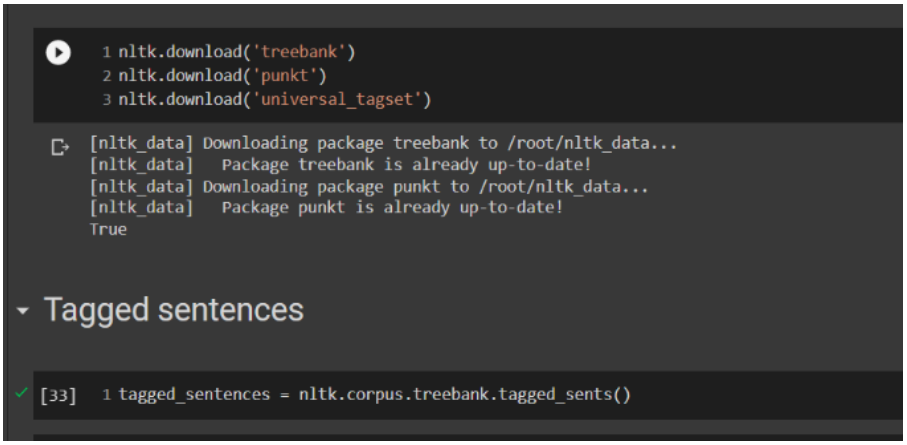
- John saw the moon with the telescope.

  VP => VBD NP PP

همانطور که در مثال بالا می بینیم مشخص نیست که کدام یک از گزینه ها فوق مورد نظر است. برای حل این مشکل میتوانیم ازLexical Information و از قواعد هد استفاده کنیم . اگر مشخصه متمایز در جای دیگری رخ دهد، می تواند به روش های دیگری تقسیم شود.همچنین سیستم های برچسب گذاری مختلف می توانند به مشکل کمک کنند.

سوال عملی

برای این پیاده سازی ابتدا نیاز داریم تا یک مجموعه برچسب گذاری شده را دانلود و ایمپورت کنیم.



```
1 nltk.download('treebank')
2 nltk.download('punkt')
3 nltk.download('universal_tagset')
```

```
[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data]   Package treebank is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

Tagged sentences

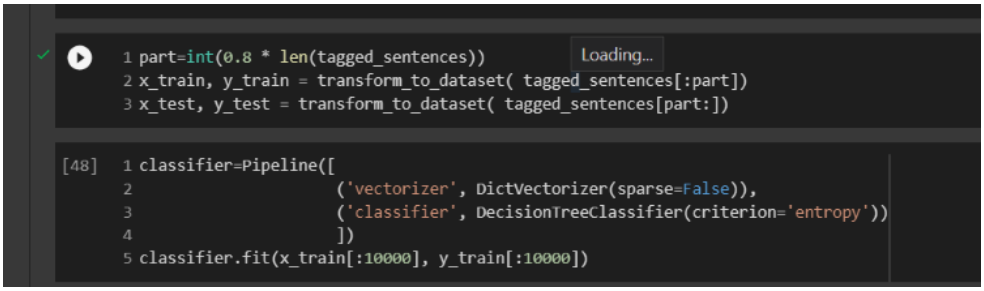
```
[33] 1 tagged_sentences = nltk.corpus.treebank.tagged_sents()
```

برای هر اصطلاح، بسته به جمله ای که عبارت از آن استخراج شده است، فرهنگ لغت ویژگی ها را ایجاد می کنیم. این ویژگی ها می تواند شامل اطلاعاتی در مورد کلمات قبلی و بعدی و همچنین پیشوندها و پسوندها باشد.

```python
def features(sentence, index):
  return {
  'word': sentence[index],
  'is_first': index == 0,
  'is_last': index == len(sentence) - 1,
  'is_capitalized': sentence[index][0].upper() == sentence[index][0],
   'is_all_caps': sentence[index].upper() == sentence[index],
  'is_all_lower': sentence[index].lower() == sentence[index],
  'prefix-1': sentence[index][0],
  'prefix-2': sentence[index][:2],
  'prefix-3': sentence[index][:3],
  'suffix-1': sentence[index][-1],
  'suffix-2': sentence[index][-2:],
  'suffix-3': sentence[index][-3:],
  'prev_word': '' if index == 0 else sentence[index - 1],
  'next_word': '' if index == len(sentence) - 1 else sentence[index + 1],
  'has_hyphen': '-' in sentence[index],
  'is_numeric': sentence[index].isdigit(),
  'capitals_inside': sentence[index][1:].lower() != sentence[index][1:]
   }
```

اکنون، مانند یادگیری ماشین، باید مجموعه داده را برای آموزش و آزمایش تقسیم کنیم.

اکنون، شبکه عصبی ما بردارها را به عنوان ورودی می گیرد، بنابراین باید ویژگی های فرهنگ لغت خود را به بردار تبدیل کنیم.



با استفاده از برچسب تریگرام یا استفاده از مجموعه داده متفاوت و بزرگتر می توانیم دقت را افزایش دهیم .