تمرین سری دوم کامیار مرادیان زه آب ۹۹۵۲۲۱۰۴

بخش تئوری

سوال اول

بخش الف) چالش‌هایی که پیش روی مسائل موجود در دستهٔ NER قرار دارند، به شرح زیر هستند:

۱. محدودهٔ هر موجودیت نام‌دار: در این دسته از مسائل، برخلاف مسائل موجود در دستهٔ POS، با کلمات به عنوان توکن سروکار نداریم، بلکه با بخش‌هایی از خود متن طرف هستیم که نیاز است مشخص کنیم که آیا بخش مورد نظر در یک موجودیت نام‌دار قرار دارد یا خیر. بنابراین نخستین چالشی که با آن مواجه هستیم مشخص کردن محدودهٔ هر موجودیت است. برای این چالش، راهکارهایی نظیر BIO Tagging و یا نسخه‌های دیگر آن وجود دارند که این امکان را می‌دهند که توکن‌های هر متن را به صورت لغت در نظر بگیریم.

۲. ابهام و چندمعنایی: برخی از لغات مانند Bank وجود دارند که چندین معنی دارند و به عبارتی در رابطه با این لغات با مشکل Polysemy سروکار داریم. این ابهام در رابطه با لغات دیگر که تنها یک معنا دارند نیز وجود دارد. به عنوان مثال کلمه‌ای مانند JFK که می‌تواند به عنوان فرودگاه یا اسم شخص بکار رود و یا کلمهٔ The که می‌تواند در اسم یک محل و یا اسم یک سازمان قرار گیرد. بنابراین معنای لغات ممکن است براساس متنی که در آن قرار گرفته شده است، متفاوت باشد.

۳. داده‌ها و آماده‌سازی آن‌ها: سیستم‌های NER نظارت‌شده به مقدار زیادی داده‌های نشانه‌گذاری‌شده برای آموزش نیاز دارند. نشانه‌گذاری داده‌ها زمان‌بر است و نیاز به تخصص حوزه‌ای قابل توجهی دارد که می‌تواند هزینه‌بر و غیرعملی باشد، به ویژه برای زبان‌های کم‌منبع یا حوزه‌های تخصصی.

۴. استفاده از اطلاعات زمینه‌ای: سیستم‌های NER مؤثر باید اطلاعات زمینه‌ای غیرمحلی را برای بهبود دقت ادغام کنند. این امر شامل درک بافت گسترده‌تری است که یک کلمه یا عبارت در آن ظاهر می‌شود، به جای تجزیه و تحلیل آن به تنهایی. به عنوان مثال در زبان انگلیسی، این نکته که لغات موجود در یک موجودیت نام‌گذاری‌شده اغلب به صورت کپیتال نوشته می‌شوند، می‌تواند کمک‌کننده باشد.

۵. سازگاری با نام‌های موجود جدید و تکامل زبان: سیستم‌های NER باید به طور مداوم برای شناسایی نام‌های موجود جدید و تغییرات در استفاده از زبان با گذر زمان سازگار شوند. این نیازمند مکانیزم‌های یادگیری و سازگاری مداوم است که اجرای مؤثر آن‌ها می‌تواند چالش‌برانگیز باشد.

بخش ب) بافت متن و مفهوم کلی آن، نقش مهمی در دقت سیستم شناسایی موجودیت‌های نام‌دار (NER) که در حال طراحی آن هستیم، ایفا می‌کند. از جملهٔ تاثیرات می‌توان به موارد زیر اشاره کرد:

- بافت خاص حوزه: سیستم‌های NER معمولاً خاص حوزه هستند، به این معنی که بر روی نوع خاصی از متن، مانند اسناد مالی یا پرونده‌های پزشکی، آموزش داده شده‌اند. اگر بافت متن مورد پردازش با حوزه‌ای که سیستم NER بر آن آموزش دیده متفاوت باشد، ممکن است دقت آن کاهش یابد. به عنوان مثال، سیستم NER آموزش دیده بر روی اسناد مالی ممکن است در شناسایی نام نهادها در یک سند پزشکی با مشکل مواجه شود.
- ابهام در بافت: همانطور که گفته شد، بافت و مفهوم متن بر روی سیستم طراحی شده تاثیر می‌گذارد. از جملهٔ این تاثیرات آن است که متن می‌تواند اصطلاحات یا عبارات مبهمی داشته باشد که بسته به بافت، به شکل‌های مختلفی

تفسیر شوند. به عنوان مثال، کلمه "Bank" می‌تواند به یک موسسه مالی یا کنار رودخانه اشاره کند. سیستم NER که به بافت متن توجه نکند، ممکن است این اصطلاحات مبهم را به درستی طبقه‌بندی نکند.

- چندزبانگی: سیستم‌های NER می‌توانند برای کار با چند زبان طراحی شوند. اگر بافت متن مورد پردازش به زبانی باشد که سیستم NER بر آن آموزش ندیده است، ممکن است دقت آن کاهش یابد. به عنوان مثال، سیستم NER آموزش دیده بر روی زبان انگلیسی با احتمال بالایی در شناسایی نام نهادها در متنی به زبان اسپانیایی با مشکل مواجه شود.

بخش ج) با توجه به محدودیت‌های موجود در HMM، الگوریتم CRF ارائه شد که در آن به جای اتکا به چند ویژگی از پیش تعریف شده، ویژگی‌های متنوعی را می‌توان ایجاد کرد تا به جواب بهتری برای مسائل دست یافت. همچنین الگوریتم CRF به عنوان حالت جامع‌تر و کامل‌تر روش HMM شناخته می‌شود که می‌توان با اندکی تنظیم ویژگی‌های آن، ان را به روش HMM نیز تبدیل کرد. از جمله محدودیت‌های موجود در الگوریتم HMM و تفاوت آن با الگوریتم CRF در زیر لیست شده‌اند.

- تمایز بین مدل تفکیکی و مدل مولد: CRF یک مدل تفکیکی است، در حالی که HMM یک مدل مولد است. این بدان معنی است که CRF احتمال شرطی $P(y|x)$ را که در آن y برچسب و x بردار ویژگی است، مستقیماً مدل می‌کند. HMM، از طرف دیگر، احتمال مشترک $P(x, y)$ را مدل می‌کند که ممکن است برای برخی از وظایف کمتر قابل تفسیر و پیچیده‌تر باشد. به عبارتی در این مدل‌ها به دنبال چگونگی ساخت داده‌ای که در حال حاضر با آن سروکار داریم، به همان نحوی که در متن به ما داده شده است سروکار داریم. در حالی که در روش CRF صرفاً با ارتباط بین تگ و لغات داده شده سروکار داریم. بنابراین CRF اغلب برای وظایفی که احتمال شرطی مرتبط‌تر است، مانند برچسب‌گذاری توالی، که هدف آن اختصاص دادن برچسب به هر عنصر در یک توالی با توجه به بافت عناصر اطراف آن است، مناسب‌تر است.
- ویژگی‌های ثابت موجود در مدل‌های HMM و انعطاف‌پذیری مدل‌های CRF: CRF امکان تعریف توابع ویژگی انعطاف‌پذیرتری را فراهم می‌کند که می‌توانند روابط پیچیده‌تری بین متغیرهای ورودی و خروجی را به دست آورند. در مقابل، HMMها به انواع خاصی از توابع ویژگی، مانند احتمالات انتقال و احتمالات انتشار، محدود هستند. این می‌تواند CRF ها را در مدل‌سازی وابستگی‌ها و روابط پیچیده‌تر قدرتمندتر کند.
- تاثیرپذیری از متن: مدل HMM آموزش‌دیده، هیچ توجهی به محل قرارگیری یک تگ و یا لغت متناظر با خود نمی‌کند. به عبارتی برای چنین مدلی، اهمیت ندارد که وقتی دو تگ Noun و Verb پشت سر هم داریم، در کجای جمله قرار گرفته است و هر کجا که باشد ویژگی Transition Probability متناظر با آن دو را به یک میزان تغییر می‌دهد. این مورد برای Emission Probability نیز صادق است. به عبارتی برای این مدل اهمیتی ندارد که یک واژهٔ منتسب به یک تگ در کجای متن قرار دارد و در هر کجا که باشد بدون توجه به کلمات و تگ‌های اطراف آن، احتمال آن را تغییر می‌دهد. این در حالی است که با توجه به این که در CRF می‌توانیم ویژگی‌های مختلفی را ایجاد کنیم، پس می‌توان مشکل ذکرشده را مرتفع کرد و ارتباط متنی را نیز مدل کرد.
- مقابله با سوگیری برچسب: CRFها نشان داده‌اند که در حضور سوگیری برچسب، که مشکل رایجی در وظایف برچسب‌گذاری توالی است، عملکرد بهتری نسبت به HMM ها دارند. سوگیری برچسب زمانی رخ می‌دهد که توزیع برچسب‌ها در داده‌های آموزشی با توزیع برچسب‌ها در داده‌های آزمایشی متفاوت باشد. CRFها می‌توانند این مسئله را به طور موثرتری با مدل‌سازی احتمال شرطی برچسب‌ها با توجه به ویژگی‌ها، که می‌تواند رابطه واقعی بین برچسب‌ها و ویژگی‌ها را بهتر منعکس کند، مدیریت کنند.

بخش د) مجموعه تگ‌های موجود در Penn Treebank به صورت زیر است:

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	infinitive to	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	*+, %, &*	WRB	wh-adverb	*how, where*

Figure 8.2 Penn Treebank part-of-speech tags.

با توجه به این جدول هر یک از موارد گفته شده را بررسی می‌کنیم:

۱. Atlanta غلط است. همانطور که در جدول مشخص است، اسم‌های خاص مانند Atlanta دارای برچسب NNP هستند.
۲. dinner غلط است. NNS برای لغات جمع است در حالی که این لغت مفرد بوده و برچسب درست برای آن، برچسب NN است که برای لغات مفرد و یا اسم‌های عام استفاده می‌شود.
۳. have غلط است. چرا که فعل استفاده شده یک فعل برای مفرد غیرسوم شخص است. و با توجه به جدول نیاز است که برای آن، از VBP استفاده شود.
۴. can غلط است. چرا که can یک Modal بوده و نیاز است که برای آن از MD استفاده کنیم و نه VBP که برای افعال غیر سوم شخص استفاده می‌شود.

بخش ه) همانطور که گفته شد در مسائل NER برخلاف مسائل موجود در دستهٔ POS، با کلمات به عنوان توکن سروکار نداریم، بلکه با بخش‌هایی از خود متن طرف هستیم که نیاز است مشخص کنیم که آیا بخش مورد نظر در یک موجودیت نام‌دار قرار دارد یا خیر. بنابراین نخستین چالشی که با آن مواجه هستیم مشخص کردن محدودهٔ هر موجودیت است. برای این چالش، راهکارهایی نظیر BIO Tagging و یا نسخه‌های دیگر آن نظیر BIOES و IO وجود دارند که این امکان را می‌دهند که توکن‌های هر متن را به صورت لغت در نظر بگیریم.

در روش BIO است که به ما امکان می‌دهد تا NER را مانند یک وظیفه برچسب‌گذاری توالی کلمه به کلمه در نظر بگیریم، از طریق برچسب‌هایی که هم مرز و هم نوع نام نهاد را به دست می‌دهند. در برچسب‌گذاریBIO ، هر توکنی که شروع کننده یک بازه مورد علاقه باشد را با برچسب 'B' برچسب‌گذاری می‌کنیم، توکن‌هایی که در داخل یک بازه قرار دارند را با برچسب 'I' برچسب‌گذاری می‌کنیم، و هر توکنی که خارج از هر بازه مورد علاقه باشد را با برچسب 'O' برچسب‌گذاری می‌کنیم. برچسب‌گذاری BIO می‌تواند همان اطلاعات مورد نیاز برای POS و یا NER را به ما تحویل دهد، اما با این حال این مزیت را برای ما فراهم می‌آورد که می‌توانیم با استفاده از آن فرآیند NER را به فرآیند POS شبیه کرده و آن را آسان کنیم. به طوری که در آن به هر یک از کلمات ورودی یک برچسب را نسبت می‌دهیم. دو روش دیگر که در کنار BIO وجود دارد، روش‌های IO و نیز BIOES هستند. برچسب گذاریIO ، که با حذف برچسب B برخی اطلاعات را از دست می دهد، و برچسب گذاری

BIOES، که برچسب پایان E را برای پایان یک بازه و برچسب S را برای بازه ای که تنها یک کلمه دارد، اضافه می کند. در زیر یک مثال یکسان که با کمک این سه روش برچسب‌گذاری شده‌اند آورده شده است که از داخل کتاب مرجع Jurafski گرفته شده است.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Figure 8.7 NER as a sequence model, showing IO, BIO, and BIOES taggings.

بخش عملی

سوال اول

در این مسئله به دنبال آموزش یک مدل جهت پیش‌بینی برچسب‌های POS برای یک دادهٔ متنی هستیم. برای این منظور از مجموعهٔ دادهٔ brown استفاده شده است. در وب‌سایت nltk در خصوص این مجموعهٔ داده به صورت زیر توضیح داده شده است:

> مجموعه دادهٔ *Brown* اولین مجموعهٔ دادهٔ الکترونیکی یک میلیون کلمه‌ای انگلیسی بود که در سال ۱۹۶۱ در دانشگاه *Brown* ایجاد شد. این مجموعهٔ داده شامل متن از ۵۰۰ منبع است و منابع بر اساس ژانر مانند خبر، سرمقاله و غیره دسته‌بندی شده‌اند.

در این مجموعهٔ دادگان متنی هر لغت به همراه برچسب POS مخصوص خود قرار گرفته است که می‌توانیم از آن‌ها استفاده کنیم. جهت استفاده از این مجموعهٔ داده نیاز است که در ابتدا، تابع generate_dict را تکمیل کنیم که با دریافت لیستی از داده به شکل گفته شده، برای هر لغت تعداد دفعاتی که یک برچسب خاص خورده است شمرده شده و نتیجه را برمی‌گرداند.

برای پیش‌بینی برچسب هر لغت نیز به این صورت عمل می‌کنیم که برای یک واژه، برچسبی که بیشترین میزان استفاده را داشته است را برای آن انتخاب می‌کنیم. برای لغات ناشناخته نیز دو رویکرد وجود دارد: اول اینکه بدون توجه به کلمه، برچسب NN را به آن بدهیم از آنچه که تعداد بیشتری از لغات در این دسته قرار می‌گیرند، و رویکرد دوم آن است که بر اساس ساختار لغت، برچسب مناسب را به آن لغت بدهیم. این قوانین به صورت زیر هستند:

```
- 'VBG' (verb, gerund) for words ending in 'ing'
```

```
- 'NP$' (noun, possessive) for words ending in "'s"
- 'NNS' (noun, plural) for words ending in 's'
- 'RB' (adverb) for words ending in 'ly'
- 'VBN' (verb, past participle) for words ending in 'ed'
- 'JJ' (adjective) for words matching certain patterns like
'ble', 'ish', 'ful', etc.
- 'CD' (cardinal numeral) for numeric strings
- 'NP' (noun, proper singular) for capitalized words
```

نتایج نهایی بدست آمده به صورت زیر است:

```
length of training set:     75415
length of testing set:      25139
intersection:               3429
Assuming that all unknown words are NN
>> accuracy: 0.8312184255539202
With additional rules for unknown words
>> accuracy: 0.8754127053582084
1110 more words got correctly classified.
```

بنابراین با استفاده از قوانینی که از یک دانش پیشین نشئت گرفته شده‌اند می‌توانیم بهتر عمل کنیم و مدل قوی‌تری را ایجاد کنیم.

سوال دوم

برای این مسئله مانند سوال پیشین از مجموعه دادهٔ Brown استفاده می‌کنیم با این تفاوت که این بار از مجموعه برچسب‌های Universal Tag برای برچسب‌گذاری هر یک از این دادگاه استفاده می‌نماییم.

این مجموع برچسب به صورت زیر است:

```
VERB - verbs (all tenses and modes)
NOUN - nouns (common and proper)
PRON - pronouns
ADJ - adjectives
ADV - adverbs
ADP - adpositions (prepositions and postpositions)
CONJ - conjunctions
DET - determiners
NUM - cardinal numbers
PRT - particles or other function words
X - other: foreign words, typos, abbreviations
. - punctuation
```

در گام نخست تابع collect_probabilities را جهت آموزش مدل ایجاد می‌کنیم. با اجرای این قطعه کد بر روی مجموعه داده‌های ورودی هر دو مجموعه احتمالات Transition و Emission بدست می‌آیند. علاوه بر این دو، احتمالات اولیهٔ هر یک از برچسب‌ها که بر اساس تعداد دفعات حضور آن‌ها در متن بدست آمده است، به غیر از توکن اول محاسبه می‌شود.

تابع بعدی، تابع create_confusion_matrices است. در این تابع، یک مجموعه تگ‌های صحیح را به همراه تگ‌های پیشبینی شده برای یک جمله را به عنوان ورودی دریافت می‌کنیم. ماتریس مذکور برای هر یک از تگ‌های موجود در مجموعهٔ تگ‌های در دسترس، چهار مقدار را دربردارد. این چهار مقدار عبارتند از:

```
True Positive (TP): The number of positive instances that were
correctly predicted as positive by the model.
True Negative (TN): The number of negative instances that were
correctly predicted as negative by the model.
False Positive (FP): The number of negative instances that were
incorrectly predicted as positive by the model.
False Negative (FN): The number of positive instances that were
incorrectly predicted as negative by the model.
```

بنابراین برای این که هر یک از این مقادیر را محاسبه کنیم به این صورت عمل می‌کنیم:

- هر لغت که به درستی پیشبینی شده است، TP برچسب آن را افزایش می‌دهیم و TN تمام برچسب دیگر را نیز یک واحد افزایش می‌دهیم.
- در غیر این صورت، FP برچسب پیش‌بینی شده را افزایش می‌دهیم، و همینطور FN برچسب واقی لغت را نیز یک واحد بیشتر می‌کنیم. در پایان TN سایر برچسب‌ها را یک واحد بیشتر می‌کنیم.

پس از آموزش مدل، برای انجام عملیات نتیجه‌گیری و پیش‌بینی برای هر لغت موجود در جملات قرار گرفته در مجموعه داده‌ها، از Virtebi استفاده می‌کنیم. طبق این روش، برای هر جمله یک ماتریس Viterbi را تشکیل می‌دهیم. در ابتدا برای سطر ابتدایی از این ماتریس با توجه به اینکه لغت پیشینی وجود ندارد، با استفاده از احتمالات اولیهٔ pi و نیز احتمال emission هر لغت به ازای هر برچسب، برای هر برچسب احتمال منتسب به آن را بدست می‌آوریم. سپس برای هر برچسب بعدی، به ازای هر لغت، از فرمول زیر استفاده می‌کنیم:

```
viterbi[s,t] = viterbi[s', t-1] * a(s|s') * b_s(o_t)
```

که در آن b_s همان emission و نیز a همان transition است. برای بهینه کردن این رابطه به جای استفاده از ضرب عادی از لاگ آن استفاده می‌کنیم تا به صورت یک مجموع درآید.

در این بین که ماتریس را محاسبه می‌کنیم بهترین مسیر را نیز ذخیره‌سازی می‌کنیم. در پایان با شروع از بهترین برچسب برای آخرین واژه با استفاده از مسیر محاسبه شده به سمت عقب حرکت می‌کنیم تا به ابتدای جمله برسیم. به این ترتیب مجموعهٔ برچسب‌های پیش‌بینی شده را برای جملهٔ مورد نظر بدست می‌آوریم.

با اجرای قطعه کدهای پیاده‌سازی شده، به نتایج زیر می‌رسیم:

```
DET: {'TP': 2970, 'FP': 243, 'TN': 22139, 'FN': 30}
NOUN: {'TP': 6395, 'FP': 1407, 'TN': 18545, 'FN': 199}
ADJ: {'TP': 1309, 'FP': 119, 'TN': 23140, 'FN': 690}
VERB: {'TP': 2923, 'FP': 125, 'TN': 21529, 'FN': 687}
ADP: {'TP': 3159, 'FP': 632, 'TN': 21948, 'FN': 32}
.: {'TP': 3089, 'FP': 13, 'TN': 22050, 'FN': 0}
ADV: {'TP': 755, 'FP': 15, 'TN': 24061, 'FN': 323}
CONJ: {'TP': 716, 'FP': 3, 'TN': 24408, 'FN': 15}
PRT: {'TP': 294, 'FP': 12, 'TN': 24476, 'FN': 369}
```

```
PRON: {'TP': 608, 'FP': 2, 'TN': 24487, 'FN': 44}
NUM: {'TP': 345, 'FP': 1, 'TN': 24648, 'FN': 146}
X: {'TP': 4, 'FP': 0, 'TN': 25098, 'FN': 37}

Tag with the most false positives is: NOUN with 1407 counts.
Tag with the most false negative is:  ADJ with 690 counts.

model got 22567 samples correct out of 25139
accuracy: 0.8976888499940332
```

با بررسی نتایج به دست آمده به نتایج زیر می‌رسیم:

- مدل تعداد FPهای بیشتری را برای Nounها شناخته است که این مورد با توجه به حضور بیشتر آن‌ها در متون ارتباط دارد. به عبارتی آن‌ها در داده‌های متنی آموزشی حضور بیشتری نسبت به سایر لغات داشتند و بنابراین در صورتی که یک لغت دارای دو معنی یکی ADJ و دیگری Noun بوده باشد، برچسب Noun برای آن برگزیده خواهد شد.
- به دلیل آنکه بخشی از ADJها دارای چند معنی هستند یکی از آن‌ها حداقل Noun است بنابراین تعدادی از آن‌ها به عنوان Noun پیش‌بینی شده‌اند. دو مورد زیر از جمله مواردی است که در میان مجموعه داده‌های تستی بوده است:

```
sentence:    ['Only', 'public', 'understanding', 'and',
'support', 'can', 'provide', 'that', 'service', '.']
hidden s:    ['ADJ', 'ADJ', 'NOUN', 'CONJ', 'NOUN', 'VERB',
'VERB', 'DET', 'NOUN', '.']
predictions: ['ADJ', 'NOUN', 'NOUN', 'CONJ', 'NOUN', 'VERB',
'VERB', 'ADP', 'NOUN', '.']
```

همانطور که مشخص است public به اشتباه به عنوان Noun شناخته شده است.

- نمونهٔ دیگری از اشتباهات عمده اعداد هستند. اعداد به کرات به عنوان برچسب‌های دیگر مشاهده می‌شوند. به طوری که الگوی خاصی میان آن‌ها نیست. مانند نمونهٔ زیر که ۶۶ به عنوان Noun شناخته شده است:

```
sentence:       ['And', 'over', '66', 'per', 'cent', 'of', 'the',
'elementary', 'schools', 'with', '150', 'or', 'more', 'pupils',
'do', 'not', 'have', 'any', 'library', 'at', 'all', '.']
hidden s:       ['CONJ', 'PRT', 'NUM', 'ADP', 'NOUN', 'ADP',
'DET', 'ADJ', 'NOUN', 'ADP', 'NUM', 'CONJ', 'ADJ', 'NOUN',
'VERB', 'ADV', 'VERB', 'DET', 'NOUN', 'ADP', 'PRT', '.']
predictions:    ['CONJ', 'ADP', 'NOUN', 'ADP', 'NOUN', 'ADP',
'DET', 'ADJ', 'NOUN', 'ADP', 'NUM', 'CONJ', 'ADJ', 'NOUN',
'VERB', 'ADV', 'VERB', 'DET', 'NOUN', 'ADP', 'PRT', '.']
```

- نمونهٔ دیگر از اشتباهات، میان دو برچسب PRT و ADP است. وقتی که یک حرف اضافه به عنوان PRT پس از یک فعل قرار می‌گیرد اما الگوریتم آن را به عنوان یک ADP در نظر گرفته است. مانند نمونهٔ زیر:

```
sentence:       ['In', 'every', 'aspect', 'of', 'service', '--',
'to', 'the', 'public', ',', 'to', 'children', 'in', 'schools',
',', 'to', 'colleges', 'and', 'universities', '--', 'the',
'library', 'of', 'today', 'is', 'failing', 'to', 'render',
'vitally', 'needed', 'services', '.']
hidden s:       ['ADP', 'DET', 'NOUN', 'ADP', 'NOUN', '.', 'ADP',
'DET', 'NOUN', '.', 'ADP', 'NOUN', 'ADP', 'NOUN', '.', 'ADP',
```

```
'NOUN', 'CONJ', 'NOUN', '.', 'DET', 'NOUN', 'ADP', 'NOUN',
'VERB', 'VERB', 'PRT', 'VERB', 'ADV', 'VERB', 'NOUN', '.']
predictions:    ['ADP', 'DET', 'NOUN', 'ADP', 'NOUN', '.', 'ADP',
'DET', 'NOUN', '.', 'ADP', 'NOUN', 'ADP', 'NOUN', '.', 'ADP',
'NOUN', 'CONJ', 'NOUN', '.', 'DET', 'NOUN', 'ADP', 'NOUN',
'VERB', 'VERB', 'ADP', 'DET', 'NOUN', 'VERB', 'NOUN', '.']
```

- نمونهٔ دیگری که تعداد FNهای آن زیاد بوده است، برچسب X است که با توجه به آنکه در مجموعهٔ دادگان احتمال حضور آن بسیار کم بوده است، تعداد FNهای آن به مراتب بیشتر از TPهای آن بوده است که مشخص کنندهٔ این است که مدل به خوبی آن را آموزش ندیده است.
- همچنین برچسبی که تمام نمونه‌های آن به درستی پیش‌بینی شده است، علائم نگارشی بوده است که می‌توان گفت با توجه به این که محل آن‌ها در متون تا حدودی مشخص بوده و پس از افعال یا اسم‌ها است، مدل به درستی توانسته است که آن‌ها را تشخیص دهد.
- ADVها دستهٔ دیگری از برچسب‌ها هستند که FN آن‌ها بالا بوده است. به طوری که بیشترین میزان اشتباهات بر می‌گردد به حالاتی که ADVها به عنوان ADP پیش‌بینی شده‌اند. با بررسی‌های لازم مشخص شد، که در اکثر موارد اشتباه ADVهایی که پس از Nounها می‌آیند به عنوان ADP شناخته می‌شوند که این نیز باز می‌گردد به داده‌های آموزشی و این نکته که این لغات هم می‌توانستند که ADV باشند و هم ADP اما در داده‌های آموزشی بیشتر پس از Noun ظاهر شده‌اند. مانند as در مثال زیر:

```
sentence:       ['Food', ':', 'stew', 'a', 'la', 'Mulligatawny',
'Most', 'members', 'of', 'the', 'U.S.', 'Senate', ',', 'because',
'they', 'are', 'human', ',', 'like', 'to', 'eat', 'as', 'high',
'on', 'the', 'hog', 'as', 'they', 'can', '.']
hidden s:       ['NOUN', '.', 'NOUN', 'X', 'X', 'NOUN', 'ADJ',
'NOUN', 'ADP', 'DET', 'NOUN', 'NOUN', '.', 'ADP', 'PRON', 'VERB',
'ADJ', '.', 'VERB', 'PRT', 'VERB', 'ADV', 'ADV', 'ADP', 'DET',
'NOUN', 'ADP', 'PRON', 'VERB', '.']
predictions:    ['NOUN', '.', 'ADP', 'DET', 'NOUN', 'ADP', 'ADJ',
'NOUN', 'ADP', 'DET', 'NOUN', 'NOUN', '.', 'ADP', 'PRON', 'VERB',
'NOUN', '.', 'VERB', 'ADP', 'NOUN', 'ADP', 'NOUN', 'ADP', 'DET',
'NOUN', 'ADP', 'PRON', 'VERB', '.']
```

- همچنین ADVها نمونه‌هایی داشته است که به اشتباه Noun و یا ADJ شناخته شده است که دلیلش مانند آنچه که بالا ذکر شده است می‌باشد.
- تعداد زیادی از Verb هایی که به عنوان FN شمرده شده‌اند به عنوان Noun شناخته شده بودند. به عنوان مثال ممکن است که یک verb به صورت ingدار در جمله حضور داشته باشد، و یا حتی سوم شخص اما کلمه‌ای مشابه با یک کلمهٔ جمع تشکیل داده است. مانند دو نمونهٔ زیر:

```
sentence:       ['Crime', ':', "'", 'skyjacked', "'", 'From',
'International', 'Airport', 'in', 'Los', 'Angeles', 'to',
'International', 'Airport', 'in', 'Houston', ',', 'as', 'the',
'great', 'four-jet', 'Boeing', '707', 'flies', ',', 'is', 'a',
'routine', 'five', 'hours', 'and', '25', 'minutes', ',',
'including', 'stopovers', 'at', 'Phoenix', ',', 'El', 'Paso',
',', 'and', 'San', 'Antonio', '.']
hidden s:       ['NOUN', '.', '.', 'VERB', '.', 'ADP', 'ADJ',
'NOUN', 'ADP', 'NOUN', 'NOUN', 'ADP', 'ADJ', 'NOUN', 'ADP',
'NOUN', '.', 'ADP', 'DET', 'ADJ', 'ADJ', 'NOUN', 'NUM', 'VERB',
'.', 'VERB', 'DET', 'ADJ', 'NUM', 'NOUN', 'CONJ', 'NUM', 'NOUN',
```

```
'.', 'ADP', 'NOUN', 'ADP', 'NOUN', '.', 'NOUN', 'NOUN', '.',
'CONJ', 'NOUN', 'NOUN', '.']
predictions:    ['NOUN', '.', '.', 'NOUN', '.', 'ADP', 'ADJ',
'NOUN', 'ADP', 'NOUN', 'NOUN', 'ADP', 'ADJ', 'NOUN', 'ADP',
'NOUN', '.', 'ADP', 'DET', 'ADJ', 'NOUN', 'NOUN', 'NOUN', 'NOUN',
'.', 'VERB', 'DET', 'NOUN', 'NUM', 'NOUN', 'CONJ', 'NUM', 'NOUN',
'.', 'ADP', 'NOUN', 'ADP', 'NOUN', '.', 'NOUN', 'NOUN', '.',
'CONJ', 'NOUN', 'NOUN', '.']
```

```
sentence:       ['What', 'the', 'man', 'wanted', 'was', 'four',
'persons', 'to', 'volunteer', 'as', 'hostages', ',', 'along',
'with', 'the', 'crew', '.']
hidden s:       ['DET', 'DET', 'NOUN', 'VERB', 'VERB', 'NUM',
'NOUN', 'PRT', 'VERB', 'ADP', 'NOUN', '.', 'ADP', 'ADP', 'DET',
'NOUN', '.']
predictions:    ['DET', 'DET', 'NOUN', 'VERB', 'VERB', 'NUM',
'NOUN', 'ADP', 'NOUN', 'ADP', 'NOUN', '.', 'ADP', 'ADP', 'DET',
'NOUN', '.']
```

نتایج به دست آمده از این روش در مقایسه با آنچه که در سوال اول پیاده‌سازی کرده‌ایم دارای بهبود ۲-۳ درصدی است. این در حالی است که اگر هر یک از این دو مورد بهتر پیاده‌سازی می‌شدند، به احتمال بالاتری دست می‌یافتند. چرا که به طور معمول حالت پایه دارای درصد دقت ۹۲ درصدی است.

اما خب با توجه به این که احتمالات را از حالت unigram به حالت bigram افزایش دادیم باعث شده است که دقت نهایی بدست آمده بهبود بخشیده شود.

سوال سوم

بخش الف) از جمله چالش‌هایی که ممکن است به وجود آیند عبارتند از:

- همپوشانی میان موجودیت‌ها: در برخی از موارد ممکن است که دو یا چند نام با یک دیگر همپوشانی داشته باشند. در این صورت موجودیتی با طول کوچک‌تر به عنوان موجودیت اصلی برای یک عبارت انتخاب شده و به آن نسبت داده خواهد شد.
- همپوشانی متن با بخشی از موجودیت‌ها: ممکن است مواردی پیش آید که یک بخش از متن با یک بخش از یک موجودیت عنوان فیلم، همپوشانی داشته باشد. به عنوان مثال Black در مجموعهٔ متنی داده شده با واژهٔ Black Swan همپوشانی داشته است. در این موارد نیاز است که سراسر بخش بررسی شود. همچنین در صورتی که این همپوشانی بیشتر شده و سراسر متن را در برگیرد می‌توانیم بررسی کنیم که آیا لغت دیگری بعد از این بخش انتخاب شده وجود دارد که ادامه‌روی عبارت تشکیل شده است یا خیر. برای این منظور می‌توان از ویژگی واژگان کپیتال در انگلیسی استفاده کرد.
- تطابق متن با موجودیت‌های انتخاب شده: نیاز است که مجموعهٔ موجودیت‌های انتخاب شده به متن انتخابی و حوزهٔ مورد نظر مربوط باشد. درست مانند آنچه که در اینجا اتفاق افتاده است و هر دو مربوط به فیلم‌ها هستند.

- تناقضات حاصل از چندزبانی و یا چندشکلی بودن کلمه: ممکن است که یک فیلم در زبان‌های مختلف دارای اسم‌های مختلفی باشد. به عنوان مثال یک فیلم در زبان اسپانیایی یک اسم و در زبان انگلیسی یک اسم دیگر داشته باشد. بنابراین نیاز است که هر دو مجموعه چه مجموعهٔ هدف و چه مجموعهٔ انتخابی برای موجودیت‌ها از نظر زبانی شبیه هم باشند. همچنین مشکل دیگری که ممکن است رخ دهد، برمیگردد به این نکته که ممکن است یک کلمه در یک زبان دارای چند شکل املایی مختلف باشد. به عنوان مثال کلمه‌ای مانند رنگ به دوشکل در زبان انگلیسی بسته به لهجهٔ مورد استفاده نوشته می‌شود. بنابراین علاوه بر اینکه زبان متن با زبان مجموعه دادهٔ استفاده شده نیاز است که یکسان باشد، نیاز است معادل‌سازی‌های دیگر نظیر لهجه و مانند آن‌ها را نیز در نظر گرفت.
- موجودیت‌های چند کلمه‌ای: یک موجودیت ممکن است یک عبارت چند کلمه‌ای باشد. بنابراین نیاز است که برای مقابله با چنین مشکلی از راهکاری استفاده کنیم که هر کلمه را به عنوان یک توکن بررسی کرده و یک لیبل را به آن اختصاص دهیم مانند آنچه که در POS انجام می‌دهیم. برای این منظور می‌توانیم از BIO استفاده کنیم که پیش از این در سوالات دیگر توضیح داده شده است و همچنین برای این مسئله نیز مورد استفاده قرار گرفته است.
- وجود علائم نگارشی در موجودیت: ممکن است که در که در یک موجودیت، علائم نگارشی نیز مانند , یا . وجود داشته باشند. در این صورت جهت شناسایی چنین موجودیت‌هایی نیاز است که از روش‌های توکن‌سازی قوی‌تری استفاده کنیم و تنها نمی‌توانیم بر جداسازی یک عبارت با استفاده فاصلهٔ میان کلمات اتکا کنیم. در این مسئله، همین کار را کرده و از روش توکن‌سازی موجود در کتابخانهٔ nltk استفاده کرده‌ایم.
- یکسان‌سازی کردن توکن‌ها: توکن‌هایی که از هر دو بخش هدف و مجموعهٔ موجودیت‌های نام‌گذاری‌شده استفاده می‌کنیم نیاز است که یکسان باشد. در صورتی که تفاوتی میان این دو وجود داشته باشد با مشکل مواجه خواهیم بود.
- مواجهه با کلمات OOV: ممکن است که در متن موجودیت‌هایی وجود داشته باشد که ما نتوانیم ان‌ها را شناسایی کنیم. این موضوع برمی‌گردد به محدودیت‌های موجود در مجموعه دادهٔ موجودیت‌های نام‌گذاری شده. برای برطرف کردن این مشکل می‌توانیم از یک مجموعه دادهٔ بزرگ‌تر برای موجودیت‌های نام‌گذاری شده استفاده کنیم. هر چند که در این مسئله و با توجه به استفاده از یک مجموعه دادهٔ ۱۰۰۰ تایی نتوانستیم این مشکل را برطرف کنیم و برای رفع آن نیاز است که از یک مجموعه دادهٔ بزرگ‌تر برای موجودیت‌های نام‌گذاری شده استفاده کنیم.