Assignment NO.6 Solutions

NLP | Fall 1401 | Dr.Minayi

Student name : **Amin Fathi**

Student id : **400722102**

Problem 1

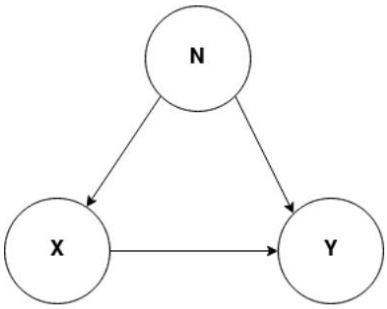
a)

از منظر علی سوگیری‌های نادرست از دو عامل مخدوش‌کننده سرچشمه می‌گیرند. ۱ pre-context (و ۲.entity-order)

pre-context به این معناست که مدل تحت تاثیر کلمات پیش‌زمینه قرار می‌گیرد که ممکن است در هنگام تولید یک کلمه موجودیت خاص، کلمات خارج از موجودیت باشند. برای مثال، در جمله‌ی I Still have much muscle pain and fatigue ، شیوه‌ی مولد autoregressive باعث تولید کلمه‌ی fatigue از muscle fatigue به شرط کلمات muscle و pain که extra-entity هستند شده است. این مسئله باعث می‌شود که مدل اشتباهاً وابستگی بین fatigue که intra-entity است و muscle و pain که extra-entityهستند را در نظر بگیرد؛ درحالی که وابستگی بین muscle و fatigue که intra-entity هستند را نادیده می‌گیرد. در نتیجه زمانی که تنها موجودیت masucle fatigue داده می‌شود، مدل به علت اینکه بایاس وابستگی نادرستی را یاد گرفته است، قادر به پیش‌بینی دقیق entity نخواهد بود .

entity-order به این واقعیت اشاره دارد که مدل در هنگام تولید یک دنباله موجودیت تحت تاثیر یک ترتیب از پیش تعیین شده موجودیت‌ها قرار می‌گیرد. موجودیت در یک جمله اساساً یک ساختار مجموعه‌ای بدون نظم رمزگشایی در میانشان وجود دارد. در مقابل، مدل NER مولد ترتیب رمزگشایی موجودیت‌ها را از قبل مشخص می‌کند تا سوگیری نادرست را معرفی می‌کند و وابستگی دوطرفه بین موجودیت‌ها را نادیده می‌گیرد. همان‌طور که در جمله‌ی Stallone is the actor of “Rocky” and “Rambo” ، پس از تثبیت مجموعه موجودیت‌ها، مدل تنها وابستگی یک طرفه‌ی Rambo به Stallone و Rocky بدون درنظر گرفتن معکوس مدل می‌کند. اگر در ابتدا Rambo رمزگشایی شود، به دلیل عدم وابستگی معکوس، رمزگشایی در موجودیت دیگر Stallone و Racky برای مدل دشوار است.

می‌توان علیت‌ها را در فرآیند تولید دنباله موجودیت با یک مدل علی ساختاری (SCM) فرموله کنیم. در شکل زیر پیوندهای مستقیم علیت بین دو گره نشان داده شده است.



معلول←علت. X→Y نشان‌دهنده‌ی فرآیند تولید دنباله‌ی هدف است که می‌توان آن را با توجه به مکان کلمات تولید شده به دو حالت inter-entity generation و intra-entity generation تقسیم کرد. N بیانگر کلمات pre-context است که می‌تواند بر روی تولید کلمات بعدی (N→Y) تاثیر بگذارد. پس از ورودی X توسط یک backdoor path به شکل X←N→Y، contaminated شده است؛ بنابراین N یک مخدوش‌کننده برای فرآیند X→Y است که یک سوگیری نادرست را به مدل معرفی می‌کند.

b)

ابتدا بر روی تولید کلمات در داخل موجودیت تمرکز می‌شود. رمزگشای autoregressive نیاز به رمزگشایی کلمه در مرحله‌ی فعلی دارد که مشروط به کلمات پیش‌زمینه که همان کلمات تولیدی هستند، است. کلمات پیش‌زمینه ممکن است در موجودیت‌های دیگری باشند که با موجودیت در حال تولید مرتبط نیستند؛ بنابراین، وابستگی‌های اشتباه را یاد می‌گیرد و بایاس را به مدل وارد می‌کند که در ادامه deconfounding درون موجودیتی با استفاده از داده افزایی معرفی شده است تا مخدوش‌کننده پیش‌زمینه را از بین ببرد.

طبق رابطه‌ی بیان شده در ادامه، مخدوش‌کننده‌ی N، کلمات پیش‌زمینه را طبقه‌بندی کرده و مدل را روی هر طبقه آموزش می‌دهیم.

$$P(Y \mid do(X)) = \sum_{n} P(Y \mid X, n)P(n)$$

برای جلوگیری از تاثیر سایر کلمات موجودیت، دنباله‌های هدف نمونه بر اساس هر موجودیت تقسیم می‌شوند و دنباله‌های هدف جداگانه برای هر موجودیت ساخته می‌شوند. به عنوان نمونه، به صورت تصادفی از یک کلمه متن [CW] از موجودیت e^i از X نمونه‌برداری شده و در مقابل موجودیت به عنوان دنباله هدف y’ الحاق می‌شود.

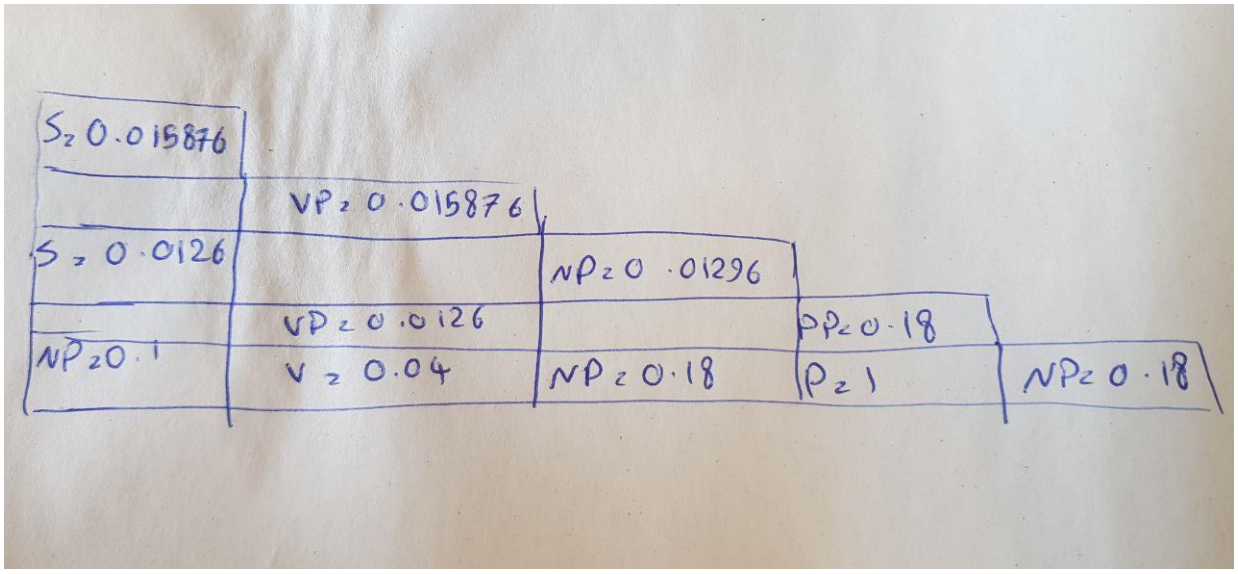
$$\{[CW], y_{e_1^i}, y_{e_2^i}, \cdots, y_{e_\varepsilon^i}\}$$

حال جایی که E نشان‌دهنده‌ی طول موجودیت e^i است، اگر در یک جمله M ، Xموجودیت وجود داشته باشد، می‌توان نمونه‌های افزوده شده M را ساخت (X’,Y) یعنی نمونه تکمیل‌کننده که با اضافه شدن آن‌ها و بررسی موجودیت‌های آن‌ها اثر pre-context از بین می‌رود و به ازای مدل‌های دیگر نیز نمونه وجود داشته و احتمال آن مدل‌های صفر نیست .

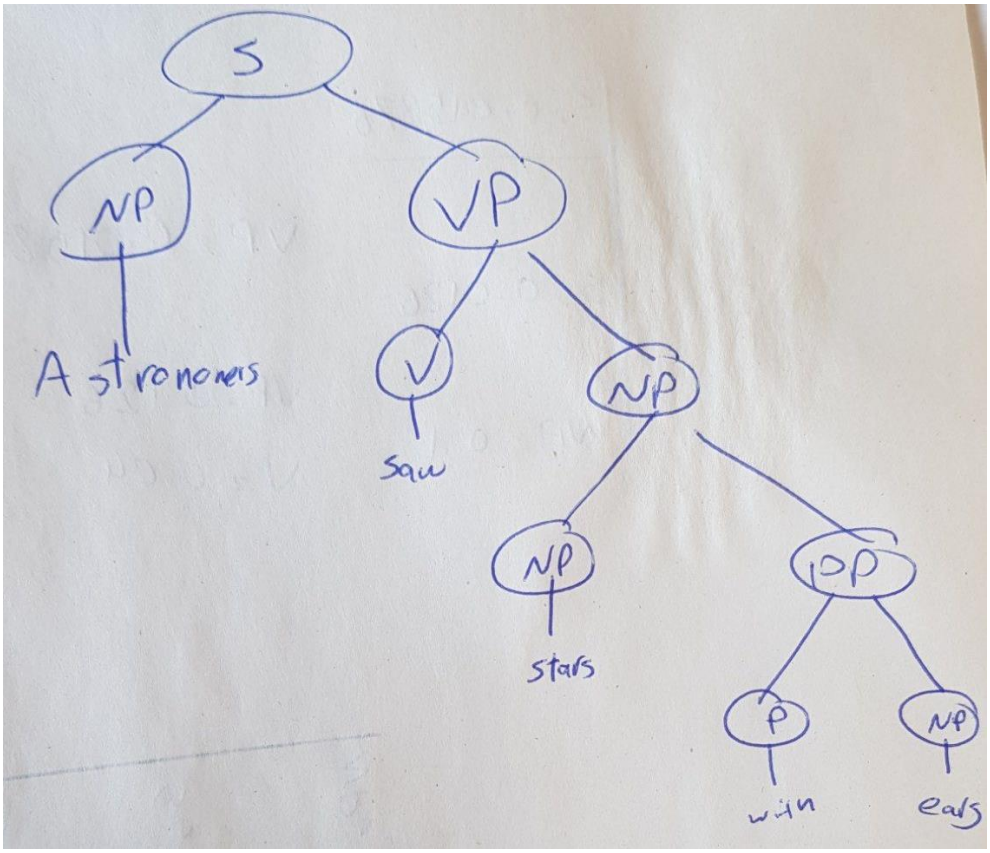
همچنین می‌توان گفت که در مقایسه با دنباله هدف Y نمونه اصلی، دنباله هدف در نمونه تقویت‌شده حاوی برچسب‌هایی نیست که شروع و پایان دنباله را نشان دهد، یعنی [ss] و .[ee] این برای این است که به مدل بگوییم که جای همه موجودیت‌ها، تنها یک موجودیت منفرد در نمونه تقویت‌شده ایجاد کند، تا از خروجی مدل آموزش‌دیده شده توسط نمونه‌های تقویت‌شده در اوایل پیش‌بینی عملی جلوگیری شود.

Inter-entity Deconfounding DA: مورد دیگر generation case این است که پس از تولید موجودیت فعلی، انتظار می‌رود مدل اولین کلمه موجودیت بعدی را تولید کند. در مدل‌های NER مولد سنتی، دنباله هدف به ترتیب موجودیت‌ها ثابت می‌شود. ترتیب موجود با توجه به وقوع از پیش تعیین شده است. با این حال، موجودیت‌ها اساساً ساختارهای مجموعه‌ای هستند و توالی رمزگشایی قرار نیست ثابت شود. یک موجودیت از پیش تعیین شده می‌تواند یک سوگیری نادرست را به مدل معرفی کند.

Problem 2



درخت حاصل از جدول بالا به شکل زیر است.



Problem 3

a)

ابتدایی‌ترین روش این است که از search بر روی متن استفاده کنیم. مثلا با استفاده از regex در جمله قسمت‌هایی که یکسان هستند و یا با استفاده از تابع find در پایتون شرایط را بررسی کرده و در کنار هر شرط با شرط بعدی یک and می‌گذاریم تا حاصل حاوی تمامی شروط خواسته شده باشد.

b)

این روش برای زمانی که متن طولانی باشد به خوبی عمل نمی‌کند؛ زیرا مدت زمان پردازش آن بسیار زیاد می‌شود. همچنین برای حالت‌هایی که کوئری پیچیده باشد و یا کلمات مترادف مدنظر باشد نیز عملی نخواهد بود. اگر ورودی تغییر کند نیز باید کل کتاب‌ها از ابتدا بررسی شوند.

c)

می‌توان برای هر کتاب یک id درنظر گرفت. برای هر کلمه نیز یک لیست در نظر گرفت تا id کتاب‌هایی که در آن موجود هستند را شامل شود. بنابراین یک دیکشنری خواهیم داشت که کلیدهای آن کلمات و مقادیر ما id کتاب‌هایی است که این کلمه در آن‌ها موجود است .

برای ساخت این دیکشنری ابتدا هر کتاب را tokenize کرده و پس از نرمال‌سازی و انجام عملیات Stemming و Lemmatization، Stopwordها را حذف کرده و عملیات indexing در نهایت انجام می‌شود. کلمات ابتدا به صورت الفبایی مرتب شده. تکرار کلمات نیز درنظر گرفته می‌شود و کلمات تکراری حذف شده و یک بار می‌آیند. اینگونه لیست کلمات ساخته می‌شود.

Problem 4

a)

در PCFGs اطلاعات واژگان درنظر گرفته نمی‌شود. یعنی اهمیت نمی‌دهد که برای چه کلمه‌ای چه قانونی استفاده شده است و به نوعی تمام کلمات را یکسان درنظر می‌گیرد

قوانین را بدون درنظر گرفتن مکان اعمال آن‌ها با احتمال یکسانی محاسبه می‌کند، یعنی اهمیت نمی‌دهد که در کجای درخت قرار گرفته است .

یک نوع سوگیری دارد به طوری که احتمال درخت‌های کوچک بیشتر از درخت‌های بزرگ محاسبه می‌کند.

برای دو درخت متفاوت با مجموعه قوانین یکسان احتمال برابر را محاسبه می‌کند و ترتیب اعمال قوانین را درنظر نمی‌گیرد.

b)

در روش lexicalized سعی شده است تا این مشکلات برطرف شود. برای مشکل نخست اطلاعات لغوی واژگان درنظر گرفته می‌شود که مهم‌ترین ایده در این روش است و احتمال بیشینه برای هر قانون وابسته به کلمه عمل می‌کند. برای حل مشکل دوم، برچسب گره پدر را نیز در قوانین اضافه می‌کند تا یک مرحله وابستگی بیشتر شده و خطای بیان شده کاهش یابد. در این روش نیز با توجه به ذات احتمالی بودن باز هم احتمال درخت‌های کوچک بیشتر از درخت‌های بزرگ می‌شود.

Problem 5

a)

محققان در جامعه پردازش اطلاعات چین (CIPSC) برنامه‌ای با عنوان CCKS 2020 Entity and Event Extraction از سوابق پزشکی الکترونیکی چین ایجاد کردند. هدف این برنامه شناسایی entity های پزشکی از EMR های آرشیو شده‌ی چینی است که در قالب

متن ساده هستند. علاوه بر این، این برنامهentity های پزشکی را در شش دسته از پیش تعریف شده گروه‌بندی می‌کند. بیماری و تشخیص، معاینه و تصویربرداری، معاینه آزمایشگاهی، جراحی، دارو و آناتومیک. در نتیجه یکی از کاربردهای استفاده از NER در داده‌های پزشکی شناسایی و دسته‌بندی اطلاعات است. همچنین به صورت پیشرفته‌تر می‌توان از این داده‌های دسته‌بندی شده و برچسب خورده با استفاده از شبکه‌های عمیق، مدلی را آموزش داد تا بتواند با مشاهده اطلاعات پزشکی، بیماری فرد را پیش‌بینی کند.

به‌طور کلی می‌توان بیان کرد که اطلاعاتی که با NER می‌توان از یک مجموعه داده‌ی پزشکی استخراج کرد شامل نام بیماری‌ها، نام داروها، نام ویروس‌ها، نام بیمارستان و ... است. البته هر نوع اطلاعاتی که ماهیت نام، مکان و سازمان داشته و قابل استخراج باشد. اطلاعات متنوعی همچون نوع بیماری، روند تشخیص، روند درمان، دارو مورد استفاده و از این گونه موارد که به صورت متنی هستند را می‌توان استخراج کرد.

b)

برای ساخت مدل NER به صورت زیر عمل می‌کنیم:

ابتدا نیاز به مجموعه‌ای از داده برای آموزش مدل خود داریم. مجموعه داده باید برچسب‌دار باشد. از رمزگذار IO به دلیل راحت‌تر، سریع‌تر و عملکرد بهتر از IOB استفاده می‌کنیم. پس از آنکه داده‌ها را برچسب زدیم باید اقدام به استخراج ویژگی‌ها کنیم. می‌توان از POS کلمه‌های قبل و بعد، خود کلمه‌های قبل و بعد و tag NER کلمه‌های قبلی را به عنوان یک سری ویژگی استفاده کرد. همچنین برخی ویژگی‌های دست نوشت نظیر مواری مانند داشتن خطش در کلمه، داشتن field در انتهای کلمه و ...

می‌توان از فرم کلی کلمات نظر داشتن اعداد- داشتن خط تیره- کوچک یا بزرگ بودن آن‌ها یک شکل خاص برای کلمات استخراج کرد و تحت این قوانین ویژگی‌ها را بررسی کرد. مثلاً کلمه‌ی CAP1 به صورت XXXd خواهد شد که X بیانگر حرف بزرگ و d بیانگر عدد است. درصورتی که طول کلمه بیشتر از ۴ حرف باشد دو حرف اول و آخر را به این فرم درآورده و حرف‌های میانی را صرفاً به صورت فرم در آورده و تنها نوع‌های موجود را به صورت Canonical مرتب می‌کنیم. برای کلمات طولانی داریم:

دو حرف اولXx :

دو حرف آخرxx :

حرف‌های میانی شامل x- هستند که اگر مرتب شوند به صورت -x خواهند شد و در نهایت شکل به صورت زیر خواهد شد که از چسباندن سه بخش بالا به دست می‌آیدXx-xxx-Xx .

ویژگی‌های به دست آمده در نهایت در Maximum Entropy Markov Models (MEMM) استفاده می‌شوندMEMM .ها برای مدل‌هایی به کار می‌روند که در آن‌ها با دنباله سر و کار داریم. می‌توان از سه نوع اموزش Greedy – Beam – Viterbi برای مدل مارکوفی استفاده کرد Viterbi .از بین این نوع الگوریتم‌ها پیشنهاد می‌شود. چون بهترین دنباله سراسری را انتخاب می‌کند. همچنین می‌توان از CRF به عنوان یک مدل دنباله‌ای دیگر استفاده کرد.

پس از آموزش مدل باید به ارزیابی آن پرداخت. ارزیابی standford برای NER بر اساس توکن‌ها نبوده و بر اساس خود نوع موجودیت است. به این معنا که یک موجودیت ممکن است از چند توکن تشکیل شده باشد که باید تمامی آن‌ها درنظر گرفته شود. برای این کار می‌توان از معیارهای ارزیابیPrecision ، Recallو F1 score استفاده کرد. مشاهده شده است که در این نوع ارزیابی حتی اگر یک توکن از یک نوع موجودیت بزرگ اشتباه برچسب زده شود، معیار ارزیابی آن را اشتباه تشخیص داده و خطا درنظر گرفته می‌شود. می‌توان از معیارهای دیگر نظیر MUC که به صورت زیر بخش و بخشی امتیازدهی می‌کنند، استفاده کرد. اینگونه با یک خطای کوچک بقیه‌ی تشخیص‌های درست سوخته نمی‌شوند. در نتیجه معیار پیشنهادی MUC است.

منابع:

اسلایدهای درسی و ویدیوهای دانشگاه استنفورد

https://direct.mit.edu/dint/article/3/3/402/102637/Medical-Named-Entity-Recognition-from-Un-labelled