به نام خدا

# تمرین سری سوم درس مبانی پردازش زبان و گفتار دکتر مرضیه داودآبادی فراهانی

# فرزان رحمانی 99521271

## سوالات تئوری

الف)

NER به دلیل پیچیدگی های زبان انسانی با چالش های متعددی مواجه است. در اینجا برخی از موارد رایج ذکر شده است:

* ابهام(Ambiguity): کلمات می توانند معانی متعددی داشته باشند. به عنوان مثال، "Apple" می تواند میوه یا شرکت فناوری باشد. مدل‌های NER باید زمینه را برای رمزگشایی نوع صحیح موجود در نظر بگیرند.
* وابستگی زمینه(Context Dependency): معنای یک کلمه اغلب به کلمات اطراف بستگی دارد. "Washington" بسته به زمینه می تواند یک ایالت یا نام خانوادگی رئیس جمهور آمریکا باشد.
* تنوع زبان(Language Variations): زبان عامیانه، گویش ها و تفاوت های منطقه ای می تواند مدل های NER را اشتباه بگیرد. «Doc» بسته به زمینه و نحوه نگارش ممکن است به یک پزشک یا یک سند اشاره داشته باشد.
* پراکندگی داده ها(Data Sparsity): آموزش مدل های NER به داده های برچسب گذاری شده زیادی نیاز دارد، جایی که موجودیت ها از قبل شناسایی و دسته بندی شده اند. ایجاد این امر می تواند گران و زمان بر باشد، به خصوص برای دامنه های خاص.
* تعمیم مدل(Model Generalization): یک مدل NER که بر روی یک نوع متن (مثلاً مقالات خبری) آموزش دیده است، ممکن است با دیگری (مثلاً سوابق پزشکی) مشکل داشته باشد. مدل ها باید با انواع مختلف زبان سازگار باشند.
* موجودیت های خارج از واژگان(Out-of-Vocabulary Entities): مدل های NER ممکن است موجودیت هایی را که قبلاً در طول آموزش ندیده اند از دست بدهند. به عنوان مثال، یک مدل آموزش دیده بر روی نام مکان های رایج ممکن است جزیره تازه کشف شده را از دست بدهد.
* مقیاس پذیری و عملکرد(Scalability and Performance): پردازش مقادیر زیاد متن می تواند از نظر محاسباتی برای سیستم های NER گران باشد. ایجاد تعادل بین دقت و کارایی یک چالش مداوم است.
* تغییرپذیری(Variability): موجودیت های نامگذاری شده می توانند اشکال مختلفی مانند اختصارات، کلمات اختصاری، غلط املایی یا نام های جایگزین(abbreviations, acronyms, misspellings, or alternative names) داشته باشند. به عنوان مثال، "United States" را می توان به صورت اختصاری "US" یا "U.S.A." و "New York City" را می توان به عنوان "NYC" یا "The Big Apple" نامید. شناخت این تغییرات برای NER دقیق ضروری است.
* موجودات با نام پیچیده(Complex Named Entities): برخی از موجودات نامگذاری شده پیچیده و از چند کلمه تشکیل شده اند. به عنوان مثال، "Bill Gates" نام یک فرد است که از دو کلمه تشکیل شده است. شناسایی چنین موجودیت های چند کلمه ای به عنوان یک موجودیت واحد به جای موجودیت های جداگانه یک چالش است.
* همپوشانی موجودیت نامگذاری شده(Named Entity Overlap): موجودیتهای نامگذاری شده می توانند با یکدیگر همپوشانی داشته باشند. به عنوان مثال، در عبارت " Apple's headquarters are in Cupertino, California,"، "Apple" هم یک شرکت و هم یک سازمان است، در حالی که "Cupertino" یک مکان است. حل چنین همپوشانی ها و طبقه بندی صحیح موجودیت ها برای NER دقیق بسیار مهم است.
* موجودیت‌های نادر و جدید(Rare and Novel Entities): مدل‌های NER ممکن است همه موجودیت‌های نام‌گذاری‌شده را در طول آموزش مشاهده نکرده باشند، به‌ویژه برای دامنه‌هایی با اصطلاحات تخصصی یا موجودیت‌های نوظهور. رسیدگی به موجودیت های کمیاب و جدید نیاز به قابلیت های تعمیم قوی دارد.
* NER بین زبانی (Cross-language NER): موجودیت های نام گذاری شده در یک زبان ممکن است معادل مستقیم در زبان های دیگر نداشته باشند، که NER بین زبانی را چالش برانگیز می کند. مسائل مربوط به نویسه‌گردانی(Transliteration) و ترجمه برای شناسایی دقیق موجودیت بین زبانی باید مورد توجه قرار گیرد.

همچنین در ادامه جواب Claude آمده است که خالی از لطف نیست:

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that aims to identify and classify named entities in text, such as persons, organizations, locations, dates, and various other types of entities. While NER has seen significant advancements in recent years, it still faces several challenges due to the complexities of human language. Here are some of the common ones:

1. Ambiguity and context-dependence:
   * Named entities can have multiple meanings or interpretations depending on the context in which they appear. For example, "Paris" can refer to a person, a city, or a brand.
   * Resolving the correct entity type based on the context can be challenging, especially for entities that have multiple potential types.
2. Heterogeneity of named entities:
   * Named entities can take various forms, including single words, multiple words, abbreviations, acronyms, and even nested entities (e.g., "University of California, Berkeley").
   * Handling the diversity of named entity structures and formats can be complex for NER systems.
3. Lack of comprehensive knowledge bases:
   * NER systems often rely on knowledge bases or gazetteers to recognize and classify named entities. However, these resources may not be exhaustive, especially for domains with rapidly evolving entities (e.g., new companies, products, or emerging technologies).
   * Maintaining and updating these knowledge bases can be a continuous challenge.
4. Domain-specific challenges:
   * The performance of NER systems can vary significantly across different domains (e.g., news, scientific literature, social media, medical records) due to the use of domain-specific terminology, named entity variations, and specialized context.
   * Adapting NER models to new domains requires additional training data and fine-tuning, which can be time-consuming and resource-intensive.
5. Handling multi-lingual and multi-script entities:
   * Named entities can appear in different languages and scripts, which can pose challenges for NER systems designed for a single language or script.
   * Adapting NER models to handle multiple languages and scripts requires addressing issues such as character encoding, transliteration, and language-specific entity variations.
6. Evolving entity mentions:
   * Named entities and their representations can evolve over time, with new entities emerging and existing entities changing their names, abbreviations, or contexts.
   * Keeping NER models up-to-date and robust to such changes is an ongoing challenge.

Addressing these challenges requires continuous research and development in areas such as deep learning, transfer learning, few-shot learning, and the incorporation of external knowledge sources. Additionally, domain-specific adaptations and multi-lingual capabilities are important focus areas for advancing the state-of-the-art in Named Entity Recognition.

ب)

مفهوم متن به چندین دلیل نقش مهمی در دقت سیستم های شناسایی موجودیت نامگذاری شده (NER) دارد:

* بهبود ابهام‌زدایی(Improved Disambiguation): کلماتی که دارای چندین معانی هستند را می‌توان بر اساس زمینه به درستی شناسایی کرد. به عنوان مثال، " Apple" در " I bought some apples from the store" به میوه اشاره دارد، در حالی که " Apple is releasing a new phone" به شرکت اشاره دارد. مفهوم متن(context) سرنخ هایی را برای محتمل ترین معنی ارائه می دهد.
* تغییرات مفهوم متنی(Contextual Variations): موجودیت های نامگذاری شده بسته به زمینه می توانند اشکال و عبارات مختلفی داشته باشند. به عنوان مثال، " New York" می تواند در یک زمینه به عنوان "NY" و در زمینه دیگر " New York City" ظاهر شود. درک متن اطراف(context) به سیستم‌های NER کمک می‌کند تا این تغییرات زمینه‌ای را شناسایی کرده و آنها را به نوع موجودیت صحیح ترسیم کند.
* رفع ابهام(Resolving Ambiguity): جملاتی که دارای ابهام ذاتی هستند از مفهوم متن بهره می برند. " Time flies." را در نظر بگیرید. آیا « Time» مجله است یا اشاره ای به گذر زمان؟ متن اطراف می تواند موجودیت مورد نظر را روشن کند.
* درک روابط موجودیت ها(Understanding Entity Relationships): زمینه کمک می کند تا چگونگی ارتباط موجودیت های نامگذاری شده با یکدیگر را شناسایی کنید. به عنوان مثال، در " President Biden visited France,"، " France " مکانی است مربوط به عمل " visited " توسط " President Biden ".
* Coreference Resolution: حل و فصل همبستگی ها، که در آن ضمایر یا سایر عبارات ارجاع به نهادهای ذکر شده قبلی اشاره می کنند، برای NER دقیق ضروری است. برای مثال، در جمله “Steve Jobs co-founded Apple. He revolutionized the tech industry,”، ضمیر «He» به «Steve Jobs» برمی‌گردد و این نهاد را به برچسب صحیح آن مرتبط می‌کند.
* Named Entity Span Identification: زمینه به شناسایی دقیق مرزهای موجودیت های نامگذاری شده کمک می کند. گاهی اوقات، موجودیت‌های نام‌گذاری شده چندین کلمه یا عبارات را در بر می‌گیرند و درک بافت اطراف به تعیین درست مرزهای موجودیت کمک می‌کند.
* چالش‌های ناشی از عدم وجود مفهوم متن(Challenges due to Missing Context): قطعه‌های کوتاه یا عبارات مجزا می‌توانند برای سیستم‌های NER دشوار باشند. بدون زمینه، " Washington D.C." می تواند یک مکان یا اشاره به یک شخص خاص باشد.
* محدودیت های وابستگی به زمینه(Limitations of Context-Dependence): اتکای بیش از حد به زمینه می تواند منجر به خطا شود. طعنه یا زبان مجازی(Sarcasm or figurative language) می تواند سیستم NER را گیج کند. به عنوان مثال، در " The king is dead, long live the king, "، متن نشان می دهد که "king" علی رغم معنای مجازی به همان شخص اشاره دارد.
* انواع موجودیت نامگذاری شده(Named Entity Types): زمینه اغلب سرنخ هایی در مورد نوع موجودیت نامگذاری شده ارائه می دهد. به عنوان مثال، در یک گزارش مالی، عباراتی مانند " revenue "، " profit " و " expenses " احتمالا به عنوان واحدهای مالی شناسایی می شوند، در حالی که در یک سند پزشکی، عباراتی مانند " patient "، " disease " و " treatment ". احتمالاً به عنوان نهادهای پزشکی شناخته می شوند.

تاثیر کلی:

گنجاندن اطلاعات مفهوم متن در سیستم‌های NER می‌تواند دقت آن‌ها را به طور قابل‌توجهی بهبود بخشد و آنها را قادر می‌سازد تا تصمیمات آگاهانه‌تری در مورد شناسایی و طبقه‌بندی موجودیت نام‌گذاری شده بگیرند. با این حال، مهم است که اذعان کنیم که زمینه(context) همچنین می‌تواند پیچیدگی‌هایی را معرفی کند که نیاز به توسعه بیشتر در مدل‌های NER دارند. در مجموع، درک متنی یک جنبه اساسی از درک زبان طبیعی است و برای ساختن سیستم های NER قوی و موثر ضروری است.

همچنین در ادامه جواب Claude آمده است که خالی از لطف نیست:

The text context plays a crucial role in the accuracy of Named Entity Recognition (NER) systems. Here's a more detailed explanation of the impact of text context on NER:

1. Ambiguity resolution:
   * As mentioned earlier, named entities can have multiple meanings or interpretations depending on the context in which they appear. The surrounding text context helps NER systems disambiguate the correct entity type.
   * For example, the word "Washington" could refer to a person, a city, or a state, depending on the context.
2. Specialized terminology:
   * Domain-specific texts often contain specialized terminology and named entities that may not be present in general-purpose knowledge bases or training data used by NER systems.
   * The surrounding context in these specialized domains provides important clues to identify and classify the named entities accurately.
3. Entity boundary detection:
   * Determining the exact boundaries of a named entity (i.e., where the entity starts and ends) can be challenging, especially for multi-word entities.
   * The text context, including sentence structure, grammar, and semantic relationships, helps NER systems identify the appropriate entity boundaries.
4. Anaphoric references:
   * Entities are often referred to using pronouns or shortened forms (e.g., "the company" instead of the full company name) in subsequent mentions within a text.
   * Understanding the coreference and anaphoric relationships between these entity references requires analyzing the broader text context.
5. Entity type disambiguation:
   * As mentioned earlier, some named entities can belong to multiple entity types (e.g., "Paris" as a person or a location). The surrounding text context provides vital information to determine the correct entity type.
6. Inference and reasoning:
   * In some cases, identifying named entities may require inference and reasoning based on the broader text context, rather than just relying on surface-level patterns or lexical features.
   * For example, recognizing that a person is a scientist or a politician may depend on the surrounding text describing their professional activities and affiliations.

To address these challenges, modern NER systems often employ advanced techniques, such as contextual language models (e.g., BERT, GPT), multi-task learning, and joint modeling of entity recognition and entity typing. These approaches allow the NER systems to better leverage the rich text context and improve their overall accuracy in identifying and classifying named entities.

ج)

HMM ها (مدل های پنهان مارکوف) یک رویکرد محبوب برای کارهایی مانند POS tagging و NER هستند، اما در هنگام ثبت روابط پیچیده بین کلمات در یک دنباله محدودیت هایی دارند. CRF ها (فیلدهای تصادفی شرطی) این محدودیت ها را با معرفی محدودیت های اضافی برطرف می کنند که منجر به بهبود عملکرد در وظایف POS tagging و NER می شود.

کاستی های HMM:

* فرض استقلال(Independence Assumption): HMM ها فرض می کنند خروجی در هر مرحله (به عنوان مثال، شناسایی یک موجودیت نامگذاری شده) فقط به حالت پنهان فعلی (به عنوان مثال، " part of speech") بستگی دارد و نه به خروجی های قبلی (موجودات با نام قبلی). این می تواند در زبان غیر واقعی باشد، جایی که شناسایی یک موجودیت می تواند بر شناسایی موجودیت بعدی تأثیر بگذارد (به عنوان مثال، " New York City " به احتمال زیاد یک مکان است تا فقط "New York").
* بازنمایی ویژگی های محدود(Limited Feature Representation): HMM ها معمولاً از ویژگی های اساسی مانند خود کلمه فعلی استفاده می کنند. CRF ها امکان نمایش ویژگی های غنی تری را فراهم می کنند که کلمات اطراف، برچسب های part of speech یا حتی حروف بزرگ را در نظر می گیرند، که منجر به درک دقیق تری از زمینه می شود.

بهبودهای CRF:

* Discriminative Modeling: CRF ها مدل های متمایز (Discriminative) هستند، در حالی که HMM ها مدل های مولد(generative models) هستند. مدل‌های متمایز مستقیماً احتمال شرطی دنباله خروجی را با توجه به توالی ورودی مدل می‌کنند، در حالی که مدل‌های مولد احتمال توأم هر دو توالی ورودی و خروجی را مدل‌سازی می‌کنند. CRF ها صرفاً بر یادگیری روابط بین ویژگی های ورودی و برچسب های خروجی تمرکز می کنند و به آنها امکان می دهد وابستگی های پیچیده تری را دریافت کنند و در کارهای برچسب گذاری دنباله ای مانند NER به عملکرد بهتری دست یابند.
* Feature-Rich Models: CRF ها می توانند طیف وسیع تری از ویژگی ها را فراتر از کلمه فعلی ترکیب کنند. این کمک می‌کند تا تعامل پیچیده بافت و ویژگی‌های کلمه‌ای را که NER را تحت تأثیر قرار می‌دهند، ثبت کنید.  
  به عبارت دیگر، CRF ها امکان ادغام ویژگی های غنی و گویا را می دهند که از دنباله ورودی مشتق شده اند. این ویژگی‌ها می‌توانند جنبه‌های مختلفی از داده‌های ورودی، از جمله هویت‌های کلمه، part-of-speech tags ، جاسازی‌های کلمه، اطلاعات نحوی و ویژگی‌های متنی را به تصویر بکشند. با استفاده از این ویژگی‌ها، CRFها می‌توانند تفاوت‌های ظریف و الگوهای موجود در داده‌ها را بهتر دریافت کنند، که منجر به بهبود دقت در وظایف NER در مقایسه با نمایش ویژگی‌های محدود در HMM می‌شود.
* Global Inference: برخلاف HMM که پیش بینی های محلی را انجام می دهند، CRF ها می توانند کل توالی را در طول استنتاج در نظر بگیرند. این به آن‌ها اجازه می‌دهد تا موجودیت‌های منحصر به فرد متقابل را شناسایی کنند (به عنوان مثال، a person cannot be a location) و این محدودیت‌ها را اعمال کنند، که منجر به نتایج سازگارتر و دقیق‌تر می‌شود.  
  به عبارت دیگر، بر خلاف HMM ها که وابستگی های توالی را با استفاده از احتمالات انتقال محلی بین حالت های مجاور مدل می کنند، CRF ها وابستگی های توالی جهانی را مدل می کنند. CRF ها کل توالی ورودی را هنگام اختصاص برچسب ها به عناصر جداگانه در نظر می گیرند و به آنها امکان می دهد وابستگی های دوربرد و اطلاعات زمینه ای را که ممکن است چندین کلمه یا عبارات را شامل می شود، ضبط کنند. این دیدگاه جهانی CRF ها را قادر می سازد تا تصمیمات برچسب گذاری آگاهانه تری بگیرند و پدیده های پیچیده زبانی مانند مرزهای موجودیت نامگذاری شده و موجودیت های تو در تو را بهتر مدیریت کنند.
* محدودیت‌های برچسب‌گذاری انعطاف‌پذیر(Flexible Labeling Constraints): CRF‌ها محدودیت‌های برچسب‌گذاری انعطاف‌پذیر و قابل تنظیم را امکان‌پذیر می‌کنند، که می‌تواند با ویژگی‌های خاص وظیفه NER در دست طراحی شود. این محدودیت‌ها می‌توانند قوانین خاص حوزه، الگوهای زبانی یا ساختارهای نحوی را برای هدایت فرآیند برچسب‌گذاری و بهبود کیفیت پیش‌بینی‌های خروجی اعمال کنند. در مقابل، HMM ها احتمالات انتقال سختی را بین حالت ها تحمیل می کنند، که ممکن است همیشه با پیچیدگی های داده های زبان طبیعی هماهنگ نباشد.
* احتمال شرطی: CRF ها احتمال یک دنباله خروجی را مدل می کنند (به عنوان مثال، برچسب های موجودیت نامگذاری شده برای هر کلمه) با توجه به یک دنباله ورودی (کلمات واقعی در جمله). این به آنها اجازه می دهد تا کل توالی و روابط بین خروجی ها را در نظر بگیرند.

در اصل، CRF ها بر روی نقاط قوت HMM ها ساخته می شوند و در عین حال محدودیت های آن ها را برطرف می کنند. آنها امکان نمایش ویژگی های پیچیده تر و احتمالات مشروط را فراهم می کنند، که منجر به درک دقیق تری از چگونگی تأثیر زمینه بر شناسایی موجودیت نامگذاری شده می شود.

در کل، CRFها چارچوبی انعطاف‌پذیر، متمایز و گویاتر برای وظایف برچسب‌گذاری دنباله‌ای مانند NER در مقایسه با رویکرد مدل‌سازی محدود HMMs ارائه می‌دهند. با استفاده از نمایش‌های ویژگی‌های غنی‌تر، زمینه توالی جهانی، و محدودیت‌های برچسب‌گذاری قابل تنظیم، CRFها می‌توانند دقت و استحکام بالاتری در برنامه‌های NER به دست آورند.

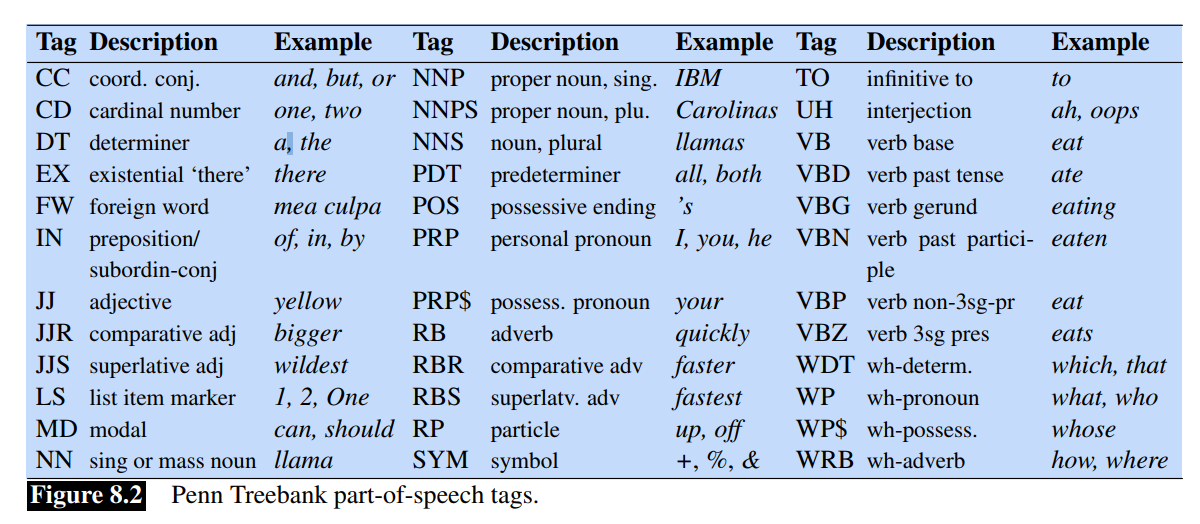
همچنین در ادامه جواب Claude آمده است که خالی از لطف نیست:

Conditional Random Fields (CRFs) address some of the key limitations of Hidden Markov Models (HMMs) in the context of structured prediction tasks, such as part-of-speech tagging and named entity recognition. Here's how CRFs improve upon the constraints of HMMs:

1. Relaxation of independence assumptions:
   * HMMs make strong assumptions about the independence of observations and state transitions, which can be restrictive in real-world scenarios.
   * CRFs are a discriminative model that relaxes these independence assumptions by allowing the model to capture dependencies between the input features and the output labels, as well as dependencies between the output labels themselves.
2. Ability to use rich, overlapping features:
   * HMMs are typically limited to using local, independent features, such as the current word and its part-of-speech tag.
   * CRFs can leverage rich, overlapping features that may capture more complex dependencies in the data, such as the surrounding words, morphological features, and contextual information.
3. Global optimization:
   * HMMs optimize the model parameters locally, which can lead to suboptimal solutions.
   * CRFs perform a global optimization of the model parameters, considering the entire sequence of output labels, which can lead to better generalization and more accurate predictions.
4. Handling of label bias:
   * HMMs suffer from the label bias problem, where the model tends to prefer state transitions with higher probabilities, even if they are not the most optimal choices.
   * CRFs do not have the same label bias issue, as they model the conditional probability of the output labels given the input features, rather than the joint probability of the input and output, as in HMMs.
5. Flexibility in modeling dependencies:
   * CRFs can capture more complex dependencies between the input features and output labels, as well as between the output labels themselves, through the use of arbitrary, potentially non-independent features.
   * This flexibility allows CRFs to better model the structure and correlations present in the data, leading to improved performance on structured prediction tasks.
6. Effective handling of sparse features:
   * CRFs can effectively handle sparse features, which are common in natural language processing tasks, through the use of regularization techniques.
   * This allows CRFs to learn from high-dimensional feature spaces without overfitting, leading to better generalization.

In summary, CRFs address the limitations of HMMs by relaxing the independence assumptions, allowing the use of rich, overlapping features, performing global optimization, handling label bias, and effectively modeling complex dependencies in the data. These improvements make CRFs a more powerful and flexible tool for structured prediction tasks in natural language processing.

د)   
tag های مجموعه Penn Treebank به همراه تعریف آنها را در زیر مشاهده می‌کنید.



* کلمه Atlanta به اشتباه NN برچسب گذاری شده است. باید NNP باشد چرا که اسم شهر است پس اسم خاص محسوب می شود.

I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NNP

* کلمه dinner به اشتباه NNS برچسب گذاری شده است. باید NN باشد چرا اسم جمع نیست.

Does/VBZ this/DT flight/NN serve/VB dinner/NN

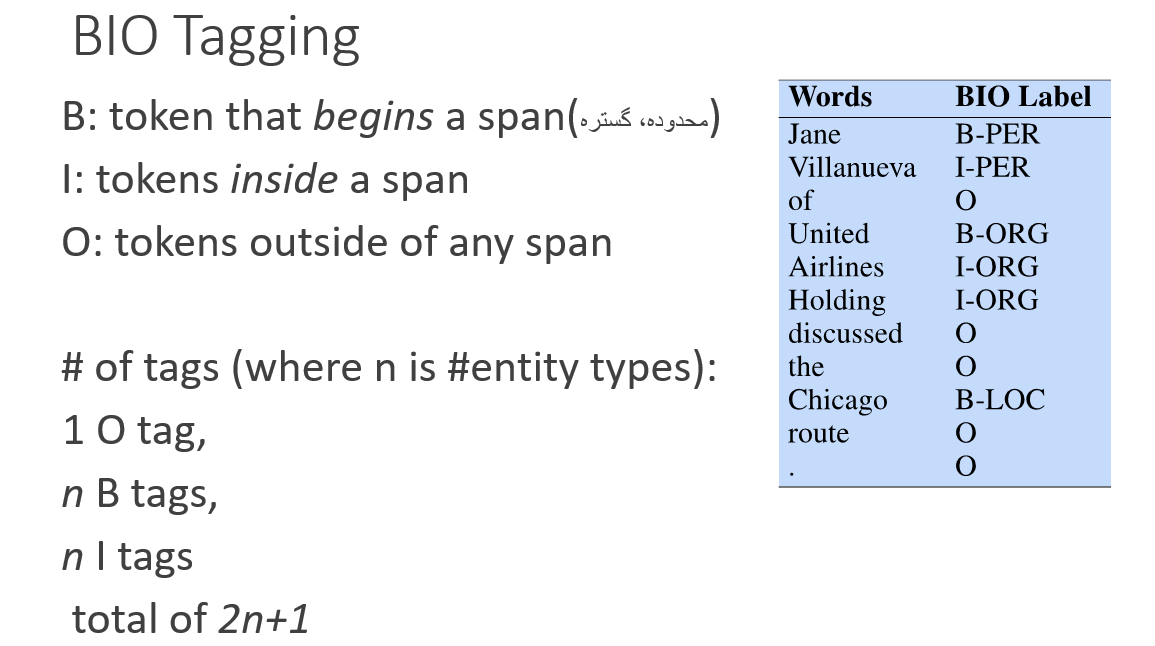
* کلمه have به اشتباه VB برچسب گذاری شده است. باید VBP باشد.

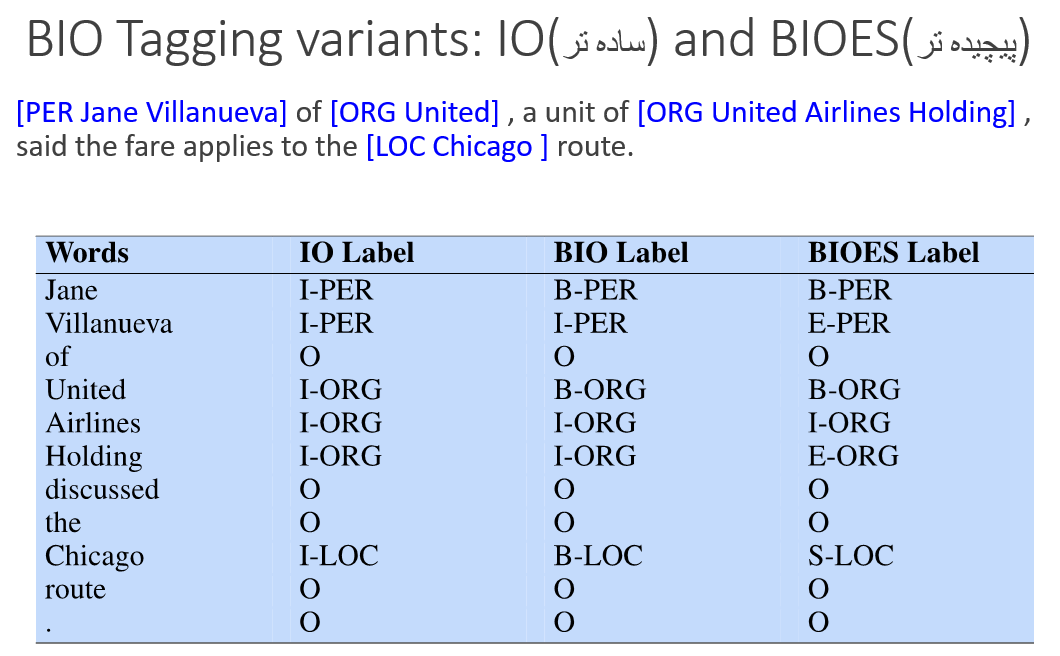
I/PRP have/VBP a/DT friend/NN living/VBG in/IN Denver/NNP

* کلمه Can به اشتباه VBP برچسب گذاری شده است. باید MD باشد چرا که فعل کمکی modal می‌باشد.

Can/MD you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

ه)  
ابتدا به بیان مطالب بیان شده در اسلاید های درس می پردازیم و سپس جواب سوال کامل تر بیان می کنیم.





روش برچسب‌گذاری BIO یک رویکرد رایج است که برای برچسب‌گذاری موجودیت‌های نام‌گذاری‌شده در وظایف برچسب‌گذاری دنباله‌ای، مانند شناسایی موجودیت نام‌گذاری شده (NER) استفاده می‌شود. در طرح برچسب‌گذاری BIO، به هر توکن در یک دنباله، یکی از سه برچسب B، I یا O اختصاص داده می‌شود که به ترتیب نشان می‌دهد که آیا یک موجودیت با نام شروع می‌شود، داخل یک موجودیت نام‌گذاری شده است یا خارج از هر موجودیت نام‌گذاری شده است. روش تگ گذاری BIO چگونه کار می کند:

1. B (Beginning): این برچسب به اولین نشانه یک موجودیت نامگذاری شده در دنباله اختصاص داده می شود. هر موجودیت نامگذاری شده با یک برچسب B شروع می شود.
2. I (Inside): این برچسب به دنبال برچسب B به نشانه های درون یک موجودیت نامگذاری شده اختصاص داده می شود. اگر دنباله‌ای حاوی موجودیت‌های چند کلمه‌ای باشد، تمام نشانه‌های بعدی پس از نشانه اولیه با I برچسب‌گذاری می‌شوند تا زمانی که موجودیت به پایان برسد.
3. O (Outside): این برچسب به نشانه هایی اختصاص داده می شود که بخشی از هیچ موجودیت نامگذاری شده نیستند.

به عنوان مثال، جمله "Steve Jobs co-founded Apple Inc." را در نظر بگیرید. با استفاده از طرح BIO برچسب گذاری شده است:

* "Steve" - B-PER (Beginning of a person entity)
* "Jobs" - I-PER (Inside the person entity)
* "co-founded" - O (Outside any entity)
* "Apple" - B-ORG (Beginning of an organization entity)
* "Inc." - I-ORG (Inside the organization entity)

اکنون، بیایید تفاوت‌های بین طرح‌های برچسب‌گذاری BIO، IO و BIOES را مورد بحث قرار دهیم:

1. BIO در مقابل IO (Inside-Outside):
   * در طرح BIO، هر موجودیت با یک تگ B شروع می شود و پس از آن یک یا چند تگ I برای توکن های بعدی درون موجودیت قرار می گیرد.
   * در طرح IO، موجودیت های نامگذاری شده فقط با تگ I (اگر داخل یک موجودیت) یا تگ O (اگر خارج از هر موجودیتی هستند) برچسب گذاری می شوند. هیچ برچسب شروع خاصی (B) وجود ندارد.
   * طرح BIO اطلاعات صریح تری در مورد مرزهای موجودیت های نامگذاری شده ارائه می دهد و یادگیری و پیش بینی دقیق مرزهای موجودیت را برای مدل ها آسان تر می کند. با این حال، در مقایسه با طرح IO به برچسب های بیشتری نیاز دارد.
2. BIO در مقابل BIOES (Beginning, Inside, Outside, End, Single):
   * طرح BIOES طرح BIO را با افزودن دو تگ اضافی گسترش می دهد: E (پایان) و S (تک).
   * E برای علامت گذاری پایان یک موجودیت چند نشانه ای و S برای علامت گذاری یک موجودیت تک نشانه استفاده می شود.
   * به عنوان مثال، در طرح BIOES، موجودیت "استیو جابز" به عنوان B-PER، و E-PER برچسب گذاری می شود، در حالی که یک موجودیت تک نشانه مانند "Apple" به عنوان S-ORG برچسب گذاری می شود.
   * طرح BIOES جزئیات بیشتری را در نمایش مرزهای موجودیت، به ویژه برای موجودیت هایی که از چندین توکن تشکیل شده اند، فراهم می کند. این ابهام را کاهش می دهد و برچسب گذاری را آموزنده تر می کند. با این حال، به برچسب های اضافی نیاز دارد و ممکن است پیاده سازی آن پیچیده تر باشد.

انتخاب روش مناسب:

* BIO به دلیل سادگی و کارایی گزینه خوبی برای اکثر کارهای NER است.
* IO برای سناریوهای ساده تری که در آن تمایز موجودیت های تک کلمه ای و چند کلمه ای ضروری نیست مفید است.
* BIOES اطلاعات دقیق تری ارائه می دهد اما ممکن است بسته به سیستم NER نیاز به تنظیمات در پردازش داشته باشد.

به طور خلاصه، در حالی که هر سه طرح برچسب‌گذاری به منظور حاشیه‌نویسی موجودیت‌های نام‌گذاری‌شده در داده‌های متنی عمل می‌کنند، آنها در جزئیات، پیچیدگی و بیان متفاوت هستند. انتخاب طرح برچسب گذاری به الزامات خاص وظیفه NER و مبادله بین سادگی و دقت بستگی دارد. روش برچسب گذاری BIO یک راه روشن و کارآمد برای نشان دادن موجودیت های نامگذاری شده در وظایف NER ارائه می دهد. در حالی که BIOES جزئیات بیشتری را ارائه می دهد، BIO به دلیل تعادل بین سادگی و اثربخشی، انتخاب محبوبی است. بهترین روش به نیازهای خاص و پیچیدگی برنامه NER شما بستگی دارد.

همچنین در ادامه جواب Claude آمده است که خالی از لطف نیست:

The BIO (Beginning, Inside, Outside) tagging method is a widely used approach for representing and labeling named entities in text. In this method, each token in the text is assigned one of the following three tags:

1. **B (Beginning)**: This tag indicates that the current token is the beginning of a named entity.
2. **I (Inside)**: This tag indicates that the current token is inside a named entity, but not at the beginning.
3. **O (Outside)**: This tag indicates that the current token is not part of a named entity.

The BIO tagging method allows NER (Named Entity Recognition) systems to identify the boundaries of named entities and classify them into different entity types (e.g., person, organization, location).

Now, let's investigate the differences between the BIO tagging method and other similar approaches:

1. **IO Labeling**:
   * The IO (Inside, Outside) labeling method is a simpler version of the BIO tagging, where only two labels are used: "I" for tokens inside a named entity, and "O" for tokens outside a named entity.
   * This method does not explicitly identify the beginning of named entities, which can make it more challenging to determine the exact boundaries of the entities.
2. **BIOES Labeling**:
   * The BIOES (Beginning, Inside, Outside, End, Single) labeling method is an extension of the BIO tagging, with the addition of two more tags:
     + **E (End)**: This tag indicates that the current token is the end of a named entity.
     + **S (Single)**: This tag indicates that the current token is a single-token named entity.
   * The BIOES tagging provides more detailed information about the entity boundaries, which can be particularly useful for identifying nested or discontinuous named entities.

The key differences between these methods are:

* **IO Labeling**: Simpler, but less informative about the entity boundaries.
* **BIO Labeling**: Provides information about the beginning and inside of named entities, but does not explicitly identify the end.
* **BIOES Labeling**: More detailed, with additional tags for the end and single-token entities, which can improve the accuracy of NER systems in certain cases.

The choice between these tagging methods often depends on the specific requirements of the NER task, the complexity of the named entities in the data, and the capabilities of the NER model being used. Many state-of-the-art NER systems employ the BIOES tagging method, as it can provide more precise information about the named entity boundaries and lead to better overall performance.

مراجع:

<https://chat.openai.com/>   
<https://bard.google.com/>   
<https://claude.ai/chats>

## سوالات عملی

1. نوت بوک تکمیل شده و در داخل آن توضیح داده شده است. در اینجا نیز به طور خلاصه آن را شرح می دهیم:

**comparing the most frequent tag baseline versions.**   
**'Assuming that all unknown words are NN' versus 'With additional rules for unknown words'**

* most frequent tag baseline: `accuracy: 0.81841` (assuming all unknown words are 'NN')
* improved most frequent tag baseline: `accuracy: 0.86276` (using the additional rules for unknown words)

Steps we have gone through are:

* Data Preparation: The Brown Corpus from the NLTK library is used for training and testing. It's divided into training and testing sets.
* The generate\_dict() function creates a dictionary where the keys are unique words, and the values are lists of dictionaries. Each dictionary in the list represents a unique tag associated with the word and its count.
* The predict\_tag() function uses the dictionary generated in the previous step to predict the tags for the test set. It assigns the 'NN' (noun) tag to unknown words and selects the tag with the highest count for known words.
* The predict\_tag\_with\_improvements() function builds on the previous one, adding more specific rules to handle unknown words. It checks for certain patterns in the word (e.g., ending in 'ing', 's', 'ly', 'ed', etc.) and assigns more appropriate tags based on these patterns.
* The script first divides the Brown corpus (news category) into a training set (75%) and a testing set (25%). It then generates the tag dictionary from the training set and uses it to predict the tags for the testing set.
* The initial accuracy, assuming all unknown words are 'NN', is printed.
* The improved accuracy, using the additional rules for unknown words, is then printed.
* The delta, which represents the number of more words that were correctly classified with the improved rules, is also printed.

The key observations from the results are:

* The initial accuracy which is 0.81841, assuming all unknown words are 'NN', is a baseline that provides a starting point for comparison.
* The improved accuracy which is 0.86276, using the additional rules for unknown words, shows an increase (approximately 5%) in performance.
* The delta value which is 1115 quantifies the improvement, indicating that the additional rules helped correctly classify a substantial number of words that were previously misclassified.

This analysis demonstrates the importance of developing more sophisticated rules and techniques to handle unknown words in part-of-speech tagging, which can lead to substantial improvements in the overall accuracy of the system.

2. نوت بوک تکمیل شده و در داخل آن توضیح داده شده است. در اینجا نیز به طور خلاصه آن را شرح می دهیم:

**comparing HMM (Q2 in HW3) performance to the most frequent tag baseline (Q1 in HW3).**

* HMM tagger: accuracy: 0.91650
* most frequent tag baseline: accuracy: 0.81841 (assuming all unknown words are 'NN')
* improved most frequent tag baseline: accuracy: 0.86276 (using the additional rules for unknown words)

1. The collect\_probabilities() function processes the training data and computes the following:
   * tag\_freq: The frequency of each tag type observed in the training data.
   * word\_per\_tag\_freq: The frequency of each word observed under each tag in the training data, with initial smoothing.
   * bigram: The transition probabilities between tag types based on bigram counts.
   * init\_dist: The initial probability distributions for tag types based on the second tag in each sample.
2. The create\_confusion\_matrix() function updates the confusion matrices based on the predicted tags and the actual (hidden) tags for each word in the test sentences. It also identifies the tags with the falsest positives and false negatives.
3. The viterbi() function implements the Viterbi algorithm for part-of-speech tagging on the test set. It processes each sentence in the test set and performs the following steps:
   * Initializes the path probability and backpointer dictionaries for the Viterbi algorithm.
   * Iterates through each word in the sentence, calculating the most likely tag sequence based on the probabilities and transition probabilities.
   * Determines the best path at the end of the sentence and reconstructs the predicted tag sequence.
   * Compares the predicted tags with the actual (hidden) tags and updates the confusion matrices accordingly.
   * Prints the sentence, hidden states, and predicted tags for each sentence, as well as the overall accuracy of the model on the test set.

The key observations from the results are:

* The Viterbi algorithm is implemented to find the most likely sequence of tags for each sentence in the test set.
* The confusion matrices are used to track the performance of the model, identifying the tags with the falsest positives and false negatives.
* The overall accuracy of the model on the test set is reported, providing a measure of the performance of the HMM-based part-of-speech tagger.

This analysis demonstrates the implementation of a basic HMM-based part-of-speech tagger and the use of the Viterbi algorithm to decode the most likely tag sequence for a given input. The confusion matrices provide valuable insights into the strengths and weaknesses of the model, which can guide further improvements and refinements.

3.  
الف)  
دلایلی که ممکن است دیتا های موجود در article-about-a-genre.txt مشکل ساز شوند:

After examining the article-about-a-genre.txt file, there are a few reasons why the named entities (specifically, the movie titles) may be problematic for a Named Entity Recognition (NER) system:

1. Cultural and linguistic context: Many of the named entities mentioned are Chinese or Hong Kong names, titles of films, production studios etc. An NER system trained primarily on English data may struggle to accurately identify and classify these culturally and linguistically distinct entities.
2. Transliteration issues: Some of the Chinese/Hong Kong names are transliterated into English, like "Jackie Chan", "Jet Li", "Donnie Yen" etc. Transliteration can introduce variations that make consistent recognition harder.
3. Name ambiguity: Some names like "Bruce Lee", "Tony Leung", "Andy Cheng" are personal names but could potentially be confused with common words/phrases by an NER system.
4. Genre-specific terminology: Terms like "wuxia", "kung fu", "Shaolin", "wing chun" etc. are domain-specific vocabulary related to martial arts films that general NER systems may not adequately recognize.
5. Multiword entities: Titles like "Shang-Chi and the Legend of the Ten Rings", "Crouching Tiger, Hidden Dragon", "Iron Monkey" etc. are multiword entities that require the system to identify boundaries correctly.
6. Code-switching: The text contains English words/sentences intermixed with Chinese names/terms, which can make language identification and entity extraction harder.
7. Variability in Entity Naming: Named entities like "Shang-Chi" and "Bruce Lee" have different variations in their representation throughout the text. For instance, "Shang-Chi" is referred to both with and without the hyphen, while "Bruce Lee" is mentioned with different variations of his name, such as "Bruce Lee", "Lee", and "Bruce".
8. Ambiguity: Some named entities might be ambiguous or context-dependent. For example, "Jet Li" and "Li" could refer to the actor Jet Li or to someone else with the surname Li. Similarly, "Chan" could refer to Jackie Chan or to another individual with the same surname.
9. Cultural Specificity: Names like "Tony Leung Chiu-wai", "Yuen Woo-ping", and "Stephen Chow" might be challenging for NER systems to recognize if they are not trained on a diverse range of cultural names.
10. Complexity of Compound Names: Certain names consist of multiple parts, like "Tony Leung Chiu-wai" and "Shaw Brothers". NER systems might struggle to correctly identify and parse these compound names.
11. Mention of Fictional Entities: The text discusses fictional characters like "Shang-Chi" and "Iron Fist", which might not be recognized by NER systems if they are not trained to identify fictional entities.
12. Mix of Proper and Common Nouns: The text includes a mix of proper nouns (e.g., "Hong Kong", "Marvel Cinematic Universe") and common nouns (e.g., "kung fu", "cinema"), which could potentially confuse NER systems.
13. Acronyms and Abbreviations: The text may contain acronyms and abbreviations that require context to resolve, such as "NER" for Named Entity Recognition and "MCU" for Marvel Cinematic Universe.
14. Mixed use of full names and partial names: Some named entities are referred to by their full names (e.g., "Tony Leung Chiu-wai"), while others are referred to by only their first names or last names (e.g., "Bruce Lee," "Chan," "Li"). This inconsistency in referencing can make it challenging for systems to identify and link entities correctly.

In summary, the highly specialized cultural context, multilingual content, transliteration issues, and domain-specific vocabulary of this article about martial arts films poses challenges that could degrade the performance of general-purpose NER systems not specifically tailored for this domain.

To address these challenges, the NER system would need to be designed with strategies such as:

* Improving the coverage and consistency of the movie title training data, including titles from diverse cultural and linguistic backgrounds.
* Incorporating contextual information and language models to better distinguish movie titles from other named entities and common phrases.
* Developing techniques to handle variations in movie title formats and structures.
* Implementing mechanisms to update the movie title knowledge base over time to keep up with the evolving movie landscape.

By addressing these potential issues, the NER system can become more robust and accurate in identifying movie titles within the given text.

دلایلی که ممکن است دیتا های موجود در IMDB-top-1000.csv مشکل ساز شوند:

Based on the data in the IMDB-top-1000.csv file, here are some potential reasons why certain named entities may be problematic for Named Entity Recognition (NER) systems:

1. Multi-word named entities: Many movie titles like "The Shawshank Redemption", "The Lord of the Rings: The Return of the King", "One Flew Over the Cuckoo's Nest" etc. are multi-word named entities. Correctly identifying the boundaries and full span of these can be challenging.
2. Foreign language names: There are movie titles, actor names, director names etc. from various languages like French (e.g. "Amelie"), Spanish (e.g. "Y Tu Mamá También"), Korean (e.g. "Parasite"), Hindi (e.g. "3 Idiots") etc. Handling names from diverse languages can increase ambiguity.
3. Ambiguity with common words: Some movie titles like "Inception", "Coco", "Joker" are common words that an NER system may fail to recognize as named entities. Similarly, names like "Will Smith", "Chris Brown" etc. could be mistaken for common phrases.
4. Inconsistent formats: There seems to be some inconsistency in name formats - e.g. for some actors only the last name is listed, while for others both first and last name are present. This variability can confuse NER models.
5. Long movie titles: Extremely long movie titles like "Everything Everywhere All at Once" may make it harder to accurately capture the full entity span.
6. Special characters: Titles with special characters like punctuation marks (e.g. "K.G.F: Chapter 2") or Roman numerals (e.g. "Spider-Man: No Way Home") add complexity.
7. Lack of context: Without metadata about whether a name refers to an actor, director, character etc., disambiguation becomes harder for an NER system working solely on this tabular data.
8. Missing Values: There are missing values in some columns. For instance, the "GROSS COLLECTION" column has missing values for certain movies. Similarly, some movies have missing values for actors or directors. This can affect the completeness and accuracy of the analysis, especially when trying to understand the financial performance or cast of each movie.
9. Certificate column: The certification column includes different rating systems ("X," "U," "PG," etc.), which might not be consistent across different countries. This could lead to confusion, especially if the dataset is used internationally.
10. Language and Cultural Considerations: The dataset includes movie titles, directors, and actors from various cultural backgrounds and languages. Language differences, cultural contexts, and translation issues may pose challenges in data analysis, especially when dealing with multilingual or culturally diverse datasets.

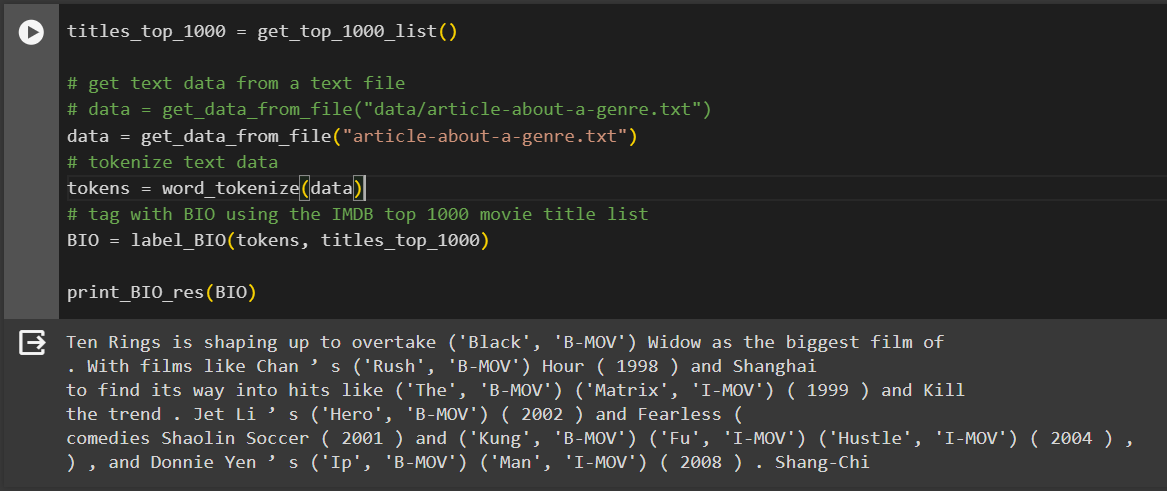
So in summary, the diversity of languages, formats, ambiguity issues and presence of very long or very short multi-word entities in this movie data poses challenges that could impact NER accuracy if not handled appropriately.

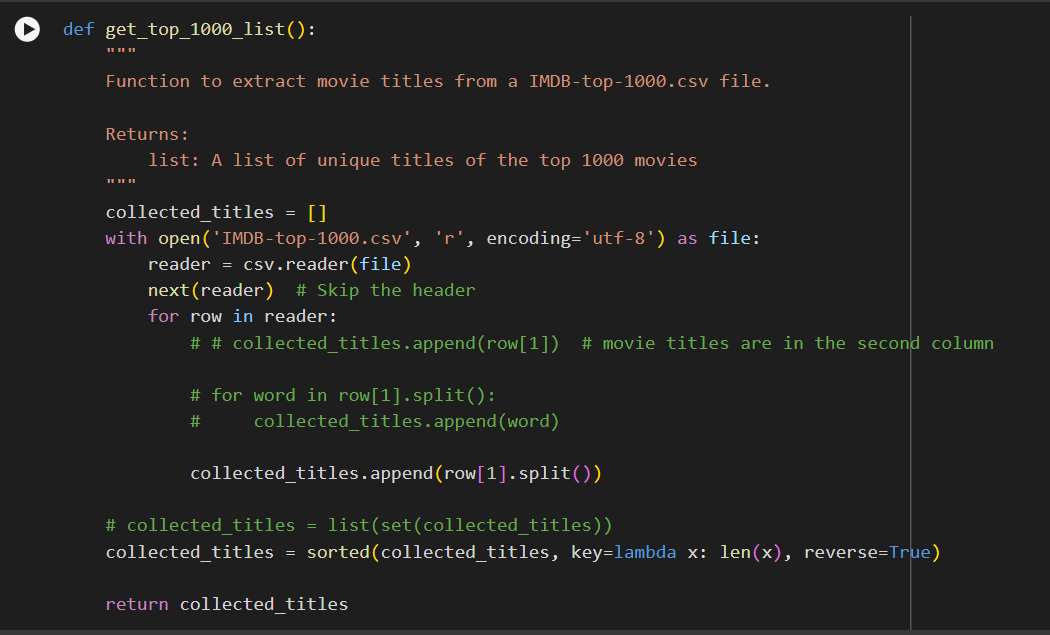
To address these challenges, the NER system would need to be designed with strategies such as:

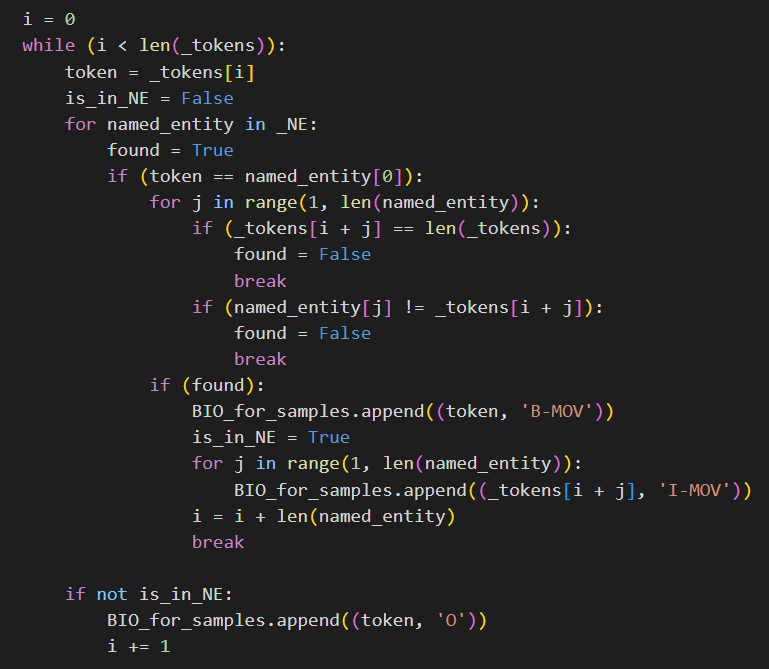
* Improving the coverage and consistency of the movie title training data, including titles from diverse cultural and linguistic backgrounds.
* Implementing techniques to handle variations in movie title formats, capitalization, and punctuation.
* Incorporating contextual information and language models to better distinguish movie titles from other named entities and common phrases.
* Developing mechanisms to update the movie title knowledge base over time to keep up with the evolving movie landscape.
* Exploring multi-lingual or language-agnostic approaches to handle non-English movie titles.

By addressing these potential issues, the NER system can become more robust and accurate in identifying movie titles within the given text.

ب) نوت بوک تکمیل شد و می توانید به آن مراجعه کنید. همان طور که میبینید فیلم هایی که عناوین آنها داخل فایل IMDB-top-1000.csv موجود است را توانسته ایم تشخیص دهیم. ولی بعضی از فیلم ها که در داده آموزشی ما نیستند نظیر ‘Shaolin Soccer’ را نتوانستیم پیدا کنیم. برای بهبود عملکرد میتوانیم عناوین فیلم های بیشتری مثلا به جای 1000 از 10000 فیلم برتر استفاده کنیم.







مراجع:

<https://chat.openai.com/>   
<https://bard.google.com/>   
<https://claude.ai/chats>

پایان