Academic Integrity

1:

I'm a student doing a Python project in a DH course. Act as a good tutor and give me advice without giving me the full answers.

**ChatGPT said:**

Updated saved memory

Got it! I’ll guide you through your Python project by asking helpful questions, explaining concepts, and pointing you in the right direction—without spoiling the learning process. What are you working on for your project right now?

4o

**You said:**

im working on the ner pipeline and need to sort through place names. does this work: doc = nlp(text) for sentence in doc.sentences: for entity in sentence.ents: if ent.type in ["GPE", "LOC"]: place = entity.text()

**ChatGPT said:**

You're very close, and it's great that you're thinking in terms of iterating through sentences and entities! A few nudges:

**Indentation**: Python is very sensitive to indentation, and it looks like your for loop under doc = nlp(text) might be over-indented. Make sure your structure aligns properly.

**Variable consistency**: You use entity in the loop (for entity in sentence.ents) but then reference ent in the if statement (if ent.type...). That will raise an error unless you change ent to entity.

**Accessing the entity type and text**:

Make sure you're using the correct method or attribute to access entity.text and entity.type. Depending on the NER library you're using (e.g., spaCy, Stanza), this might vary. Double-check the documentation.

Here’s a guiding question: Which NLP library are you using—spaCy, Stanza, or something else? That will help tailor advice on the correct attribute names.

2:

im using stanza and regex, can you give me examples of using regex to clean gpe names to remove punctuation and THE infront of names

**ChatGPT said:**

Great—Stanza for NER and regex for post-processing is a solid combo.

When cleaning **GPE (Geo-Political Entity)** names with regex, you might want to:

1. **Remove leading articles like "the" or "The"**.
2. **Strip punctuation from the beginning/end** (but maybe not *inside* the name like in "St. Louis").

Here are a few regex patterns you can experiment with:

**1. Remove leading "the"/"The" (case-insensitive)**

python

Copy code

import re

def clean\_place\_name(name):

name = re.sub(r'^\s\*the\s+', '', name, flags=re.IGNORECASE)

return name

**2. Strip common punctuation around the name**

python

Copy code

def clean\_place\_name(name):

name = re.sub(r'^\s\*the\s+', '', name, flags=re.IGNORECASE) # remove "the"

name = re.sub(r'[^\w\s.-]', '', name) # remove unwanted punctuation but keep dots and hyphens

return name.strip()

**3. Optional: combine it into a loop for entity processing**

Instead of writing the whole loop for you, think about this:

* How would you collect all entities of type GPE or LOC from your Stanza doc?
* Can you apply clean\_place\_name() inside that loop before storing or printing the name?

Would you like a quick review of how to extract entity type and text correctly in Stanza, just to confirm you're looping right?

4o

**You said:**

clean\_counts = {} for place, count in place\_counts.items(): place = re.sub(r"[']s\b", "", place) place = re.sub(r"[^\w\s]", "", place) place = re.sub(r"^the\s+","", place, flags=IGNORECASE) clean\_counts[place] = clean\_counts.get(place, 0) + count will this do? why am i getting an error in the line where in cleaning for the "THE"?

**ChatGPT said:**

You're **really close**, and this is a clean approach to normalizing your place names!

The issue is in this part of your code:

python

Copy code

place = re.sub(r"^the\s+","", place, flags=IGNORECASE)

You're using IGNORECASE without importing it. In Python's re module, IGNORECASE is actually re.IGNORECASE.

**✅ Fix:**

python

Copy code

import re # make sure this is at the top of your file

place = re.sub(r"^the\s+", "", place, flags=re.IGNORECASE)

**Extra tip:**

Your earlier regex for [']s\b` handles possessives (like “London’s”), which is smart. Keep an eye out for:

* Titles like “City of London” (might not want to remove “City of”).
* All-uppercase vs title case — sometimes normalizing case can help (e.g., place.lower()), but sometimes it erases important distinctions.

Want to add a step to normalize capitalization too?

4o