Scrambling in Persian from English Subtitles

John R. Starr

Presentation Outline:

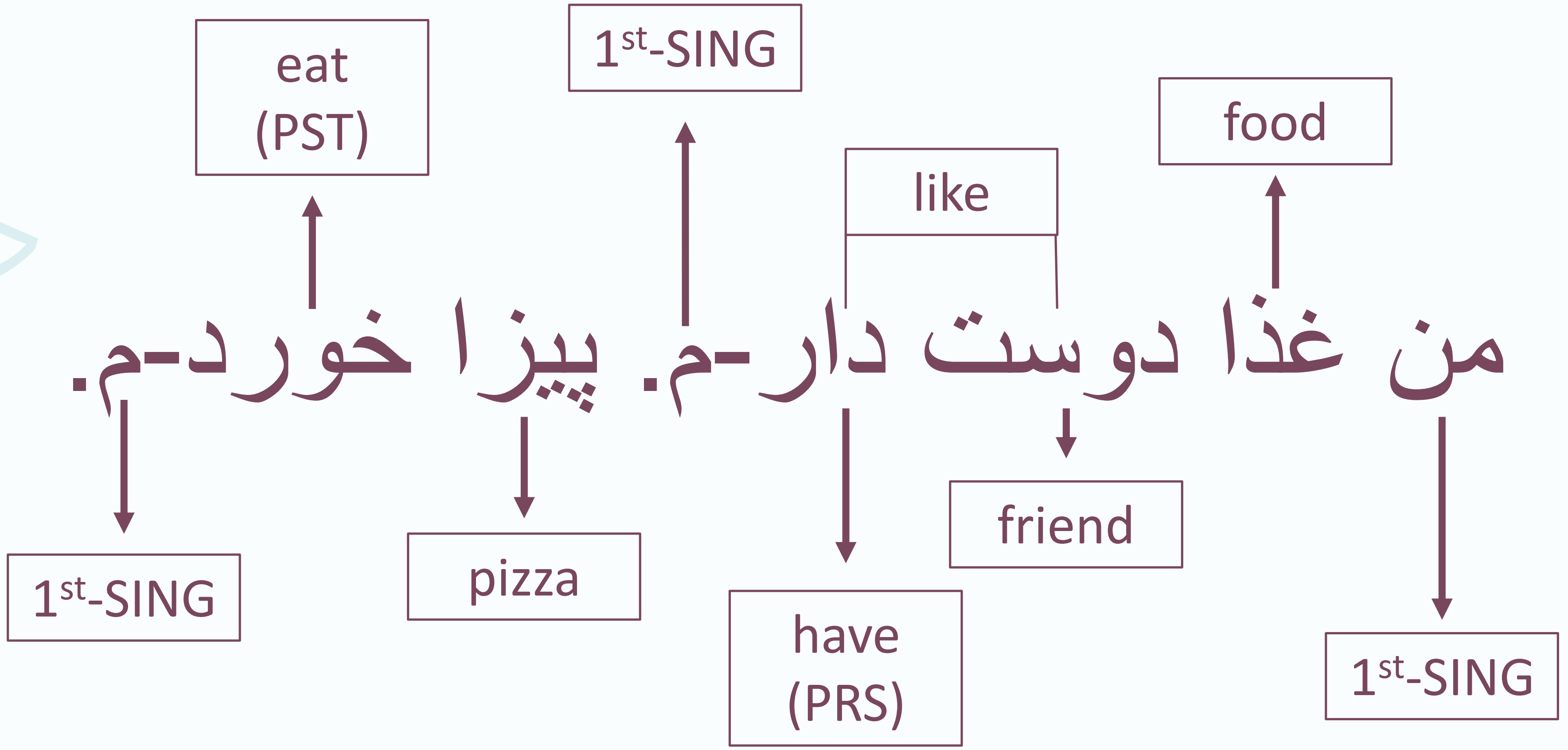
Part I:
On Persian

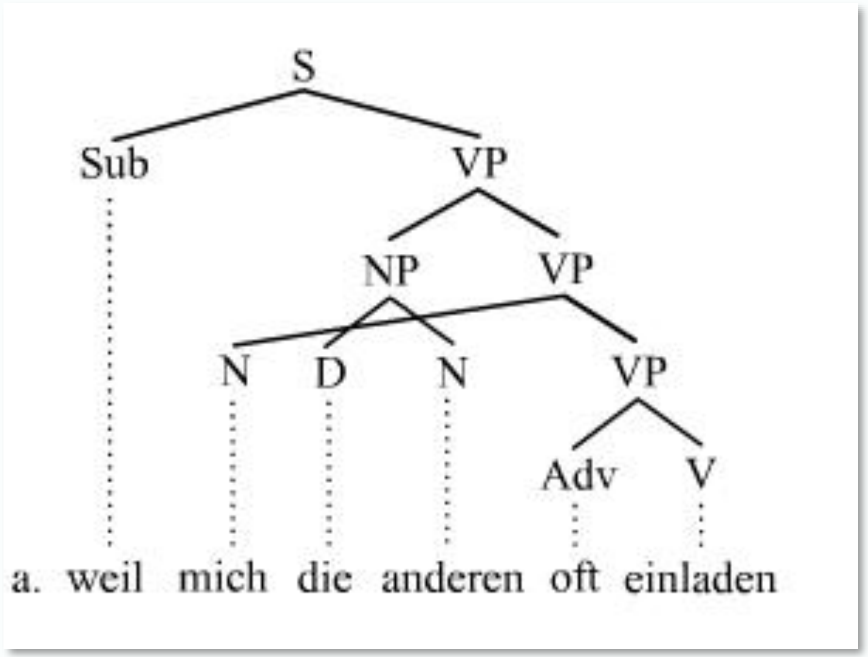
General characteristics, example sentence, scrambling, project motivation

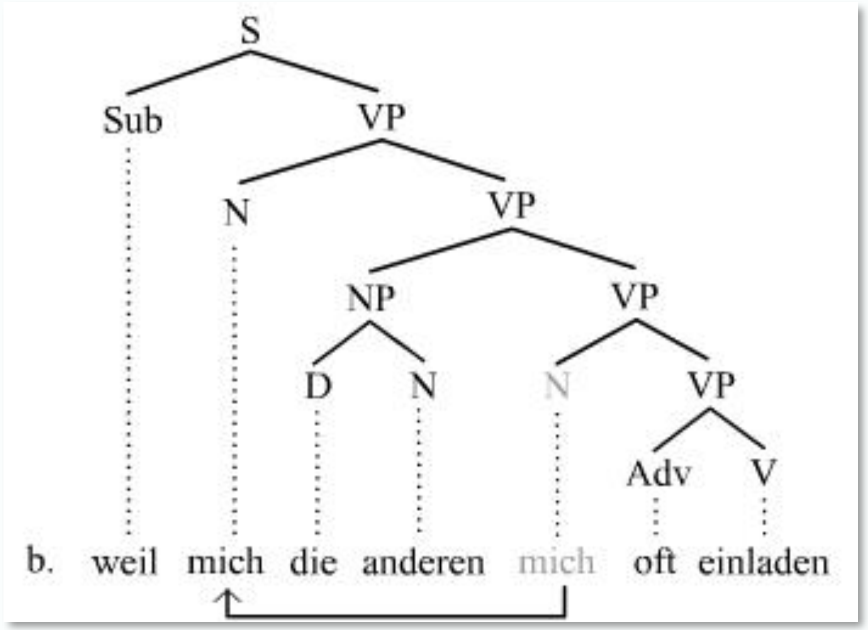
Persian: A Brief Description

- Indo-Iranian language spoken in Iran, Afghanistan, and Tajikistan
- Language subfamilies: Farsi (this project), Dari, and Tajik
- Reads right-to-left, using the Arabic alphabet (+4 letters)
- Many verbs are compound and include non-verbal components
- Is “pro-drop”: subject pronouns optional if already stated
- Underlyingly SOV, but is open to scrambling (DEFINE SCRAMBLING)

An Example Sentence:







What is scrambling?

An example from German: “Because the others invite me”

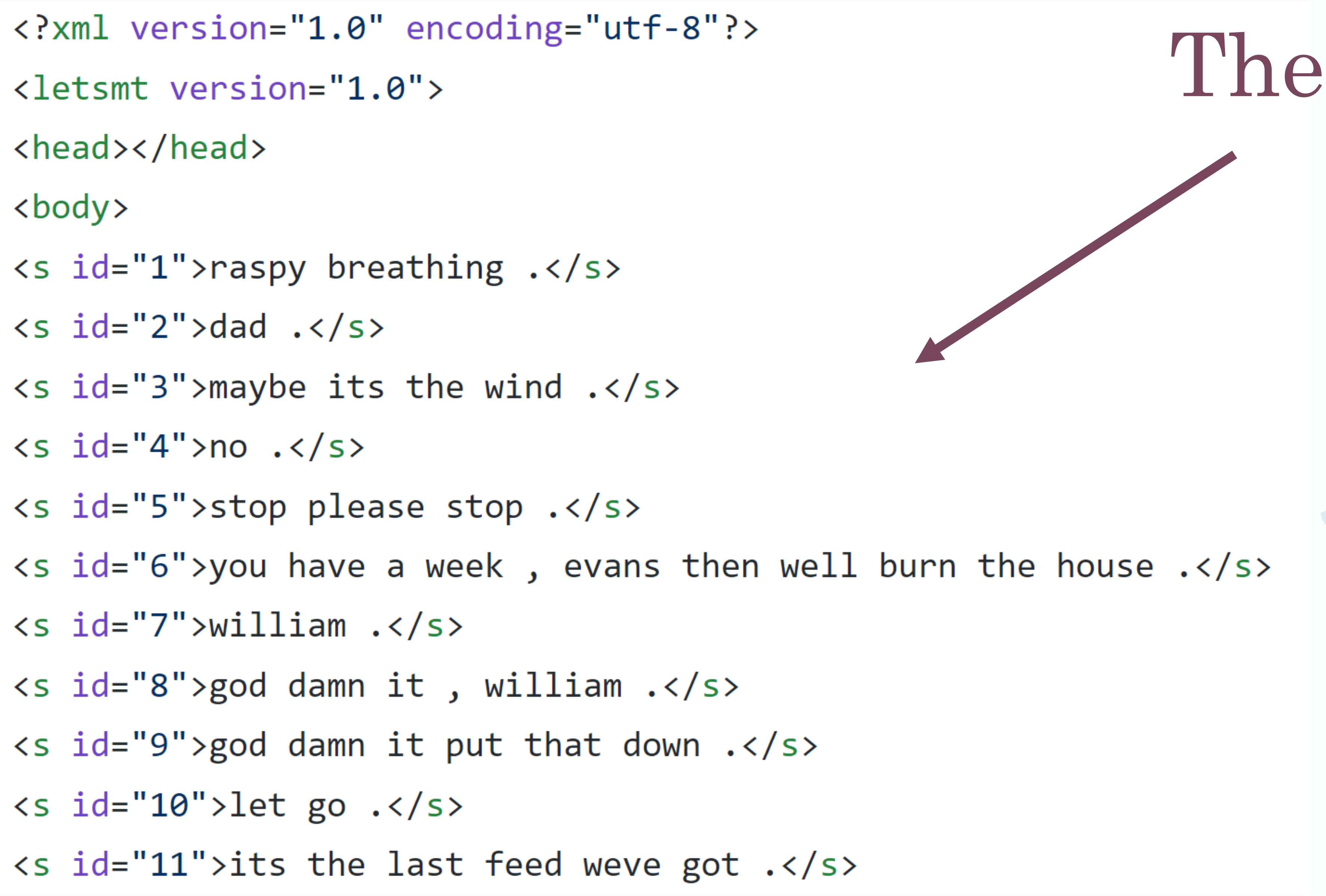
Part II: The Tehran English-Persian Parallel Corpus

Layout of corpus, example of raw data, positives/negatives of corpus

The Tehran English-Persian Parallel Corpus: What's it all about?

- Parallel corpus of English-Persian subtitles
- 554, 621 lines total
  - About 3.8 million words for each language
- Obtained from approx. 21,000 movie files from Open-subtitles (free online collection of movie subtitles in various languages)
- One large .xml file for each language:

The Raw Data



```xml
<?xml version="1.0" encoding="utf-8"?>
<letsmt version="1.0">
<head></head>
<body>
<s id="1">raspy breathing .</s>
<s id="2">dad .</s>
<s id="3">maybe its the wind .</s>
<s id="4">no .</s>
<s id="5">stop please stop .</s>
<s id="6">you have a week , evans then well burn the house .</s>
<s id="7">william .</s>
<s id="8">god damn it , william .</s>
<s id="9">god damn it put that down .</s>
<s id="10">let go .</s>
<s id="11">its the last feed weve got .</s>
```

The Ups and Downs of the TEP

+

- Data intended to sound like regular speech, rather than formal writing
- Accurate alignment of speakers due to timestamps
- .xml files clearly indexed

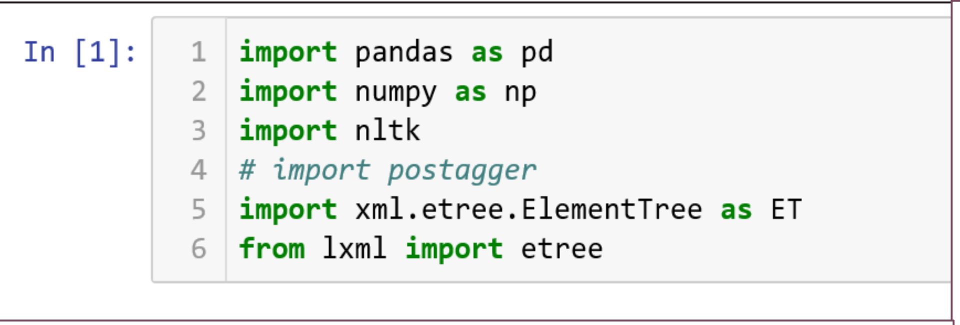
-

- Lack of punctuation
- Spoken Persian behaves very differently than written Persian
- Persian spelling varies greatly (but can have a great effect)
- Many incomplete sentences

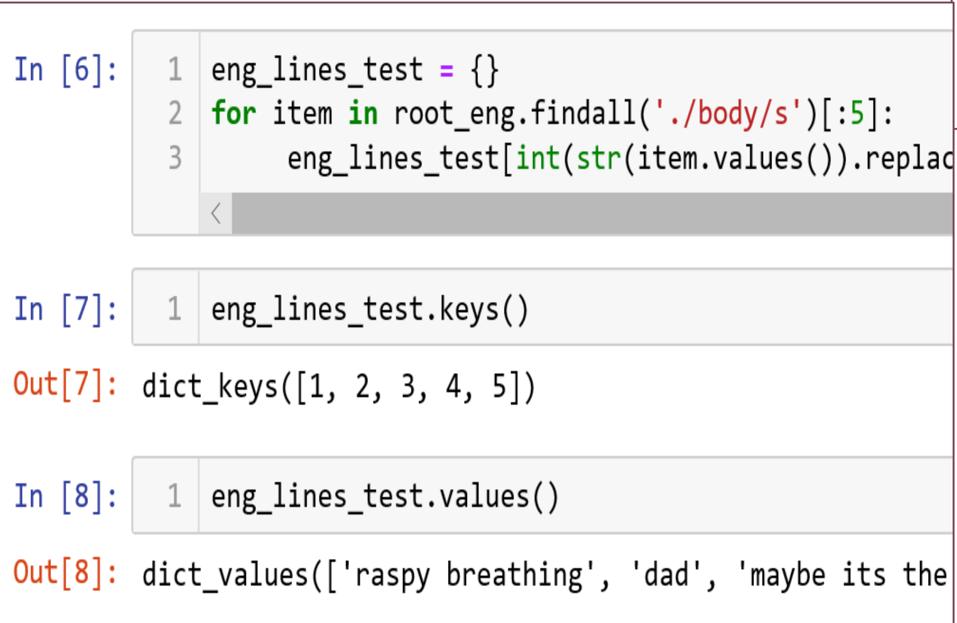
Part III: Organizing the Data

Extracting data from .xml files, moving to DFs, combining data into one big DF

The Process:



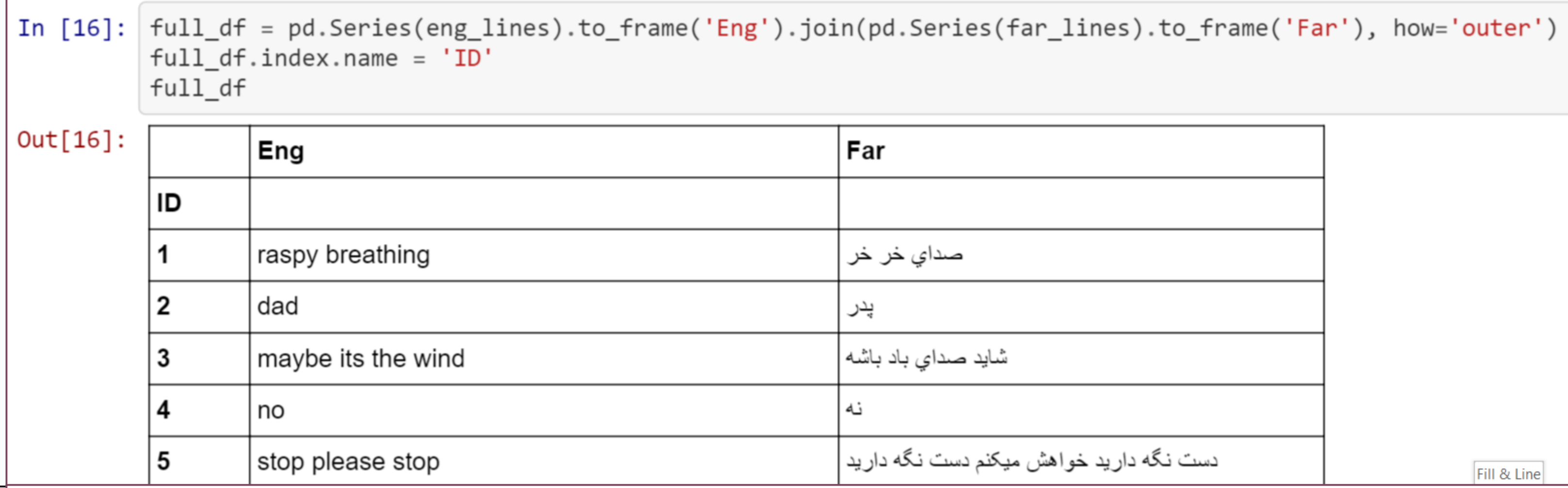
```
In [1]:
import pandas as pd
import numpy as np
import nltk
# import postagger
import xml.etree.ElementTree as ET
from lxml import etree
```



```
In [6]:
eng_lines_test = {}
for item in root_eng.findall('./body/s')[:5]:
    eng_lines_test[int(str(item.values()).replac
```

```
In [7]:
eng_lines_test.keys()
Out[7]: dict_keys([1, 2, 3, 4, 5])
```

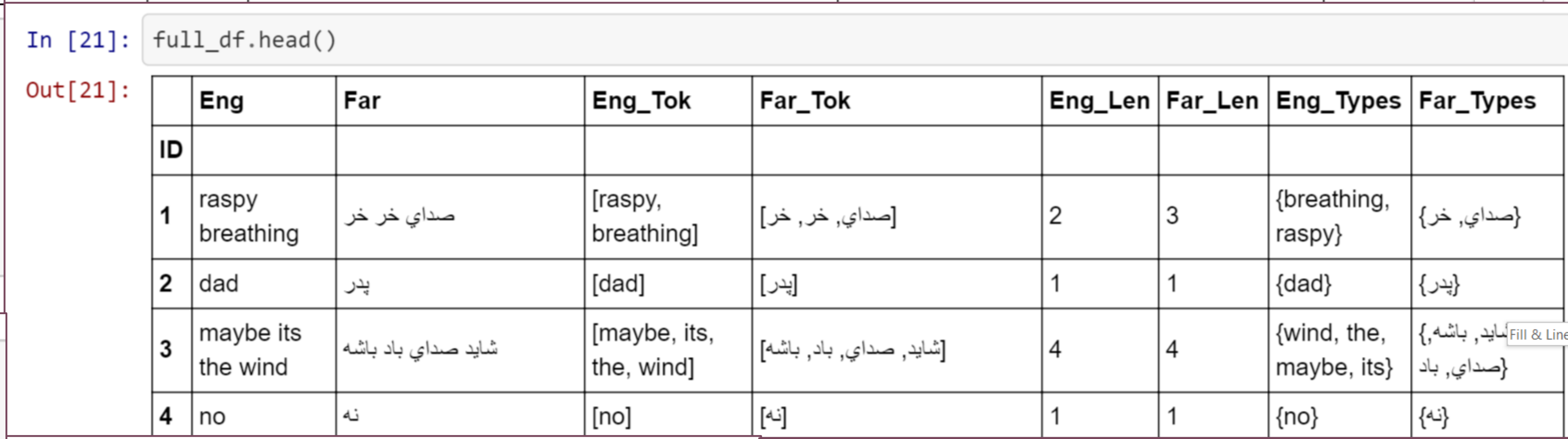
```
In [8]:
eng_lines_test.values()
Out[8]: dict_values(['raspy breathing', 'dad', 'maybe its the
```



```
In [16]: full_df = pd.Series(eng_lines).to_frame('Eng').join(pd.Series(far_lines).to_frame('Far'), how='outer')
         full_df.index.name = 'ID'
         full_df
```

Out[16]:

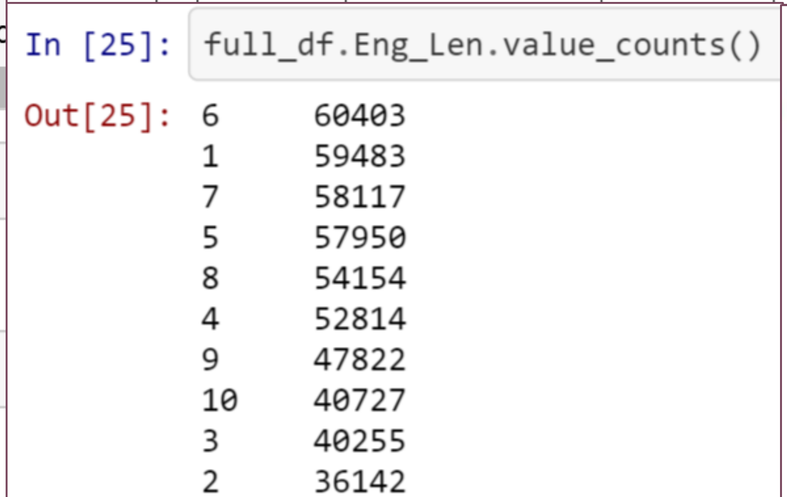
	Eng	Far
ID		
1	raspy breathing	صداي خر خر
2	dad	پدر
3	maybe its the wind	شايد صداي باد باشه
4	no	نه
5	stop please stop	دست نگه داريد خواهش ميكنم دست نگه داريد



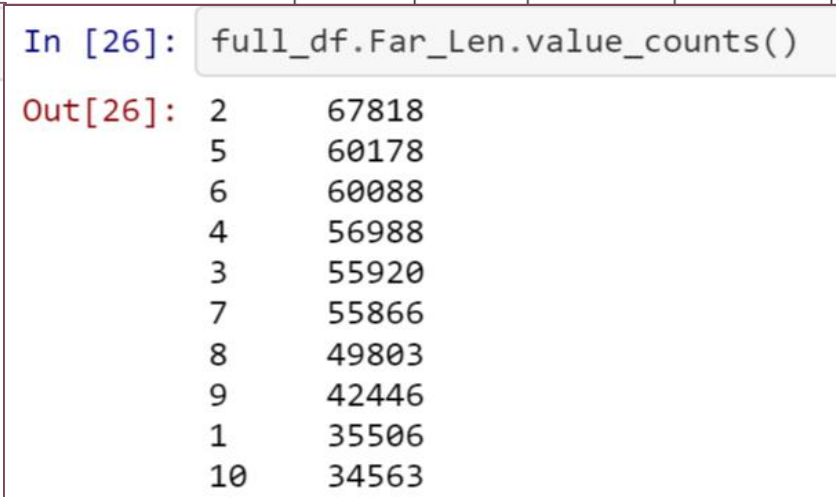
```
In [21]: full_df.head()
```

Out[21]:

	Eng	Far	Eng_Tok	Far_Tok	Eng_Len	Far_Len	Eng_Types	Far_Types
ID								
1	raspy breathing	صداي خر خر	[raspy, breathing]	[صداي, خر, خر]	2	3	{breathing, raspy}	{صداي, خر}
2	dad	پدر	[dad]	[پدر]	1	1	{dad}	{پدر}
3	maybe its the wind	شايد صداي باد باشه	[maybe, its, the, wind]	[شايد, صداي, باد, باشه]	4	4	{wind, the, maybe, its}	{شايد, باشه, صداي, باد}
4	no	نه	[no]	[نه]	1	1	{no}	{نه}



```
In [25]: full_df.Eng_Len.value_counts()

Out[25]: 6      60403
         1      59483
         7      58117
         5      57950
         8      54154
         4      52814
         9      47822
         10     40727
         3      40255
         2      36142
```

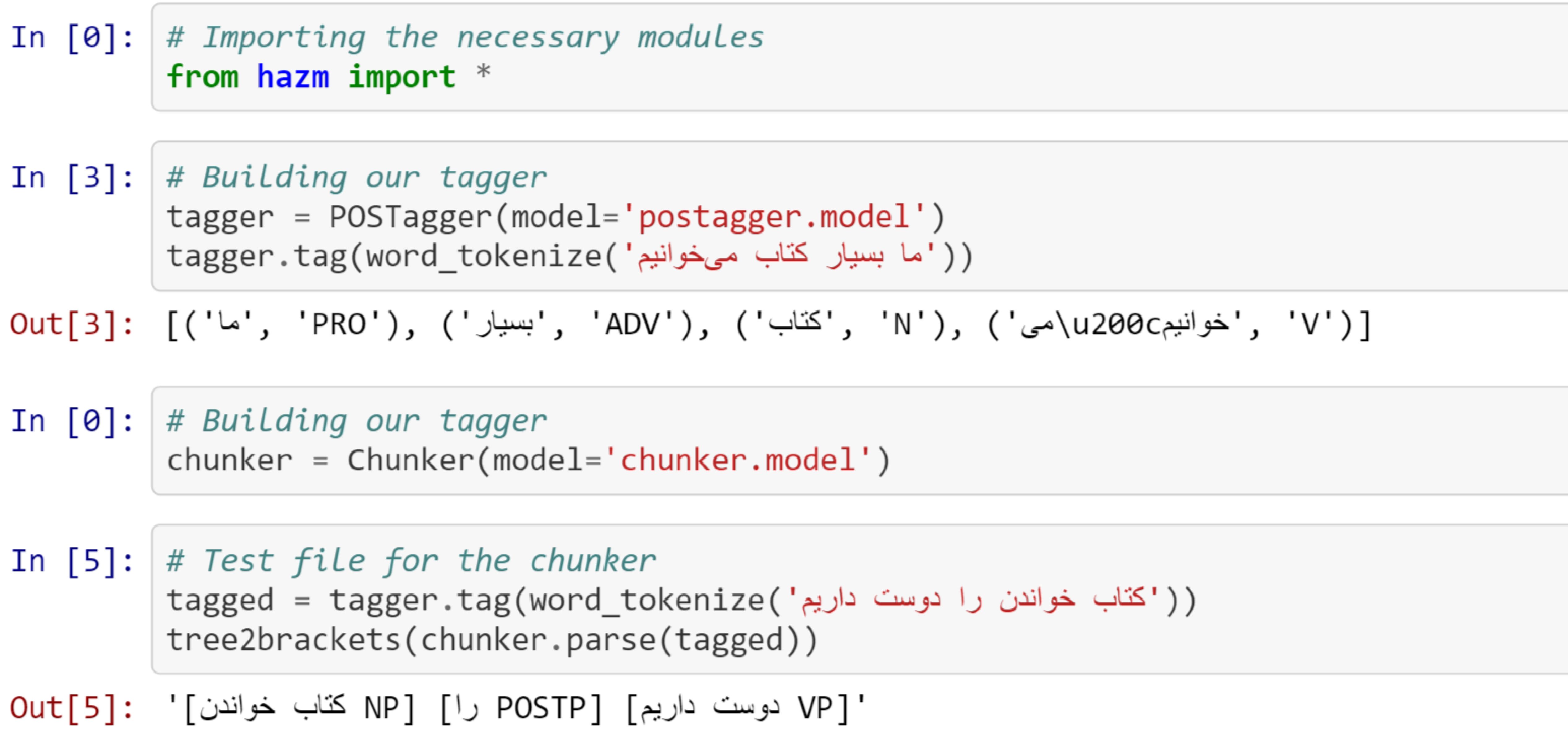


```
In [26]: full_df.Far_Len.value_counts()

Out[26]: 2      67818
         5      60178
         6      60088
         4      56988
         3      55920
         7      55866
         8      49803
         9      42446
         1      35506
         10     34563
```

Part IV: Modifying the Data

POS-tagging, chunking, generalizing

POS-tagging and Chunking Persian Text



```
In [0]:  # Importing the necessary modules
         from hazm import *
```

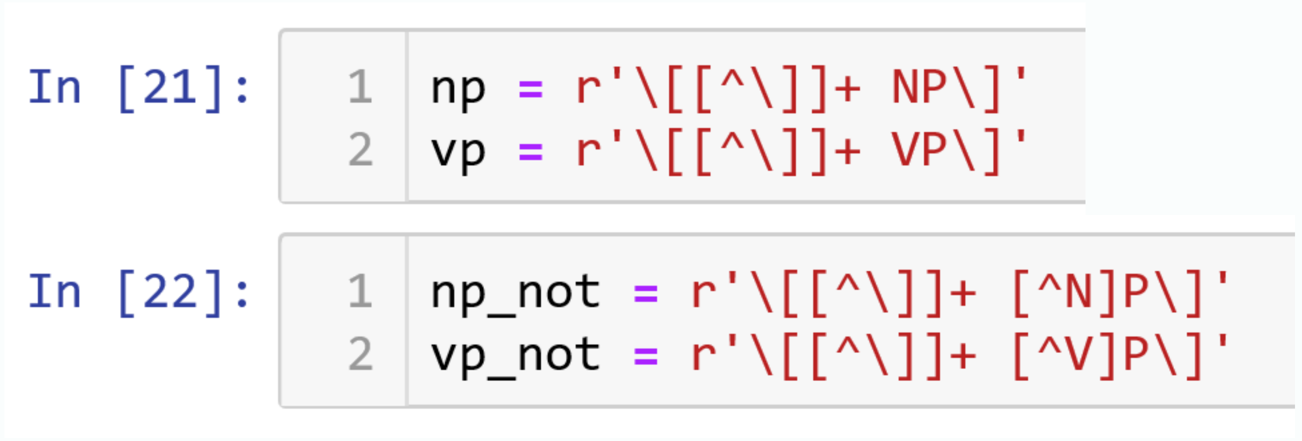
```
In [3]:  # Building our tagger
         tagger = POSTagger(model='postagger.model')
         tagger.tag(word_tokenize('ما بسیار کتاب می‌خوانیم'))

Out[3]:  [('ما', 'PRO'), ('بسیار', 'ADV'), ('کتاب', 'N'), ('می\u200cخوانیم', 'V')]
```

```
In [0]:  # Building our tagger
         chunker = Chunker(model='chunker.model')
```

```
In [5]:  # Test file for the chunker
         tagged = tagger.tag(word_tokenize('کتاب خواندن را دوست داریم'))
         tree2brackets(chunker.parse(tagged))

Out[5]:  '[کتاب خواندن NP] [را POSTP] [دوست داریم VP]'
```

How to properly generalize the WO?

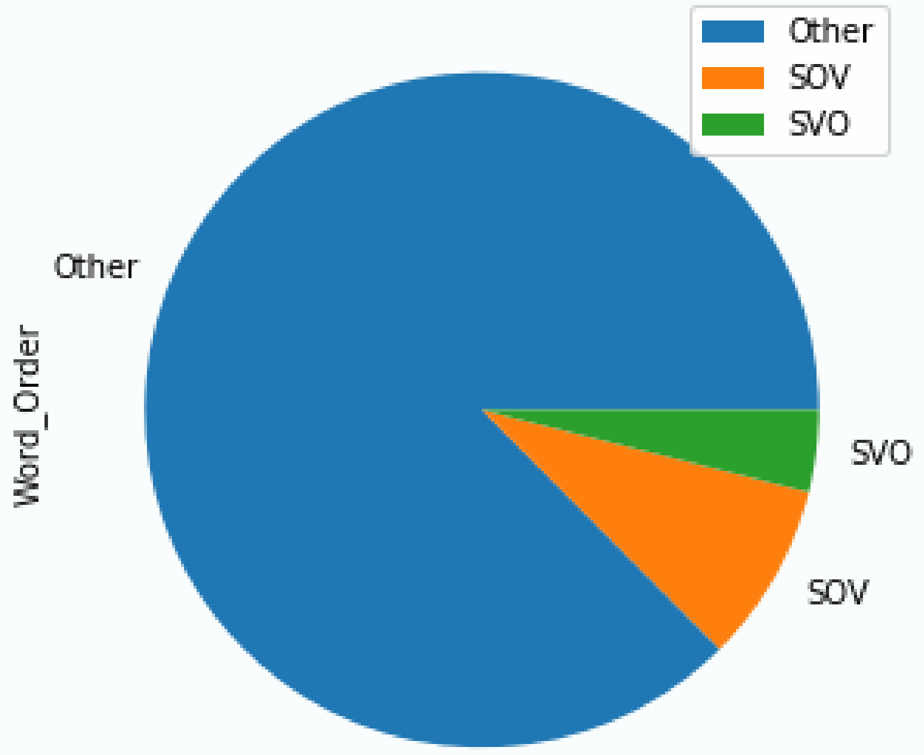
- Data is in strings → Determine WO with regex
  - Alternative methods?
- Defining NPs/VPs <u>and</u> defining what they're *not*:
  - What is:
  - What isn't:



```
In [21]:  np = r'\[[^\]]+ NP\]'
          vp = r'\[[^\]]+ VP\]'

In [22]:  np_not = r'\[[^\]]+ [^N]P\]'
          vp_not = r'\[[^\]]+ [^V]P\]'
```

Part V: Analyzing the Data

How the data is split up, general remarks, looking at data using the word “like”



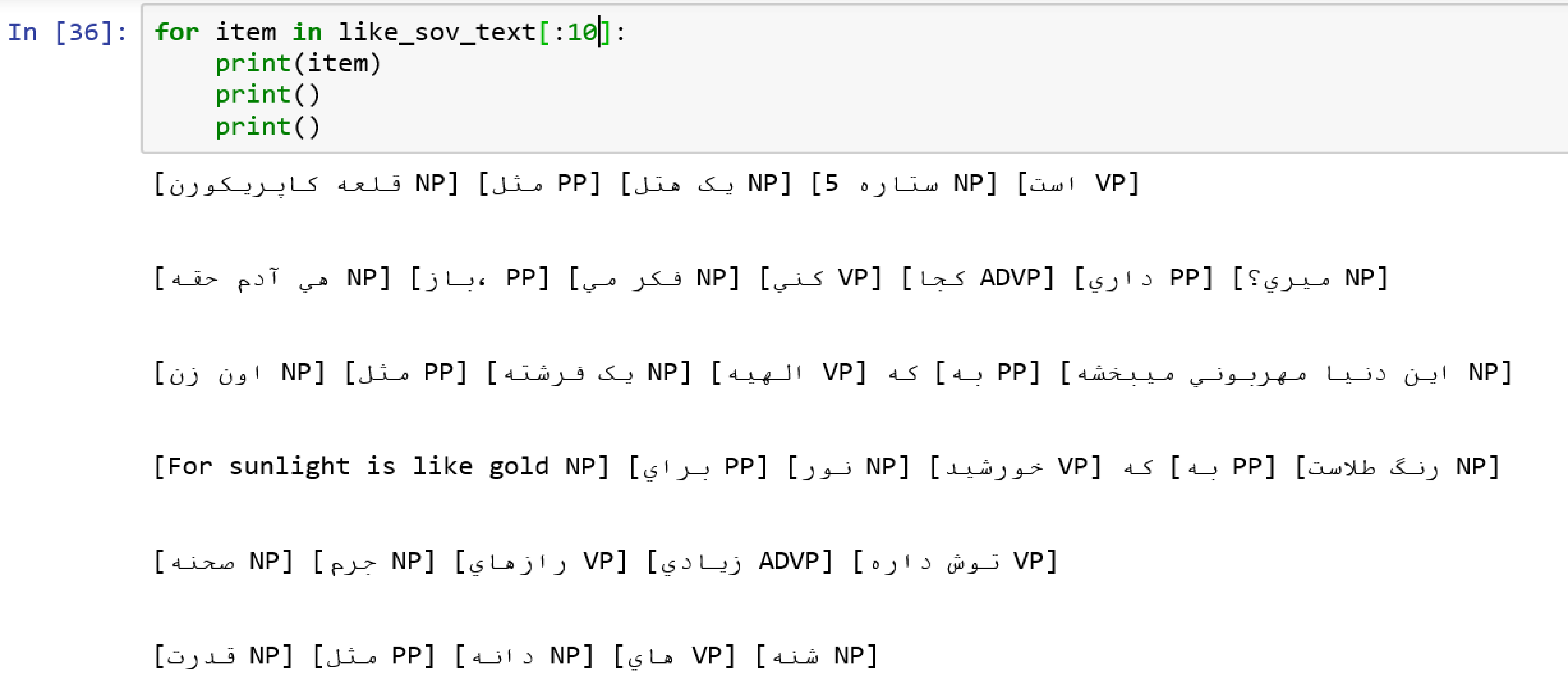
How do the data look?

Yikes…

Some Qualities of Generalizations:

- The chunker is… not great.
- SOV/SVO sentences generally correct (counts not yet discovered)
- “Other sentences” are either too simple, too complex, or have a weird quirk about them
- Looking into “is like” sentences:

Example SOV Sentences:



```
In [36]: for item in like_sov_text[:10]:
             print(item)
             print()
             print()
```

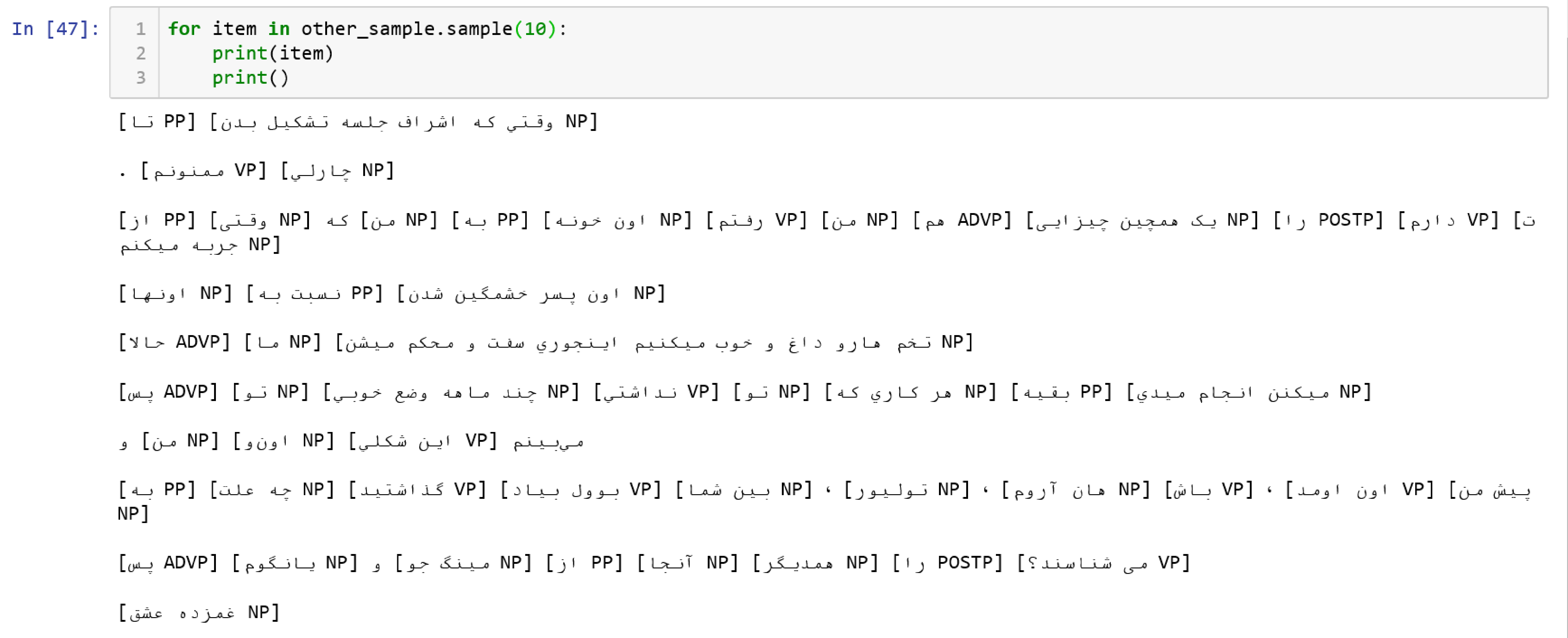
```
[قلعه کاپریکورن NP] [مثل PP] [یک هتل NP] [5 ستاره NP] [است VP]


[هي آدم حقه NP] [باز، PP] [مي فکر NP] [کني VP] [کجا ADVP] [داري PP] [ميري؟ NP]


[اون زن NP] [مثل PP] [یک فرشته NP] [الهيه VP] که [به PP] [اين دنيا مهربوني ميبخشه NP]


[For sunlight is like gold NP] [براي PP] [نور NP] [خورشيد VP] که [به PP] [رنگ طلاست NP]


[صحنه NP] [جرم NP] [رازهاي VP] [زيادي ADVP] [توش داره VP]


[قدرت NP] [مثل PP] [دانه NP] [هاي VP] [شنه NP]
```

Example SVO Sentences:

```
In [38]:
for item in like_svo_text[:10]:
    print(item)
    print()
    print()
```

```
[ویزیتها این NP] [شبیه VP] [تلوزیونیه مسابقه یه اغاز NP]


[این NP] [آهنگه VP] [میمونه خوره مثله NP]


[این NP] [مسته VP] [کفش هنر NP]


[کریسمس NP] [توي VP] [من کورس NP] [بین VP] [مامان NP] و [بابا NP] [میشه خلاصه NP]


[عشقمون NP] [شبیه VP] [قرمزه رز گل یک NP]


[پایین این زندگي NP] [شبیه VP] [تو زندگي NP] [فضاست VP] [نباید VP] [اشتباه زیاد NP] [کني VP]


[رفتار NP] [هاي VP] [آور شرم VP] [شهدا خون که اینه شبیه درست NP] [را POSTP] [کنیم پایمال NP]
```

What about the “Other” sentences?



```
In [47]:
for item in other_sample.sample(10):
    print(item)
    print()
```

```
[PP تا] [NP وقتي که اشراف جلسه تشکيل بدن]

. [VP ممنونم] [NP چارلي]

[PP از] [NP وقتی] که [NP من] [PP به] [NP اون خونه] [VP رفتم] [NP من] [ADVP هم] [NP یک همچین چیزایی] [POSTP را] [VP دارم] [ت
جربه میکنم NP]

[NP اونها] [PP نسبت به] [NP اون پسر خشمگین شدن]

[ADVP حالا] [NP ما] [NP تخم هارو داغ و خوب ميکنيم اينجوري سفت و محکم ميشن]

[ADVP پس] [NP تو] [NP چند ماهه وضع خوبي] [VP نداشتي] [NP تو] [NP هر کاري که] [PP بقيه] [NP ميکنن انجام ميدي]

و [NP من] [NP اونو] [VP اين شکلي] مي‌بينم

[PP به] [NP چه علت] [VP گذاشتيد] [VP بوول بياد] [NP بين شما] ، [NP توليور] ، [NP هان آروم] [VP باش] ، [VP اون اومد] [پيش من
NP]

[ADVP پس] [NP يانگوم] و [NP مينگ جو] [PP از] [NP آنجا] [NP همديگر] [POSTP را] [VP مى شناسند؟]

[NP غمزده عشق]
```

Part VI: Conclusions

(Preliminary) results, further directions, possible corrections

Preliminary Results

- Constructions from similes lean towards SOV, whereas metaphorical constructions lean towards SVO
- More abstract comparisons also lean towards SVO
- Many different constructions will be analyzed

Limitations of Project:

- Persian’s inconsistent transcription and subject availability
- Persian parser/chunker/generalizer are limited and do not appear to have high levels of accuracy
- Nearly 80% of the data is categorized as “Other”

Further Directions:

- Machine learning program that categorizes sentences for scrambling
- Working with the speech data itself
- Sentiment marking

THE END

Questions?

References:

– Link to Tehran-Parallel Corpus: http://opus.nlpl.eu/TEP.php

  – Information on the TEP: https://link.springer.com/content/pdf/10.1007%2F978-3-642-19437-5.pdf

  – M. T. Pilevar, H. Faili, and A. H. Pilevar, â€œTEP: Tehran English-Persian Parallel Corpusâ€�in proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2011).

– Information on Hazm: https://github.com/sobhe/hazm

– Project Github: https://github.com/Data-Science-for-Linguists-2019/Scrambling-in-English-to-Persian-Subtitles