Connections Project

Monday, October 2, 2023 9:40 PM

Cogan's suggestion :

Regarding the connections problem:

- A particular word can be expanded into sets of other words. These could be
  - semantic (perhaps, in the notion of hypernymy/hyponymy) -- this could come from WordNet
  - context (perhaps, in the notion of scenes "read" vs "read" or "ma (slang)" vs "ma (last name)") -- this could come from Wikidata

Perhaps we can identify possibly connections by intersecting the clouds that result from this. Happy to see what you think about!

-Cogan

Find groups of four items that share something in common.

- Select four items and tap 'Submit' to check if your guess is correct.
- Find the groups without making 4 mistakes!

Category Examples

- FISH: Bass, Flounder, Salmon, Trout
- FIRE ___: Ant, Drill, Island, Opal

Categories will always be more specific than "5-LETTER WORDS," "NAMES" or "VERBS."

Each puzzle has exactly one solution. Watch out for words that seem to belong to multiple categories!

Each group is assigned a color, which will be revealed as you solve:

Straightforward

↓

Tricky

Definitions:

Hypernymy	A word with a broad meaning that more specific words fall under; a superordinate. For example, *color* is a hypernym of *red*.
Hyponymy	A word of more specific meaning than a general or superordinate term applicable to it. For example, *spoon* is a hyponym of *cutlery*.
WordNet	https://wordnet.princeton.edu/ http://wordnetweb.princeton.edu/perl/webwn https://wordnet.princeton.edu/frequently-asked-questions
Wikidata	https://www.wikidata.org/wiki/Wikidata:Main_Page

Python

- Natural Language Toolkit has taken over the development of **pywordnet**. There is now a Python package, nltk_lite.wordnet, which incorporates pywordnet and which supports WordNet 2.1. It is included in NLTK Lite.

From <https://wordnet.princeton.edu/related-projects#Python>

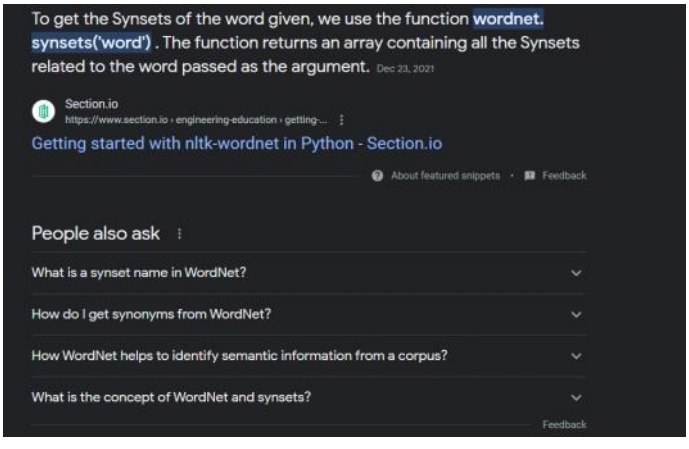
https://stackoverflow.com/questions/35117028/synonyms-join-with-a-string-of-words-in-python

https://www.holisticseo.digital/python-seo/nltk/wordnet

GENERAL STEPS

1. Take 16 English words as an input
2. Expand each individual word into a set of synonyms
3. Compare the sets of synonyms?
4. Determine the words that connect most closely

Tackle Wikidata tomorrow



. Lemmatization links similar meaning words as one word, making tools such as chatbots and search engine queries more effective and accurate.

From <https://www.google.com/search?q=what+does+the+lemmas+fuunction+do&rlz=1C1UEAD_enUS1057US1057&oq=what+does+the+lemmas+fuunction+do&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIJCAEQIRgKGKABMgkIAhAhGAoYoAEyCQgDECEYChigATIHCAQQIRirAjIHCAUQIRirAjIHCAYQIRirAjIKCAcQIRgWGB0YHjIKCAgQIRgWGB0YHtIBCTEwOTA3ajBqN6gCALACAA&sourceid=chrome&ie=UTF-8#cobssid=s>

```
#import Wordnet and the Natural Language Toolkit
from nltk.corpus import wordnet
#Create empty list add synonyms to
synonyms = []
#Loop through all the words in the synset and bring them down to the basic
level(lemma), add each synonym to list
for syn in wordnet.synsets(""):
    for i in syn.lemmas():
        synonyms.append(i.name())
print(set(synonyms))
```

```
#import Wordnet and the Natural Language Toolkit
from nltk.corpus import wordnet
inp = input("What is the words you want synonyms for? ")
inps = inp.split(' ')
#Create empty list add synonyms to
synonyms = []
#Loop through all the words in the synset and bring them down to the basic
level(lemma), add each synonym to list
for h in inps:
    for syn in wordnet.synsets(h):
        for i in syn.lemmas():
            synonyms.append(i.name())
print(set(synonyms))
```