```
# This script serves for finding the lemmas of each words that belongs in the ontology of
accommodation
# Since a word can have several meanings, in the initial phase the script will ask the owner of
the system
# to distinguish the potential candidate lemmas based on their definition in WordNet. This
script aims to create
# a list of synonyms/hyponyms/hypernyms which will serve as a comparison basis for finding the
features in opinions

## The term 'user' below refers to the service owner


from nltk.corpus.reader.wordnet import Synset
from nltk.corpus import wordnet as wn
import os

feature = raw_input('Choose a feature to base your ranking: ')

#After the user inputs the feature, list all the synsets of this feature
f = open('C:/Python27/ontology/'+ feature + '.txt','w')
L = wn.synsets(feature)
for item in L:
    print item, item.definition()

    #Ask the user if this was the definition that matches his search
    answer = raw_input('Does this definition match your search (Y/N): ')
    if answer=='Y' or answer=='y':
        print [str(lemma.name()) for lemma in item.lemmas()]
        thelist1= [str(lemma.name()) for lemma in item.lemmas()]
        for ele in thelist1: f.write(ele + '\n')


        #If the user answers yes then find the hyponyms for this word
        K1 = item.hyponyms()
        for hypo in K1:
            print hypo, hypo.definition()
            answer1 = raw_input('Does this definition match your search (Y/N): ')
            if answer1=='Y' or answer1=='y':
                print [str(lemma.name()) for lemma in hypo.lemmas()]
                thelist2=[str(lemma.name()) for lemma in hypo.lemmas()]
                for ele in thelist2: f.write(ele + '\n')

        #Also find the hypernyms of the word
        K2 = item.hypernyms()
        for hyper in K2:
            print hyper, hyper.definition()
            answer2 = raw_input('Does this definition match your search (Y/N): ')
            if answer2=='Y' or answer2=='y':
                print [str(lemma.name()) for lemma in hyper.lemmas()]
                thelist3=[str(lemma.name()) for lemma in hyper.lemmas()]
                for ele in thelist3: f.write(ele + '\n')

                #And for the hypernyms find the hyponyms (nested loop to ensure enrichment of
                ontology)
                K3 = hyper.hyponyms()
                for hypohyper in K3:
                    print hypohyper, hypohyper.definition()
                    answer3 = raw_input('Does this definition match your search (Y/N): ')
                    if answer3=='Y' or answer3=='y':
                        print [str(lemma.name()) for lemma in hypohyper.lemmas()]
                        thelist4=[str(lemma.name()) for lemma in hypohyper.lemmas()]
                        for ele in thelist4: f.write(ele + '\n')

raw_input("Press Enter to continue ..." )
```