

nlTK

October 15, 2021

```
[20]: import nltk
```

```
[21]: from nltk.tokenize import sent_tokenize
text="""My Wonderful Family. I live in a house near the mountains. I have two
↳brothers and one sister, and I was born last. My father teaches mathematics,
↳and my mother is a nurse at a big hospital. My brothers are very smart and
↳work hard in school. My sister is a nervous girl, but she is very kind. My
↳grandmother also lives with us. She came from Italy when I was two years old.
↳ She has grown old, but she is still very strong. She cooks the best food!
My family is very important to me. We do lots of things together. My brothers
↳and I like to go on long walks in the mountains. My sister likes to cook
↳with my grandmother. On the weekends we all play board games together. We
↳laugh and always have a good time. I love my family very much."""
tokenized_text=sent_tokenize(text)
print(tokenized_text)
```

```
['My Wonderful Family.', 'I live in a house near the mountains.', 'I have two
brothers and one sister, and I was born last.', 'My father teaches mathematics,
and my mother is a nurse at a big hospital.', 'My brothers are very smart and
work hard in school.', 'My sister is a nervous girl, but she is very kind.', 'My
grandmother also lives with us.', 'She came from Italy when I was two years
old.', 'She has grown old, but she is still very strong.', 'She cooks the best
food!', 'My family is very important to me.', 'We do lots of things together.',
'My brothers and I like to go on long walks in the mountains.', 'My sister likes
to cook with my grandmother.', 'On the weekends we all play board games
together.', 'We laugh and always have a good time.', 'I love my family very
much.']
```

```
[22]: from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['My', 'Wonderful', 'Family', '.', 'I', 'live', 'in', 'a', 'house', 'near',
'the', 'mountains', '.', 'I', 'have', 'two', 'brothers', 'and', 'one', 'sister',
',', 'and', 'I', 'was', 'born', 'last', '.', 'My', 'father', 'teaches',
'mathematics', ',', 'and', 'my', 'mother', 'is', 'a', 'nurse', 'at', 'a', 'big',
'hospital', '.', 'My', 'brothers', 'are', 'very', 'smart', 'and', 'work',
'hard', 'in', 'school', '.', 'My', 'sister', 'is', 'a', 'nervous', 'girl', ',',
'but', 'she', 'is', 'very', 'kind', '.', 'My', 'grandmother', 'also', 'lives',
```

```
'with', 'us', '.', 'She', 'came', 'from', 'Italy', 'when', 'I', 'was', 'two',
'years', 'old', '.', 'She', 'has', 'grown', 'old', ',', 'but', 'she', 'is',
'still', 'very', 'strong', '.', 'She', 'cooks', 'the', 'best', 'food', '!',
'My', 'family', 'is', 'very', 'important', 'to', 'me', '.', 'We', 'do', 'lots',
'of', 'things', 'together', '.', 'My', 'brothers', 'and', 'I', 'like', 'to',
'go', 'on', 'long', 'walks', 'in', 'the', 'mountains', '.', 'My', 'sister',
'likes', 'to', 'cook', 'with', 'my', 'grandmother', '.', 'On', 'the',
'weekends', 'we', 'all', 'play', 'board', 'games', 'together', '.', 'We',
'laugh', 'and', 'always', 'have', 'a', 'good', 'time', '.', 'I', 'love', 'my',
'family', 'very', 'much', '.']
```

```
[38]: from nltk.probability import FreqDist
fdist = FreqDist(tokenized_word)
print(fdist)
print(fdist.freq("long"))
print(fdist.most_common(2))
print(fdist.max())
print(fdist.tabulate())
```

```
<FreqDist with 93 samples and 167 outcomes>
```

```
0.005988023952095809
```

```
[('.', 16), ('My', 8)]
```

```
.
.
My I and a is
very the , in brothers sister my
She to mountains have two was but
she grandmother with old family We together
Wonderful Family live house near one
born last father teaches mathematics mother nurse
at big hospital are smart work hard
school nervous girl kind also lives us
came from Italy when years has grown
still strong cooks best food ! important
me do lots of things like go
on long walks likes cook On weekends
we all play board games laugh always
good time love much
16 8 6 6 5 5
5 4 4 3 3 3 3
3 3 2 2 2 2 2
2 2 2 2 2 2 2
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
```

```

1          1          1          1          1          1          1
1          1          1          1          1          1          1
1          1          1
None

```

```
[24]: from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```

{"won't", 'most', 'the', 'not', 'out', 'it', 'off', "that'll", 'while',
"hadn't", 'yourself', 'his', 'on', 'no', 'hadn', 'all', "it's", 'there', 'can',
'isn', 'but', 'of', 'herself', 'if', 'be', 'down', "haven't", 'before', 'nor',
'this', "you'd", 'her', 'here', 'only', 'at', 'been', 'themselves', 'have',
"mustn't", 'against', 'needn', 'above', 'too', 'didn', "isn't", 's', 'mightn',
'weren', 'he', "needn't", "wasn't", "couldn't", 'about', 'did', 've', 'or',
'up', 'theirs', "she's", 'was', 'them', 'during', 'mustn', 'for', 'do', 'does',
're', 'wouldn', 'and', 'your', 'that', 'doesn', 'because', 'below', "aren't",
'as', 'yours', 'between', "shouldn't", 'some', 'to', 'doing', "weren't",
'where', 'again', 'should', 'has', 'its', 'then', 'being', 'd', 'so', 'is',
'himself', 'him', 'from', 'whom', 'me', 'you', 'haven', "mightn't", 'aren',
'hers', 'when', 'such', 'through', 'itself', 'o', 'once', 'more', "you're",
'into', 'are', 'each', 'after', 'shan', 'couldn', 'm', "shan't", 'until',
'myself', 'ours', 'ourselves', 'ma', 'yourselves', 'why', 'won', 'these', 'y',
'wasn', 'our', 'under', 'their', 'who', 'by', 'other', 'shouldn', 'any', 'had',
'than', 'll', "didn't", 'ain', 'we', 'they', "don't", 'am', 'will', 'a', 'she',
'same', "should've", 'in', "you'll", 'don', 'were', 'having', "wouldn't",
'with', 'just', 'few', 'over', 'i', "hasn't", 'both', 'very', 'hasn', 'further',
'what', 'those', "doesn't", "you've", 'own', 'now', 'how', 't', 'which', 'an',
'my'}

```

```
[25]: filtered_sent=[]
for w in tokenized_word:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Word:",tokenized_word)
print("Filtered Word:",filtered_sent)
```

```

Tokenized Word: ['My', 'Wonderful', 'Family', '.', 'I', 'live', 'in', 'a',
'house', 'near', 'the', 'mountains', '.', 'I', 'have', 'two', 'brothers', 'and',
'one', 'sister', ',', 'and', 'I', 'was', 'born', 'last', '.', 'My', 'father',
'teaches', 'mathematics', ',', 'and', 'my', 'mother', 'is', 'a', 'nurse', 'at',
'a', 'big', 'hospital', '.', 'My', 'brothers', 'are', 'very', 'smart', 'and',
'work', 'hard', 'in', 'school', '.', 'My', 'sister', 'is', 'a', 'nervous',
'girl', ',', 'but', 'she', 'is', 'very', 'kind', '.', 'My', 'grandmother',
'also', 'lives', 'with', 'us', '.', 'She', 'came', 'from', 'Italy', 'when', 'I',
'was', 'two', 'years', 'old', '.', 'She', 'has', 'grown', 'old', ',', 'but',
'she', 'is', 'still', 'very', 'strong', '.', 'She', 'cooks', 'the', 'best',
'food', '!', 'My', 'family', 'is', 'very', 'important', 'to', 'me', '.', 'We',
'do', 'lots', 'of', 'things', 'together', '.', 'My', 'brothers', 'and', 'I',

```

```
'like', 'to', 'go', 'on', 'long', 'walks', 'in', 'the', 'mountains', '.', 'My',
'sister', 'likes', 'to', 'cook', 'with', 'my', 'grandmother', '.', 'On', 'the',
'weekends', 'we', 'all', 'play', 'board', 'games', 'together', '.', 'We',
'laugh', 'and', 'always', 'have', 'a', 'good', 'time', '.', 'I', 'love', 'my',
'family', 'very', 'much', '.']
Filtered Word: ['My', 'Wonderful', 'Family', '.', 'I', 'live', 'house', 'near',
'mountains', '.', 'I', 'two', 'brothers', 'one', 'sister', ',', 'I', 'born',
'last', '.', 'My', 'father', 'teaches', 'mathematics', ',', 'mother', 'nurse',
'big', 'hospital', '.', 'My', 'brothers', 'smart', 'work', 'hard', 'school',
.', 'My', 'sister', 'nervous', 'girl', ',', 'kind', '.', 'My', 'grandmother',
'also', 'lives', 'us', '.', 'She', 'came', 'Italy', 'I', 'two', 'years', 'old',
.', 'She', 'grown', 'old', ',', 'still', 'strong', '.', 'She', 'cooks', 'best',
'food', '!', 'My', 'family', 'important', '.', 'We', 'lots', 'things',
'together', '.', 'My', 'brothers', 'I', 'like', 'go', 'long', 'walks',
'mountains', '.', 'My', 'sister', 'likes', 'cook', 'grandmother', '.', 'On',
'weekends', 'play', 'board', 'games', 'together', '.', 'We', 'laugh', 'always',
'good', 'time', '.', 'I', 'love', 'family', 'much', '.']
```

```
[26]: from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
```

```
ps = PorterStemmer()

stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))

print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
```

```
Filtered Sentence: ['My', 'Wonderful', 'Family', '.', 'I', 'live', 'house',
'near', 'mountains', '.', 'I', 'two', 'brothers', 'one', 'sister', ',', 'I',
'born', 'last', '.', 'My', 'father', 'teaches', 'mathematics', ',', 'mother',
'nurse', 'big', 'hospital', '.', 'My', 'brothers', 'smart', 'work', 'hard',
'school', '.', 'My', 'sister', 'nervous', 'girl', ',', 'kind', '.', 'My',
'grandmother', 'also', 'lives', 'us', '.', 'She', 'came', 'Italy', 'I', 'two',
'years', 'old', '.', 'She', 'grown', 'old', ',', 'still', 'strong', '.', 'She',
'cooks', 'best', 'food', '!', 'My', 'family', 'important', '.', 'We', 'lots',
'things', 'together', '.', 'My', 'brothers', 'I', 'like', 'go', 'long', 'walks',
'mountains', '.', 'My', 'sister', 'likes', 'cook', 'grandmother', '.', 'On',
'weekends', 'play', 'board', 'games', 'together', '.', 'We', 'laugh', 'always',
'good', 'time', '.', 'I', 'love', 'family', 'much', '.']
Stemmed Sentence: ['my', 'wonder', 'famili', '.', 'i', 'live', 'hous', 'near',
'mountain', '.', 'i', 'two', 'brother', 'one', 'sister', ',', 'i', 'born',
'last', '.', 'my', 'father', 'teach', 'mathemat', ',', 'mother', 'nurs', 'big',
'hospit', '.', 'my', 'brother', 'smart', 'work', 'hard', 'school', '.', 'my',
'sister', 'nervou', 'girl', ',', 'kind', '.', 'my', 'grandmoth', 'also', 'live',
'us', '.', 'she', 'came', 'itali', 'i', 'two', 'year', 'old', '.', 'she',
```

```
'grown', 'old', ',', 'still', 'strong', '.', 'she', 'cook', 'best', 'food', '!',  
'my', 'famili', 'import', '.', 'we', 'lot', 'thing', 'togeth', '.', 'my',  
'brother', 'i', 'like', 'go', 'long', 'walk', 'mountain', '.', 'my', 'sister',  
'like', 'cook', 'grandmoth', '.', 'on', 'weekend', 'play', 'board', 'game',  
'togeth', '.', 'we', 'laugh', 'alway', 'good', 'time', '.', 'i', 'love',  
'famili', 'much', '.']
```

```
[27]: nltk.pos_tag(tokenized_word)
```

```
[27]: [('My', 'PRP$'),  
      ('Wonderful', 'JJ'),  
      ('Family', 'NNP'),  
      ('.', '.'),  
      ('I', 'PRP'),  
      ('live', 'VBP'),  
      ('in', 'IN'),  
      ('a', 'DT'),  
      ('house', 'NN'),  
      ('near', 'IN'),  
      ('the', 'DT'),  
      ('mountains', 'NNS'),  
      ('.', '.'),  
      ('I', 'PRP'),  
      ('have', 'VBP'),  
      ('two', 'CD'),  
      ('brothers', 'NNS'),  
      ('and', 'CC'),  
      ('one', 'CD'),  
      ('sister', 'NN'),  
      (',', ','),  
      ('and', 'CC'),  
      ('I', 'PRP'),  
      ('was', 'VBD'),  
      ('born', 'VBN'),  
      ('last', 'JJ'),  
      ('.', '.'),  
      ('My', 'PRP$'),  
      ('father', 'NN'),  
      ('teaches', 'NNS'),  
      ('mathematics', 'NNS'),  
      (',', ','),  
      ('and', 'CC'),  
      ('my', 'PRP$'),  
      ('mother', 'NN'),  
      ('is', 'VBZ'),  
      ('a', 'DT'),  
      ('nurse', 'NN'),
```

('at', 'IN'),
('a', 'DT'),
('big', 'JJ'),
('hospital', 'NN'),
('.', '.'),
('My', 'PRP\$'),
('brothers', 'NNS'),
('are', 'VBP'),
('very', 'RB'),
('smart', 'JJ'),
('and', 'CC'),
('work', 'VB'),
('hard', 'RB'),
('in', 'IN'),
('school', 'NN'),
('.', '.'),
('My', 'PRP\$'),
('sister', 'NN'),
('is', 'VBZ'),
('a', 'DT'),
('nervous', 'JJ'),
('girl', 'NN'),
(',', ', ', ', '),
('but', 'CC'),
('she', 'PRP'),
('is', 'VBZ'),
('very', 'RB'),
('kind', 'NN'),
('.', '.'),
('My', 'PRP\$'),
('grandmother', 'NN'),
('also', 'RB'),
('lives', 'VBZ'),
('with', 'IN'),
('us', 'PRP'),
('.', '.'),
('She', 'PRP'),
('came', 'VBD'),
('from', 'IN'),
('Italy', 'NNP'),
('when', 'WRB'),
('I', 'PRP'),
('was', 'VBD'),
('two', 'CD'),
('years', 'NNS'),
('old', 'JJ'),
('.', '.'),

('She', 'PRP'),
('has', 'VBZ'),
('grown', 'VBN'),
('old', 'JJ'),
(',', ' '),
('but', 'CC'),
('she', 'PRP'),
('is', 'VBZ'),
('still', 'RB'),
('very', 'RB'),
('strong', 'JJ'),
('.', ' '),
('She', 'PRP'),
('cooks', 'VBZ'),
('the', 'DT'),
('best', 'JJS'),
('food', 'NN'),
('!', ' '),
('My', 'PRP\$'),
('family', 'NN'),
('is', 'VBZ'),
('very', 'RB'),
('important', 'JJ'),
('to', 'TO'),
('me', 'PRP'),
('.', ' '),
('We', 'PRP'),
('do', 'VBP'),
('lots', 'NNS'),
('of', 'IN'),
('things', 'NNS'),
('together', 'RB'),
('.', ' '),
('My', 'NNP'),
('brothers', 'NNPS'),
('and', 'CC'),
('I', 'PRP'),
('like', 'VBP'),
('to', 'TO'),
('go', 'VB'),
('on', 'IN'),
('long', 'JJ'),
('walks', 'NNS'),
('in', 'IN'),
('the', 'DT'),
('mountains', 'NNS'),
('.', ' '),

```

('My', 'PRP$'),
('sister', 'JJ'),
('likes', 'NNS'),
('to', 'TO'),
('cook', 'VB'),
('with', 'IN'),
('my', 'PRP$'),
('grandmother', 'NN'),
('.', '.'),
('On', 'IN'),
('the', 'DT'),
('weekends', 'NNS'),
('we', 'PRP'),
('all', 'DT'),
('play', 'VBP'),
('board', 'NN'),
('games', 'NNS'),
('together', 'RB'),
('.', '.'),
('We', 'PRP'),
('laugh', 'VBP'),
('and', 'CC'),
('always', 'RB'),
('have', 'VB'),
('a', 'DT'),
('good', 'JJ'),
('time', 'NN'),
('.', '.'),
('I', 'PRP'),
('love', 'VBP'),
('my', 'PRP$'),
('family', 'NN'),
('very', 'RB'),
('much', 'RB'),
('.', '.')]
```

```
[41]: from nltk.corpus import cmudict
prondict = nltk.corpus.cmudict.dict()
print(prondict['fire'])
print(prondict['language'])
```

```

[['F', 'AY1', 'ER0'], ['F', 'AY1', 'R']]
[['L', 'AE1', 'NG', 'G', 'W', 'AHO', 'JH'], ['L', 'AE1', 'NG', 'G', 'W', 'IHO', 'JH']]
```

```
[42]: entries = nltk.corpus.cmudict.entries()
for entry in entries[39943:39951]:
    print(entry)
```



```

('explorer', ['IH0', 'K', 'S', 'P', 'L', 'A01', 'R', 'ERO'])
('explorers', ['IH0', 'K', 'S', 'P', 'L', 'A01', 'R', 'ERO', 'Z'])
('explores', ['IH0', 'K', 'S', 'P', 'L', 'A01', 'R', 'Z'])
('exploring', ['IH0', 'K', 'S', 'P', 'L', 'A01', 'R', 'IH0', 'NG'])
('explosion', ['IH0', 'K', 'S', 'P', 'L', 'OW1', 'ZH', 'AHO', 'N'])
('explosions', ['IH0', 'K', 'S', 'P', 'L', 'OW1', 'ZH', 'AHO', 'N', 'Z'])
('explosive', ['IH0', 'K', 'S', 'P', 'L', 'OW1', 'S', 'IH0', 'V'])
('explosively', ['EH2', 'K', 'S', 'P', 'L', 'OW1', 'S', 'IH0', 'V', 'L', 'IYO'])

```

```

[43]: for word, pron in entries:
        if len(pron) == 3:
            ph1, ph2, ph3 = pron
            if ph1=='P' and ph3 == 'T':
                print( word , ph2)

```

```

pait EY1
pat AE1
pate EY1
patt AE1
peart ER1
peat IY1
peet IY1
peete IY1
pert ER1
pet EH1
pete IY1
pett EH1
piet IY1
piette IY1
pit IH1
pitt IH1
pot AA1
pote OW1
pott AA1
pout AW1
puett UW1
purt ER1
put UH1
putt AH1

```

```

[46]: syllable = ['IH0', 'K', 'S']
[word for word, pron in entries if pron[-3:] == syllable]

```

```

[46]: ['academics',
       'acoustics',
       'acrobatics',
       'acrylics',
       'addicks',

```

'aerobatics',
'aerobics',
'aerodynamics',
'aeronautics',
'aesthetics',
'affix',
'agnostics',
'agrigenetics',
'alcoholics',
'alix',
'alphagraphics',
'alphametrics',
'amyx',
'analgesics',
'analytics',
'anesthetics',
'angiomedics',
'anorexics',
'antibiotics',
'antibiotics',
'antics',
'apparatchiks',
'appendix',
'asics',
'asthmatics',
'astronautics',
'astrophysics',
'athletics',
'atlantic's",
'atlantic's",
'atmospharics',
'attics',
'audiotronics',
'automatics',
'automatix',
'avionics',
'ballistics',
'baltics',
'barrick's",
'barrick's",
'basic's",
'basics',
'basix',
'batiks',
'beatniks',
'beatrix',
'beatrix',

'bendix',
'benedix',
'biddix',
'bioethics',
'biologics',
'bionetics',
'biophysics',
'biotherapeutics',
'bishopricks',
"brunswick's",
"budick's",
"buick's",
'buicks',
"butterick's",
'cadnetix',
'calisthenics',
'calix',
'calyx',
"carpencic's",
'catholics',
'celtics',
'centronics',
'ceramics',
'cervix',
"chadwick's",
'chamonix',
'characteristics',
'charismatics',
'chemfix',
'chetniks',
'civics',
'classics',
'clerics',
"clinic's",
'clinics',
'clintonomics',
'cognetics',
'comics',
'conflicts',
'conics',
'conics',
'cosmetics',
'cowlicks',
'criminalistics',
"critic's",
'critics',
"critics",

'cryogenics',
"cubic's",
'cynics',
'cyrix',
'datagraphix',
'datametrics',
'datametrics',
'dederichs',
"deedrick's",
"democratic's",
'democratics',
'demographics',
'diabetics',
'diagnostics',
'dianetics',
'diasonics',
"dimick's",
'diuretics',
'domestics',
"dominic's",
'dramatics',
'dynamics',
"dynamics'",
"dynamics'",
'ebonics',
'eccentrics',
'eccentrics',
'econometrics',
'economics',
'economics',
"ehrlich's",
"elbrick's",
"electric's",
'electrics',
'electrodynamics',
'electromedics',
'electronics',
"electronics'",
"endotronics'",
'endotronics',
'enix',
'environics',
"epic's",
'epics',
'epidemics',
'epileptics',
'ergonomics',

"eric's",
"erik's",
'essex',
'esthetics',
'ethics',
'ethnics',
'eugenics',
'exotics',
'fabrics',
'fanatics',
'felix',
'ferrofluidics',
'fibronics',
'flextronics',
'forensics',
'forensics',
'fralix',
"frederick's",
'fredericks',
"frederick's",
'fredricks',
'frerichs',
'frericks',
'friedrichs',
'generics',
'genetics',
"geographic's",
'geometrics',
'geopolitics',
'geriatrics',
'gimmicks',
'graphics',
"graphics'",
'gunlicks',
'gymnastics',
'harmonics',
'heinrichs',
'helionetics',
'helix',
'hemodynamics',
'hendricks',
'hendriks',
'hendrix',
'hendryx',
'henrichs',
'henricks',
'heroics',

'heuristics',
'hieroglyphics',
'hinrichs',
'hispanics',
'histrionics',
'hydraulics',
'hypnotics',
'hysterics',
'identics',
'identics',
'immunomedics',
'informatics',
'informatics',
'informix',
'intermagnetics',
'intermedics',
"interpublic's",
'ionics',
'isaacs',
'isomedix',
"isomedix's",
'italics',
"karadzic's",
'kendricks',
'kerekes',
'kibbutzniks',
'kinetics',
'klezematics',
"komarik's",
'lasersonics',
"limerick's",
'limericks',
'linguistics',
'logistics',
'lumonics',
'lyrics',
'macroeconomics',
'macroeconomics',
"magic's",
'magnetics',
'mannix',
"mastrich's",
'mathematics',
'matrix',
"maverick's",
'mavericks',
"maverick's",

'mavericks',
'maverix',
"mccormick's",
'mechanics',
"mechanics'",
"medic's",
'medics',
'merricks',
'mesbics',
"metallic's",
'metaphysics',
'metics',
'metrics',
'microeconomics',
'microelectronics',
'micrographics',
'mimics',
'minix',
'minnix',
'mnemonics',
'mnemonics',
'molonicks',
"morovcic's",
'mosaics',
"mubarak's",
"mubarak's",
'mullenix',
'mullenix',
'mullinix',
'mulnix',
"munich's",
"music's",
'musics',
'mystics',
'narcotics',
'non-catholics',
'novamatrix',
'nucleonics',
'numismatics',
'obstetrics',
'odetics',
"olympic's",
'olympics',
'onomastics',
'onyx',
"opic's",
'optics',

'organics',
'orix',
'orthodontics',
'oryx',
"ozick's",
"pacific's",
"panic's",
'panics',
'paramedics',
"patrick's",
'patricks',
'pediatrics',
'penix',
'pennix',
'personics',
'pharmakinetics',
'phenix',
"philharmonic's",
'phoenix',
'phonetics',
'phonics',
'photovoltaics',
'photronics',
"physics",
'physics',
'pinnix',
'plantronics',
'plastics',
"plavsic's",
'pneumatics',
'polemics',
'polyconomics',
'prefix',
'prosthetics',
'psychics',
"public's",
'publics',
'publix',
'pyrotechnics',
'reaganomics',
'refuseniks',
'relics',
'remics',
'reprographics',
"republic's",
'republics',
"resnick's",

'respironics',
'robotics',
"roderick's",
'rogernomics',
'romantics',
"rubik's",
"scientific's",
'scientifics',
'sconnix',
'semantics',
'semiotics',
'shaddix',
"silajdzic's",
'siliconix',
'skeptics',
'skolniks',
'sonics',
'specifics',
'sputniks',
'statistics',
"statistics'",
'stoics',
'suffix',
'summagraphics',
'sussex',
'symbolics',
'synbiotics',
'synbiotics',
'synoptics',
'synthetics',
'systematics',
'systemix',
'systemix',
'tactics',
'technics',
'tectonics',
'tektronix',
'telectronics',
'telematics',
'telephonics',
'theatrics',
'therapeutics',
'thermedics',
'thermoplastics',
"thorndike's",
'thorndikes',
'tonics',

```
'topics',  
'toxics',  
"traffic's",  
'tropics',  
'unix',  
"vinick's",  
"vinnick's",  
'vitronics',  
"wallich's",  
"wolkanczik's",  
'workaholics',  
'xylogics',  
"zurich's"]
```

[]: