# -\*- coding: utf-8 -\*-

"""

Written by Evan Lalopoulos <evan.lalopoulos.2017@my.bristol.ac.uk>

University of Bristol, May 2018

Copyright (C) - All Rights Reserved

"""

from .liwc import Liwc

WORD\_CAT\_DICT\_1 = {

"love": [

"affect",

"posemo"

],

"loved": [

"affect",

"posemo"

]

}

WORD\_CAT\_DICT\_2 = {

"abandon\*": [

"affect",

"negemo"

],

"absolute": [

"cogmech"

],

"love": [

"affect",

"posemo"

]

}

def test\_build\_trie():

expected = {

"l": {

"o": {

"v": {

"e": {

"$": [

"affect",

"posemo"

],

"d": {

"$": [

"affect",

"posemo"

]

}

}

}

}

}

}

assert Liwc.\_build\_char\_trie(WORD\_CAT\_DICT\_1) == expected

def test\_build\_trie\_wildcard():

expected = {

"a": {

"b": {

"a": {

"n": {

"d": {

"o": {

"n": {

"\*": [

"affect",

"negemo"

],

"$": [

"affect",

"negemo"

]

}

}

}

}

},

"s": {

"o": {

"l": {

"u": {

"t": {

"e": {

"$": [

"cogmech"

]

}

}

}

}

}

}

}

},

"l": {

"o": {

"v": {

"e": {

"$": [

"affect",

"posemo"

]

}

}

}

}

}

assert Liwc.\_build\_char\_trie(WORD\_CAT\_DICT\_2) == expected

def test\_search\_trie():

trie = Liwc.\_build\_char\_trie(WORD\_CAT\_DICT\_1)

assert Liwc.\_search\_trie(trie, 'love') == ["affect", "posemo"]

assert Liwc.\_search\_trie(trie, 'loved') == ["affect", "posemo"]

def test\_search\_wildcard():

trie = Liwc.\_build\_char\_trie(WORD\_CAT\_DICT\_2)

assert Liwc.\_search\_trie(trie, 'abandon') == ["affect", "negemo"]

assert Liwc.\_search\_trie(trie, 'abandonment') == ["affect", "negemo"]