**Nuclear Data Section**

**International Atomic Energy Agency**

**P.O.Box 100, A-1400 Vienna, Austria**

**Memo CP-D/1107**

**Date:** 22 April 2024

**To:** Distribution

**From:** N. Otsuka

**Subject: Python spell checker for free text in EXFOR**

I wrote a simple spell checker for free text in EXFOR entries for review of preliminary tapes. This script checks each set of characters consisting of lower cases. Compilers can run it by (1) copy & paste the Python script x4\_spells.py (174 lines) from the appendix of this memo, and (2) create x4\_spells.dic (collection of words not in a standard English dictionaries). It also requires installation of the Python module “pyspellchecker”:

pip install pyspellchecker

***Example***: Spell check of PRELIM.D141:

X4\_SPELLS (Ver-2024-04-22) run on 22-04-2024

--------------------------------------------

Name of EXFOR file [exfor.txt] --> prelim.d141

 isobutane gas of about 2.5 mbar pressure. D642700100020

 ^^^^^^^^^ ^^^^

TITLE Examining the correlation between multi-neutron D644400100003

 ^^^^^

FACILITY (VDGT,3INDNSD) pelletron accelerator D644400100014

 ^^^^^^^^^

 of 5 mbar of isobutane gas, was placed at the focal D644400100017

 ^^^^ ^^^^^^^^^

…

 counting statics, deadtime (8%) D644800100024

 ^^^^^^^^

TITLE Systematic study of prescission neutron multiplicity: D645600100003

 ^^^^^^^^^^^

 revealing the role of entrance channel magicity D645600100004

 ^^^^^^^^

DETECTOR (HPGE) 100 cc detector, standard 152Eu reference D645800100016

 ^^

 - Statistical uncertainty in the photopeak counts D645900100019

 ^^^^^^^^^

Detected typos:

 cc x 2

 ck

 das

 deadtime

 hr

 isobutane x 2

 linac

 magicity

 mbar x 2

 multi

 pelletron x 2

 photopeak

 prescission

Spell checking completed. Good bye!

The dictionary file used for this checking is

atm

cm

deg

min

mg

mm

multiwire

nsec

sec

yr

**Appendix: x4\_spells.py**

#!/usr/bin/python3

ver="2024-04-22"

######################################################

# X4\_SPELLS Ver. 2024-04-22

# (Spellchecker for free text in EXFOR)

#

# Naohiko Otuka (IAEA Nuclear Data Section)

######################################################

# Install pyspellchecker by the following command:

# pip install pyspellchecker

######################################################

import datetime

import os

import re

import sys

from spellchecker import SpellChecker

spell = SpellChecker()

def main():

 global exfor, dict\_full

 time=datetime.datetime.now()

 date=time.strftime("%Y%m%d")

 time=time.strftime("%Y-%m-%d %H:%M:%S")

 (file\_exfor,file\_dict)=get\_input()

 known\_words=[]

 lines=get\_file\_lines(file\_dict)

 for line in lines:

 array=re.split("(,|\s)", line)

 for item in array:

 known\_words.append(item)

 lines=get\_file\_lines(file\_exfor)

 typos=read\_system\_identifier(lines,known\_words)

 num=dict()

 if (len(typos)>0):

 print("Detected typos:")

 typos.sort()

 typos\_unique=list(dict.fromkeys(typos))

 for word in typos\_unique:

 num[word]=typos.count(word)

 for typo in typos\_unique:

 if (num[typo]==1):

 print(" ",typo)

 else:

 print(" ",typo," x",num[typo])

 print()

 print("Spell checking completed. Good bye!")

def get\_input():

 time=datetime.datetime.now()

 date=time.strftime("%d-%m-%Y")

 print()

 print("X4\_SPELLS (Ver-"+ver+") run on "+date)

 print("--------------------------------------------")

 file\_exfor=""

 file\_dict="x4\_spells.dic"

 while not os.path.exists(file\_exfor):

 file\_exfor=input("Name of EXFOR file [exfor.txt] --> ")

 if file\_exfor=="":

 file\_exfor="exfor.txt"

 if not os.path.exists(file\_exfor):

 print(" \*\* File "+file\_exfor+" does not exist.")

 if not os.path.exists(file\_dict):

 print(" \*\* File "+file\_dict+" does not exist.")

 print()

 return file\_exfor, file\_dict

def print\_underline\_multi(col1s,col2s):

 char=""

 ioff=0

 for i, col1 in enumerate(col1s):

 length=col2s[i]-col1

 char+=" "\*(col1-ioff)+"^"\*length

 ioff=col1+length

 print(char)

def get\_file\_lines(file):

 if os.path.exists(file):

 f=open(file, "r")

 lines=f.readlines()

 f.close()

 else:

 msg="File "+file+" does not exist."

 print\_error\_fatal(msg)

 return lines

# Checking and extraction of system identifier lines

def read\_system\_identifier(lines,known\_words):

 sys\_id="ENDENTRY"

 all\_typos=[]

 for line in lines:

 if re.compile("^BIB ").search(line):

 sys\_id="BIB"

 nparen=0

 code=""

 elif re.compile("^ENDBIB ").search(line):

 sys\_id="ENDBIB"

 elif sys\_id=="BIB":

 content=line[11:66]

 (code,text,nparen)=code\_extraction(content,code,nparen)

 if nparen==0:

 code=""

 words=re.split("(,|-|:|\s|\(|\))", text)

 typos=[]

 for i, word in enumerate(words):

 if re.compile("^[a-z]+$").search(word):

 result=spell.unknown([word])

 if len(result)==1 and word not in known\_words:

 typos.append(word)

 all\_typos.append(word)

 col1s=[]

 col2s=[]

 char=line

 for typo in typos:

 col1=char.find(typo)

 col2=col1+len(typo)

 col1s.append(col1)

 col2s.append(col2)

 length=len(typo)

 char=char.replace(typo,"X"\*length,1)

 if len(typos)>0:

 line=line.rstrip("\n")

 print(line)

 print\_underline\_multi(col1s,col2s)

 return all\_typos

def code\_extraction(content,code,nparen):

 chars=list(content)

 nchar=0

 text=""

 for char in chars:

 if char=="(":

 nparen+=1

 elif char==")":

 nparen-=1

 if (nparen>0):

 nchar+=1

 if nparen==0:

 if nchar!=0:

 nchar+=1

 code+=content[0:nchar]

 text=content[nchar:]

 text=text.rstrip()

 return code, text, nparen

 else:

 text+=" "

 code+=content.rstrip()

 return code, "", nparen

if \_\_name\_\_ == "\_\_main\_\_":

 exfor=dict()

 main()

 exit()

**Distribution:**

a.koning@iaea.org

abhihere@gmail.com

aloks279@gmail.com

daniela.foligno@oecd-nea.org

dbrown@bnl.gov

draj@barc.gov.in

exfor@oecd-nea.org

fukahori.tokio@jaea.go.jp

ganesan555@gmail.com

gezg@ciae.ac.cn

iwamoto.osamu@jaea.go.jp

jmwang@ciae.ac.cn

kaltchen@ukr.net

kimdh@kaeri.re.kr

kimura.atsushi04@jaea.go.jp

l.vrapcenjak@iaea.org

manuel.bossant@oecd-nea.org

masaaki@nucl.sci.hokudai.ac.jp

marina-03-08@yandex.ru

michael.fleming@oecd-nea.org

mmarina@ippe.ru

nicolas.soppera@oecd-nea.org

n.otsuka@iaea.org

nrdc@jcprg.org

odsurenn@gmail.com

ogritzay@ukr.net

ogrudzevich@ippe.ru

otto.schwerer@aon.at

pikulina@expd.vniief.ru

pritychenko@bnl.gov

s.okumura@iaea.org

scyang@kaeri.re.kr

selyankina@expd.vniief.ru

sonzogni@bnl.gov

stakacs@atomki.mta.hu

stanislav.hlavac@savba.sk

sv.dunaeva@gmail.com

tada@nucl.sci.hokudai.ac.jp

taova@expd.vniief.ru

tarkanyi@atomki.hu

v.devi@iaea.org

v.zerkin@iaea.org

vidyathakur@yahoo.co.in

vsemkova@inrne.bas.bg

vvvarlamov@gmail.com

yolee@kaeri.re.kr

zholdybayev@inp.kz