For normalizing the short forms : created a dictionary by name short_forms
“U” at starting of sentence is converted to “You”
“u” in middle of sentence is converted to “you”
“r” is converted to “are”
“&amp” is converted to “and”

[1]To handle date strings like 1st, 2nd, 3rd, 4th : created a list by name date
Validated below rules on dates (1-31) using regular expression :

1. “st” should have only 1 as prior to it
2. “nd” can have only 2 as prior to it
3. “rd” can have only 3 as prior to it
4. “th” can have [4-9] as prior to it

Clitics handled :

won’t -> will not
shan’t -> shall not
an’t (n’t has “a” prior to it)(eg: c<u>an’t</u>) -> a not
Else n’t -> not
‘m -> am
‘d -> would
‘re -> are
‘ll -> will
‘ve -> have
‘s -> us if “let” is prior to ‘s
‘s -> is if “e/t” is prior to ‘s (eg: here’s, there’s, it’s)

Possessive apostrophe:
s’ -> “ ‘s” (eg: bts’ -> bts’s)

Regular expressions used for tokenization(separate tokens matching the pattern):

Punctuation : regex : [.!*-....etc] used string.punctuation for the list

Hashtag : regex : starts with #, followed by numbers or alphabets

User Handle : starts with @, followed by numbers or alphabets

Emoji : starting with X, <, O, B, |, =, ;, \ followed by middle : or - ending with D, P, 3, O, (,) etc
eg: <3, :), ;)

URLs : start with http/https, ://, combination of (., /, alphabest, -, digits, passkeys[0-9A-F])
Eclipse(...)
apostrophe('s)

Date format : [dd-mm-yyyy, yyyy\mm\dd] or 1st August, 15th November, etc mentioned above in [1]
Number : format - dd,dd,ddd eg : 90,00,000

Time : [hh:mm am/pm, h:mm am/pm, h am/pm, hh am/pm]
Eg: [24:50 pm, 7:00 pm, 7 pm, 8am etc]

Contact number : dddd ddd ddd or ddd ddd dddd(eg 1800 324 3248 etc)

Floating point number : dd.dd (eg: 3.25, 5555.90 etc)

Separating words attached by eclipse(".....") : then... Odd -> then, …, Odd
Separating hyphen words : Covid-19 -> Covid, -, 19

Other words : any pattern with digits or alphabets
Normalizing:
Converting Date to Canonical form : function by name - date_to_cfd

    Validates month and date
    Conversion to date whenever month name is detected in the sentence
    If any one of the year, month, date missing puts '?' at the place
    Eg : 15th August - CF:D:????-08-15

Converting time to Canonical form : function by name - time_to_cft

    Validates minutes
    Conversion to time to IST according to am/pm
    Eg : 7pm -> CF:T:1900:IST

Run using ./111708049_LabAssign1_code.py
Checking it by using "diff 111708049_Assign1_GoldStandard.txt output.txt"
Submitted files:

    111708049_Assgn1Dataset-1.txt
    111708049_LabAssign1_code.py
    111708049_Assign1_Writeup.pdf
    111708049_Assign1_GoldStandard.txt
    111708049_Assign1Output.txt