

Annex 2. Unit tests

```
"""
```

```
Author: Daria Alexander
```

```
We test measures' calculations of 10 random messages taken from the data  
with Python unit testing framework.
```

```
"""
```

```
import unittest
import turnFactory

#Use methods from the class TurnFactory that computes all the measures
matches = turnFactory.turnFactory()
matches.initialiseFactory('logs_events.json')
matches.getTurns()
matches.identifyTurns()
messagelist = matches.messagelist #dictionary

#Prepare answers as dictionary of dictionaries
answers = {} # the answers for 10 random messages
message = {} # each message

#12269
message = {'tot_char_user_mess_duration': 2.326, 'tot_char_us_turn_duration': 1.819,
'final_char_us_mess_duration': 1.984, 'final_char_us_turn_dur': 1.552,
'nb_words_us_mess_duration': 0.41, 'nb_words_us_turn_duration': 0.321, 'mean_pause_duration':
0.521, 'percentage_pause': 67, 'revisions_us_mess_duration': 0.274,
'revisions_us_turn_duration': 0.214, 'revisions_per_words': 0.667, 'revision_proportion': 0.853,
'mean_duration_p_bursts': 0.248, 'mean_duration_r_bursts': 2.14, 'mean_length_of_p_bursts':
1.789, 'mean_length_of_r_bursts': 5.667}
answers[12269] = message

#41108
message = {'tot_char_user_mess_duration': 1.958, 'tot_char_us_turn_duration': 1.326,
'final_char_us_mess_duration': 1.958, 'final_char_us_turn_dur': 1.326,
'nb_words_us_mess_duration': 0.318, 'nb_words_us_turn_duration': 0.215, 'mean_pause_duration':
0.761, 'percentage_pause': 76, 'revisions_us_mess_duration': 0.0, 'revisions_us_turn_duration':
0.0, 'revisions_per_words': 0.0, 'revision_proportion': 1.0, 'mean_duration_p_bursts': 0.234,
'mean_duration_r_bursts': 18.895, 'mean_length_of_p_bursts': 1.947, 'mean_length_of_r_bursts':
37.0}
answers[41108] = message

#42398
message = {'tot_char_user_mess_duration': 1.96, 'tot_char_us_turn_duration': 1.21,
'final_char_us_mess_duration': 1.91, 'final_char_us_turn_dur': 1.179,
'nb_words_us_mess_duration': 0.302, 'nb_words_us_turn_duration': 0.186, 'mean_pause_duration':
0.853, 'percentage_pause': 81, 'revisions_us_mess_duration': 0.05, 'revisions_us_turn_duration':
0.031, 'revisions_per_words': 0.167, 'revision_proportion': 0.974, 'mean_duration_p_bursts':
0.195, 'mean_duration_r_bursts': 9.681, 'mean_length_of_p_bursts': 2.053,
'mean_length_of_r_bursts': 19.5}
answers[42398] = message

#57405
message = {'tot_char_user_mess_duration': 0.952, 'tot_char_us_turn_duration': 0.684,
'final_char_us_mess_duration': 0.688, 'final_char_us_turn_dur': 0.494,
'nb_words_us_mess_duration': 0.146, 'nb_words_us_turn_duration': 0.105, 'mean_pause_duration':
1.264, 'percentage_pause': 90, 'revisions_us_mess_duration': 0.205,
'revisions_us_turn_duration': 0.147, 'revisions_per_words': 1.4, 'revision_proportion': 0.723,
'mean_duration_p_bursts': 0.13, 'mean_duration_r_bursts': 3.26, 'mean_length_of_p_bursts':
1.327, 'mean_length_of_r_bursts': 3.421}
answers[57405] = message

#56495
message = {'tot_char_user_mess_duration': 1.792, 'tot_char_us_turn_duration': 1.509,
'final_char_us_mess_duration': 1.792, 'final_char_us_turn_dur': 1.509,
'nb_words_us_mess_duration': 0.368, 'nb_words_us_turn_duration': 0.31, 'mean_pause_duration':
1.035, 'percentage_pause': 80, 'revisions_us_mess_duration': 0.0, 'revisions_us_turn_duration':
0.0, 'revisions_per_words': 0.0, 'revision_proportion': 1.0, 'mean_duration_p_bursts': 0.245,
'mean_duration_r_bursts': 21.76, 'mean_length_of_p_bursts': 2.294, 'mean_length_of_r_bursts':
```

```

39.0}
answers[56495] = message

#11712
message = {'tot_char_user_mess_duration': 1.238, 'tot_char_us_turn_duration': 0.137,
'final_char_us_mess_duration': 1.032, 'final_char_us_turn_dur': 0.115,
'nb_words_us_mess_duration': 0.413, 'nb_words_us_turn_duration': 0.046, 'mean_pause_duration':
0.647, 'percentage_pause': 80, 'revisions_us_mess_duration': 0.206,
'revisions_us_turn_duration': 0.023, 'revisions_per_words': 0.5, 'revision_proportion': 0.833,
'mean_duration_p_bursts': 0.161, 'mean_duration_r_bursts': 2.171, 'mean_length_of_p_bursts':
1.0, 'mean_length_of_r_bursts': 3.0}
answers[11712] = message

#34423
message = {'tot_char_user_mess_duration': 3.01, 'tot_char_us_turn_duration': 1.154,
'final_char_us_mess_duration': 3.01, 'final_char_us_turn_dur': 1.154,
'nb_words_us_mess_duration': 0.78, 'nb_words_us_turn_duration': 0.299, 'mean_pause_duration':
0.57, 'percentage_pause': 63, 'revisions_us_mess_duration': 0.0, 'revisions_us_turn_duration':
0.0, 'revisions_per_words': 0.0, 'revision_proportion': 1.0, 'mean_duration_p_bursts': 0.327,
'mean_duration_r_bursts': 8.969, 'mean_length_of_p_bursts': 2.7, 'mean_length_of_r_bursts':
27.0}
answers[34423] = message

#33861
message = {'tot_char_user_mess_duration': 0.681, 'tot_char_us_turn_duration': 0.625,
'final_char_us_mess_duration': 0.582, 'final_char_us_turn_dur': 0.533,
'nb_words_us_mess_duration': 0.061, 'nb_words_us_turn_duration': 0.056, 'mean_pause_duration':
2.009, 'percentage_pause': 76, 'revisions_us_mess_duration': 0.077,
'revisions_us_turn_duration': 0.07, 'revisions_per_words': 1.25, 'revision_proportion': 0.854,
'mean_duration_p_bursts': 0.604, 'mean_duration_r_bursts': 8.964, 'mean_length_of_p_bursts':
1.78, 'mean_length_of_r_bursts': 6.357}
answers[33861] = message

#56869
message = {'tot_char_user_mess_duration': 2.19, 'tot_char_us_turn_duration': 1.073,
'final_char_us_mess_duration': 1.894, 'final_char_us_turn_dur': 0.928,
'nb_words_us_mess_duration': 0.414, 'nb_words_us_turn_duration': 0.203, 'mean_pause_duration':
0.671, 'percentage_pause': 71, 'revisions_us_mess_duration': 0.237,
'revisions_us_turn_duration': 0.116, 'revisions_per_words': 0.571, 'revision_proportion': 0.865,
'mean_duration_p_bursts': 0.268, 'mean_duration_r_bursts': 2.609, 'mean_length_of_p_bursts':
2.056, 'mean_length_of_r_bursts': 6.167}
answers[56869] = message

#74213
message = {'tot_char_user_mess_duration': 3.293, 'tot_char_us_turn_duration': 1.773,
'final_char_us_mess_duration': 3.12, 'final_char_us_turn_dur': 1.68,
'nb_words_us_mess_duration': 0.52, 'nb_words_us_turn_duration': 0.28, 'mean_pause_duration':
0.436, 'percentage_pause': 60, 'revisions_us_mess_duration': 0.173,
'revisions_us_turn_duration': 0.093, 'revisions_per_words': 0.333, 'revision_proportion': 0.947,
'mean_duration_p_bursts': 0.285, 'mean_duration_r_bursts': 2.704, 'mean_length_of_p_bursts':
2.375, 'mean_length_of_r_bursts': 9.5}
answers[74213] = message

def compare_values(key):
    """
    :param key:
    :return: logical result of compare
    """
    for number, message in answers.items(): # our 10 messages
        if message[key] != messagelist[number].result[key]: # if the value of the message key is
not equal to the value
            # of each key of the dictionary result.
            return False
    return True

# we test all the methods that compute fluency measures
class Test_TurnFactory(unittest.TestCase):

    def test_tot_char_user_mess_dur(self):
        key = 'tot_char_user_mess_duration'
        self.assertTrue(compare_values(key), key)

```

```

def test_tot_char_user_turn_dur(self):
    key = 'tot_char_us_turn_duration'
    self.assertTrue(compare_values(key), key)

def test_fin_char_us_mess_dur(self):
    key = 'final_char_us_mess_duration'
    self.assertTrue(compare_values(key), key)

def test_fin_char_us_turn_dur(self):
    key = 'final_char_us_turn_dur'
    self.assertTrue(compare_values(key), key)

def test_words_mess_dur(self):
    key = 'nb_words_us_mess_duration'
    self.assertTrue(compare_values(key), key)

def test_words_turn_dur(self):
    key = 'nb_words_us_turn_duration'
    self.assertTrue(compare_values(key), key)

def test_mean_pause_duration(self):
    key = 'mean_pause_duration'
    self.assertTrue(compare_values(key), key)

def test_percentage_pause_time(self):
    key = 'percentage_pause'
    self.assertTrue(compare_values(key), key)

def test_revision_episodes_mess_dur(self):
    key = 'revisions_us_mess_duration'
    self.assertTrue(compare_values(key), key)

def test_revision_episodes_turn_dur(self):
    key = 'revisions_us_turn_duration'
    self.assertTrue(compare_values(key), key)

def test_revision_episodes_per_words(self):
    key = 'revisions_per_words'
    self.assertTrue(compare_values(key), key)

def test_revision_proportion(self):
    key = 'revision_proportion'
    self.assertTrue(compare_values(key), key)

def test_mean_duration_p_bursts(self):
    key = 'mean_duration_p_bursts'
    self.assertTrue(compare_values(key), key)

def test_mean_duration_r_bursts(self):
    key = 'mean_duration_r_bursts'
    self.assertTrue(compare_values(key), key)

def test_mean_length_p_bursts(self):
    key = 'mean_length_of_p_bursts'
    self.assertTrue(compare_values(key), key)

def test_mean_length_r_bursts(self):
    key = 'mean_length_of_r_bursts'
    self.assertTrue(compare_values(key), key)

if __name__ == '__main__':
    unittest.main()

```