Text Preprocessing Pipeline and Proofreading Results COMP 6751 Project 1 Demo File

Haochen Zou (40158179)

0. Expectations of originality

I certify that this submission is my original work and meets the Faculty's Expectations of Originality.
Name: Haochen Zou; I.D: 40158179; Date: 2021.10.1

1. Test Scenarios

1.1. Test Scenario 1

Given: User enters the Reuters corpus text file name **training/267**.
When: The file name is correctly input.
Result: Program can run successfully, and the text preprocess pipeline and proofreading results given.
Conclusion: The program can preprocess and proofread the text file training/267. Tokenized, sentences split, pos tag, numbers normalized, and date recognized and parsed as **Figures 1**, **2** and **3** shown below.
Appendix: Content of text file **training/267**:
INDONESIA UNLIKELY TO IMPORT PHILIPPINES COPRA
Indonesia is unlikely to import copra from the Philippines in 1987 after importing 30,000 tonnes in 1986, the U.S. Embassy's annual agriculture report said.
The report said the 31 pct devaluation of the Indonesian rupiah, an increase in import duties on copra and increases in the price of Philippines copra have reduced the margin between prices in the two countries.
Indonesia's copra production is forecast at 1.32 mln tonnes in calendar 1987, up 1.30 mln tonnes in 1986.

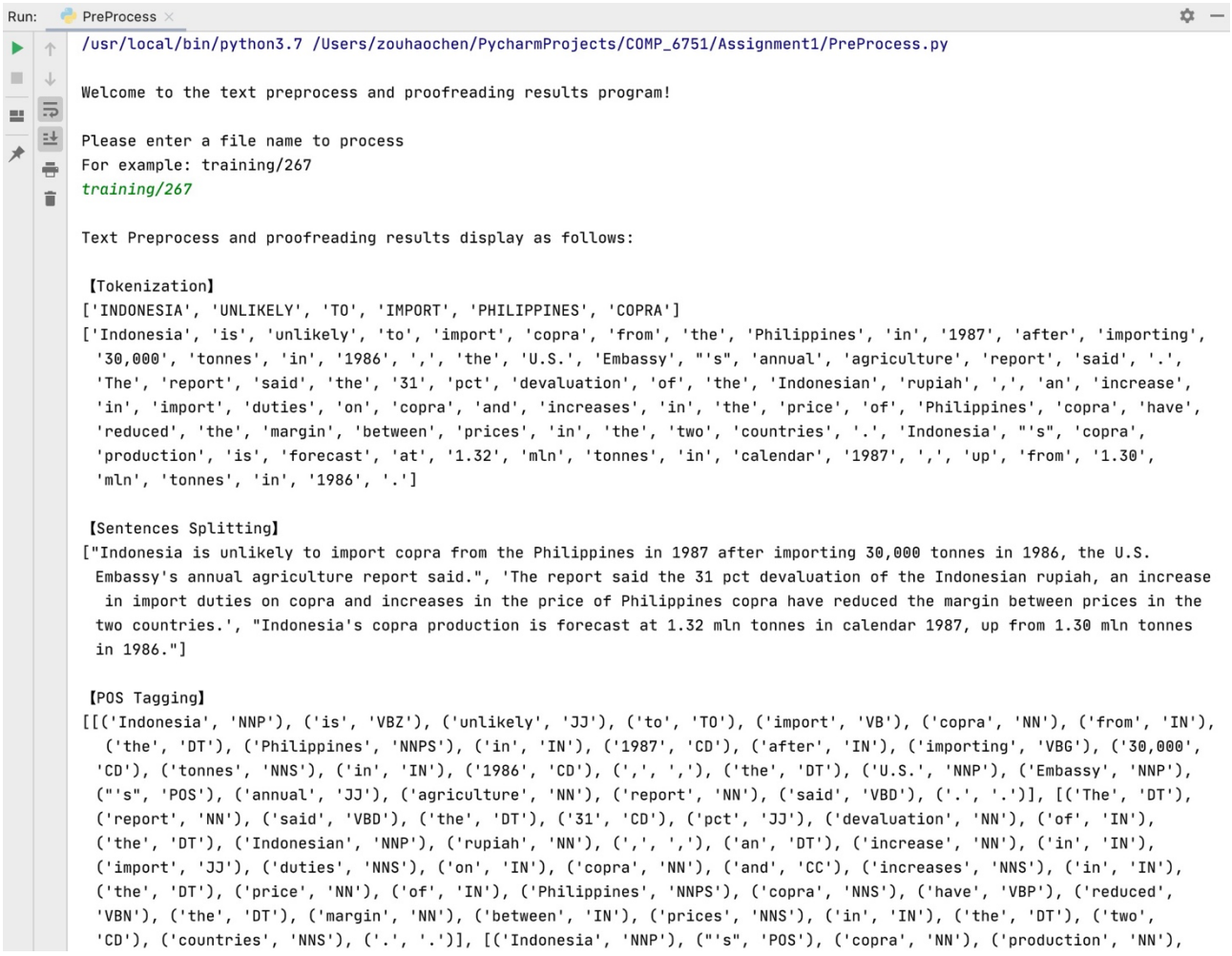

Figure 1. Test scenario 1 result: input file name training/267.

```
Run:  PreProcess
 '30,000', 'tonnes', 'in', '1986', ',', 'the', 'U.S.', 'Embassy', "'s", 'annual', 'agriculture', 'report', 'said', '.',
 'The', 'report', 'said', 'the', '31', 'pct', 'devaluation', 'of', 'the', 'Indonesian', 'rupiah', ',', 'an', 'increase',
 'in', 'import', 'duties', 'on', 'copra', 'and', 'increases', 'in', 'the', 'price', 'of', 'Philippines', 'copra', 'have',
 'reduced', 'the', 'margin', 'between', 'prices', 'in', 'the', 'two', 'countries', '.', 'Indonesia', "'s", 'copra',
 'production', 'is', 'forecast', 'at', '1.32', 'mln', 'tonnes', 'in', 'calendar', '1987', ',', 'up', 'from', '1.30',
 'mln', 'tonnes', 'in', '1986', '.']

【Sentences Splitting】
["Indonesia is unlikely to import copra from the Philippines in 1987 after importing 30,000 tonnes in 1986, the U.S.
 Embassy's annual agriculture report said.", 'The report said the 31 pct devaluation of the Indonesian rupiah, an increase
  in import duties on copra and increases in the price of Philippines copra have reduced the margin between prices in the
 two countries.', "Indonesia's copra production is forecast at 1.32 mln tonnes in calendar 1987, up from 1.30 mln tonnes
 in 1986."]

【POS Tagging】
[[('Indonesia', 'NNP'), ('is', 'VBZ'), ('unlikely', 'JJ'), ('to', 'TO'), ('import', 'VB'), ('copra', 'NN'), ('from', 'IN'),
  ('the', 'DT'), ('Philippines', 'NNPS'), ('in', 'IN'), ('1987', 'CD'), ('after', 'IN'), ('importing', 'VBG'), ('30,000',
 'CD'), ('tonnes', 'NNS'), ('in', 'IN'), ('1986', 'CD'), (',', ','), ('the', 'DT'), ('U.S.', 'NNP'), ('Embassy', 'NNP'),
 ("'s", 'POS'), ('annual', 'JJ'), ('agriculture', 'NN'), ('report', 'NN'), ('said', 'VBD'), ('.', '.')], [('The', 'DT'),
 ('report', 'NN'), ('said', 'VBD'), ('the', 'DT'), ('31', 'CD'), ('pct', 'JJ'), ('devaluation', 'NN'), ('of', 'IN'),
 ('the', 'DT'), ('Indonesian', 'NNP'), ('rupiah', 'NN'), (',', ','), ('an', 'DT'), ('increase', 'NN'), ('in', 'IN'),
 ('import', 'JJ'), ('duties', 'NNS'), ('on', 'IN'), ('copra', 'NN'), ('and', 'CC'), ('increases', 'NNS'), ('in', 'IN'),
 ('the', 'DT'), ('price', 'NN'), ('of', 'IN'), ('Philippines', 'NNPS'), ('copra', 'NNS'), ('have', 'VBP'), ('reduced',
 'VBN'), ('the', 'DT'), ('margin', 'NN'), ('between', 'IN'), ('prices', 'NNS'), ('in', 'IN'), ('the', 'DT'), ('two',
 'CD'), ('countries', 'NNS'), ('.', '.')], [('Indonesia', 'NNP'), ("'s", 'POS'), ('copra', 'NN'), ('production', 'NN'),
 ('is', 'VBZ'), ('forecast', 'VBN'), ('at', 'IN'), ('1.32', 'CD'), ('mln', 'NN'), ('tonnes', 'NNS'), ('in', 'IN'),
 ('calendar', 'NN'), ('1987', 'CD'), (',', ','), ('up', 'RB'), ('from', 'IN'), ('1.30', 'CD'), ('mln', 'NN'), ('tonnes',
 'NNS'), ('in', 'IN'), ('1986', 'CD'), ('.', '.')]]

【Number Normalization】
['1987', '30,000', '1986', '31', 'two', '1.32', '1987', '1.30', '1986']
['1987', '30,000', '1986', '31', '2', '1.32', '1987', '1.30', '1986']
['one thousand, nine hundred and eighty-seven', 'thirty thousand', 'one thousand, nine hundred and eighty-six',
 'thirty-one', 'two', 'one point three two', 'one thousand, nine hundred and eighty-seven', 'one point three', 'one
 thousand, nine hundred and eighty-six']

【Date recognition】
{'in 1986', 'in 1987'}

【Date parsing】
(DATE (IN in) (YEAR (DIGIT 1) (DIGIT 9) (DIGIT 8) (DIGIT 6)))
(DATE (IN in) (YEAR (DIGIT 1) (DIGIT 9) (DIGIT 8) (DIGIT 7)))

Process finished with exit code 0
```

Figure 2. Test scenario 1 result: input file name training/267.

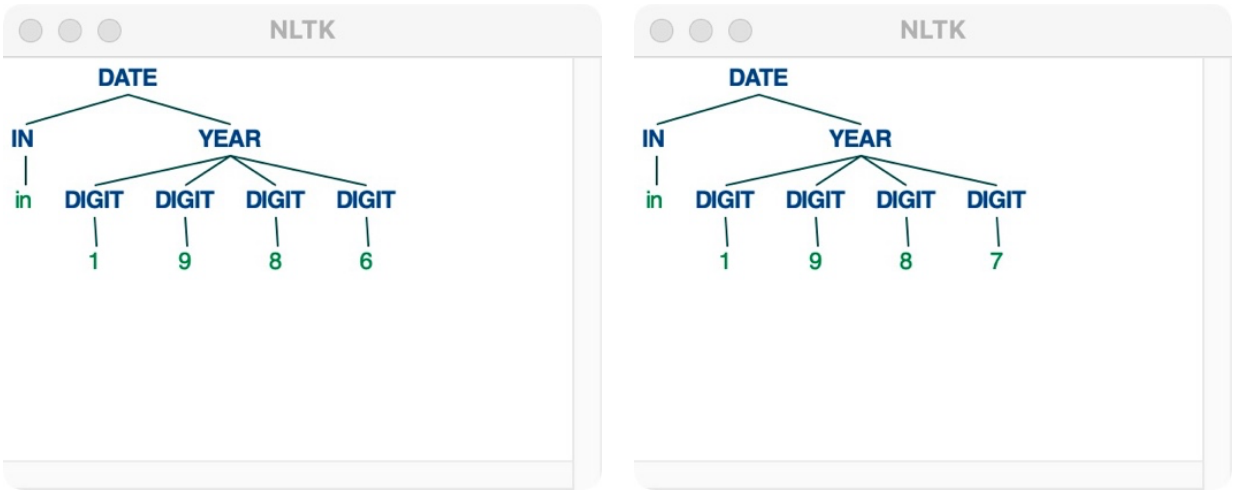

Figure 3. Parsing tree generated by the program.

1.2. Test Scenario 2

Given: User enters a text file name which does not in the Reuters corpus.
When: The file name is incorrectly input.
Result: Program will print error information to the user as **Figure 4** shown.
Conclusion: The program cannot process the file name which not in the Reuters corpus.

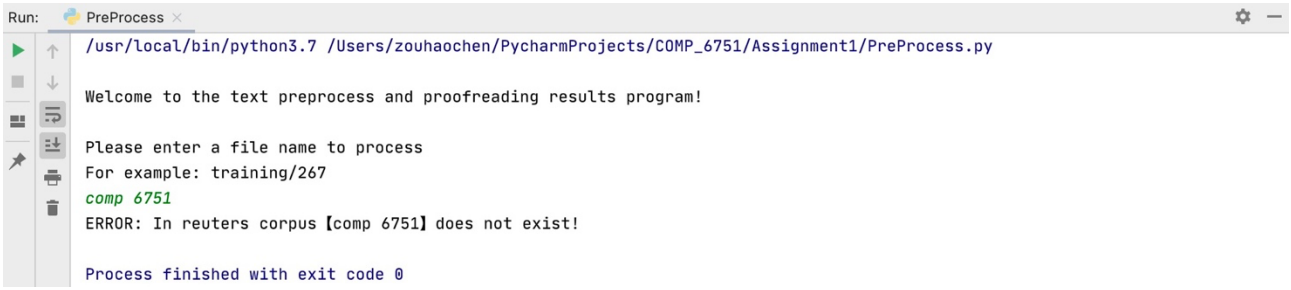

Figure 4. User enters a text file name which does not in the Reuters corpus.

1.3. Test Scenario 3

Given: User enters the Reuters corpus text file name **test/14832**.

When: The file name is correctly input.

Result: Program can run successfully, and the text preprocess pipeline and proofreading results given.

Conclusion: The program can preprocess and proofread the text file test/14832. Tokenized, sentences split, pos tag, numbers normalized, and date recognized and parsed as **Figures 5**, **6**, **7**, and **8** shown below.

Appendix: Content of text file **test/14832**:

THAI TRADE DEFICIT WIDENS IN FIRST QUARTER

Thailand's trade deficit widened to 4.5 billion baht in the first quarter of 1987 from 2.1 billion a year ago, the Business Economics Department said.

It said Janunary/March imports rose to 65.1 billion baht from 58.7 billion. Thailand's improved business climate this year resulted in a 27 pct increase in imports of raw materials and semi-finished products.

The country's oil import bill, however, fell 23 pct in the first quarter due to lower oil prices.

The department said first quarter exports expanded to 60.6 billion baht from 56.6 billion.

Export growth was smaller than expected due to lower earnings from many key commodities including rice whose earnings declined 18 pct, maize 66 pct, sugar 45 pct, tin 26 pct and canned pineapples seven pct.

Products registering high export growth were jewellery up 64 pct, clothing 57 pct and rubber 35 pct.

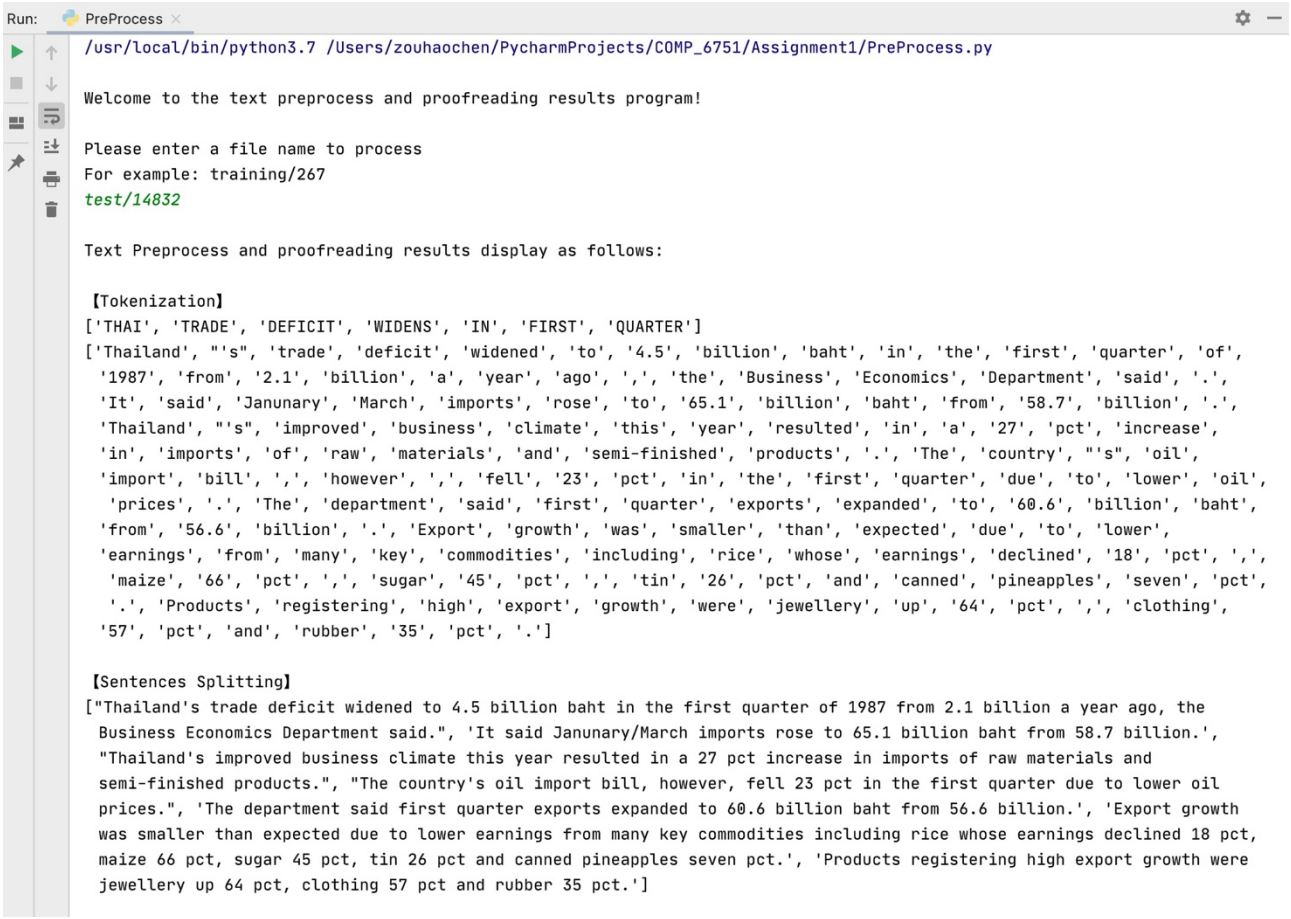

Figure 5. Test scenario 3 result: input file name test/14832.

Figure 6. Test scenario 3 result: input file name test/14832.

Figure 7. Test scenario 3 result: input file name test/14832.

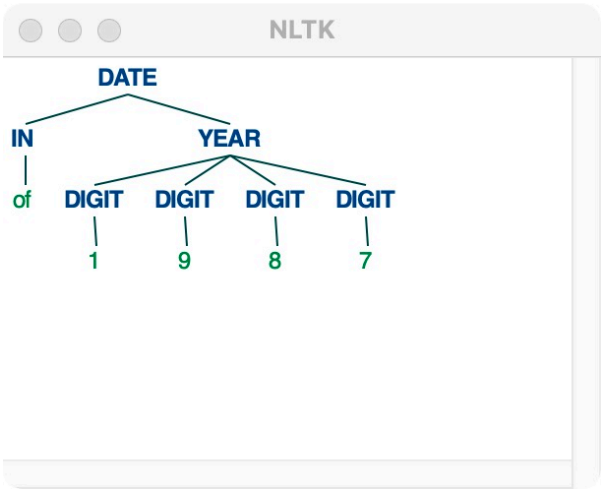

Figure 8. Parsing tree generated by the program.

1.4. Test Scenario 4

Given: User enters the Reuters corpus text file name **training/9880**.

When: The file name is correctly input.

Result: Program can run successfully, and the text preprocess pipeline and proofreading results given.

Conclusion: The program can preprocess and proofread the text file training/9880. Tokenized, sentences split, pos tag, numbers normalized, and date recognized and parsed as **Figures 9**, and **10** shown below.

Appendix: Content of text file **training/9880**:

U.K. MONEY MARKET GETS 25 MLN STG LATE HELP

The Bank of England said it provided about 25 mln stg in late help to the money market, bringing the total assistance today to 266 mln stg.

This compares with the bank's revised estimate of a 350 mln stg money market shortfall.

```
Run:  PreProcess
/usr/local/bin/python3.7 /Users/zouhaochen/PycharmProjects/COMP_6751/Assignment1/PreProcess.py

Welcome to the text preprocess and proofreading results program!

Please enter a file name to process
For example: training/267
training/9880

Text Preprocess and proofreading results display as follows:

【Tokenization】
['U.K.', 'MONEY', 'MARKET', 'GETS', '25', 'MLN', 'STG', 'LATE', 'HELP']
['The', 'Bank', 'of', 'England', 'said', 'it', 'provided', 'about', '25', 'mln', 'stg', 'in', 'late', 'help', 'to', 'the',
 'money', 'market', ',', 'bringing', 'the', 'total', 'assistance', 'today', 'to', '266', 'mln', 'stg', '.', 'This',
 'compares', 'with', 'the', 'bank', "'s", 'revised', 'estimate', 'of', 'a', '350', 'mln', 'stg', 'money', 'market',
 'shortfall', '.']

【Sentences Splitting】
['The Bank of England said it provided about 25 mln stg in late help to the money market, bringing the total assistance
 today to 266 mln stg.', "This compares with the bank's revised estimate of a 350 mln stg money market shortfall."]

【POS Tagging】
[[('The', 'DT'), ('Bank', 'NNP'), ('of', 'IN'), ('England', 'NNP'), ('said', 'VBD'), ('it', 'PRP'), ('provided', 'VBD'),
 ('about', 'RB'), ('25', 'CD'), ('mln', 'NNS'), ('stg', 'RB'), ('in', 'IN'), ('late', 'JJ'), ('help', 'NN'), ('to', 'TO'),
  ('the', 'DT'), ('money', 'NN'), ('market', 'NN'), (',', ','), ('bringing', 'VBG'), ('the', 'DT'), ('total', 'JJ'),
 ('assistance', 'NN'), ('today', 'NN'), ('to', 'TO'), ('266', 'CD'), ('mln', 'NN'), ('stg', 'NN'), ('.', '.')], [('This',
 'DT'), ('compares', 'VBZ'), ('with', 'IN'), ('the', 'DT'), ('bank', 'NN'), ("'s", 'POS'), ('revised', 'JJ'), ('estimate',
  'NN'), ('of', 'IN'), ('a', 'DT'), ('350', 'CD'), ('mln', 'NN'), ('stg', 'JJ'), ('money', 'NN'), ('market', 'NN'),
 ('shortfall', 'NN'), ('.', '.')]]
```

Figure 9. Test scenario 4 result: input file name training/9880.

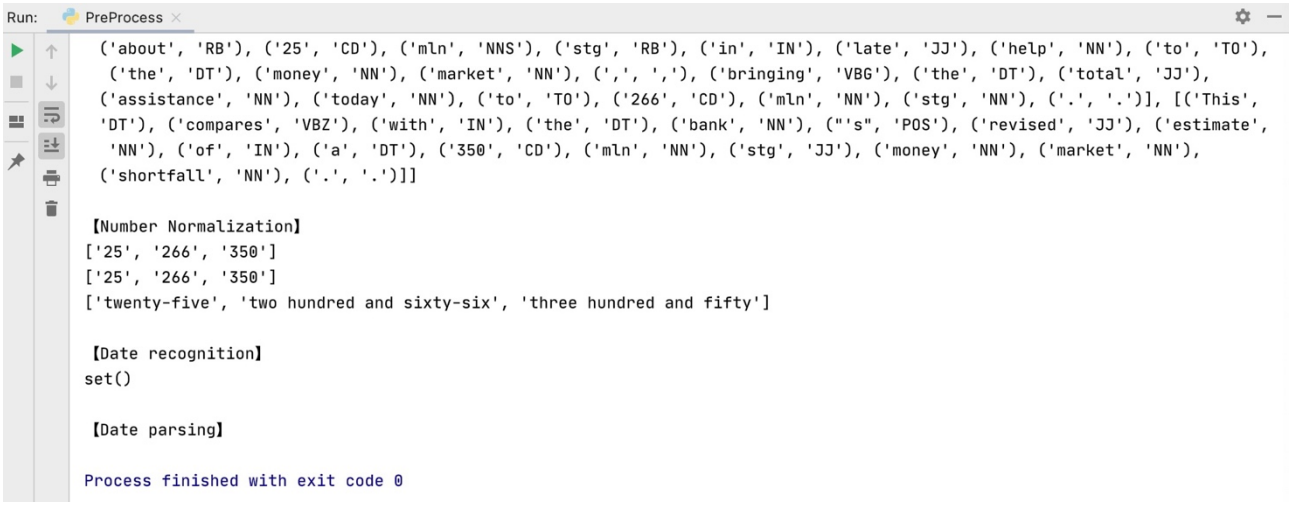

Figure 10. Test scenario 4 result: input file name training/9880.

1.5. Test Scenario 5

Given: User enters the Reuters corpus text file name **test/14828**.

When: The file name is correctly input.

Result: Program can run successfully, and the text preprocess pipeline and proofreading results given.

Conclusion: The program can preprocess and proofread the text file test/14828. Tokenized, sentences split, pos tag, numbers normalized, and date recognized and parsed as **Figures 11**, and **12** shown below.

Appendix: Content of text file **test/14828**:

CHINA DAILY SAYS VERMIN EAT 7-12 PCT GRAIN STOCKS

A survey of 19 provinces and seven cities showed vermin consume between seven and 12 pct of China's grain stocks, the China Daily said.

It also said that each year 1.575 mln tonnes, or 25 pct, of China's fruit output are left to rot, and 2.1 mln tonnes, or up to 30 pct, of its vegetables. The paper blamed the waste on inadequate storage and bad preservation methods.

It said the government had launched a national programme to reduce waste, calling for improved technology in storage and preservation, and greater production of additives. The paper gave no further details.

Figure 11. Test scenario 5 result: input file name test/14828.

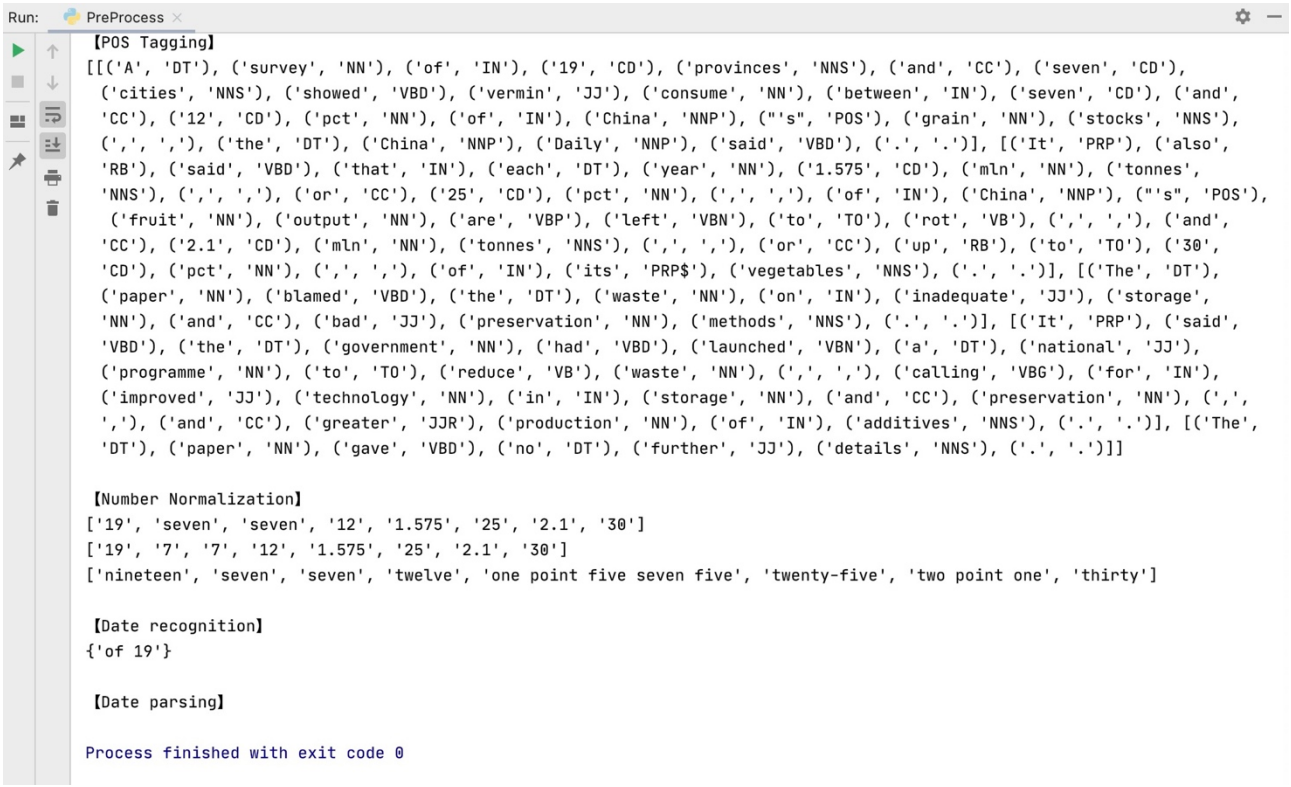

Figure 12. Test scenario 5 result: input file name test/14828.

2. Errors and Limitations

2.1. Words2num package cannot convert 'billion'

As results shown in test scenario 3, the package words2num cannot transfer billion to 1,000,000,000. The same as some other words such as trillion, million etc. Also, the accurate number should be 2,100,000,000 instead of 2.1, billion. However, the program just split these two numbers with attribute <CD> and convert them separately which indicate both package words2num and number normalization should be improved.

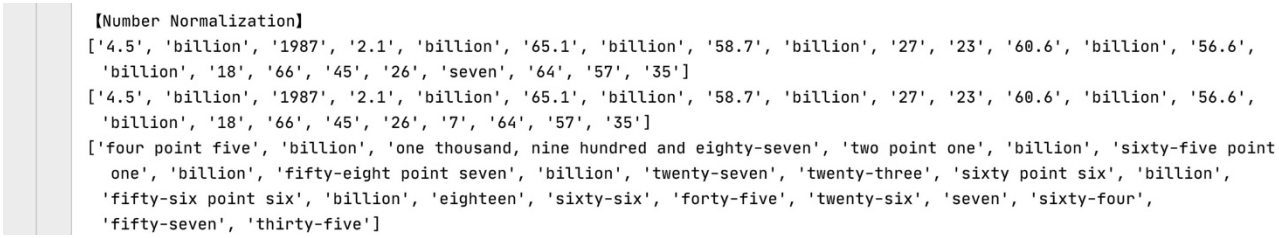

Figure 13. 'Billion' in test scenario 3 has not convert to number.

2.2. Number normalization display part does not consolidate duplicate data

As results shown in test scenario 1, 3, and 5. The program display all the numbers after finishing the number normalization part. However, it does not consolidate duplicate number data.

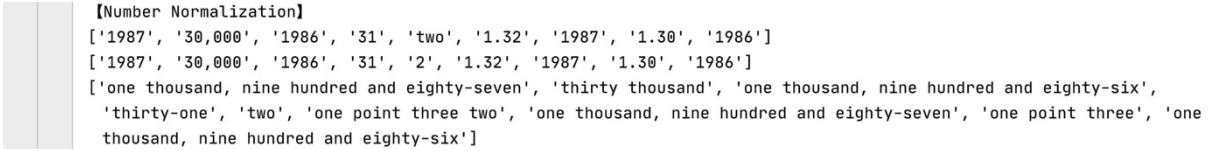

Figure 14. Duplicate number data '1987' and '1986' displayed in test scenario 1.

```
【Number Normalization】
['4.5', 'billion', '1987', '2.1', 'billion', '65.1', 'billion', '58.7', 'billion', '27', '23', '60.6', 'billion', '56.6',
 'billion', '18', '66', '45', '26', 'seven', '64', '57', '35']
['4.5', 'billion', '1987', '2.1', 'billion', '65.1', 'billion', '58.7', 'billion', '27', '23', '60.6', 'billion', '56.6',
 'billion', '18', '66', '45', '26', '7', '64', '57', '35']
['four point five', 'billion', 'one thousand, nine hundred and eighty-seven', 'two point one', 'billion', 'sixty-five point
  one', 'billion', 'fifty-eight point seven', 'billion', 'twenty-seven', 'twenty-three', 'sixty point six', 'billion',
 'fifty-six point six', 'billion', 'eighteen', 'sixty-six', 'forty-five', 'twenty-six', 'seven', 'sixty-four',
 'fifty-seven', 'thirty-five']
```

Figure 15. Duplicate number data 'billion' displayed in test scenario 3.

```
【Number Normalization】
['19', 'seven', 'seven', '12', '1.575', '25', '2.1', '30']
['19', '7', '7', '12', '1.575', '25', '2.1', '30']
['nineteen', 'seven', 'seven', 'twelve', 'one point five seven five', 'twenty-five', 'two point one', 'thirty']
```

Figure 16. Duplicate number data '7' displayed in test scenario 3.

2.3. Date recognition print empty set when there is no date information in text file

As results shown in test scenario 4. The program displays empty set when there is no date information in text file instead of informing the user that 'There is no date in the text file'.

```
【Date recognition】
set()

【Date parsing】

Process finished with exit code 0
```

Figure 17. Date recognitions print empty set when there is no date information in text file in test scenario 4.

2.4. Wrong information recorded as date

As results shown in test scenario 5. The program recognizes and displays a date information: 'of 19' which is not a date.

```
【Date recognition】
{'of 19'}

【Date parsing】

Process finished with exit code 0
```

Figure 18. Wrong information recorded as date in test scenario 5.

2.4. Limitation in recognize and parse all types of date information

Due to the limitation of definition for date recognition context free grammar and date parse context free grammar, the program can only recognize and parse limit types of date information which cannot cover all types of them.