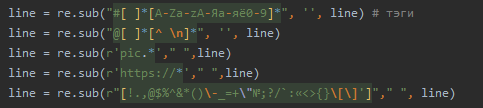
Майнор «Интеллектуальный анализ данных»

«Введение в программирование»

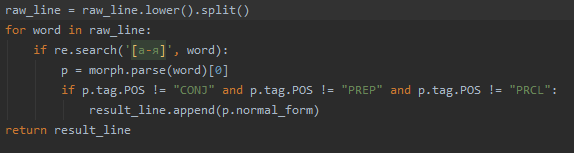
Лабораторная работа №2

1. Подготовка и обработка данных (1. preprocessing.py)

На первом этапе предложенные данные очищались от знаков препинания, ссылок, отметок с помощью регулярных выражений



Затем с помощью библиотеки pyMorphy2 удалялись служебные части речи (предлоги, союзы, частицы), а остальные слова приводились к начальной форме.



В результате данного этапа формировалось 2 файла с очищенными строками с датой и без даты.

«Сырые строки»

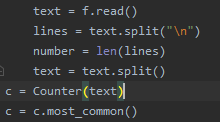
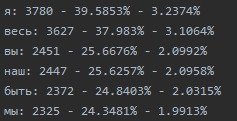
 file.txt

Строки после обработки

 result\_with\_time.txt

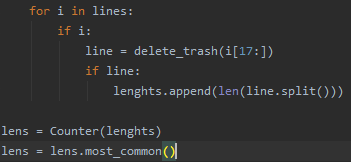
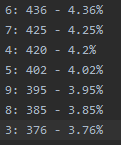
1. Частотный анализ (2.1 frequency\_words.py 2.2 frequency\_lengths.py)

На этом этапе лабораторной работы был использован класс Counter, с помощью которого формируется словарь, где ключ - это слово/количество слов в строке, а значение – количество, сколько раз этот ключ встречается.

  Frequency.txt

В первой части частотного анализа создается файл, в который записываются все встреченные слова с порядке убывания частоты использования, а так же само количество встреч, доля твитов, в которых слово используется и доля от общего количества слов в очищенных твитах.

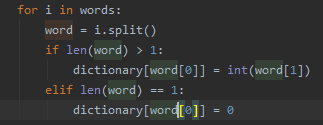
Во второй части частотного анализа был использован файл с «сырыми» строками, из которого был сформирован список с длинами соответствующей строки.

  twits\_length.txt

А также создаётся файл, куда записываются длины в порядке убывания количества твитов с такими длинами, количество твитов с таким количеством слов и процент от общего числа строк.

1. Эмпирическая оценка/разметка отдельных слов (3. estimations.py)

На данном этапе были вручную выставлены оценки настроения 3500 отдельных слов, для остальных слов с частотой встречаемости 1-2 была выставлено нейтральное настроение.

  estimations.txt

1. Правила классификации. Оценка твитов. Сравнительный анализ (classification.py)

Правила оценивания:

1. Сумма оценок

Оценка строки определяется как сумма оценок слов в этом твите

Классификация оценивается по пороговому правилу:

* Негативный: оценка < 0
* Нейтральный: 0 <= оценка <= 2
* Позитивный: оценка > 2

1. Для каждого типа слов определить долю слов такого типа в твите. Тональность твита определяется по типу с наибольшей долей.
2. Классификация оценивается по пороговому правилу по доле слов разных типов:

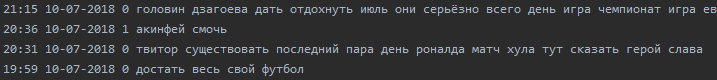
* Негативный: доля негативных слов > 0.1
* Положительный: доля положительных слов > 0.2
* Нейтральный: не подошёл под 2 первых пункта

1. Общее настроение твита оценивается по количеству слов разных типов:

* Негативный: количество негативных слов > 1
* Положительный: количество положительных слов > 1
* Нейтральный: не подошёл под 2 первых пункта

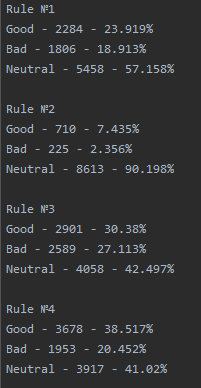
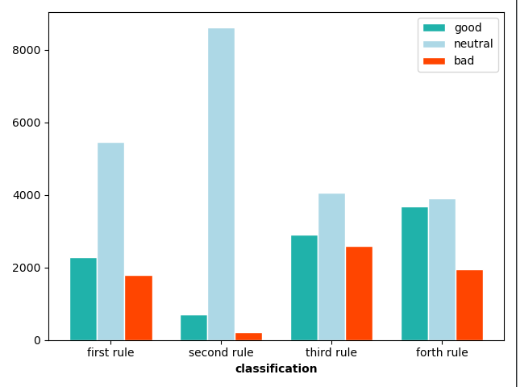
По данным правилам была проведена классификация всех твитов, результаты которой записаны в файл classifications.txt, а во оценки каждого твита записаны в файлы twit\_estimation<i>.txt

 twit\_estimation1.txt

 twit\_estimation2.txt

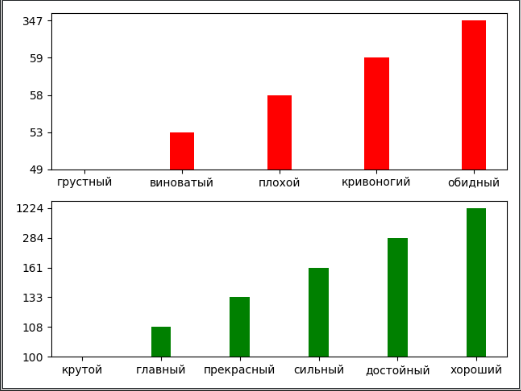
 twit\_estimation3.txt

 twit\_estimation4.txt

 classifications.txt 

5. Части речи (5. adjectives.py)

С помощью уже ранее составленного файла frequency.txt были найдены первые 5 положительных и негативных прилагательных, информация о них была записана в файл и по ней составлен график.

 (Adjectives.txt) 

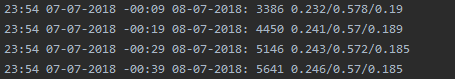
6. Оценить распределение положительных/отрицательных/нейтральных твитов по времени (6. distribution\_in\_time.py)

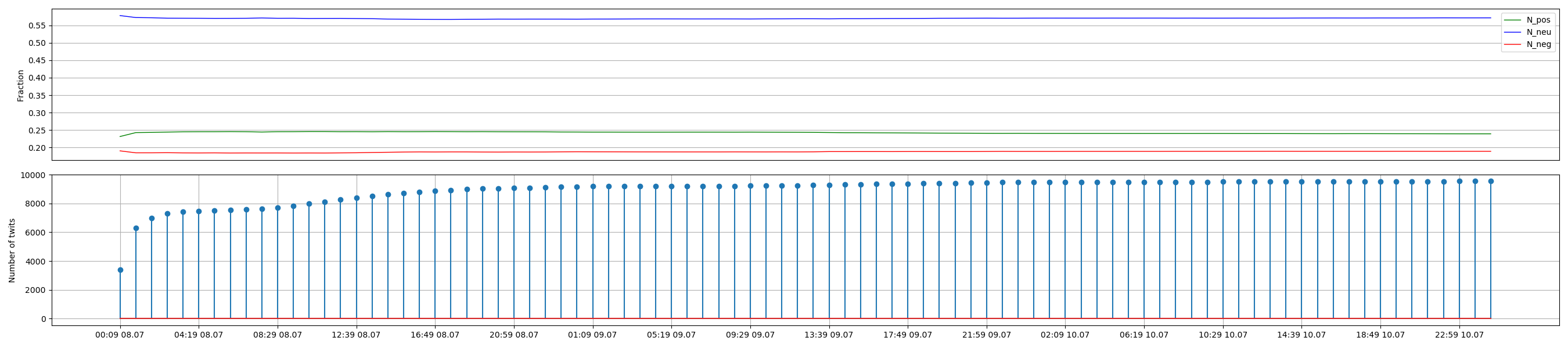
Так как твиты изначально и так отсортированы по времени, то следует лишь посчитать количество твитов разного типа в каждом их временных окон и их общее количество, записать эту информацию в файл hours rool#<i>.txt и сохранить в список для последующего построения графиков.

Для этого этапа лабораторной работы на этапе классификации и оценивания твитов были созданы файла twit\_estimation<i>.txt, в которых записаны время твита, его оценка согласно одному из правил и сама строка твита.

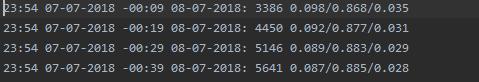
Полученные результаты:

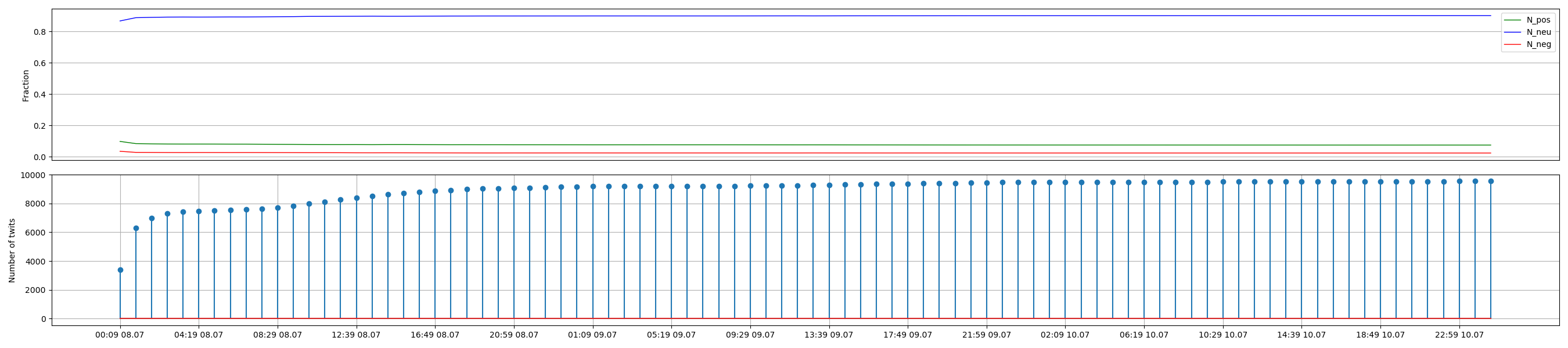
Первое правило:

 hours rool#1.txt

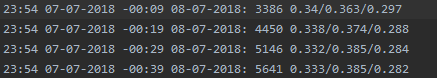


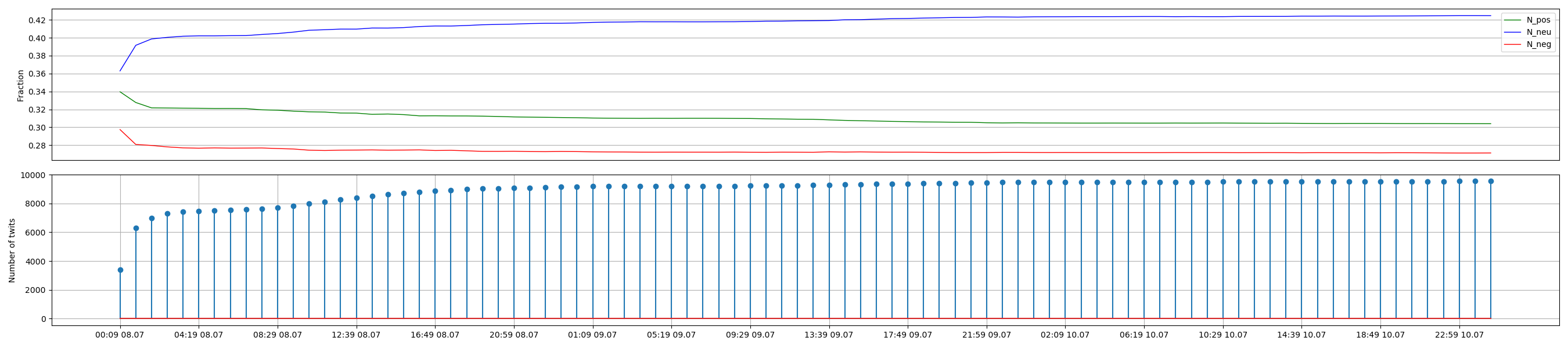
Второе правило:

 hours rool#2.txt

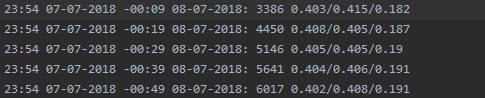


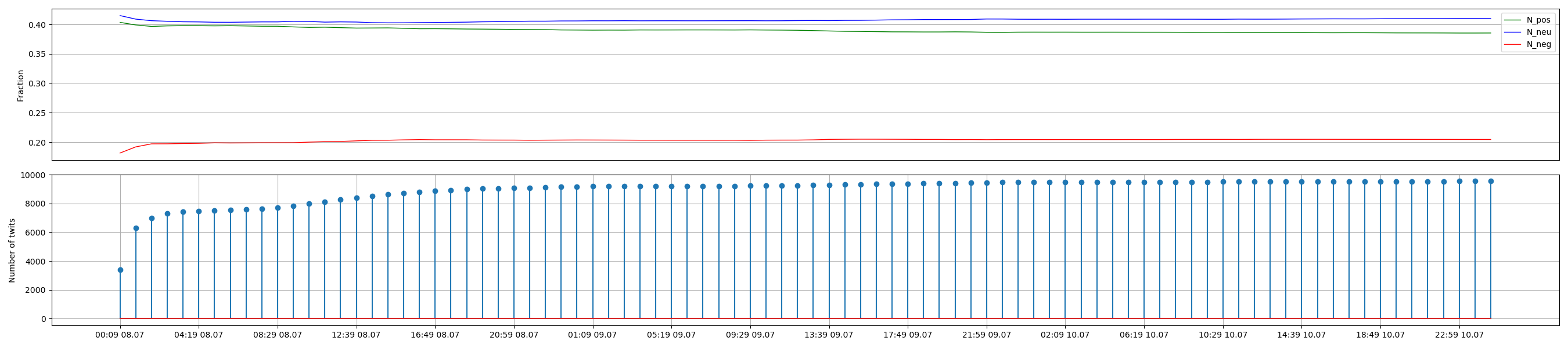
Третье правило:

 hours rool#3.txt



Четвёртое правило:

 hours rool#4.txt



7. Определение точности эмпирической оценки твитов (7. estimation\_check.py)

Для каждого правила классификации были найдены слова с наименьшим и наибольшим расхождением оценки. Слова, частота встречаемости которых меньше 3 не учитывались. Общая оценка точности считалась верной, если расхождение оценков было < 0.4. Результаты были записаны в файлы estimation check rool#<i>.txt

 estimation\_check rool#1.txt  estimatimation\_check rool#2.txt

 estimation\_check rool#3.txt  estimatimation\_check rool#4.txt

8. Частота встречаемости матных слов в разные интервалы времени (8. distribution\_in\_time\_swear\_words.txt)

Изначально бранные и оскорбительные слова были помечены отдельной оценкой в файле estimations.txt. Составив список этих слов, была создана статистика употребления матных слов относительно общего числа твитов в разные интервалы времени

