CS 412: Introduction to Data Mining **Spring 2020**

Homework

Handed Out: March 6, 2020 *Due: March* 27, 2020 *11:59 pm*

1 General Instructions

- The programming assignment will be hosted on hackerrank (`https://www.hackerrank.com/`) as a programming contest. To participate in this contest, please open a hackerrank account with your illinois.edu email netid. If you already have a username in hackerrank that is different from your netid, please fill out your netid and username in this google form (`https://forms.gle/bWSeytDD54nEdWBN8`).
- It is OK to discuss the problems with your classmates. However, it is NOT OK to work together or share code. You need to write your code independently and the TAs will not do the code debugging. Plagiarism is an academic violation to copy, to include text from other sources, including online sources, without proper citation. To get a better idea of what constitutes plagiarism, consult the CS Honor code (`http://cs.illinois.edu/academics/honor-code`) on academic integrity violations, including examples, and recommended penalties. There is a zero tolerance policy on academic integrity violations. Any student found to be violating this code will be subject to disciplinary action.
- Please use Piazza if you have questions about the homework. Also feel free to send us e-mails and come to office hours.

2 Programming Assignment Instructions

In this programming assignment, we will implement a contiguous sequential pattern mining algorithm and apply it on text data to mine potential phrase candidates. Participate in the programming contest hosted at hackerrank: `www.hackerrank.com/cs412-hw2`.

- Please read the problem description carefully.
- The input will always be valid. We are mainly testing your understanding of frequent sequential pattern mining, not your coding skills.
- Please pay special attention to the output format. We will be using the hackerrank based autograder and it is extremely important that your generated output satisfies the requirement.
- We don't have specific constrains for this programming question. The only constrains are the standard environment constraints in hackerrank: `https://www.hackerrank.com/environment`.

- The grading will be based on if you pass the test cases. You are provided with one sample test case to debug your code. For the final grading, we will use additional test cases to test your code.
- Please use Piazza first if you have questions about the homework.

3 Programming Assignment (50 points)

This question aims to provide you a better understanding of adapting a pattern mining algorithm on real-world applications based on what you learned in class.

1. Understand a new problem and design an algorithm to solve it.
2. Implement a frequent contiguous sequential pattern mining algorithm to mine the frequent phrases from a text corpus.

Problem Definition

A contiguous sequential pattern is a sequence of items that frequently appears as a consecutive subsequence in a database of many sequences. For example, if the corpus is

```
good fish sandwich and french fries
disgusting fish sandwich but good french fries
their fish sandwich is the best fish sandwich
```

and the minimum support is 2, then patterns like “fish sandwich” and “french fries” are both frequent contiguous sequential patterns, while the pattern “sandwich fries” is not a frequent contiguous sequential pattern because these two words are not consecutive to each other. Notice that “sandwich fries” is still a frequent sequential pattern though.

In this problem, the frequency is also defined differently: multiple appearances of a subsequence in a single sequence record count multiple times. The reason is that we want to find phrases, and a phrase can be repeated in a single sentence. For example, the pattern “fish sandwich” appears once in the first sequence, once in the second sequence and twice in the last, so its support should be calculated as 4. Another example: “A B A B A”. Subsequence “A B A” actually occurs twice, so the support is 2.

Input Format

The input dataset is a text corpus. Each line is basically a sequence of strings separated by spaces. Note, punctuations are also considered as words, and are also separated by spaces.

Space is not a word. The lower case letters and upper case letters are different (i.e., 'A' and 'a' are two different words).

Be aware of the size of the input.

Constraints

Minimum length is 2, maximum length is 5, and minimum support is 2. That is, the patterns have to contain at least two words, but no more than 5. The frequency of the patterns is no less than 2.

Output Format

The output are the **most frequent 20** patterns you mined out from the input dataset. The frequent patterns should be ordered according to their support from largest to smallest. Ties should be resolved by ordering the frequent patterns according to the ASCII order (increasing order).

Each line of the output should be in the format:

```
[Support frequent-pattern]
[Support frequent-pattern]
......
```

Please refer to the sample input and output below.

Sample Input

```
good grilled fish sandwich and french fries , but the service is bad
disgusting fish sandwich , but good french fries
their grilled fish sandwich is the best fish sandwich , but pricy
A B A B A B A
```

Sample Output

```
[4, 'fish sandwich']
[3, ', but']
[3, 'A B']
[3, 'A B A']
[3, 'B A']
[2, 'A B A B']
[2, 'A B A B A']
[2, 'B A B']
[2, 'B A B A']
[2, 'fish sandwich ,']
[2, 'fish sandwich , but']
```

```
[2, 'french fries']
[2, 'grilled fish']
[2, 'grilled fish sandwich']
[2, 'sandwich ,']
[2, 'sandwich , but']
```