CS Games 2017

Relay Programming III

Participants	3
Workstations	3
Value	7%
Duration	3 hours

Moderation robot

Your company just released a new version of their popular software, Ideji. However, this version changes the licensing model from a one time cost to a monthly subscription. Users' anger is being felt on social networks and the person responsible for comment moderation is overwhelmed by the volume of negative comments. You're tasked with the creation of an application that will read all comments and hide the negative ones.

Dictionary

An English dictionary (dictionary.txt, a utf-8 encoded text file) is supplied to you. Each of its lines is made up of a word and its type(s), separated by "\".

The following is the list of possible types (case-sensitive):

- **N** : Noun
- **p** : Plural
- **h** : Noun phrase
- **V** : Verb (usu participle)
- **t** : Verb transitive (Requires one or more objects)
- **i** : Verb intransitive (Doesn’t require object)
- **A** : Adjective
- **v** : Adverb
- **C** : Conjunction (and, but, etc...)
- **P** : Preposition (in, at, on)
- **!** : Interjection (word or expression that expresses a spontaneous feeling or reaction)
- **r** : Pronoun
- **D** : Definite article
- **I** : Indefinite article
- **o** : Nominative

Some filtering-related types were also added to the dictionary:

- **b** : Word present in a negative comment.
- **g** : Word present in a positive comment.
- **s** : Word that may represent our product..
- **c** : Word that may represent our competitor.

Sentence analysis

Included is **Natural Language Processing with Python** (nltk_ch08.html) to help you develop a sentence analysis algorithm. However, the NLTK library is not available.

Application execution

The application shall receive a single argument, the path to the file containing the social network messages.

```
$ automod path_to_message.csv
```

This file is a CSV (**Comma-Separated Values**) file, where each line is made up of 2 elements: the message ID and the message itself. There is an example **message.csv** file included, containing each sentence of the problem statement.

Example:

```
41,Ideji is so awful with their new monthly subscription.
```

The application shall write the message ID and its classification to standard output. There are 4 possible classes: "**positive**", "**negative**", "**neutral**", and "**not related**".

Example:

```
41,negative
```

Features

- ❏ A sentence mentioning the product name "**Ideji**" and a type "**b**" word is **negative**. (2 pts)
    - **Example** : "Ideji is bad with their new business model."

- ❏ A sentence mentioning the product name "**Ideji**" and a type "**g**" word is **positive**. (2 pts)
    - **Example** : "Ideji is my favorite ide."

- ❏ A sentence mentioning the product using a type "**s**" word, preceded by "**Your**" (Your product, Your software, Your application) and a type "**b**" word is **negative**. (3 pts)
    - **Example** : "Your product subscription is so bad."

- ❏ A sentence mentioning the product using a type "**s**" word, preceded by "**Your**" (Your product, Your software, Your application) and a type "**s**" word is **positive**. (3 pts) (3 points)
    - **Example** : "I really like your application."

- ❏ A message that doesn't contain any word referencing your product is **not related**. (2 pts)
    - **Example** : "I saw your booth at the lastest computer science meeting."

- ❏ A sentence containing a type "**c**" word and a type "**b**" word is **positive**. (2 pts)
    - **Example** : "The alternative is so bad."

- ❏ A sentence containing a type "**c**" word and a type "**g**" word is **negative**. (2 pts)
    - **Example** : "The alternative is better than you."

- ❏ A sentence containing a type “**c**” word and a type “**g**” word, but also a conjunction with a type “**s**” word and a type “**b**” is **neutral**. (2 pts)
    - **Example** : "Your competitor pricing is good but Ideji is so much better."

- ❏ Also support the inverse sentence, the conjunction contains a type “**s**” word and a type “**g**” word and the other part of the sentence contains a type “**c**” word and a type “**g**” word is **neutral**. (2 pts)
    - **Example** : "Ideji is great but I like Solarclipse too."

- ❏ A sentence containing a type "**c**" word and a type "**g**" word but also a conjunction with a type "**s**" word and a type "**b**" word is **negative**. (2 pts)
    - **Example** : "Your competitor pricing is good but Ideji subscription is awful."

- ❏ Also support the inverse case, the conjunction contains a type "**s**" word and a type "**b**" word and the other part of the sentence contains a type "**c**" word and a type "**g**" word is **negative**. (2 pts)
    - **Example** : "Ideji is bad and I really like the alternative."

- ❏ Take into account the negation (not or n't) in the sentence. A type "**g**" word become a type "**b**" word and a type "**b**" word become a type "**g**" word. (3 pts)
    - **Example** : "Your ide isn't that bad." is **positive**. "Your software isn't really good." is **negative**.

- ❏ Take into account the negation with a conjunction into the sentence. (3 pts)
    - **Example** : "Your competitor isn't good and I really like Ideji." is **positive**. "Your product isn't really the best and I really like Netfruit." is **negative**.

- ❏ Support messages with many sentences. Evaluate each sentence and calculate a score for the message. (5 pts)
    - Negative sentence : -1
    - Positive sentence : +1
    - Neutral sentence and not related : 0
    - The message score will determine the output. If score is greater than 0, the message is **positive**, lesser than 0, the message is **negative**. If score is equals to 0, the message is **neutral**.

- ❏ Include a “readme” file in your solution, document completed features and the execution of your software. (1 pt)
- ❏ Include a “run.sh” file to start your software. (1 pt)