Refactoring Report 002

Refactoring ID: 002
Title: Enhance the functionality of regular expression
Date: 2014/10/20
Vender: Acumen **Product**: News Classification & Recommendation
Platform: All platform it supports
PIC: FANG Zhou

Situation (Code Smell):
- Original regular expression is not sufficient to cover all situations

Refactoring Plan:
- Optimize the regular expression to ensure only a-z characters will be reserved

Diff:
Original

```
private void readAndProcess(File file){
    try{
        Scanner scanner=new Scanner(new FileReader(file));
        String[] s;
        String line;

        while(scanner.hasNextLine())
        {
            line=scanner.nextLine();
            s=line.split(",|\\.|\\s+|\t|\"|\'|[illegible]|[illegible]");_____ copyArray
(s);
        }

        scanner.close();
    }catch(Exception e){
        System.out.println(e.getMessage());
    }
}

private void copyArray(String[] s) {
    for(int i=0;i<s.length;i++)
    {
        if(!isInteger(s[i])&&s[i].length()!=0)
        {
            words[index]=s[i];
            index++;
        }
    }
}

private boolean isInteger(String s){
    try{
        int n = 0;
        n = Integer.parseInt(s);
        return true;
    }catch(Exception e){
        return false;
    }
}
```

In buildDictionary function:

```
if(content[i].matches(".*\\d.*") || !content[i].matches("[a-zA-Z-]+"))
    continue;
```

Updated

```
private void readAndFilter(File srcFile){
    FileReader fr;
    try {
        fr = new FileReader(srcFile);
         BufferedReader br = new BufferedReader(fr);
        String line = null;
        while((line = br.readLine()) != null){
            //split by non a-z, A-Z, -, _ ,0-9 characters
            String tokens[] = line.split("[^a-zA-Z-_0-9]");
            arrayCopy(tokens);
        }
        br.close();
    }catch (FileNotFoundException e){
        e.printStackTrace();
    }catch (IOException e){
        e.printStackTrace();
    }
}

private void arrayCopy(String [] arr){
    for(String token: arr){
        if(!token.trim().isEmpty() && token.matches("[a-zA-Z-_]+"))
            this.words.add(token.toLowerCase());//all words in lower case
    }
}
```