SentiStrength Manual

感觉强度手册

SentiStrength is a sentiment analysis tool developed by Mike Thelwall etc [1] in 2010. In this manual, we will first briefly introduce this tool, describe its functions, and show how to use this tool. More details can be found in its paper and SentiStrength’s official website [2].

SentiStrength是Mike Thelwall等[1]在2010年开发的一款情感分析工具。在本手册中，我们将首先简要介绍该工具，描述其功能，并展示如何使用该工具。更多细节可以在它的论文和SentiStrength的官网上找到[2]。

Introduce

介绍

SentiStrength is a tool developed from comments on social networking sites (MySpace [3]). Its core function is to use dictionary-based algorithms to analyse sentiment for text. Specifically, it first assigns priori sentimental scores to words according to the sentiment dictionary, and then adjusts the assignation result with several heuristic rules. It can give a sentimental score pair (ρ, η) for each input text, where ρ represents the positive score of the text, and η represents the negative score. The scale and meaning for ρ and η are as below:

SentiStrength是从社交网站(MySpace [3])上的评论发展而来的工具。它的核心功能是使用基于字典的算法来分析文本的情感。具体来说，它首先根据情感词典给词分配先验情感分数，然后用若干启发式规则调整分配结果。它可以为每个输入文本给出一个情感得分对(ρ，η)，其中ρ代表文本的正得分，η代表负得分。ρ和η的标度和含义如下:

[no positive sentiment] “+**1”– “+2” – “+3” – “+4” – “+5”** [very strong positive sentiment]

【没有正面情绪】“+1”“+2”“+3”“+4”“+5”【非常强烈的正面情绪】

[no negative sentiment] “-**1” – “ -2” – “-3” – “-4” – “-5”** [very strong negative sentiment]

[没有负面情绪]"-1 "--2 "--3 "--4 "--5 "[非常强烈的负面情绪]

Core Function

核心功能

SentiStrength's core function is to use a set of dictionaries and several heuristic rules to conduct sentimental analysis on the text. The key elements of SentiStrength are listed below.

SentiStrength的核心功能是使用一套字典和若干启发式规则对文本进行情感分析。下面列出了SentiStrength的关键要素。

* **UC-1 Assigning** **Sentiment** **Scores for Words**: The core of the algorithm is the **sentiment** **word** **strength** **list** (EmotionLookupTable in SentiStrength). EmotionLookupTable contains 2546 words or wildcards (hereinafter collectively referred to as items). Each item was preseted a sentimental score, which is an integer ranging from -5 to 5. SentiStrength will assign sentimental scores to each word in the sentence based on the EmotionLookupTable. If the word does not exist in the EmotionLookupTable, it defaults to neutral. It is noteworthy that the word “**miss**” was allocated a positive and negative strength of 2. This was the only word classed as both positive and negative. It was typically used in the phrase “I miss you”, suggesting both sadness and love.
* UC-1为词赋情感分:算法的核心是情感词强度列表(SentiStrength中的EmotionLookupTable)。EmotionLookupTable包含2546个单词或通配符(以下统称为项目)。每一项都有一个情感分数，它是一个从-5到5的整数。SentiStrength将根据情绪表为句子中的每个单词分配情绪分数。如果该单词在EmotionLookupTable中不存在，则默认为中性。值得注意的是，单词“miss”被分配了正负强度2。这是唯一一个既被归类为肯定又被归类为否定的词。它通常用在短语“我想你”中，暗示悲伤和爱。
* **UC**-**2** **Assigning Sentiment Scores for Phrases**: EmotionLookupTable is used to assign sentimental scores for uni-grams, while IdiomLookupTable is used to assign scores for the phrases which often contain multiple words. When an idiom is recognized, the sentimental score of the idiom will cover the sentimental score of the single word that constitutes the idiom. For example, In the text “It’s a killer feature.”, “killer feature” is a phrase in the dictionary with positive score 02. Although the word ‘kill’ carries negative sentiment, its effect is overridden by the sentimental score of the enclosing phrase. So the text is analyzed as positive finally.
* UC-2为短语分配情感分数:EmotionLookupTable用于为单字分配情感分数，而IdiomLookupTable用于为通常包含多个单词的短语分配分数。当习语获得认可时，习语的情感得分将覆盖构成习语的单个单词的情感得分。例如，在文本“这是一个杀手级功能。”、“杀手级特征”是字典里一个正分02的短语。虽然“kill”这个词带有负面情绪，但它的效果会被后面短语的情绪得分所掩盖。因此，本文最后进行了实证分析。
* UC-3拼写纠正:一种算法通过包含重复字母来识别拼写错误的单词的标准拼写。例如，hell 11000将被该算法识别为“hello”。算法(a)自动删除两次以上的重复字母(例如hello-> hello)；(b)对于在英语中很少出现两次的字母，删除出现两次的重复字母(例如，niice -> nice)，以及(c)删除出现两次的字母，如果它们不是标准单词，但如果被删除将形成标准单词(例如，nnice -> nice但不是hoop -> hop nor baaz -> baz)。英语单词列表将用于检查单词的拼写是否正确。
* **UC-4 Booster Word Rule**: A booster word list (BoosterWordList) contains words that boost or reduce the emotion of subsequent words, whether positive or negative. Each word increases emotion strength by 1 or 2 (e.g., very, extremely) or decreases it by 1 (e.g., some).
* UC-4增强词规则:一个增强词列表(booster word list)包含增强或降低后续词的情绪的词，无论是积极的还是消极的。每个单词增加1或2的情感强度(例如，非常，极端)或减少1(例如，一些)。
* **UC-5 Negating Word Rule**: A negating word list (NegatingWordList) contains words that invert subsequent emotion words (including any preceding booster words). For example, if "very happy" had positive strength 4 then "not very happy" would have negative strength 4. The possibility that some negating terms do not negate was not incorporated as this did not seem to occur often in the pilot data set.
* UC-5否定词规则:一个否定词表(negating word list)包含反转后续情感词的词(包括任何前面的助词)。例如，如果“非常快乐”的强度为4，那么“不太快乐”的强度为4。一些否定术语不否定的可能性没有被纳入，因为这在试点数据集中似乎不经常出现。
* **UC-6 Repeated** **Letter** **Rule**: Repeated letters above those needed for correct spelling are used to give a strength boost of 1 to sentimental words, as long as there are at least two additional letters. The use of repeated letters is a common device for expressing emotion or energy in MySpace comments, but one repeated letter often appeared to be a typing error.
* UC-6重复字母规则:只要有至少两个额外的字母，超过正确拼写所需的重复字母被用来给情感单词增加1的强度。重复字母的使用是在MySpace评论中表达情感或能量的常用手段，但一个重复的字母经常会出现打字错误。
* **UC-7 Emoji** **Rule**: An emoticon list (EmotionLookupTable) with associated strengths (positive or negative 2) supplements the sentiment word strength list (and punctuation included in emoticons is not processed further for the purposes below).
* UC-7表情符号规则:具有关联强度(正或负2)的表情符号列表(EmotionLookupTable)补充了情感单词强度列表(并且表情符号中包含的标点符号不会出于以下目的而被进一步处理)。
* **UC-8 Exclamation** **Mark** **Rule**: Any sentence with an exclamation mark was allocated a minimum positive strength of 2.
* UC-8感叹号规则:任何带有感叹号的句子都被分配了最低为2的正强度。
* **UC-9 Repeated** **Punctuation** **Rule**: Repeated punctuation including at least one exclamation mark gives a strength boost of 1 to the immediately preceding emotion word (or sentence).
* UC-9重复标点规则:包括至少一个感叹号的重复标点给紧接在前面的情感词(或句子)增加1的强度。
* **UC-10 Negative** **Sentiment** **Ignored** **in** **Questions**. For example, the question "are you angry?" would be classified as not containing sentiment, despite the presence of the word "angry". This was not applied to positive sentiment because many question sentences appeared to contain mild positive sentiment. In particular, sentences like "whats up?" were typically classified as containing mild positive sentiment (strength 2). QuestionWord is used to identify question words.
* 问题中忽略了UC-10负面情绪。比如问题“你生气了吗？”将被归类为不包含情绪，尽管存在“愤怒”一词。这不适用于积极情绪，因为许多疑问句似乎包含温和的积极情绪。特别是像“怎么了？”通常被归类为包含轻度积极情绪(强度2)。QuestionWord用于识别疑问词。

The above factors were applied separately to each sentence, with the sentence being assigned with both the most positive and most negative sentiments identified in it. Each overall text was assigned with the most positive of its sentence sentiments and the most negative of its sentence sentiments. Sentences were split either by line breaks in comments or after punctuation other than emoticons. The example in Table 1 shows how SentiStrength analyses.

上述因素分别应用于每个句子，句子被分配了最积极和最消极的情绪。每篇文章都被赋予了最积极的句子情感和最消极的句子情感。句子要么被注释中的换行符分开，要么被除表情符号以外的标点符号分开。表1中的例子显示了如何进行强度分析。

# 其他功能

Table 1 The Sample of SentiStrength

| **Sample** | **ρ** | **η** | **Dictionary/Rule**  **Used** | **Explanation** |
| --- | --- | --- | --- | --- |
| It’s a good feature. | 2 | -1 | EmotionLookupTable | The sentimental score of the word *‘good’* is pre-assigned to 02; so the sentence is assigned positive score 02. |
| It’s a very good feature. | 3 | -1 | EmotionLookupTable BoosterWordList | As booster word ‘very’ is used before the sentimental word, the sentence is assigned a positive score 03. |
| It’s not good feature. | 1 | -2 | EmotionLookupTable  NegatingWordList | Sentimental polarity of the sentimental word is inverted in here due to the use of the negation word ‘not’ before sentimental word. |
| It’s a good feature! | 3 | -1 | EmotionLookupTable  “！” Rule | “!”will strengthen the sentimental strength |
| It’s a gooood feature. | 3 | -1 | Repeated Letter Rule | Repeated letters above those needed for correct spelling are used to give a strength boost of 1 to sentimental words. |

SentiStrength was initially released in the form of a jar package. In this section, we will introduce some non-core functions and explain how to set options to use these functions when analysing. For a more comprehensive description of options, please refer to the manual on the official website.

SentiStrength最初是以jar包的形式发布的。在本节中，我们将介绍一些非核心函数，并解释如何设置选项以在分析时使用这些函数。关于选项更全面的描述，请参考官网上的手册。

# Complete Different Classification Tasks (6)

# 完成不同的分类任务(6)

SentiStrength can classify individual texts or multiple texts and can be invoked in many different ways. This section covers these methods although most users only need one of them.

SentiStrength可以对单个文本或多个文本进行分类，并且可以以多种不同的方式调用。本节将介绍这些方法，尽管大多数用户只需要其中一种。

# UC-11 Classify a single text

# UC-11对单个文本进行分类

text [text to process]

文本[要处理的文本]

The submitted text will be classified and the result returned in the form +ve –space- -ve. If the classification method is trinary, binary or scale then the result will have the form +ve –space- -ve –space- overall. E.g.,

提交的文本将被分类，结果以+ve-space-ve的形式返回。如果分类方法是三进制、二进制或比例，那么结果将具有+ve-空间-ve-空间-整体的形式。例如，

java -jar SentiStrength.jar sentidata C:/SentiStrength\_Data/ text i+love+your+dog.

Java-jar senti strength . jar senti Data C:/senti strength \_ Data/text我+爱+你+狗。

The result will be: 3 -1

结果将是:3 -1

# UC-12 Classify all lines of text in a file for sentiment [includes accuracy evaluations]

# UC-12对文件中的所有文本行进行情感分类[包括准确性评估]

input [filename]

输入[文件名]

Each line of [filename] will be classified for sentiment. Here is an example.

[文件名]的每一行都将被分类为情感。这里有一个例子。

java -jar SentiStrength.jar sentidata C:/SentiStrength\_Data/ input myfile.txt

Java-jar senti strength . jar senti Data C:/senti strength \_ Data/input my file . txt

A new file will be created with the sentiment classifications added to the end of each line.

将创建一个新文件，在每行的末尾添加情感分类。

如果任务是测试SentiStrength的准确性，那么文件可能在第一列有+ve代码，然后在第二列有负代码，在最后一列有文本。如果使用二进制/三进制/比例分类，则第一列可以包含人类编码值。列必须用制表符分隔。如果人类编码的情感分数包含在文件中，那么SentiStrength的准确性将与它们进行比较。

# UC-13 Classify texts in a column within a file or folder

# UC-13对文件或文件夹中某一栏的文本进行分类

For each line, the text in the specified column will be extracted and classified, with the result added to an extra column at the end of the file (all three parameters are compulsory).

对于每一行，指定列中的文本将被提取和分类，并将结果添加到文件末尾的额外列中(这三个参数都是必需的)。

annotateCol [col # 1..] (classify text in col, result at line end)

注释栏[栏# 1..](对列中的文本进行分类，结果位于行尾)

inputFolder [foldername] (all files in folder will be \*annotated\*)

inputFolder [foldername](文件夹中的所有文件都将被\*注释\*)

fileSubstring [text] (string must be present in files to annotate)

fileSubstring [text](字符串必须存在于要注释的文件中)

Ok to overwrite files [overwrite]

确定覆盖文件[覆盖]

If a folder is specified instead of a filename (i.e., an input parameter) then all files in the folder are processed as above. If a fileSubstring value is specified, then only files matching the substring will be classified. The parameter overwrite must be specified to explicitly allow the input files to be modified. This is a purely safety feature. E.g.,

如果指定了文件夹而不是文件名(即输入参数)，则文件夹中的所有文件都将按上述方式处理。如果指定了fileSubstring值，则只有与该子字符串匹配的文件才会被分类。必须指定参数overwrite，以显式允许修改输入文件。这是一个纯粹的安全功能。例如，

java -jar SentiStrength.jar sentidata C:/SentiStrength\_Data/ annotateCol 1 inputFolder C:/textfiles/ fileSubstring txt

Java-jar senti strength . jar senti Data C:/senti strength \_ Data/annotate col 1 input folder C:/text files/file substring txt

# UC-14 Listen at a port for texts to classify

# UC-14在一个端口监听要分类的文本

listen [port number to listen at - call OR

收听[要收听的端口号-呼叫或

This sets the program to listen at a port number for texts to classify, e.g., to listen at port 81 for texts for trinary classification:

这将程序设置为在一个端口号上监听要分类的文本，例如，在端口81上监听用于三重分类的文本:

java -jar SentiStrength.jar sentidata C:/SentiStrength\_Data/ listen 81 trinary

Java-jar senti strength . jar senti Data C:/senti strength \_ Data/listen 81 trinary

The texts must be URLEncoded and submitted as part of the URL. E.g., if the listening

文本必须进行URL编码，并作为URL的一部分提交。例如，如果收听

was set up on port 81 then requesting the following URL would trigger classification of

设置在端口81上，则请求以下URL将触发分类

the text "love you": http://127.0.0.1:81/love%20you

正文“爱你”:http://127.0.0.1:81/love%20you

The result for this would be 3 -1 1. This is: (+ve classification) (-ve classification) (trinary

结果会是3 -1 1。这是:(+ve分类) (-ve分类)(三元

classification)

分类)

# UC-15 Run interactively from the command line

# UC-15从命令行交互运行

cmd (can also set options and sentidata folder). E.g.,

cmd(也可以设置选项和sentidata文件夹)。例如，

java -jar c:\SentiStrength.jar cmd sentidata C:/SentiStrength\_Data/

Java-jar C:\ senti strength . jar cmd senti Data C:/senti strength \_ Data/

This allows the program to classify texts from the command prompt. After running this every line you enter will be classified for sentiment. To finish enter @end

这允许程序从命令提示符分类文本。运行此程序后，您输入的每一行都将被分类为情绪。若要结束，请输入@end

# UC-16 Process stdin and send to stdout

# UC-16处理标准输入并发送到标准输出

stdin (can also set options and sentidata folder). E.g.,

stdin(也可以设置选项和sentidata文件夹)。例如，

java -jar c:\SentiStrength.jar stdin sentidata C:/SentiStrength\_Data/

Java-jar C:\ senti strength . jar stdin senti Data C:/senti strength \_ Data/

SentiStrength will classify all texts sent to it from stdin and then will close. This probably the most efficient way of integrating SentiStrength efficiently with non-Java programs. The alternatives are the Listen at a port option or dumping the texts to be classified into a file and then running SentiStrength on the file.

SentiStrength将对从stdin发送给它的所有文本进行分类，然后关闭。这可能是将SentiStrength与非Java程序有效集成的最有效方式。另一种方法是在端口上监听选项，或者将要分类的文本转储到一个文件中，然后对该文件运行SentiStrength。

The parameter textCol can be set [default 0 for the first column] if the data is sent in multiple tab-separated columns and one column contains the text to be classified. The results will be appended to the end of the input data and send to STD out. The Java loop code for this is essentially:

如果数据在多个制表符分隔的列中发送，并且一列包含要分类的文本，则可以设置参数textCol第一列默认为0]。结果将被附加到输入数据的末尾，并发送到STD out。这方面的Java循环代码本质上是:

**while**((textToParse = stdin.readLine()) != **null**) {

while((text toparse = stdin . readline())！= null) {

     //code to analyse sentiment and return results

//分析情感并返回结果的代码

}

So for greatest efficiency, null should not be sent to stdin as this will close the program.

因此，为了提高效率，不应该将null发送到stdin，因为这会关闭程序。

# Set Location of Data (4)

# 设置数据的位置(4)

# UC-17 Location of linguistic data folder

# UC-17语言数据文件夹的位置

sentidata [folder for SentiStrength data (end in slash, no spaces)]

senti data[senti strength数据的文件夹(以斜杠结尾，无空格)]

This option is used to set which folder the tool searches for the dictionary needed for analysis ( such as EmotionLookupTable, IdiomLookupTable, etc).

该选项用于设置工具在哪个文件夹中搜索分析所需的词典(如EmotionLookupTable、IdiomLookupTable等)。

# UC-18 Location of sentiment term weights

# UC-18情感术语权重的位置

EmotionLookupTable [filename (default: EmotionLookupTable.txt or

EmotionLookupTable[文件名(默认:EmotionLookupTable.txt或

SentimentLookupTable.txt)].

SentimentLookupTable.txt)]。

This option is used to set which file will be set as the core sentiment word strength list for the tool. The default value is EmotionLookupTable.txt or SentimentLookupTable.txt. This file must be in the directory specified by sentidata .

该选项用于设置哪个文件将被设置为工具的核心情感词强度列表。默认值为EmotionLookupTable.txt或SentimentLookupTable.txt，该文件必须位于sentidata指定的目录中。

# UC-19 Location of output folder

# UC-19输出文件夹的位置

outputFolder [foldername where to put the output (default: folder of input)]

output folder[放置输出的文件夹名(默认:输入的文件夹)]

This option is used to set the name of the folder to put the output.

该选项用于设置放置输出的文件夹的名称。

# UC-20 File name extension for output

# 输出的UC-20文件扩展名

resultsextension [file-extension for output (default \_out.txt)]

results extension[输出的文件扩展名(default \_out.txt)]

This option is used to set what identifier will be used to mark out the output file. Its default value is “\_out.txt”. For example, if you set "input.txt" as the input file, the first output file will be “input0\_out.txt” (input file name + index + result extension).

此选项用于设置将使用什么标识符来标记输出文件。其默认值为“\_out.txt”。例如，如果将“input.txt”设置为输入文件，则第一个输出文件将是“input0\_out.txt”(输入文件名+索引+结果扩展名)。

# Set Different Type of Output (4)

# 设置不同类型的输出(4)

# UC-21 Classify positive (1 to 5) and negative (-1 to -5) sentiment strength separately

# UC-21分别对正面(1到5)和负面(-1到5)情绪强度进行分类

This is the default and is used unless binary, trinary or scale is selected. Note that 1 indicates no positive sentiment and -1 indicates no negative sentiment. There is no output of 0.

这是默认设置，除非选择了二进制、三进制或比例。请注意，1表示没有正面情绪，而-1表示没有负面情绪。没有0的输出。

# UC-22 Use trinary classification (positive-negative-neutral)

# UC-22使用三元分类法(阳性-阴性-中性)

trinary (report positive-negative-neutral classification instead)

三元(改为报告阳性-阴性-中性分类)

The result for this would be like 3 -1 1. This is: (+ve classification) (-ve classification) (trinary classification)

结果会是3 -1 1。这就是:(+ve分类) (-ve分类)(三元分类)

# UC-23 Use binary classification (positive-negative)

# UC-23使用二元分类法(正-负)

binary (report positive-negative classification instead)

二元(改为报告正负分类)

The result for this would be like 3 -1 1. This is: (+ve classification) (-ve classification) (binary classification)

结果会是3 -1 1。这就是:(+ve分类) (-ve分类)(二元分类)

# UC-24 Use a single positive-negative scale classification

# UC-24使用单一正负标度分类

scale (report single -4 to +4 classification instead)

标度(改为报告单-4到+4分类)

这样的结果将会是3 -4 -1。这就是:(+ve分类) (-ve分类)(音阶分类)

# UC-25 Explain the classification

# UC-25解释分类

explain

解释

Adding this parameter to most of the options results in an approximate explanation being given for the classification. E.g.,

将此参数添加到大多数选项中会导致对分类给出一个大致的解释。例如，

java -jar SentiStrength.jar text i+don't+hate+you. explain

java -jar SentiStrength.jar text我+不+恨+你。解释

# UC-26 Set Classification Algorithm Parameters

# UC-26设置分类算法参数

Please note that most of these options can be mapped to the core function of SentiStrength. They can change how the sentiment analysis algorithm works.

请注意，这些选项中的大部分都可以映射到SentiStrength的核心功能。他们可以改变情感分析算法的工作方式。

* alwaysSplitWordsAtApostrophes (split words when an apostrophe is met – important for languages that merge words with ‘, like French (e.g., t’aime -> t ‘ aime with this option t’aime without))
* alwaysSplitWordsAtApostrophes(遇到撇号时拆分单词-对于将单词与'合并的语言很重要，如法语(例如，t'aime -> t'aime带有此选项t ' aime不带))
* noBoosters (ignore sentiment booster words (e.g., very))
* 无助推器(忽略情绪助推器词(例如，非常))
* noNegatingPositiveFlipsEmotion (don't use negating words to flip +ve words)
* noNegatingPositiveFlipsEmotion(不要用否定词来翻转+ve词)
* noNegatingNegativeNeutralisesEmotion (don't use negating words to neuter -ve words)
* nonegating negative neutralisesemotion(不要用否定的词来中性化的词)
* negatedWordStrengthMultiplier (strength multiplier when negated (default=0.5))
* negatedWordStrengthMultiplier(求反时的强度乘数(默认值=0.5))
* maxWordsBeforeSentimentToNegate (max words between negator & sentiment word (default 0))
* maxwordsbeforestentimenttonegate(否定者和情感词之间的最大字数(默认值为0))
* noIdioms (ignore idiom list)
* noIdioms(忽略习语列表)
* questionsReduceNeg (-ve sentiment reduced in questions)
* 问题减少负面情绪
* noEmoticons (ignore emoticon list)
* 无表情图标(忽略表情列表)
* exclamations2 (exclamation marks count them as +2 if not -ve sentence)
* 感叹号2(感叹号表示+2，如果不是-ve句)
* mood [-1,0,1](interpretation of neutral emphasis (e.g., miiike; hello!!). -1 means neutral emphasis interpreted as –ve; 1 means interpreted as +ve; 0 means emphasis ignored)
* 语气[-1，0，1](中性强调的解释(如miiike你好！！).-1表示中性强调，解释为–ve；1表示解释为+ve；0表示忽略强调)
* noMultiplePosWords (don't allow multiple +ve words to increase +ve sentiment)
* noMultiplePosWords(不允许多个+ve词来增加+ve情绪)
* noMultipleNegWords (don't allow multiple -ve words to increase -ve sentiment)
* noMultipleNegWords(不允许多个单词增加情绪)
* noIgnoreBoosterWordsAfterNegatives (don't ignore boosters after negating words)
* noignoreboosterwordssafternegatives(在否定话语后不要忽略支持者)
* noDictionary (don't try to correct spellings using the dictionary by deleting duplicate letters from unknown words to make known words)
* noDictionary(不要试图使用字典通过删除未知单词中的重复字母来更正拼写)
* noDeleteExtraDuplicateLetters(不要删除单词中多余的重复字母，即使它们是不可能的，例如heyyyy)[与上面的选项不同，此选项不检查新单词是否合法]
* illegalDoubleLettersInWordMiddle [letters never duplicate in word middles] this is a list of characters that never occur twice in succession. For English the following list is used (default): ahijkquvxyz Never include w in this list as it often occurs in www
* illegaldoublelettersinwormiddle[单词中间的字母从不重复]这是一个不会连续出现两次的字符列表。对于英语，使用以下列表(默认):ahijkquvxyz从不在此列表中包含w，因为它在www中经常出现
* illegalDoubleLettersAtWordEnd [letters never duplicate at word ends] this is a list of characters that never occur twice in succession at the end of a word. For English the following list is used (default): achijkmnpqruvwxyz
* illegalDoubleLettersAtWordEnd[单词末尾的字母从不重复]这是一个在单词末尾从不连续出现两次的字符列表。对于英语，使用以下列表(默认):achijkmnpqruvwxyz
* noMultipleLetters (don't use the presence of additional letters in a word to boost sentiment)
* noMultipleLetters(不要在一个单词中使用额外的字母来提升情绪)

# Improving the accuracy of SentiStrength(2)

# 提高SentiStrength的准确性(2)

# Basic manual improvements

# 基本手动改进

If you see a systematic pattern in the results, such as the term “disgusting” typically having a stronger or weaker sentiment strength in your texts than given by SentiStrength then you can edit the text files with SentiStrength to change this. Please edit SentiStrength’s input files using a plain text editor because if it is edited with a word processor then SentiStrength may not be able to read the file afterwards.

如果你在结果中看到一个系统的模式，例如“恶心”一词在你的文本中通常比SentiStrength给出的情感强度更强或更弱，那么你可以用SentiStrength编辑文本文件来改变这一点。请使用纯文本编辑器编辑SentiStrength的输入文件，因为如果使用文字处理器编辑，SentiStrength可能无法读取该文件。

# UC-27 Optimise sentiment strengths of existing sentiment terms

# UC-27优化现有情感术语的情感强度

SentiStrength can suggest revised sentiment strengths for the EmotionLookupTable.txt in order to give more accurate classifications for a given set of texts. This option needs a large (>500) set of texts in a plain text file with a human sentiment classification for each text. SentiStrength will then try to adjust the EmotionLookupTable.txt term weights to be more accurate when classifying these texts. It should then also be more accurate when classifying similar texts.

SentiStrength可以为EmotionLookupTable.txt建议修订的情感强度，以便为给定的一组文本给出更准确的分类。这个选项需要一个纯文本文件中的大量文本(> 500)，每个文本都有人类情感分类。SentiStrength将尝试调整EmotionLookupTable.txt术语权重，以便在对这些文本进行分类时更加准确。在对相似文本进行分类时，它也应该更加准确。

optimise [Filename for optimal term strengths (e.g. EmotionLookupTable2.txt)]

优化[文件名以获得最佳术语强度(例如，EmotionLookupTable2.txt)]

This creates a new emotion lookup table with improved sentiment weights based upon an input file with human coded sentiment values for the texts. This feature allows SentiStrength term weights to be customised for new domains. E.g.,

这基于具有文本的人类编码情感值的输入文件创建了具有改进的情感权重的新的情感查找表。此功能允许为新域定制SentiStrength术语权重。例如，

java -jar c:/SentiStrength.jar minImprovement 3 input C:/twitter4242.txt optimise C:/twitter4242OptimalSentimentLookupTable.txt

Java-jar C:/senti strength . jar minImprovement 3 input C:/Twitter 4242 . txt optimize C:/Twitter 4242 optimalsentimentlookuptable . txt

This is very slow (hours or days) if the input file is large (hundreds of thousands or millions, respectively). The main optional parameter is minImprovement (default value 2). Set this to specify the minimum overall number of additional correct classifications to change the sentiment term weighting. For example, if increasing the sentiment strength of love from 3 to 4 improves the number of correctly classified texts from 500 to 502 then this change would be kept if minImprovement was 1 or 2 but rejected if minImprovement was >2. Set this higher to have more robust changes to the dictionary. Higher settings are possible with larger input files.

如果输入文件很大(分别为数十万或数百万)，这将非常慢(数小时或数天)。主要的可选参数是minImprovement(默认值为2)。设置此项以指定更改情感术语权重的附加正确分类的最小总数。例如，如果将爱情的情感强度从3增加到4将正确分类的文本的数量从500提高到502，那么如果minImprovement是1或2，则该改变将被保持，但是如果minImprovement > 2，则该改变被拒绝。将此值设置得更高，可以对字典进行更强大的更改。对于较大的输入文件，更高的设置是可能的。

要检查新字典的性能，可以使用它而不是原始的SentimentLookupTable.txt对文件进行重新分类，如下所示:

java -jar c:/SentiStrength.jar input C:/twitter4242.txt EmotionLookupTable C:/twitter4242OptimalSentimentLookupTable.txt

Java-jar C:/senti strength . jar input C:/Twitter 4242 . txt emotion lookuptable C:/Twitter 4242 optimalsentimentlookuptable . txt

# UC-28 Suggest new sentiment terms (from terms in misclassified texts)

# UC-28建议新的情感术语(来自错误分类文本中的术语)

SentiStrength can suggest a new set of terms to add to the EmotionLookupTable.txt in order to give more accurate classifications for a given set of texts. This option needs a large (>500) set of texts in a plain text file with a human sentiment classification for each text. SentiStrength will then list words not found in the EmotionLookupTable.txt that may indicate sentiment. Adding some of these terms should make SentiStrength more accurate when classifying similar texts.

SentiStrength可以建议将一组新的术语添加到EmotionLookupTable.txt中，以便对给定的一组文本进行更准确的分类。这个选项需要一个纯文本文件中的大量文本(> 500)，每个文本都有人类情感分类。SentiStrength然后会列出在EmotionLookupTable.txt中没有找到的可能表示情绪的单词。添加一些这样的术语会使SentiStrength在对相似文本进行分类时更加准确。

termWeights

术语权重

This lists all terms in the data set and the proportion of times they are in incorrectly classified positive or negative texts. Load this into a spreadsheet and sort on the PosClassAvDiff and NegClassAvDiff to get an idea about terms that either should be added to the sentiment dictionary because one of these two values is high. This option also lists words that are already in the sentiment dictionary. Must be used with a text file containing correct classifications. E.g.,

这列出了数据集中的所有术语，以及它们出现在错误分类的正面或负面文本中的次数比例。将它加载到一个电子表格中，并对PosClassAvDiff和NegClassAvDiff进行排序，以了解哪些术语应该添加到情感词典中，因为这两个值中有一个很高。该选项还列出了已经在情感词典中的单词。必须与包含正确分类的文本文件一起使用。例如，

java -jar c:/SentiStrength.jar input C:/twitter4242.txt termWeights

Java-jar C:/senti strength . jar input C:/Twitter 4242 . txt term weights

This is very slow (hours or days) if the input file is large (tens of thousands or millions, respectively).

如果输入文件很大(分别为数万或数百万)，这将非常慢(数小时或数天)。

**Interpretation**: In the output file, the column PosClassAvDiff means the average difference between the predicted sentiment score and the human classified sentiment score for texts containing the word. For example, if the word “nasty” was in two texts and SentiStrength had classified them both as +1,-3 but the human classifiers had classified the texts as (+2,-3) and (+3,-5) then PosClassAvDiff would be the average of 2-1 (first text) and 3-1 (second text) which is 1.5. All the negative scores are ignored for PosClassAvDiff

解释:在输出文件中，PosClassAvDiff列表示包含该单词的文本的预测情感得分和人类分类情感得分之间的平均差异。例如，如果单词“nasty”出现在两个文本中，并且SentiStrength将它们都分类为+1，-3，但是人类分类器将文本分类为(+2，-3)和(+3，-5)，则PosClassAvDiff将是2-1(第一个文本)和3-1(第二个文本)的平均值，即1.5。PosClassAvDiff的所有负分数都被忽略

NegClassAvDiff is the same as for PosClassAvDiff except for the negative scores.

NegClassAvDiff与PosClassAvDiff相同，只是分数为负数。

# UC-29 Machine learning evaluations

# UC-29机器学习评估

These are machine learning options to evaluate SentiStrength for academic research. The basic command is train.

这些是机器学习选项，用于评估学术研究的SentiStrength。基本命令是训练。

*train* (evaluate SentiStrength by training term strengths on results in file). An input file of 500+ human classified texts is also needed - e.g.,

*培训(通过对文件中的结果进行培训来评估SentiStrength)。还需要500+人类分类文本的输入文件，*

java -jar SentiStrength.jar train input C:\1041MySpace.txt

java -jar SentiStrength.jar训练输入C:\1041MySpace.txt

This attempts to optimise the sentiment dictionary using a machine learning approach and 10-fold cross validation. This is equivalent to using the command optimise on a random 90% of the data, then evaluating the results on the remaining 10% and repeating this 9 more times with the remaining 9 sections of 10% of the data. The accuracy results reported are the average of the 10 attempts. This estimates the improved accuracy gained from using the optimise command to improve the sentiment dictionary.

这试图使用机器学习方法和10重交叉验证来优化情感字典。这相当于对随机的90%的数据使用优化命令，然后对剩余的10%的数据评估结果，并对剩余的10%的数据的9个部分重复9次。报告的准确度结果是10次尝试的平均值。这估计了通过使用优化命令来改进情感词典所获得的改进的准确性。

它的输出是两个文件。以in \_out.txt结尾的文件报告各种准确性统计数据(例如，正确的数量和比例，正确值的1以内的数量和比例；感觉强度和人类编码值之间的相关性。以in \_out\_termStrVars.txt结尾的文件报告了每个文件夹中情感词典的变化。这两个文件还报告了用于情感算法和机器学习的参数。参见结果意味着什么？部分了解更多信息。

# Evaluation options

# 评估选项

* *all* Test all option variations listed in Classification Algorithm Parameters above rather than use the default options
* *all测试上述分类算法参数中列出的所有选项变体，而不是使用默认选项*
* *tot* Optimise by the number of correct classifications rather than the sum of the classification differences
* *tot通过正确分类的数量而不是分类差异的总和进行优化*
* *iterations* [number of 10-fold iterations (default 1)] This sets the number of times that the training and evaluation is conducted. A value of 30 is recommended to help average out differences between runs.
* *迭代次数[10次迭代的次数(默认为1)]设置训练和评估进行的次数。建议值为30，以帮助平均运行之间的差异。*
* minImprovement [min extra correct class. to change sentiment weights (default 2)] This sets the minimum number of extra correct classifications necessary to adjust a term weight during the training phase.
* 最小额外正确类。更改情感权重(默认为2)]这将设置在训练阶段调整术语权重所需的额外正确分类的最小数量。
* *multi* [# duplicate term strength optimisations to change sentiment weights (default 1)] This is a kind of super-optimisation. Instead of being optimised once, term weights are optimised multiple times from the starting values and then the average of these weights is taken and optimised and used as the final optimised term strengths. This should in theory give better values than optimisation once. e.g.,
* *multi[# duplicate term strength optimizations to change perspective weights(默认为1)]这是一种超级优化。术语权重不是被优化一次，而是从起始值被优化多次，然后取这些权重的平均值，并被优化和用作最终优化的术语强度。这在理论上应该比优化一次更有价值。例如，*

java -jar SentiStrength.jar multi 8 input C:\1041MySpace.txt iterations 2

Java-jar senti strength . jar multi 8输入C:\1041MySpace.txt迭代2

**Example: Using SentiStrength for 10-fold cross-validation**

**示例:使用SentiStrength进行10重交叉验证**

**What is this?** This estimates the accuracy of SentiStrength *after* it has optimised the term weights for the sentiment words (i.e., the values in the file EmotionLookupTable.txt).

这是什么？这在SentiStrength已经优化了情感词的术语权重(即，文件EmotionLookupTable.txt中的值)之后估计senti strength的准确性。

**What do I need for this test?** You need an input file that is a list of texts with human classified values for positive (1-5) and negative (1-5) sentiment. Each line of the file should be in the format:

这个测试我需要什么？您需要一个输入文件，它是一个文本列表，包含积极情绪(1-5)和消极情绪(1-5)的人类分类值。文件的每一行都应该采用以下格式:

Positive <tab> Negative <tab> text

正< tab >负< tab >文本

**How do I run the test?** Type the following command, replacing the filename with your own file name.

我如何运行测试？键入以下命令，用您自己的文件名替换文件名。

java -jar SentiStrength.jar input C:\1041MySpace.txt iterations 30

java -jar SentiStrength.jar输入C:\1041MySpace.txt迭代30次

This should take up to one hour – much longer for longer files. The output will be a list of accuracy statistics. Each 10-fold cross-validation

这可能需要一个小时，对于较长的文件，时间会更长。输出将是一个精度统计列表。每个10倍交叉验证

**What does 10-fold cross-validation mean?** See the k-fold section in <http://en.wikipedia.org/wiki/Cross-validation_(statistics)>. Essentially, it means that the same data is used to identify the best sentiment strength values for the terms in EmotionLookupTable.txt as is used to evaluate the accuracy of the revised (trained) algorithm – but this isn’t cheating when it is done this way.

10重交叉验证是什么意思？请参见中的k形折叠部分<http://en.wikipedia.org/wiki/Cross-validation_(statistics)>。本质上，这意味着使用相同的数据来确定EmotionLookupTable.txt中术语的最佳情感强度值，正如用于评估修订(训练)算法的准确性一样——但如果这样做，这就不是作弊。

The first line in the results file gives the accuracy of SentiStrength with the original term weights in EmotionLookupTable.txt.

结果文件中的第一行给出了在EmotionLookupTable.txt中使用原始术语权重的SentiStrength的准确性。

结果意味着什么？阅读结果最简单的方法是复制并粘贴到Excel等电子表格中。创建的表格列出了用于对文本和结果进行分类的选项。以下是前两行关键结果的摘录。它给出了正面情绪的正确总数(位置正确)和正确比例(位置正确/总数)。它还报告了正确的或与正确相差不超过1的预测数(误差不超过1)。负面情绪也有同样的信息。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Pos  Correct | Pos Correct/  Total | Neg  Correct | Neg Correct/  Total | Pos  Within1 | Pos  Within1/  Total | Neg  Within1 | Neg Within1/  Total |
| 653 | 0.627281 | 754 | 0.724304 | 1008 | 0.9683 | 991 | 0.951969 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 刷卡机  正确的 | 位置正确/  总数 | 底片  正确的 | 阴性正确/  总数 | 刷卡机  1以内 | 刷卡机  在1/内  总数 | 底片  1以内 | 负1/  总数 |
| 653 | 0.627281 | 754 | 0.724304 | 1008 | 0.9683 | 991 | 0.951969 |

Here is another extract of the first two rows of the key results. It gives the correlation between the positive sentiment predictions and the human coded values for positive sentiment (Pos Corr) and the Mean Percentage Error (PosMPEnoDiv). The same information is given for negative sentiment.

这是前两行关键结果的另一个摘录。它给出了正面情感预测与正面情感的人类编码值(Pos Corr)和平均百分比误差(PosMPEnoDiv)之间的相关性。负面情绪也有同样的信息。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pos Corr | NegCorr | PosMPE | NegMPE | PosMPEnoDiv | NegMPEnoDiv |
| 0.638382 | 0.61354 | Ignore this | Ignore this | 0.405379 | 0.32853 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 位置更正 | 内科尔 | 波斯姆佩 | NegMPE | PosMPEnoDiv | NegMPEnoDiv |
| 0.638382 | 0.61354 | 忽略这个 | 忽略这个 | 0.405379 | 0.32853 |

If you specified 30 iterations then there will be 31 rows, one for the header and 1 for each iteration. Take the average of the rows as the value to use.

如果您指定了30次迭代，那么将有31行，一行用于标题，一行用于每次迭代。取行的平均值作为要使用的值。

Reference

参考

1. THELWALL M, BUCKLEY K, PALTOGLOU G, et al. Sentiment strength detection in short informal text[J]. Journal of the American society for information science and technology, 2010, 61(12) : 2544 – 2558.
2. 王文杰，等.短文本中情感强度的检测[J].美国信息科学与技术学会杂志，2010，61(12):2544–2558。
3. <http://sentistrength.wlv.ac.uk>
4. <http://sentistrength.wlv.ac.uk>
5. https://myspace.com/
6. https://myspace.com/