Digital Epidemiology:
Sentiment Analysis Architectures, Algorithms, & Applications for Disease Ecology

Erick Oduniyi (**eeoduniyi@gmail.com**)
This document was prepared for the members of the University Of Kansas' Agusto Epidemiology Lab

ABSTRACT

The goal of this document is to review the standard sentiment analysis methods and their associated advantages and disadvantages. In the end, this will provide a platform to sensibly implement sentiment analysis algorithms reliably for any domain and application of interest. In route and in particular, we review and provide basic examples of these algorithms along the way, and finally outline how they could be used in epidemiology and assessing the public's health through data mining platforms.

1 INTRODUCTION

The ability to understand or infer emotion signatures from an individual's information: text, speech, pictures, art works has always generated healthy amounts of engineering and scientific interest. Now, in the current information age where social-media platforms like Facebook, Instagram, and Twitter attract and interconnect individuals from around the world, understanding these emotion characteristics have become critical. Indeed, these social media platforms offer individuals the ability to express and disseminate their self expression; instantaneously highlighting their personalities to the world. Hence, social media platforms offer potentially rich digital representations of humanity at large. Because of the large amounts personality representations, anthropologist, psychologist, and sociologist might be able to glean a unique understanding of human nature through automated belief and opinion mining and subsequent analysis. Principally, this automated emotion analysis is possible because of *sentiment analysis*. As a result, it has become increasingly common practice to unleash sentiment analysis methods for the prediction of election outcomes, commerce and marketing, user satisfaction, and general behavioral analysis online.

2 SENTIMENT ANALYSIS

Sentiment analysis refers to the automated process of extracting the underlying affect in content. More, sentiment analysis makes use of tools from the field of natural language processing and computational linguistics to study polarity, affect, and subjective information (e.g., speech, text, images). Assessing the *polarity* – the degree to which a text is *positive*, *negative*, or *neutral* is often the most basic task[1] of sentiment analysis. While extracting "sentiment" – *affective states* $\in$ {"angry", "sad", "happy", ... } is a slightly more sophisticated sentiment analysis endeavor.

To avoid misunderstanding, it is necessary to specify that *emotion* and sentiment are not equivalent. Emotion is typically believed to be instinctive, triggered by psychological responses to direct experiences. We often try and capture emotion with cultural concepts like happiness, sadness, fear, and surprise. While sentiment is certainly influenced by emotion, sentiment describes the mental attitudes individuals have towards things (i.e., opinion) which is formed by cultural meaning. Thus, when talking about sentiment analysis, individuals introduce the concept of polarity to tie together emotion and sentiment, where content can have a positive (or slightly positive or very positive), neutral, and negative (or slightly negative or very negative). Despite this and other sentiment analysis conventions, there are a few general difficulties of the sentiment analysis practice. Here, they are outlined as such:

- *Extremely difficult to develop a general sentiment analysis tool. It is always necessary to train and identify domain specific sentiment.*

[1] The assumption of virtually all sentiment analysis algorithms is that people have a general understanding of things that are *very positive* and *very negative*

- *Complex sentences that contain sarcasm, irony, context, negation, qualifiers, and multiple subject attitudes are difficult to account for. In fact, many sentiment analysis tools do not account for such complexity.*
- *Human judgment is used to asses the accuracy of sentiment analysis tools. However, it is impossible to get human raters to unanimously agree on sentiment. Thus, the ballpark for accuracy is limited.*

Despite these general complications, sentiment analysis still offers the ability for one to break down human generated text and information into sequences of emotions, so that the relationship between emotional content and other variables might be explored.

2.1 Sentiment Analysis Methods

Typically sentiment analysis is done by utilizing machine learning methods or dictionary-based look up algorithms. Machine learning methods require rich domain specific data, and could take longer to implement due to training, but also return more accurate results. For text, dictionary-based methods allow users to access the polarity of specific words, which provides a granular look at the perceived emotional content. Both methods must make use of human analysis, where individual ratings of content are aggregated.

2.1.1 Dictionary-based

Dictionary-based or *lexicon-based* sentiment analysis methods use a *sentiment dictionary* or *sentiment lexicon*, where words are annotated by a *polarity score*. Then, the specified *dictionary-based sentiment analysis algorithm* takes in the content word-by-word assessing the individual word polarities and then finally producing some average or aggregated score of the entire content.

Dictionary look-up algorithms In general, the *dictionary-based sentiment analysis procedure* can be described as:

1. **Tokenization** – For each paragraph p_i, we break into a set of sentences $p_i = \{s_1, s_2, \ ... \ , s_3\}$ Then, for each sentence s_i, we break into a set of words $s_i = \{w_1, w_2, \ ... \ , w_n\}$.
2. **Rating** – We'll denote $w_{i,j,k}$ as the k^{th} word in the j^{th} sentences of the i^{th} paragraph. For each word $w_{i,j,k}$, search through the respective sentiment dictionary. Give words that are positive a score of +1 and words that are negative a score of -1.
3. **Aggregation** – Depending on the sophistication of the dictionary algorithm, for each sentence we perform a count of the positive and negative words, take the *absolute proportional difference*, or *relative proportional difference*:

$$Sentence\ Polarity = w_+ - w_- \tag{1}$$

$$Sentence\ Polarity = \frac{w_+ - w_-}{w_+ + w_- + w_0} \tag{2}$$

$$Sentence\ Polarity = \frac{w_+ - w_-}{w_+ + w_-} \tag{3}$$

Where w_-, w_+, w_0 denote the total positive, negative, and neutral words in a sentence, respectively.

2.1.2 Dictionaries

Syuzhet uses three standard sentiment dictionaries: *nrc*, *afinn*, *bing*, and a custom sentiment dictionary *syuzhet*. Syuzhet augments its sentiment analysis based on each dictionary. Sentimentr makes use of the same dictionaries as syuzhet and pattern.nlp is composed of its own list of adjectives.

A variety of lexicon-based sentiment analysis algorithms and their sentiment dictionaries can be found in many packages and libraries within the programming languages Python and R. For example, Python's *pattern.nlp* and *natural language toolkit* (NLTK) offer some of the most comprehensive library of natural language processing tools, of which include dictionary and classification-based sentiment analysis methods. More, with Python's highly popular *sckit-learn* package, students and researchers can utilize a variety of machine learning methods in combination with NLTK to generate robust sentiment extraction tools. Likewise, R has a number of dictionary-based sentiment analysis tools available in the following R libraries: *syuzhet*, *sentimentr*, *tm*.

For both Python and R, each of these lexicon-based algorithms utilize a different analysis and aggregation schema (e.g., absolute proportional difference or relative proportional difference). Even more, while the pattern.nlp package was originally developed for python, it is possible to cross-compile its functions for R. Pattern.nlp is composed of a list of adjectives in English, French, Dutch, and Spanish that occur frequently in product reviews. Worth noting, unlike the other dictionary-based sentiment analysis algorithms, pattern.nlp performs sentiment analysis by taking the **average of the adjectives within a sentence** opposed to the absolute or proportional difference of **all** words.

As an example of this difference, we will examine the following sentence with pattern.nlp and compare the results to other dictionary-based algorithms:

"The movie attempts to be surreal by incorporating various time paradoxes, but it's presented in such a ridiculous way it's seriously boring."

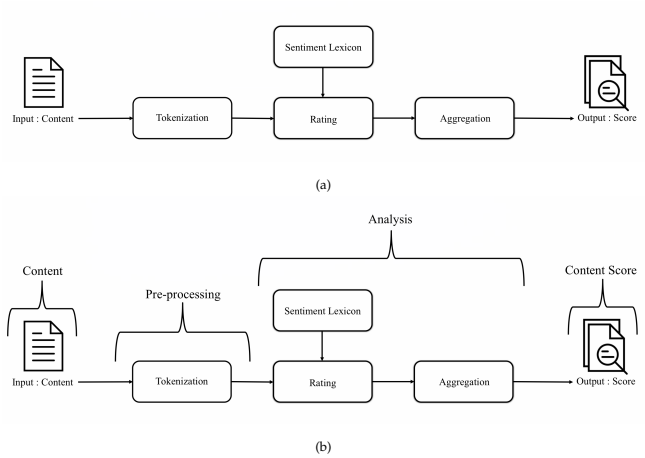

Fig. 1 – Typical systems architecture of lexicon-based sentiment analysis, where **(a)** and **(b)** show the non-annotated and annotated system process respectively.

Algorithm	pattern	nrc	afinn	syuzhet
Pattern.nlp	x	–	–	–
SentimentR	–	x	x	x
Syuzhet	–	x	x	x

Table 1 – Comparing performance on "The movie..." between pattern.nlp to sentimentr and syuzhet: nrc, afinn, bing, syuzhet

Human	Rating	Subjectivity
1	–	–
2	–	–
3	–	–

Table 2 – Comparing performance on "The movie..." between human participants

Pattern.nlp returns a sentiment score of -0.34 as it is the average sentiment score of *surreal*, *various*, *ridiculous* and *seriously boring*. Syuzhet makes use of relative proportional difference and four different sentiment dictionaries. Sentimentr is an augmentation of syuzhet, where it tries to account for *valence shifters* (negators, amplifiers, de-amplifiers, and adversative conjunctions)

Algorithm 1 *Word-Tokenization*

let $Content = \{paragraph_1, paragraph_2, \ldots, paragraph_n\}$
function TOKENIZER($Content$)
  let $\tilde{W} = \{\}$
  for $paragraph_i = 1 : n$ **do**
    let $paragraph_i = \{sentence_1, sentence_2, \ldots, sentence_m\}$
    for $sentence_j = 1 : m$ **do**
      let $sentence_j = \{w_{i,j,1}, w_{i,j,2}, \ldots, w_{i,j,k}\}$
      $\tilde{W} = \{\tilde{W}, sentence_j\}$
    end for
  end for
return $\tilde{W}$
end function

Algorithm 2 *Dictionary-based*

procedure SENTIMENTANALYSIS($Content, Dict$)
  $\tilde{W} =$ TOKENIZER($Content$)
  function RATER($\tilde{W}$)
    let $w^+ = \{\};\ w^- = \{\};\ w^0 = \{\}$
    for $w_{i,j,k} = 1 : |\tilde{W}|$ **do**
      if $w_{i,j,k} \in Dict^+$ **then**
        $w^+ = \{w^+, w_{i,j,k}\}$
      else if $w_{i,j,k} \in Dict^-$ **then**
        $w^- = \{w^-, w_{i,j,k}\}$
      else
        $w^0 = \{w^0, w_{i,j,k}\}$
      end if
    end for
    $\tilde{W}_{rated} = (w^+, w^-, w^0)$
  end function
  function AGGREGATE($\tilde{W}_{rated}$)
    $Content_{score} = \dfrac{w_+ - w_-}{w_+ + w_- + w_0}$
  end function
return $Content_{score}$
end procedure

2.1.3 Main Advantages

The strongest asset of this technique is that it does not require any training data.

2.1.4 Main Disadvantages

Dictionary-based approaches weakest points are the requirement of a large enough number of words and their associated polarity scores or annotations. Additionally, sentences, and larger-level language expressions are not included in sentiment lexicons.

2.1.5 Machine-learning-based

The sentiment analysis task is usually modeled as a classification problem where a classifier is fed with a text and returns the corresponding category, e.g. positive, negative, or neutral (in case polarity analysis is being performed).

2.1.6 Feature extraction

To perform ML-based sentiment analysis, one first must represent the sentences in some vector space. Frequency-based methods are commonly used to represent a sentence either by a bag-of-words, which is a list of the words that appear in the sentence with their frequencies, or by a term frequency-inverse document frequency (tf-idf) vector where the word frequencies in our sentences are weighted with their frequencies in the entire corpus.

Algorithm 3 *Feature Extraction*

procedure FEATUREEXTRACTION
end procedure

2.1.7 Naive Bayes

A family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text.

2.1.8 Linear Regression

The most basic algorithm in statistics for the prediction of the dependent variable Y given a set of features or dependent vairable X.

2.1.9 Support Vector Machines

A non-probabilistic model which uses a representation of text examples as points in a multidimensional space. These examples are mapped so that the examples of the different categories (sentiments) belong to distinct regions of that space.. Then, new texts are mapped onto that same space and predicted to belong to a category based on which region they fall into.

2.1.10 Deep Learning

Fig. 2 – Typical systems architecture of classification-based sentiment analysis.

Algorithm 4 *Machine-Learning-based*

1: **procedure** SENTICLASSIFICATION
2: **end procedure**

2.1.11 Main Advantages

2.1.12 Main Disadvantages

2.2 Combined Approach – Hybrid Systems

The combination of machine learning and lexicon-based approaches to address sentiment analysis is called Hybrid. Though not commonly used, this method usually produces more promising results than the approaches mentioned above.

Fig. 3 – Typical systems architecture of hybrid-based sentiment analysis.

Algorithm 5 *Hybrid*

1: **procedure** SENTIMENTHYBRID
2: **end procedure**

2.2.1 Main Advantages

2.2.2 Main Disadvantages

2.3 Evaluating Sentiment Analysis Methods

3 EMOTION RECOGNITION SYSTEMS

4 APPLYING SENTIMENT ANALYSIS

REFERENCES