Sentiment Analysis on Social Media Commentaries To Forestall Suicidal Ideation

Philip Heo
‘Iolani School

Abstract.
Sentiment analysis is defined as “a natural language processing technique used to determine whether data is positive, negative or neutral.”1 It is often used as a method to analyze the mental or emotional state of a person who might be going through mental struggles and facing personal challenges that urge them to make unhealthy or suicidal decisions. In order to prevent melancholic disorders or others of the sort, we used machine learning techniques to predict and analyze the emotions of the person by looking at that person’s comments posted on the internet, specifically on social media platforms.

We modelled a general twitter comment, processing it, and applied it with a naive bayesian model and confirmed the operation of the program. However, there were limits to the naive bayesian model of not being able to consider the word order of the sentences inputted. From this, a long short-term memory(LSTM) and recurrent neural network (RNN) were used to take into account the words in the sentences in a more precise fashion. Since the sentences contain many different forms of the same word, prepositions, and slangs, a preprocessing of removing unnecessary words and creating a list of commonly used slangs was implemented. This ultimately yielded an 81% of accuracy overall.

Introduction

Motivation

Suicides rate has increased 33% in the U.S. from 1999 to 2017, with being second leading cause of death for 10 to 34-year-olds.[2] According to a research conducted in NYU, “use of words related to negative emotions and anger significantly increased among Twitter users with major depressive symptoms compared to those otherwise”.[3]

This indicates the significant correlation between severe emotional crisis and an individual’s expressions through social media. Therefore, we try to contribute to society by analyzing the articlesposted on Twitter to understand the user's feelings and further detect depression or suicide risk early.

[1] MonkeyLearn, "Everything There Is to Know about Sentiment Analysis," MonkeyLearn, https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20(or%20opinion%20mining,feedback%2C%20and%20understand%20customer%20needs.

[2] Kirsten Weir, "Worrying Trends in U.S. Suicide Rates," American Psychological Association, last modified March 2019, https://www.apa.org/monitor/2019/03/trends-suicide.

[3] Minsu Park, Chiyoung Cha, and Meeyoung Cha, "Depressive Moods of Users Portrayed in Twitter," NYU Scholars, last modified 2012, https://nyuscholars.nyu.edu/en/publications/depressive-moods-of-users-portrayed-in-twitter.

Huge volumes of tweets are created every day. But it’s hard to analyze, understand, and sort through, as well as being time-consuming and expensive. We can use sentiment analysis as a solution. Sentiment analysis is the use of natural language processing, text analysis, and computational linguistics to systematically identify, extract, quantify, and study subjective information. Sentiment analysis is widely applied to reviews, survey responses, social media, and healthcare materials, with applications that range from marketing and customer service to clinical medicine. In this research, we used sentiment analysis to analyze 1.6 million tweets extracted using the twitter api.

Design of the Paper

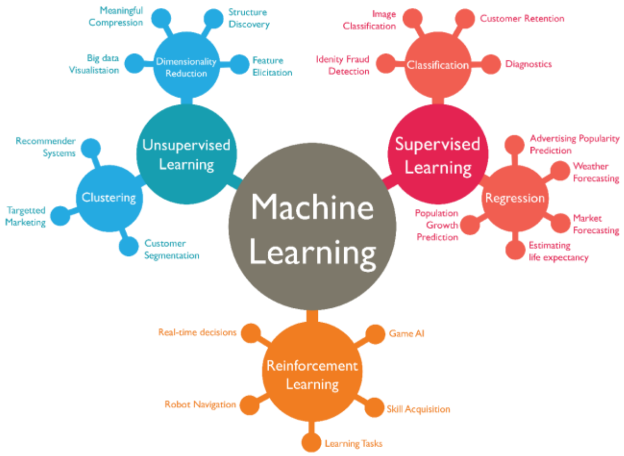
- In Section 2, there is an explanation on the background knowledge utilised in this research.
- In Section 3, the Naive Bayesian Model is introduced.
- In Section 4, an LSTM model is rendered.
- In Section 5, improving our model by preprocessing of data and preventing overfitting is shown.
- Conclusion and references

Backgrounds Knowledge

Machine Learning & artificial neural networks

Machine learning is an “application of artificial intelligence (AI)” that “provides systems the ability to automatically learn and improve from experience” without being explicitly coded or programmed first-hand. Machine learning focuses on the “development of computer programs that can access data” and use it to “learn for themselves.”

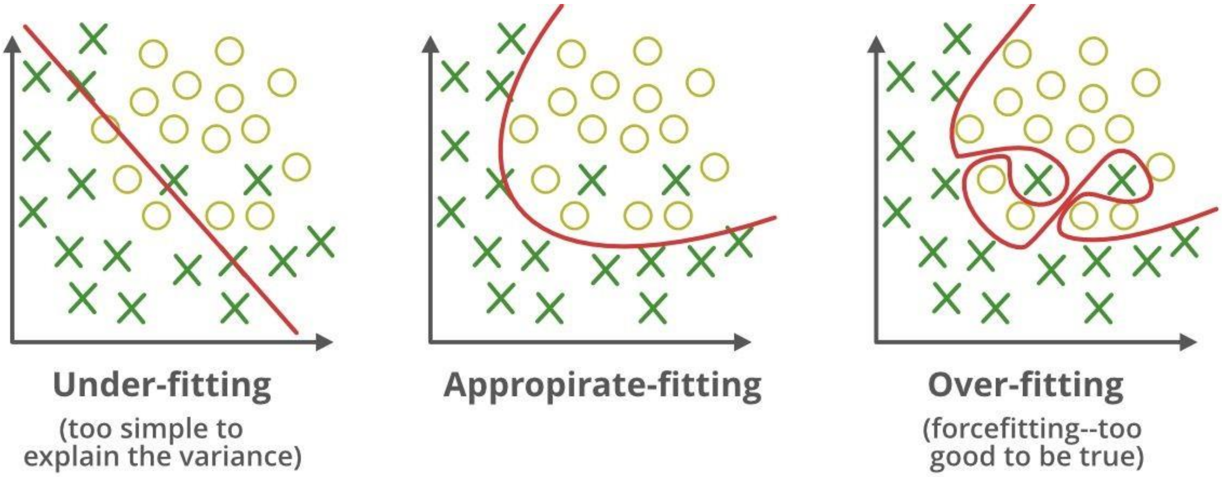
The process of learning “begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide.” The primary aim is “to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.”



However, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text. Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the modelaccordingly.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

Overfitting


Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

In statistics, overfitting is "the production of an analysis that corresponds too closely or exactly toa particular set of data, and may therefore fail to fit additional data or predict future observations reliably".[4] An overfitted model is a statistical model that contains more parameters than can be justified by the data.[5] The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.[6] Underfitting occurs when a statistical model cannot adequately capture the underlying structure of the data. An under-fitted model is a model where some parameters or terms that would appear in a correctly specified model are missing.[7] The most obvious consequence of overfitting is poor performance on the validation dataset. Other negative consequences include: A function that is overfitted is likely to request more information about each item in the validation dataset than does the optimal function; gathering this additional unneeded data can be expensive or error-prone, especially if each individual piece of information must be gathered by human observation and manual data-entry. A more complex, overfitted function is likely to be less portable than a simple one. At one extreme, a one-variable linear regression is so portable that, if necessary, it could even be done by hand. At the other extreme are models that can be reproduced only by exactly duplicating the original modeler's entire setup, making reuse or scientific reproduction difficult.[8]

[4] Lexico, https://www.lexico.com/en/definition/overfitting.

[5] Stewartschultz, last modified 2010, http://www.stewartschultz.com/statistics/books/Cambridge%20Dictionary%20Statistics%204th.pdf.

[6] Kenneth P. Burnham, "Model Selection and Multimodel Inference," Springer, last modified 2002, https://www.springer.com/gp/book/9780387953649.

[7] Stewartschultz.

[8] Douglas M. Hawkins, "The Problem of Overfitting," National Library of Medicine, last modified February 2004, https://pubmed.ncbi.nlm.nih.gov/14741005/.

Tf-idf

In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[9] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today. A survey conducted in 2015 showed that 83% of text-based recommender systems in digital libraries use tf–idf.[10]

Typically, the tf-idf weight is composed of two terms: the first computes the normalized Term Frequency(TF), aka. the number of times a word appears in a document, divided by the total number ofwords in that document; the second term is the Inverse Document Frequency(IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

TF: Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more often in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

> *TF(t)* = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and"that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

> *IDF(t)* = log_e(Total number of documents / Number of documents with term t in it)[11]

[9] Jure Leskovec, "Mining of Massive Datasets," Stanford, last modified 2014, http://infolab.stanford.edu/~ullman/mmds/book.pdf.

[10] Joeran Beel, "Research-Paper Recommender Systems: A Literature Survey," Intelligent SystemsGroup, https://isg.beel.org/pubs/2016%20IJDL%20--%20Research%20Paper%20Recommender%20Systems% 20--%20A%20Literature%20Survey%20(preprint).pdf.

[11] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, last modified 1972, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.8343&rep=rep1&type=pdf.

Naive Bayesian Model

The Naive Bayesian Model is used to determine a predictable, likely consequence or future event, most commonly applied to a weather forecast, as an example. As it is a conditional probability model, this model can predict whether or not it will rain based on the temperature, air pressure, weather, wind, etc.

The machine needs to 'learn' from a true data set that has both these factors that are initially regarded as relevant factors which cumulatively lead to the ultimate result of raining. With such true data sets being put into a computer, and the more the input the better, the computer would be able to predict based on the following model:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Where A and B are events, and P(A|B) being the probability of A given B is true.

As the data sets are based on actual occurrences, factors that are irrelevant to the predicted event are naturally likely to even out the probability for when it is satisfied and when it isn't. The model below renders factors such as 'temperature' irrelevant to predicting whether it will rain:

	Weather		Windy		Atmospheric pressure		Temperature		
	Pleasant	Grim	Yes	No	High	Low	High	Low	Total
Rain	2	6	6	2	8	0	5	3	8
No Rain	8	4	2	10	2	10	6	6	12
Total	10	10	8	12	10	10	11	9	20

The more contributing factors are recorded and therefore taken into account, the more likely for arelevant factor to be taken into account, which leads to a more accurate model. From the refined crawled tweets, numerous data sets that correlate the positive or negative state of the comment with the significant words in that comment could be put into machine learning, that is used later with the Bayesian model to predict whether a given comment is a positive or a negative one based on its composition.

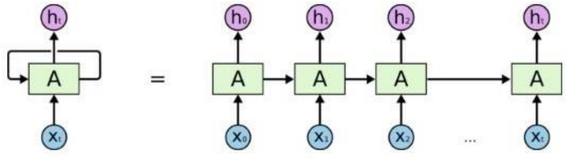
However, there are limits to the Bayesian model as well. It does not take into consideration the order in which the words are arranged in the sentence, but only whether a word appears in the sentence or not, and so accepts two different sentences of completely different meanings as that of the same meaning.

For example, the following sentences are accepted as sentences with similar meanings, while it really isn't.

- Sentence 1: I very much enjoyed this show, it was fascinating.
- Sentence 2: I did not enjoy this show, it was not as fascinating as I thought. A long short-term memory model that we will cover below will resolve this issue.

LSTM Model

LSTM model is an RNN, which unlike feedforward neural networks, can "use their internal state (memory) to process sequences of inputs that contain memory".[12] This enables it to 'remember' the inputted words, allowing to take into consideration the word order.



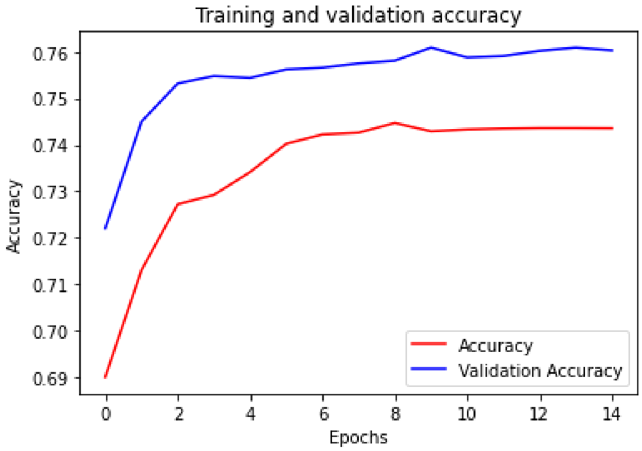
An unrolled recurrent neural network.

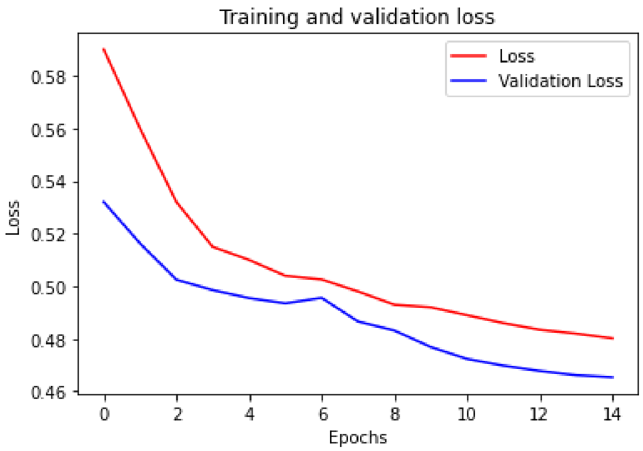
Word Embedding & Overfitting

As a typical LSTM model requires a numerical input, a 'dictionary' of words are rendered in numbers in order to successfully convert letters (alphabets) to numbers for apt reception of the machine. Normally sorting the words in alphabetical order then identifying the first word, the second word, etc. butthis has a problem. For example, the machine would ‘recognize’ the following two words similar, despitethe significant distinction between the definition of the two, because they are in semblance solely in terms of their spelling: ‘cocktail’ (12345) and ‘code’ (12346).

Thus in order to resolve this issue a technique called ‘Word Embedding’ is used. It is essentially figuring out how similar two words are to each other. For example, the word ‘excited’ and ‘thrilled’ are similar to each other in their meanings, thus they are each rendered ‘100’ and ‘101’.

With such a model taken into effect, after having done the ‘training’, a 76.6% of accuracy was shown. The major problem with this was that an ‘overfitting’ was occurring, indicated from the graph diminishing its accuracy as the amount of training increased. An **overfitting** is defined as "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably".[13] The more the machine ‘studies’ a certain given set, the more accurate it gets (virtually immaculate), while when it is given a different (or a ‘new’, ‘unprecedented’) set it does a terrible job in extrapolation. This could be easily pictured when thinking about scatter plots and regressions; while a general trendline should be extrapolated from the holistic analysis of the given data set, as the machine meticulously examines and memorizes every bit of data, it rather creates a regression that perfectly models only the given data set, which is rather a random manifestation, for it disregards the overall tendency of the set.





[12] Aditi Mittal, "Understanding RNN and LSTM," Towards Data Science, last modified October 12, 2019, https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e.
[13] Lexico.

Increasing Accuracy Through Preprocessing & Prevention of Overfitting

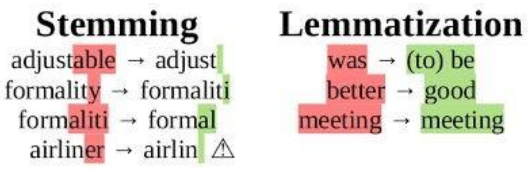
1. **URL, mention (@) :** A lot of information irrelevant to sentiment analysis is included in text data, such as URLs or mentions (@(ID)), which are obstacles to adequate analysis.

```
def RemoveURLMentions(tweet):
     tweet = re.sub(r"http\S+", "", tweet)  # Urls
    tweet = re.sub("(@[A-Za-z0-9_]+)","",
    tweet) # Mentions
    return tweet
```

2. **Stopwords :** Conjunctions, personal pronouns, and prepositions that hinder analysis focused on the meaning of the sentence (e.g. I, our, this, the, etc…).

```
def RemoveStopWords(tweet):
        tweet_words = tokenizer.tokenize(tweet)
        tweet = [w.lower() for w in tweet_words if
        not w in stop_words]
        return tweet
```

3. **Stemming / lemmatization**

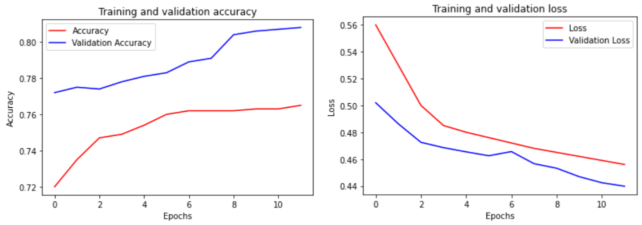


The reason for doing this is that even if the word is the same, there are multiple forms. For example, given 'adjustable' and 'adjust', the computer recognizes the two as completely different words. However, if you change this to the same stem word, to 'adjust', the computer thinks the two are the same (not that the computer understands what this means). This however facilitates the process of recognizing the meaning of the words in the sentence that are not always necessarily in the same explicit form conducive for translation.

```
def Lemmatize(tweet):
        lemmatizer =
        WordNetLemmatizer(
        )processedSentence
        = []
        for word, tag in pos_tag(tweet):
                if tag.startswith('NN'):
                        processedSentence.append(lemmatizer.lem
                        matize(word, 'n'))
                elif tag.startswith('VB'):
                        processedSentence.append(lemmatizer.lem
                        matize(word, 'v'))
                else:
                        processedSentence.append(lemmatizer.lem
                        matize(word, 'a'))
        return processedSentence
```

4. **Dropout**

This is a method that literally drops out (or neglects) some data along the way in order to prevent overfitting. Implementing this method improved 0.5%p of accuracy from the overfitted analysis.



5. **Listing**

The problem was that as colloquial language was the overwhelming majority of the composition of the tweets, it included a plethora of slangs and abbreviations, as well as any other newly created or combined words and deliberate grammatical errors. About 10% of the tweets were significantly influenced during analysis due to this use of informal language. In order to solve this issue, there must be a 'list of words' programmed explicitly

```
def Listing(tweet):
        tweet = re.sub(r"lol", "laugh out loud", tweet)
        tweet = re.sub(r"cuz", "because", tweet)
        tweet  = re.sub(r"he's",  "he is",  tweet) tweet
        = re.sub(r"there's", "there is", tweet)

        ...
        return tweet
```

converting the most used slangs and abbreviations to their actual form that embodies their literal meaning. Reference the following model:

Using this method, 2.5% of the unknown words were successfully interpreted accurately, yielding an overall 81% accuracy of the machine's determination in test data sets, a 4%p increase from the prior model without the explicit list of words created. The more commonly used slangs or abbreviations are added, the more accurate the machine would be in its translation.

6. Conclusion

In order to successfully implement the processing of mass amounts of tweets, a Naive BayesianModel was used initially. However this yielded a critical problem of not considering the word order, resulting in an inaccurate analysis of the tweets. To resolve this issue, an LSTM model was introduced, that uses an RNN to take word order into account. Then to deal with overfitting, the dropout technique was implemented, as well as several preprocessing methods such as stemming and lemmatization. Withremoving numerous obstacles to accurate analysis of the tweets such as frequent usage of slangs, that is resolved by making a list of the most-used words, stopwords such as prepositions, and hyperlinks and mentions, an 81% of accuracy was yielded with the final model.

If an actual social media company could adapt this model and augment it with additional technical developments, it could largely help those with psychological illnesses or disorders and are in desperate need of mental counseling, which significantly contributes to the society and the greater good of people's well-being and happiness.

Bibliography

- Beel, Joeran. "Research-Paper Recommender Systems: A Literature Survey." Intelligent Systems Group. https://isg.beel.org/pubs/2016%20IJDL%20--%20Research%20Paper%20Recommender%20Systems%20--%20A%20Literature%20Survey%20(preprint) .pdf.

- Burnham, Kenneth P. "Model Selection and Multimodel Inference." Springer. Last modified 2002. https://www.springer.com/gp/book/9780387953649.

- Expert System Team. "What is Machine Learning? A Definition." Expert.ai. Last modified May 6, 2020. https://www.expert.ai/blog/machine-learningdefinition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves.

- Gottschalk, Louis A. "The Measurement of Psychological States Through the Content Analysis of Verbal Behavior." Google Books. Last modified January 1, 1979. https://books.google.com/books/about/The_Measurement_of_Psychological_States.html?id=BXUrAM-e4Z4C.

- Hawkins, Douglas M. "The Problem of Overfitting." National Library of Medicine. Last modified February 2004. https://pubmed.ncbi.nlm.nih.gov/14741005/.

- Jones, K. Sparck. "A statistical interpretation of term specificity and its application in retrieval." Journal of Documentation. Last modified 1972. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.8343&rep=rep1&type=pdf.

- Leskovec, Jure. "Mining of Massive Datasets." Stanford. Last modified 2014. http://infolab.stanford.edu/~ullman/mmds/book.pdf.

- Lexico. https://www.lexico.com/en/definition/overfitting.

- Mittal, Aditi. "Understanding RNN and LSTM." Towards Data Science. Last modified October 12, 2019. https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

- MonkeyLearn. "Everything There Is to Know about Sentiment Analysis." MonkeyLearn. https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20(or%20opinion%20mining,feedback%2C%20and%20understand%20customer%20needs .

- Park, Minsu, Chiyoung Cha, and Meeyoung Cha. "Depressive Moods of Users Portrayed in Twitter." NYU Scholars.
- Last modified 2012.
- https://nyuscholars.nyu.edu/en/publications/depressive-moods-of-users-portrayed-in-twitter .

- Shewan, Dan. "10 Companies Using Machine Learning in Cool Ways." WordStream. Last modified August 12, 2019. https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications .

- Stewartschultz. Last modified 2010. http://www.stewartschultz.com/statistics/books/Cambridge%20Dictionary%20Statistics%204th.pdf .

- Stone, P., and D. Dunphy. "The General Inquirer: A Computer Approach to Content Analysis." BibSonomy. Last modified 1966. https://www.bibsonomy.org/bibtex/2cca1be66f84fee5a75c3f3afb95ae943/pdturney .

- Weir, Kirsten. "Worrying Trends in U.S. Suicide Rates." American Psychological Association. Last modified March 2019. https://www.apa.org/monitor/2019/03/trends-suicide .