

**The Story of Socrates**  
*Jay Havaladar*

# 1 Part 1: How Can A Computer Read Your Mood?

So let's start with a simple question: can a computer read your mood? This is related to the essential problem of "sentiment analysis":

**sentiment analysis:** the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic is positive, negative, or neutral.

There are a lot of basic problems here. Let me start with a couple of the most basic questions you might want to ask to get a good picture of how to solve this type of problem. - How can we reasonably grade an opinion as positive, negative, or neutral? Can we define a metric to compute opinions so that the value of the metric has some easily interpretable meaning? - How do we identify topics from the data? - Finally, how do we extract our earlier measure of opinion from the data efficiently?

There's a lot of ways of going about solving this problem. For now, we'll start with the simplest approach and build things up from there.

One way we could analyze data is to treat a sentence as simply a bag of words, without regard to the sequence the words appear in -- these are called the **unigrams** of the document. Similarly, we could consider pairs of words, called **bigrams**. For example, "I walked the dog" becomes the phrases "I walked", "walked the", and "the dog". It hopefully isn't hard to see here that number of bigrams and the number of unigrams differ only by one. Similarly, a  $k$  gram has  $n - k + 1$  elements. Now, this kind of clues us into how much work we should do in order to extract features from this space. We know from combinatorics that:

$$1 + 2 + \dots + k = \frac{k(k-1)}{2} = O(k^2)$$

Now in order to be able to compare two sentences how many  $k$ -grams should we consider? 1-grams gives us the words (likely to have a lot of overlap), the pairs of words (still likely), and the triples of words. Let's pick a dataset -- I used a bunch of Reddit comments.