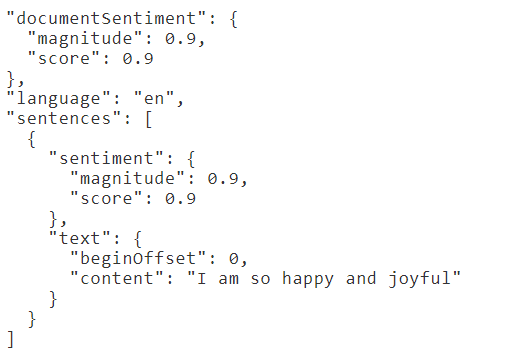


Use Google Cloud as platform to conduct sentiment analysis test.

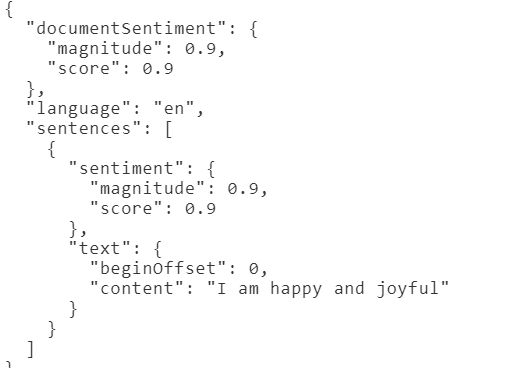
I use language\_v1 package and the main function used is analyze\_sentiment.

**Test results:**

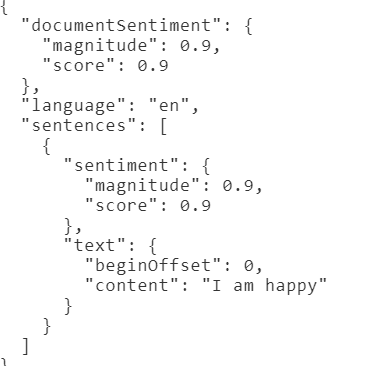
1. When input text is “I am so happy and joyful”, the magnitude is 0.9, and the score is 0.9, which is a relatively high score.



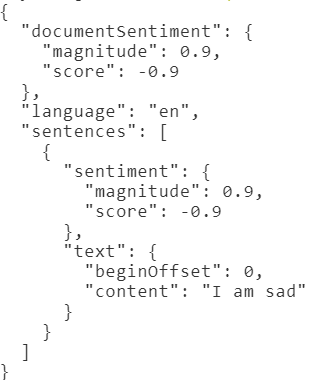
2. When input text is “I am happy and joyful”, the magnitude is 0.9, and the score is 0.9, which is a relatively high score. There is no difference with the previous one. The word “so” will not increase the score.



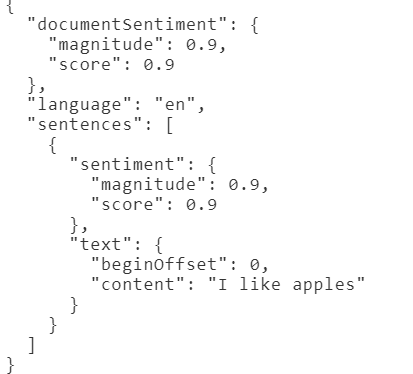
3. When input text is “I am happy”, the magnitude is 0.9, and the score is 0.9, which is a relatively high score. There is no difference with the previous one. The additional word “joyful” will not help to increase the score if there already exists a positive word, such as ”happy”.



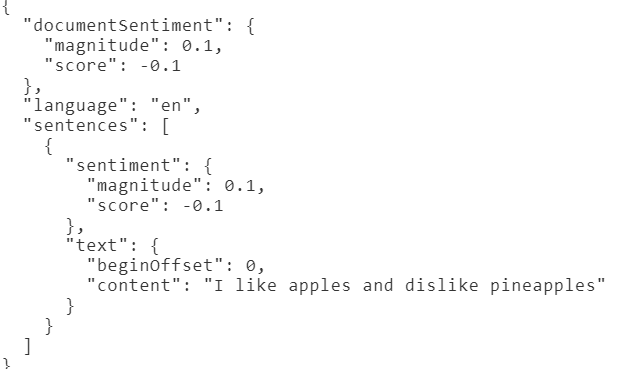
4. When input text is “I am sad”, the magnitude is 0.9, and the score is -0.9, which is a relatively low score.



5. When input text is “I like apples”, the magnitude is 0.9, and the score is 0.9, which is a relatively high score. Although there is no emotion express in the text, the word like indicates a ”high” reward to the system.



6. When input text is “I like apples and dislike pineapples”, the magnitude is 0.1, and the score is -0.1. Low magnitude means no sure about the score. It makes sense because in a single sentence I put the word “like” and “dislike”. This simple test may reveal that the algorithm use a addition-like calculation, to add 0.9 with -0.9 and get a balance one score. However, the result here is not exactly 0. The reason is to be explored.



7. When input text is “a cat”, the magnitude is 0.1, and the score is 0.1. This text should have no positive or negative sentiment, so the score is close to 0. But similarly, it is not exactly 0 but 0.1.

