Named Entity Recognition

One of the most important NLP tasks is entity recognition, which consists of locating named entities in unstructured text. For example, in the sentence

"Apple is a great company",

a good NLP engine for financial purposes should recognize that the token *Apple* is an organization (or a similar class depending on the library), and should associate the token with some ID or that organization (e.g., https://www.wikidata.org/wiki/Q312). In this open exercise, you will write a simple code that looks for two specific entities: gold and silver ONLY when they are referred to as a tradable commodity. You are highly recommended to write the code in Python, preferably in Jupyter Notebook. Besides the code itself, we are even more interested in your reasoning. Hence, use markdown to explain your choices and considerations. If you are unfamiliar with NLP or programming languages, you can still outline how you would develop this task at an architectural level.

1) A text file with about one thousand news pieces and social media comments (one article per line) is provided. Load this file into your code and have a look at a sample of such articles.

2) Write a simple string matcher function that takes an article as an argument and checks whether the strings "gold" and "silver" are found in that article. Check how many articles contain each entity. Can you see any issues with this simple approach?

3) Not everything that shines is gold (or silver). You may have noticed that simply checking for those strings could lead to many false positives. For example, "silver lining" almost certainly does not refer to silver as a commodity. Can you find other such misleading instances? Adapt your code such that it avoids false positives. Finally, can you think of ways in which your code would be generalizable out-of-sample? In other words, are there some commonalities in the false positives you first found?

4) In the previous item, you were concerned with false positives. In this item, you will be looking at false negatives. That is, there are many ways to refer to gold or silver other than their English names. In other words, there are several other **aliases** that can be used to refer to gold or silver. For example, the **XAU/USD** alias will almost certainly refer to the gold price per troy ounce in US Dollars and it is commonly used when talking about price events. Thus, even if the text does not mention the strings "gold" or "silver", the two metals may still be mentioned in the text (hence your previous code yields false negatives). Can you adapt your code such that it also looks for strings other than gold or silver (while also considering the above point on false positives)?

Sentiment Analysis

In the first part of this task, you wrote a code that tries to recognize references to gold or silver when mentioned as a commodity. However, one may also want to analyze what the articles are trying to convey about those entities. For example, whether the articles are saying something positive, neutral or negative about the entity. This is one of the definitions of sentiment analysis. In the financial markets, positiveness is usually associated with being "bullish" (expecting prices to go up), while negativeness is associated with "bearishness" (expecting prices to go down). Write a simple script to gauge the sentiment around the mentions of gold and silver. You can do it by creating two dictionaries containing words that are usually contextualized as positive or negative. For example, "rise" can be a positive word: "Gold prices rise due to inflation risk". Apply your dictionaries to estimate how positive (negative) the set of provided articles is. Write about the limitations of this approach.