Hw1: A-prior algorithm

Course: Massive Data Analytics

Professor: Dr. Iman Gholampoor

Mohammad Mehdi Zare

* **Introduction**:

In this assignment, I analyze Farsi news. This Dataset contains Persian news in JSON file. Some important features are **body**, **uid**, **title**, **URL**, and **keywords**. I implement some methods to clean data, count words, plot news timelines, plot word clouds, and A\_prior algorithm to count pair and three-set words. In the last section, find a way to understand the importance of a word by **tf-idf.**

* **Section 1: (Dataset preparation)**

In this section, considering I work on Colab, it was necessary to unzip the dataset and then read the data.

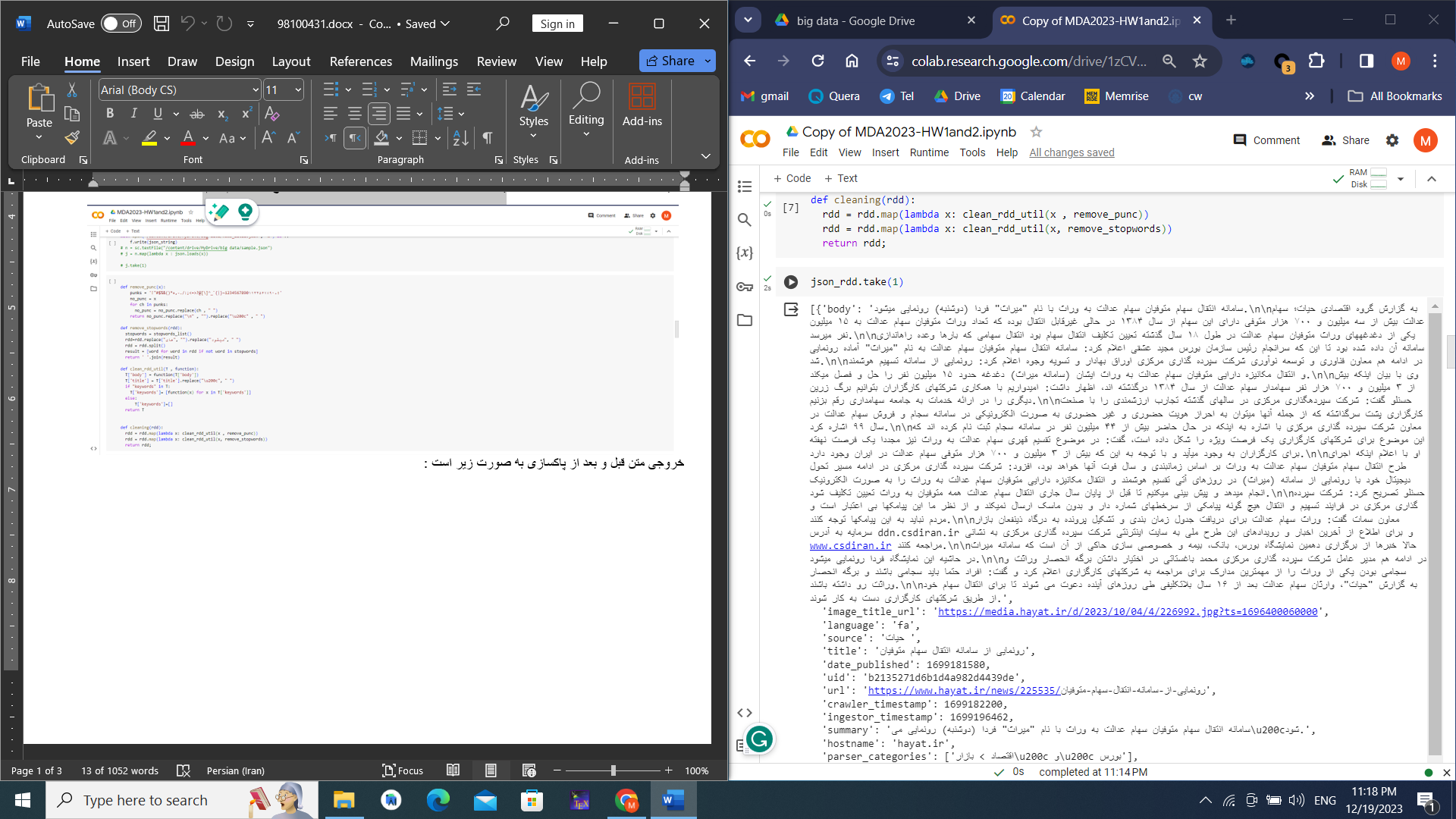
* **Section 2:(** **Preprocessing)**

In this section, I had to load the JSON file and prepare it for subsequent processing. The data includes numerous unnecessary characters and words (stopwords) that needed to be removed to avoid unnecessary loading or miscounting in our processing. For example, common words such as "هست", "از", "در", etc., are frequently occurring words that, if not removed, would be counted as high-frequency words. However, we know that these letters are very common in all texts and do not add specific information.

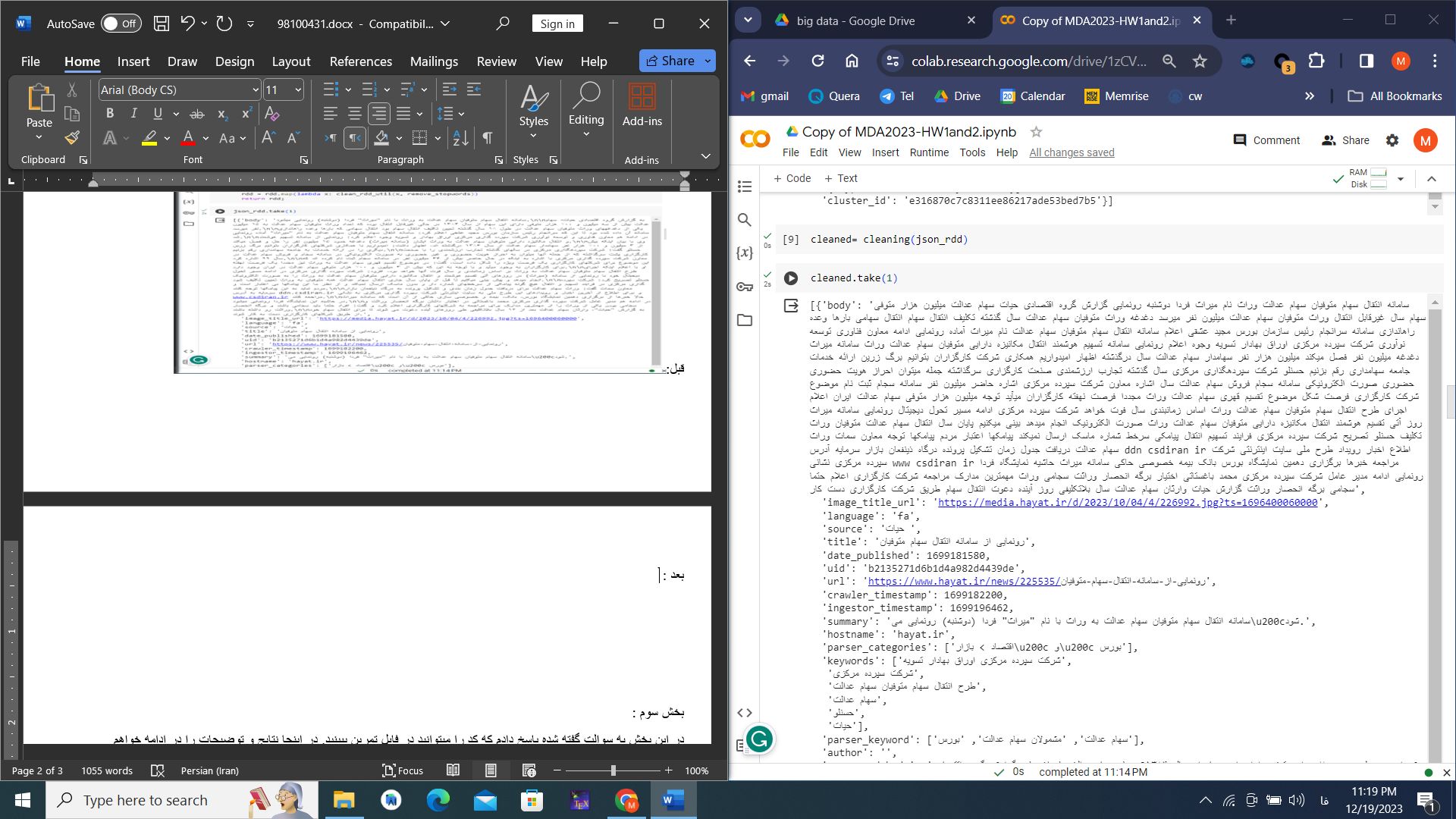
Additionally, prepositions and punctuation marks are also repetitive and need to be removed.

Below you can see the **body** before and after cleaning:

* + Before:

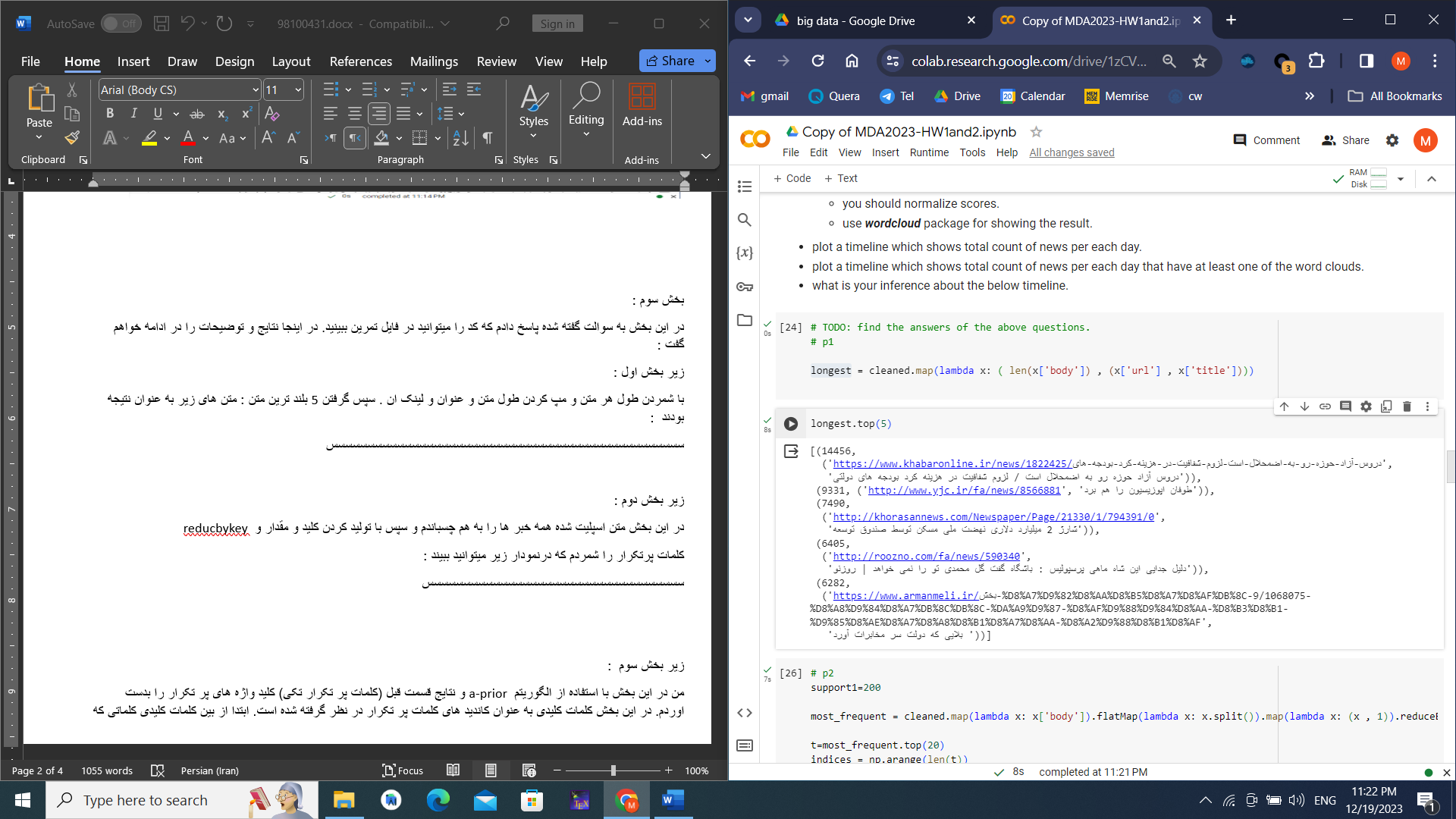
: 

* + After:



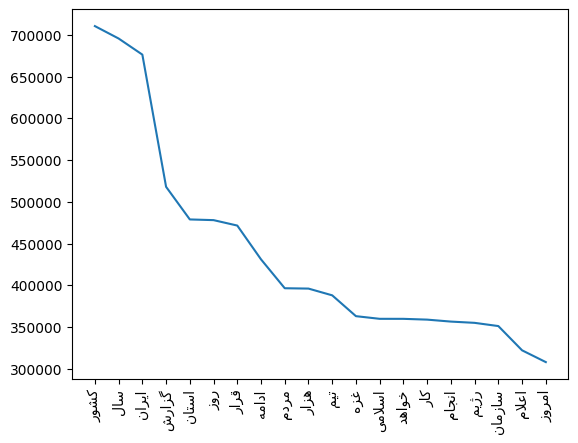
* **Section 3: Exploration**
  + **what are the titles and URLs of the 5 longest news?**

In this part, I use RDD API to map news into the length of the news and URL and title, then sort and print 5 first. Result is :



* + **what are the 20 most frequent words? (Plot the distribution of these words)**

In this part, I count word frequency in all news with key-value approach. The most frequent words are (support= 50000):

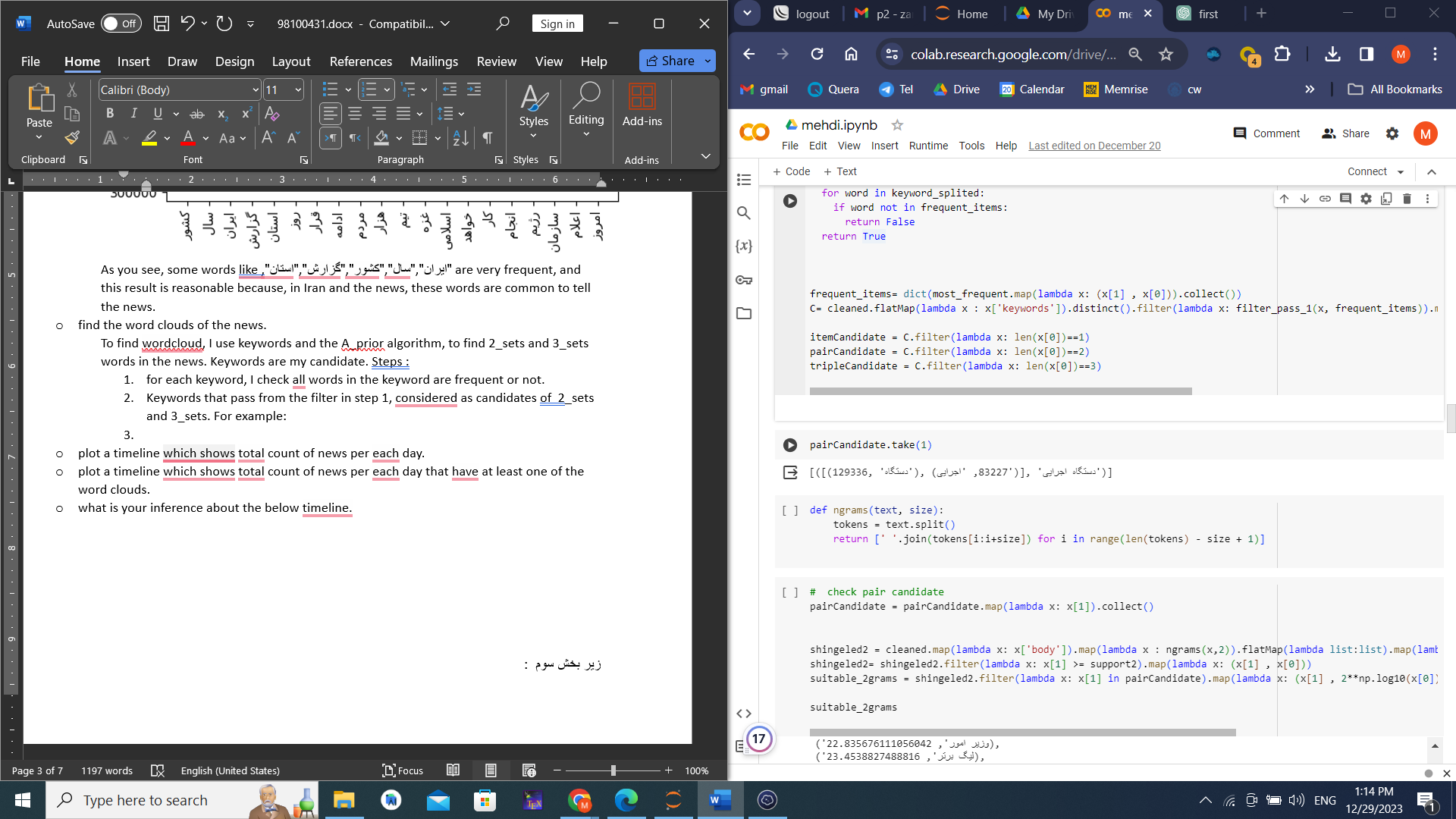


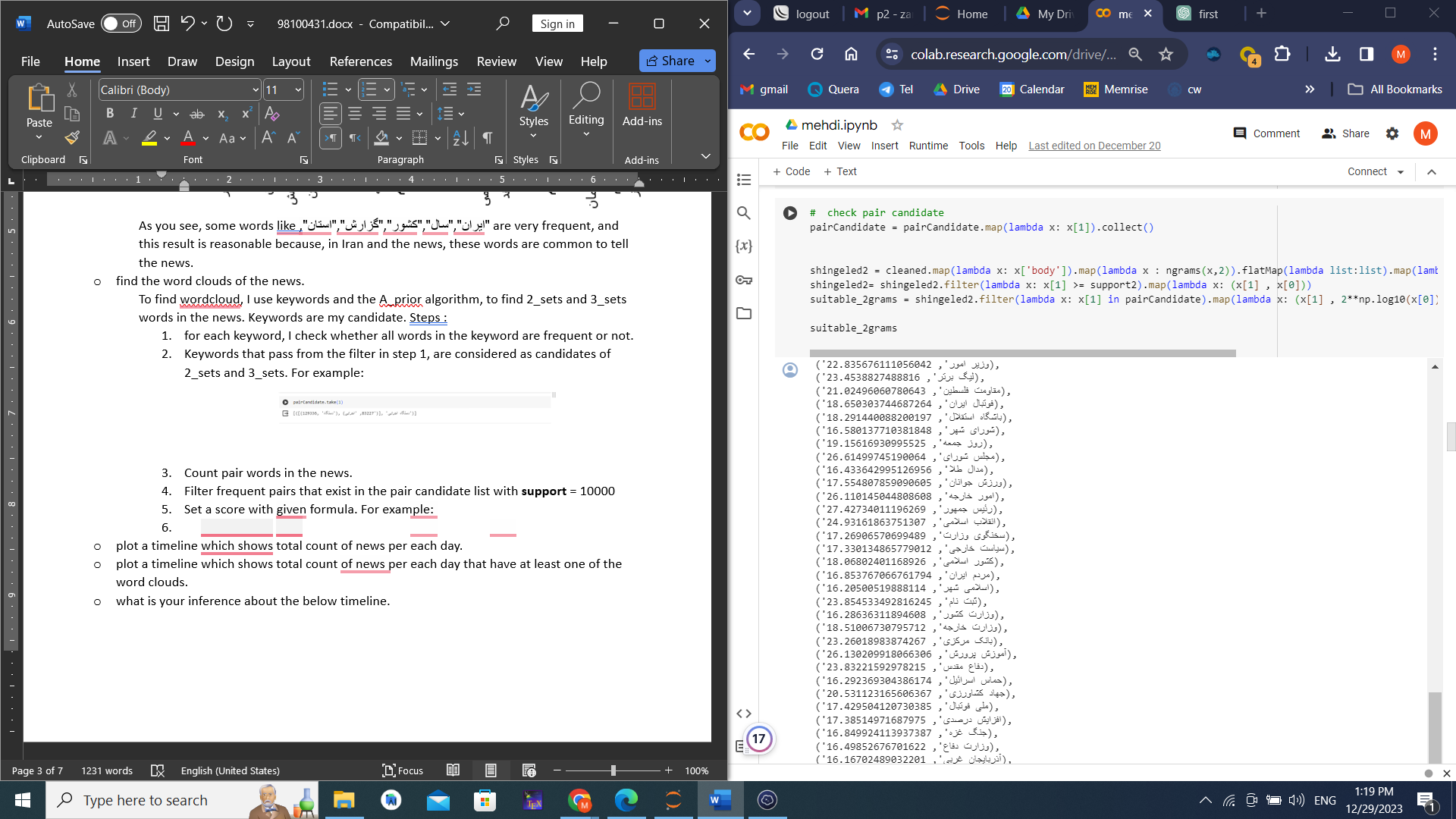
As you see, some words like, "استان", "گزارش", "کشور", "سال", "ایران" are very frequent, and this result is reasonable because, in Iran and the news, these words are common to tell the news.

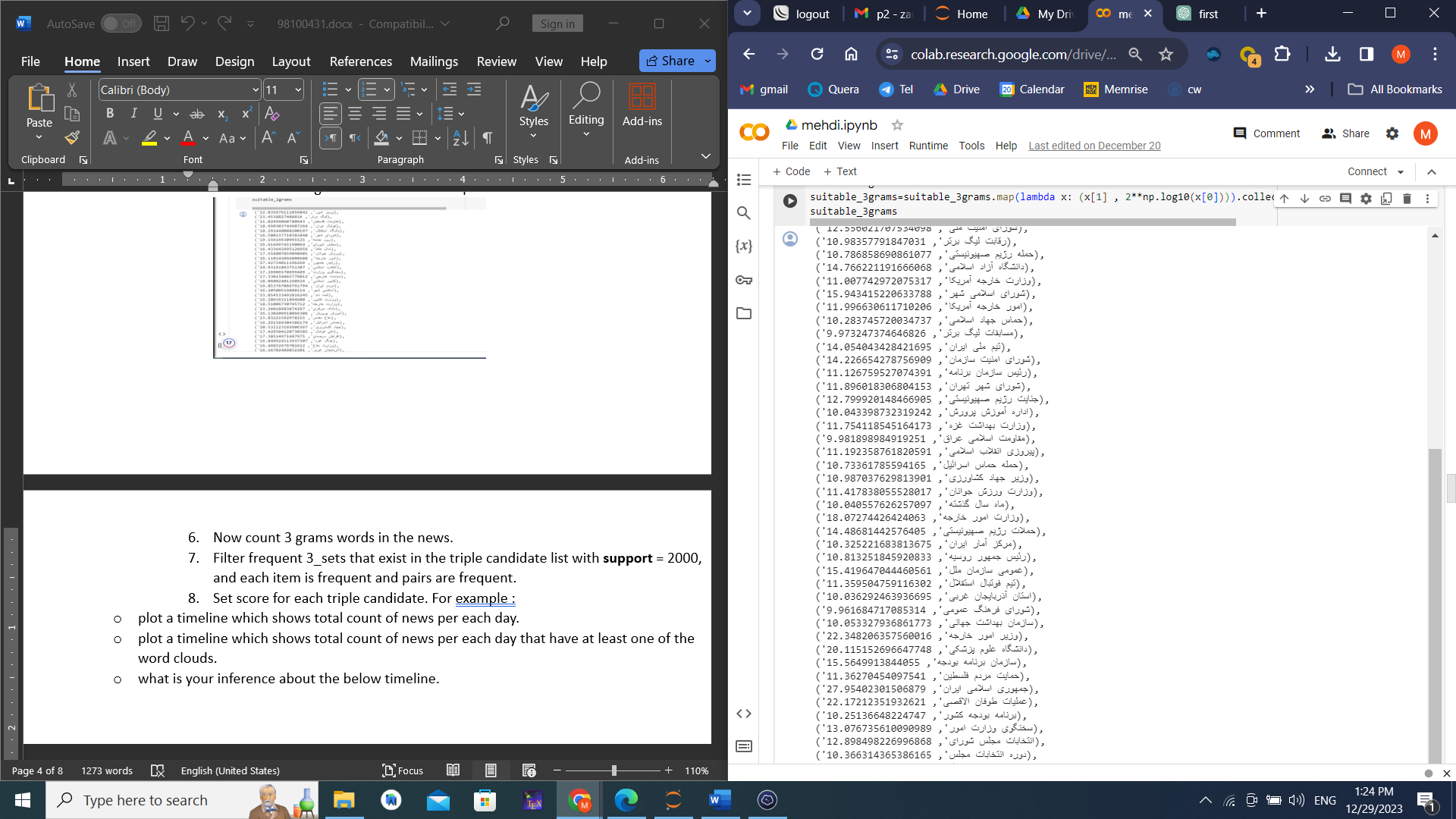
* + **find the word clouds of the news.**

To find wordcloud, I use keywords and the A\_prior algorithm, to find 2\_sets and 3\_sets words in the news. Keywords are my candidate. Steps:

1. for each keyword, I check whether all items in the keyword are frequent or not.
2. Keywords that pass from the filter in step 1, are considered as candidates of 2\_sets and 3\_sets. For example:



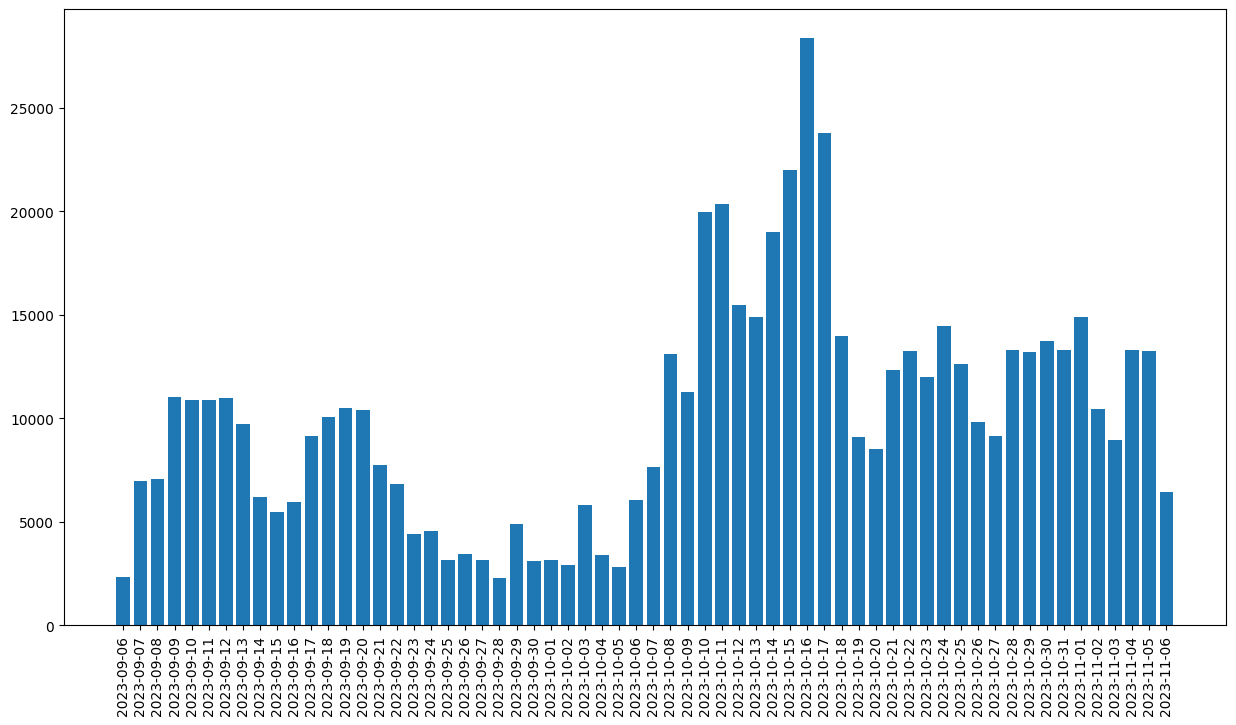
1. Count pair words in the news.
2. Filter frequent pairs that exist in the pair candidate list with **support** = 10000
3. Set a score with the given formula. For example: 
4. Now count 3 grams of words in the news.
5. Filter frequent 3\_sets that exist in the triple candidate list with **support** = 2000, and each item is frequent and pairs are frequent.
6. Set a score for each triple candidate. For example :



1. Finally, Normalize scores and plot wordcloud with libraries.

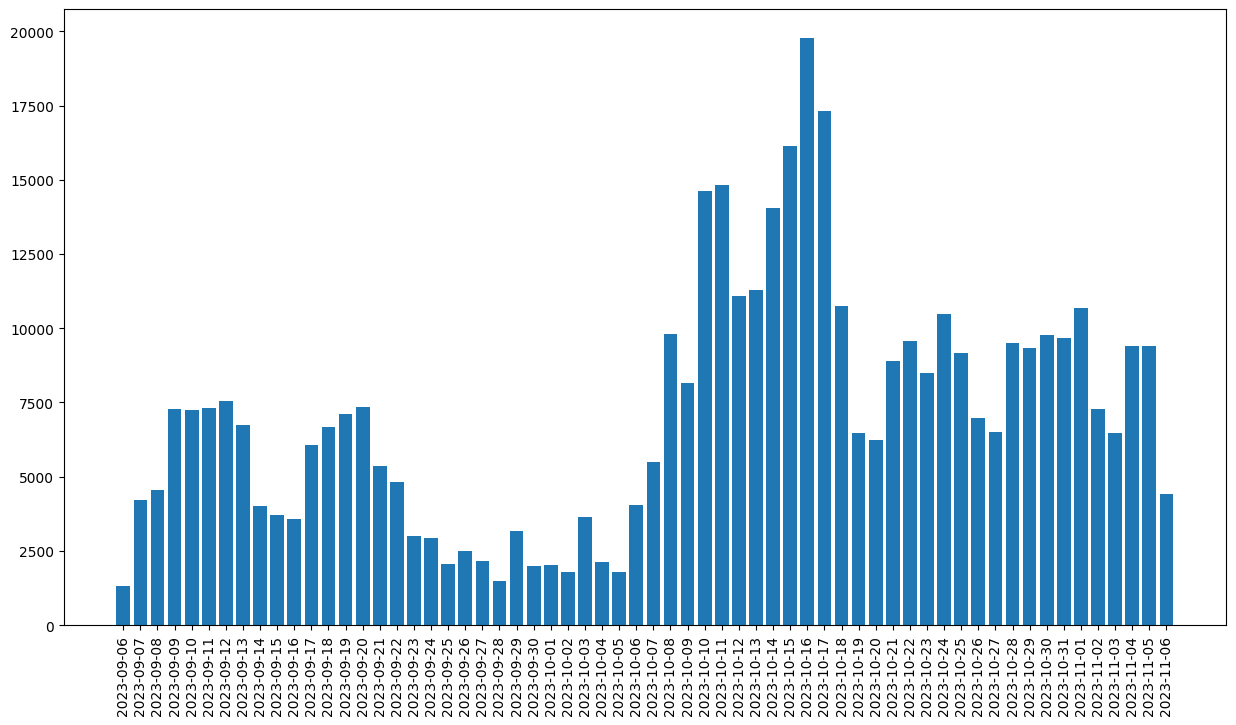


* + **plot a timeline that shows the total count of news per day.**



Considering the waves present in the chart, it can be inferred that on some days, specific events have occurred, leading to the generation and publication of more news.

* + **plot a timeline that shows the total count of news per day that has at least one of the word clouds.**

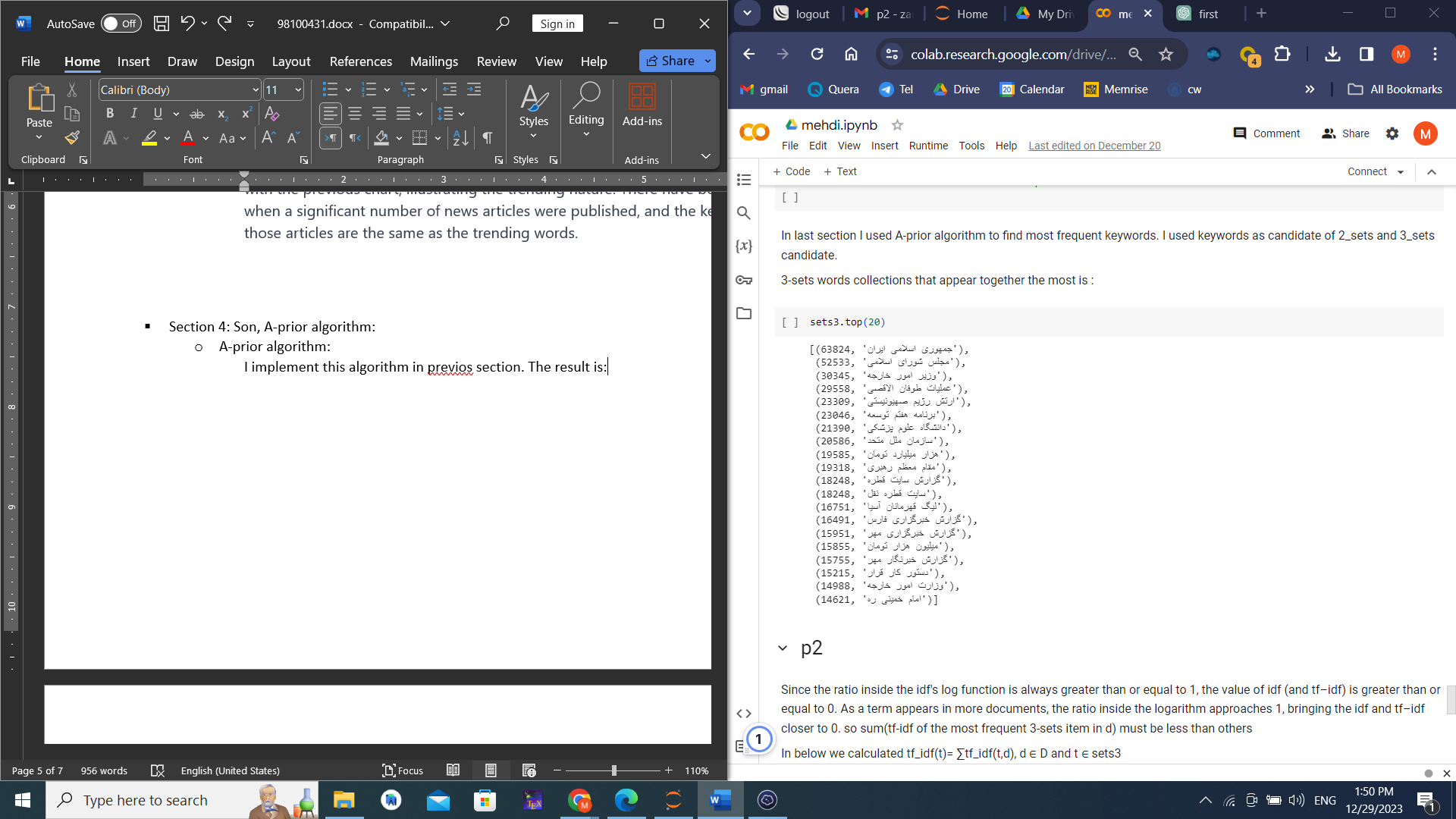


* + **what is your inference about the below timeline?**

In my opinion, on certain dates when specific events occur, repetitive keywords become prominent, indicating the trending of a particular news or incident reflected in the news. This leads to fluctuations in the chart. In other words, on certain days, news containing highly repeated keywords gradually increases, then decreases, and later, due to another event, different words become prevalent, resulting in their quantity increasing and decreasing. The pattern is consistent with the previous chart, illustrating the trending nature. There have been days when a significant number of news articles were published, and the keywords in those articles are the same as the trending words.

* **Section 4: Son, A-prior algorithm:** 
  + A-prior algorithm:

I implement this algorithm in previous section. The result is:



* + Validate model:

The frequently repeated words are typically meaningful and have high occurrences across various texts, as well as within a single text. To evaluate the validity of the identified trigrams, I use the tf-idf metric. This metric indicates the significance of a word; if it is close to one, it implies the word is highly important and hasn't been overly spammed. In other words, it exists in various texts and is also repeated frequently. Therefore, the tf-idf score is likely high or close to one for the identified trigrams. Note that in the calculation of idf, as the occurrence of a word in more texts increases, the logarithm approaches one, and the logarithm value gets closer to zero, resulting in a lower value. This signifies that the expression is common in the language and doesn't provide specific information.

The algorithm for calculation: The tf-idf value for a word-text pair is computable. By summing these values across different texts a single word (), we obtain a score for that word. The higher the score, the more important the word is. To test the algorithm, I calculate this value for 10 frequently repeated words and 5 other random words to examine my hypothesis. The results are as follows:

