Computer aided simulations and performance evaluation

Lab 8 - Fingerprinting for dictionary

Gabriele Cuni 277957

8.1 Input Parameters

- **seed**: It is used to set the random library seed, which is used to shuffle two times the list of all the words.
- **words**: It is a python list of 370101 english words coming from the words_dictionary.json [1]

8.2 Output Parameters

- **bExpMin:** It is the minimum number of bytes of the fingerprint for which there are no collisions in the **fingerprintSet** given the input list of words.
- **bTeo:** It is the theoretical number of bytes of the fingerprint for which there are no collisions in the **fingerprintSet** given the input list of words.
- **pFalsePositive:** It is the probability of false positives given the number of words and the fingerprintSet range.
- **fingerprintSetBytesSize:** It is the data structure size of **fingerprintSet** computed by means of pympler [2].
- **wordsSetBytesSize:** It is the data structure size of **wordsSet** computed by means of pympler [2].

8.3 Main Data Structures

- **experiments_result:** it is the python dictionary which contains all the simulation results.
- **fingerprintSet:** It is the python set used to store all the fingerprints and it is used to assess if a collision occurs.
- **wordsSet:** It is the python set of all words used to be compared with **fingerprintSet**. **wordsSet** is used, instead of **words** (which is a list), because given the same content, python sets take up more memory space than python lists.
- **bExp:** it is used to find the **bExpMin**. It is an incremental variable.
- **collision:** It is a flag variable used to assess if a collision is occured.

8.4 Formulas

- $bTeo = \log_2(\frac{m}{\epsilon})$ where m is the number of words and ϵ is 0.5
- $P("False\ positive") = 1 - (1 - \frac{1}{n})^m$ where m is the number of words and n is $2^{bExpMin}$
- $m \cdot b$ where m is the number of words and b is **bTeo**. It is computed inside plot8.py by plotting

8.5 Python Functions

- **pympler.asizeof.asizeof():** It is used to assess the memory occupancy in bytes of **fingerprintSet** and **wordsSet**. [2]
- **random.shuffle():** It is used two times to shuffle the words inside the list **words,**in order to make them not in the alphabetic order.
- **hashlib.md5():** It is used to obtain the hexadecimal md5 string.

8.6 Charts and Reasoning

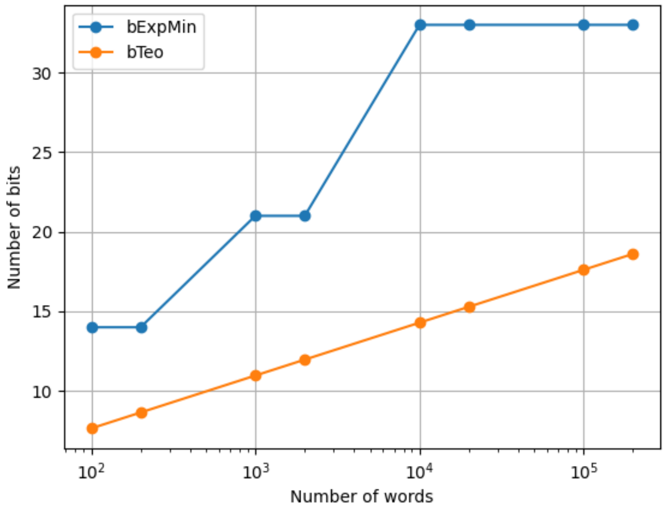

Figure 1 shows that **bTeo** is about half of **bExpMin** which it seems right given that **bTeo** is computed with $\epsilon = 0.5$, instead **bExpMin** is simulated as if $\varepsilon \to 0$.
The experiments are done with [100, 200, 1000, 2000, 10000, 20000, 100000, 200000] words.

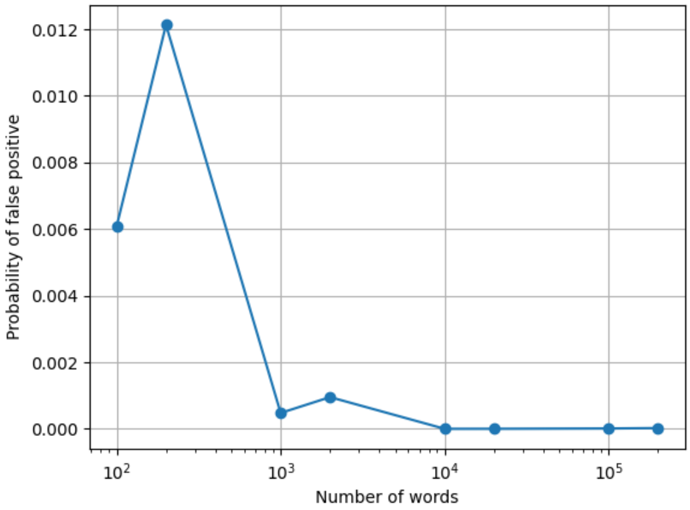

Figure 2 shows the probability of collision computed as shown in section 8.4 v.s. the number of words as before. As shown the probability of collision tends to zero by increasing the number of words, because by doing so, the fingerprint range increases a lot faster than the number of words and the wasted space is higher.

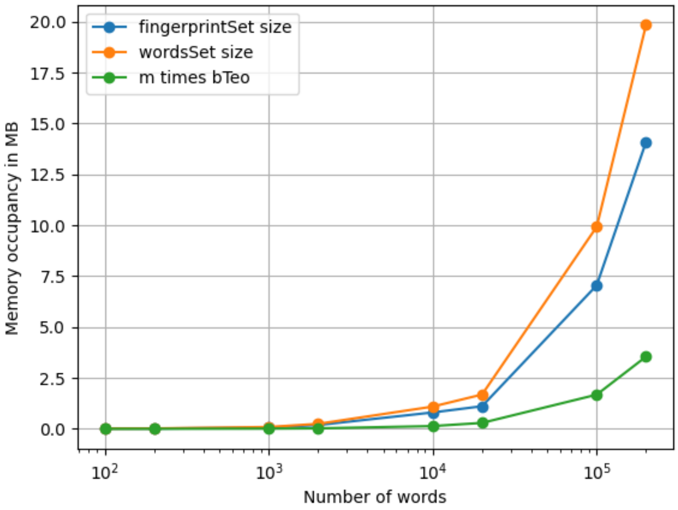

Figure 3 shows a comparison between the **fingerprintSet**, the **wordsSet** memory occupancy and the theoretical occupancy given by $m \cdot b$.
As can be shown it is worth using the fingerprint also to save memory.
The memory space of the python sets is much larger than the theoretical one because the python set object stores more data other than its content.

The important thing to notice is that all follow the same exponential trend given the number of words.

8.7 References

[1] https://github.com/dwyl/english-words
[2] https://pympler.readthedocs.io/en/latest/