**Experiment**

In the third experiment, the hypothesis is that the time taken by the probabilistic search algorithm is inversely proportional to the exactitude (an input parameter). Also, we intend to demonstrate that the exactitude used actually affects the results of the search. In both hypotheses, the exactitude is the independent variable, which implies that the information needed for both of them can be obtained by performing a single experiment instead of two different tests.

For the first hypothesis, the time will be measured in milliseconds. For the second hypothesis, the correctness of the results will be measured as the average of the cosine similarity between each of the documents and the query (which is also a document). This means that the correctness will be a number between 0 and 1. However, for additional precision, three decimals of this number will be used.

Four queries will be used: “apple red”, “blue car”, “bad good”, “confused president”. The IDF of each word included in those queries is different. The idea is to use a good sample of queries so that the results will not be biased. The IDF numbers of each word are these:

* “apple”: 156
* “red”: 585
* “blue”: 613
* “car”: 913
* “bad”: 2316
* “good”: 17007
* “confused”: 123
* “president”: 33

The frequency of each query is different. In descending order from most frequent to least frequent: “bad good”, “blue car”, “apple red” and “confused president”.

Every combination of query and exactitude percentage was tested twenty times, and the average time and average similarity were used as the data.

**Results**

As expected, the time taken by the search algorithm decreases as the exactitude sought by the user increases. This can be seen clearly in the graphic. However, the difference between going through all of the documents and only ten percent of the documents is not as much as one would expect. This is likely caused by the algorithm that chooses which documents will be analyzed. Such algorithm makes sure that no document appears twice in the list of documents to be sought. Probably, this operations takes some time to compute, but the time it takes should not grow much more with a larger corpus (this one has only 100000 documents). With a larger corpus, the difference between 100% exactitude and 10% exactitude should be more noticeable.

As for the similarity, it can be seen that better results are achieved with more exactitude. However, in some cases, the similarity remains unchanged: it cannot be improved when the average case already yields the best possible result (0.999 similarity). This corresponds with the second hypothesis.

It should be noted that even with low exactitude percentages, the best document found always had a similarity of 0.900 or more, which shows that even with low exactitudes

**Experiment 2**

Similar to the first experiment, the second experiment consists in the comparison of the empirical time taken by the search algorithm versus the theoretical time as it was calculated in the “Methodology” section. Some tests were performed with different number of documents and a graph was drawn to represent the correlation between the number of documents and the total time taken by the algorithm when it came the moment to make a query.

**Results 2**

Concerning the second experiment, the results were satisfactory and fitted the previously calculated time complexity of the search. It can be seen that the behavior of the function that describes the empirical time is the same as that of a linear function. It might not seem clear because of the spikes that are present in the graphic, but the time taken by the algorithm is directly proportional to the number of documents that are being searched.

Not much more is left to be said. While the behavior was predictable and the time complexity is O(n\* log(n)) (because of the sorting of the results), the algorithm is still not as fast as it could possibly be, but it probably has much more to do with implementation details.