Universidade de Aveiro

Departamento de Eletrónica, Comunicações e Informática



Algorithmic Information Theory (2023/24)

Lab work nº 2

27 Mar 2024

Rafael Pinto, nº 103379

Ricardo Antunes, nº 98275

Pompeu Costa, nº 103294

Contents

[Introduction 3](#_Toc163987233)

[Implementation 4](#_Toc163987234)

[Results 4](#_Toc163987235)

[Different Configurations 4](#_Toc163987236)

[Model Usage 4](#_Toc163987237)

# Introduction

This report addresses an innovative approach to determine if a text has been rewritten by ChatGPT. Instead of using the traditional feature extraction, we adopted the algorithmic information theory. We used data compression algorithms to measure the similarity between texts, eliminating the need for a separate feature extraction step. For this purpose, we implemented the "was chatted" program, which calculates the estimated number of bits required to compress the text under analysis using finite-context models. These models will be trained based on the provided texts representing both texts rewritten by ChatGPT and texts written by humans. The program will build finite-context models for each class and use these models to estimate the number of bits required to compress the text under analysis. Based on this estimate, the program determines the class to which the text belongs. The report covers all stages of the work, from the development of the models to the analysis of the results obtained. Several tests will be carried out to evaluate the effectiveness of the program in classifying texts, determining whether they were rewritten by ChatGPT or not. Additionally, we will investigate the factors that may influence this effectiveness.

# Implementation

# Results

In this section, we will present the results obtained from the tests performed with the cpm and mutate program and conduct an analysis of them. Our objectives are to observe the effect that input parameters have on the copy model performance, how our algorithm alternate between each model and how the estimated bits are affected when there is more randomness in a file. Additionally, we will also compare how general-purpose compressors, such as gzip, bzip2, and zstd, behave with different levels of randomness in a file.

## Different Configurations

We tested our implementation of the copy model algorithm using different parameters combinations for the file chry.txt. In this experiment, the window size ranged from 3 to 15, the threshold ranged from 0.3 to 0.9 and the alpha was 1. In Figure 1, we selected some of the results obtained so that it is possible to understand the effect of each parameter.

The bigger the threshold, fewer bits are estimated. This can be because the algorithm will prioritize the use of the copy model rather than the fallback model. The window size results are not that linear, it cannot be too small nor too large. In our tests, the window size that obtained the best results was 10.

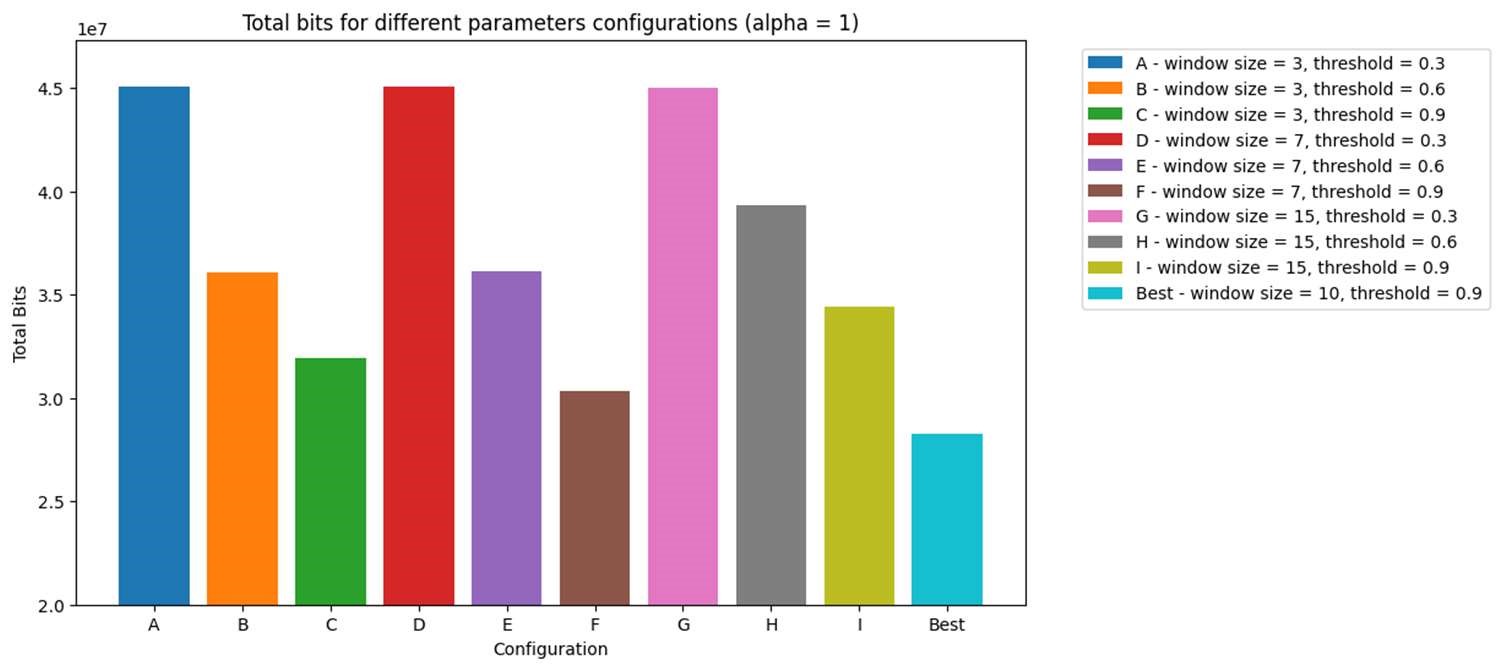


Figure 1 Comparison of total bits estimated for different parameters configurations

## Model Usage

To test the how each model was being used, it was created the example file test\_fallback.txt. We ran our cpm program with the default values (alpha = 1, window size = 10 and threshold = 0.9) and registered what model was used in each iteration. Shown in Figure 2, in the beginning, when there is not much knowledge, the fallback is the model used, but after a pattern is identified the most used model is the repeat model as expected.

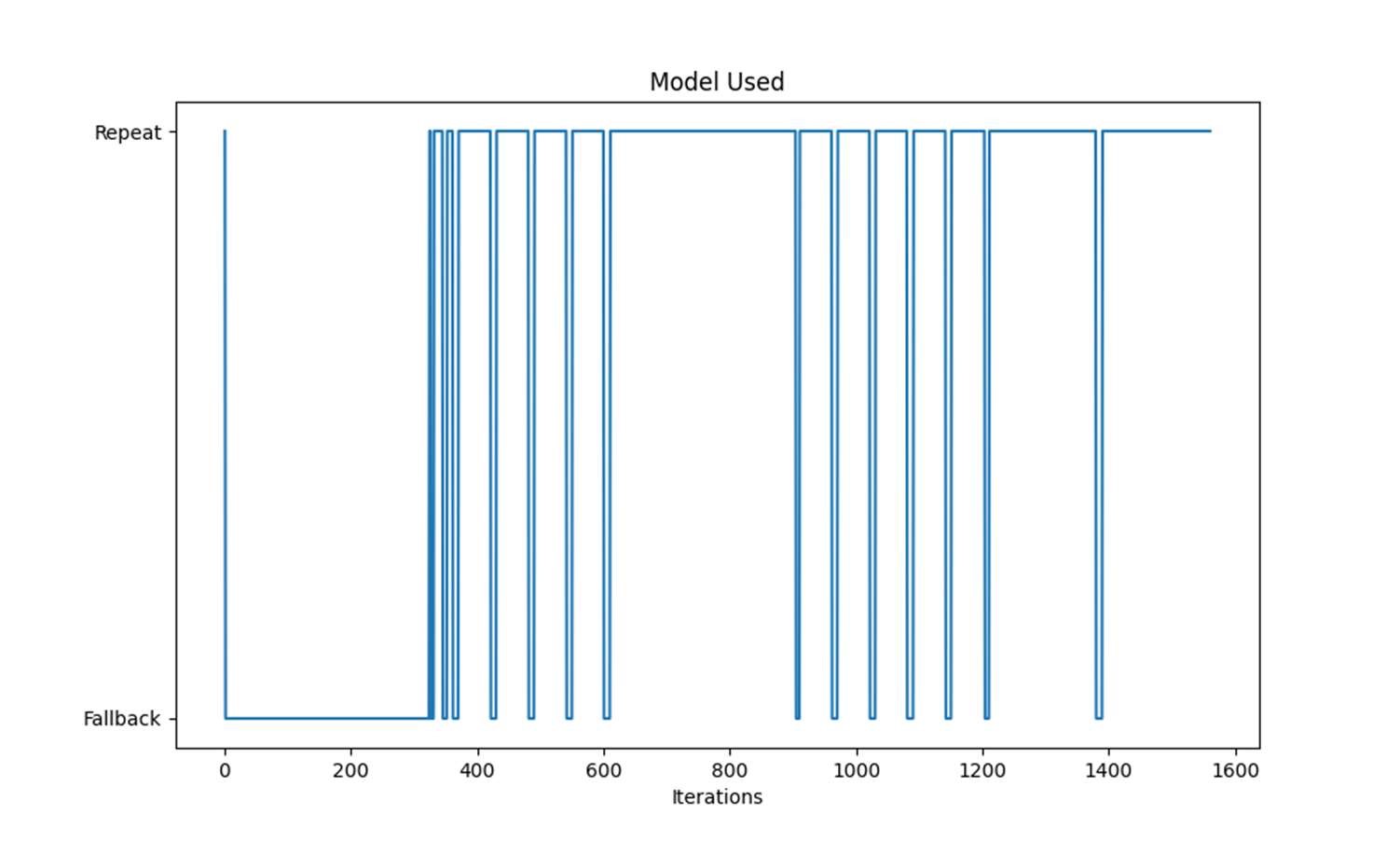


Figure 2 Model used at each iteration for test\_fallback.txt file