universidade
de aveiro

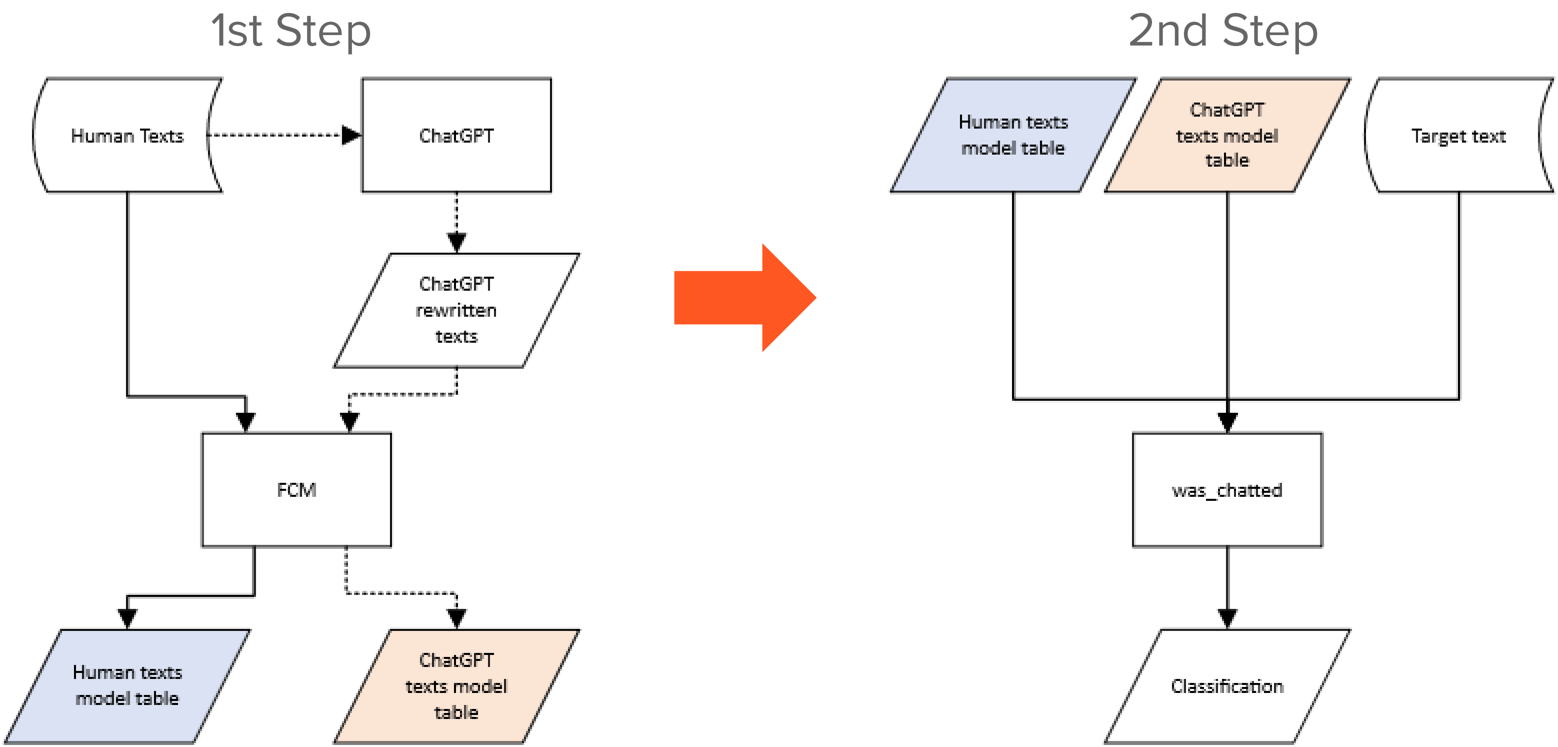
Classification using Data Compression

Gonçalo Machado | 98359
David Raposo | 93395
Catarina Marques | 81382
Bruno Nunes | 80614

Grupo 1

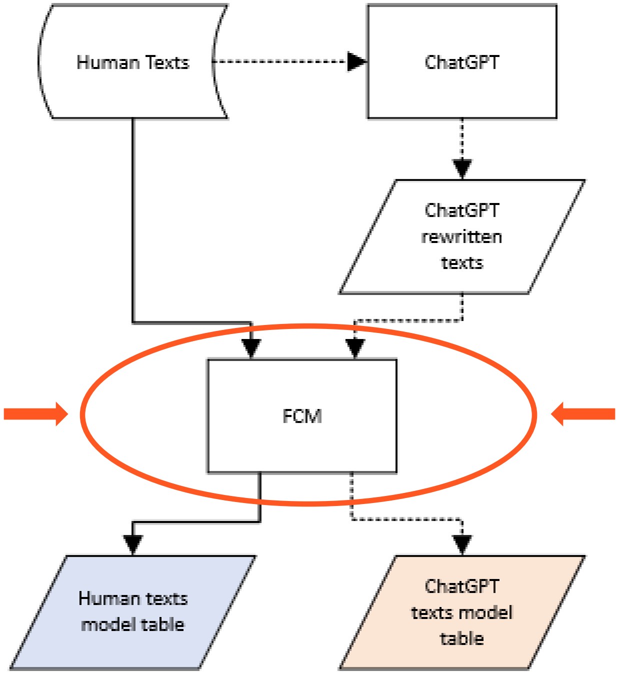
Teoria Algorítmica da Informação, 2023/2024

Methodology – Initial Decision



Methodology (FCM)

1st Step



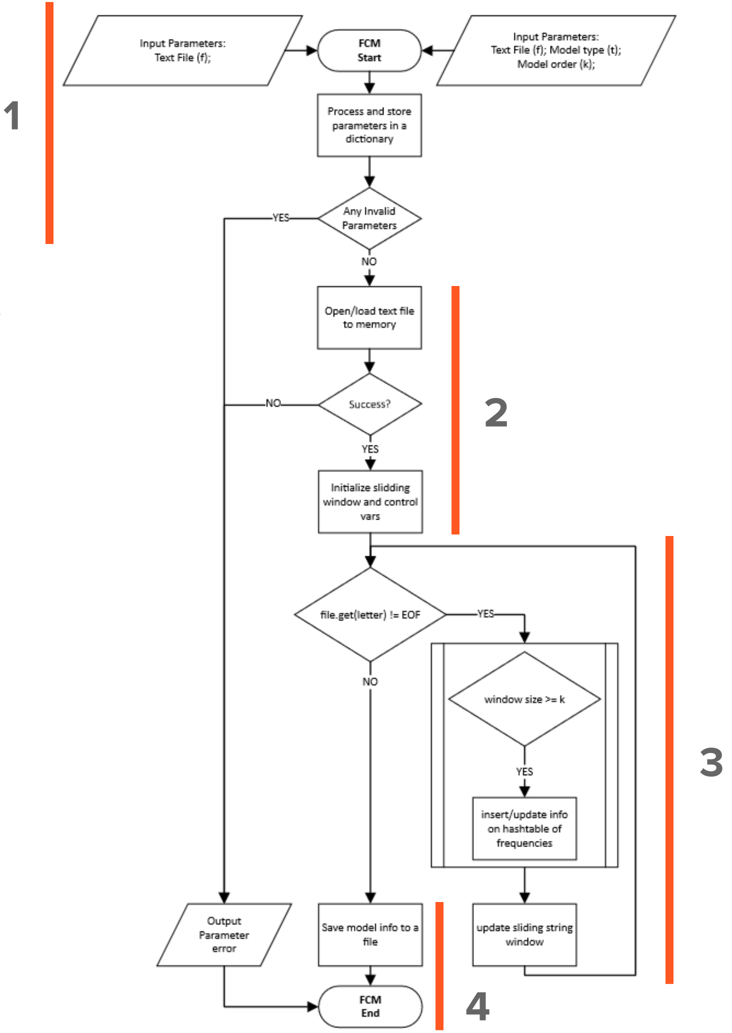
FCM

Input parameters:

- **k** - Order of the finite-context model; default value is **2**;
- **t** - Model Type (“**H**” for Human, “**A**” for AI/ChatGPT), default is “**H**”;
- **f** - Path to the file with the texts to create the model.

… On 4 steps…

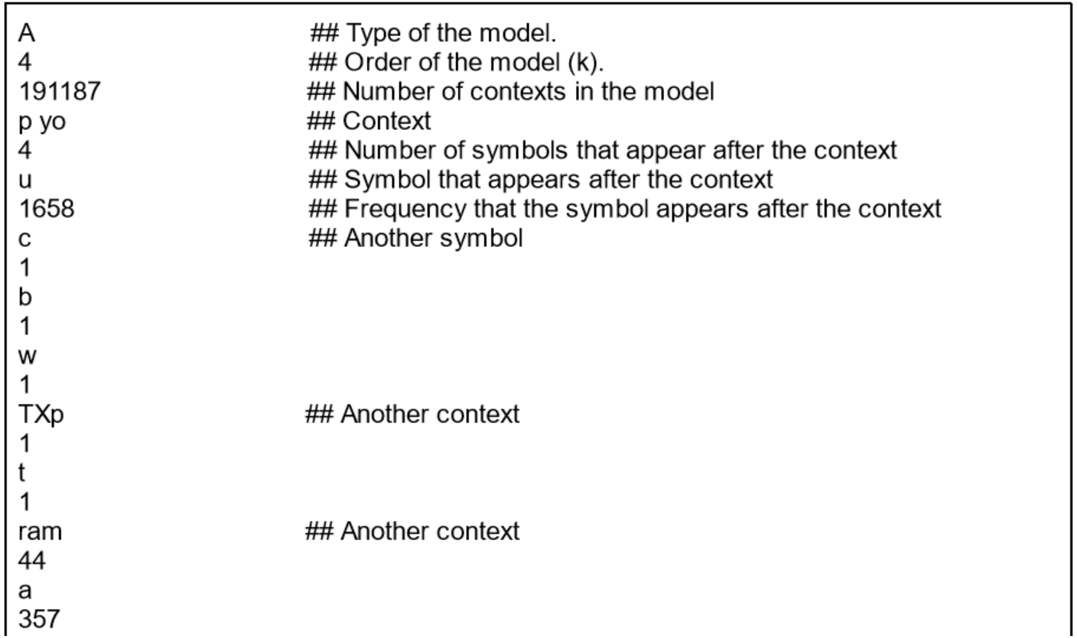
1. Read inputs parameters;
2. Load file to memory and initialize sliding window;
3. Loop through the file and collect frequencies;
4. Save frequencies model in a file.



1
Input Parameters: Text File (f);
FCM Start
Input Parameters: Text File (f); Model type (t); Model order (k);
Process and store parameters in a dictionary
Any Invalid Parameters
YES
NO
2
Open/load text file to memory
Success?
NO
YES
Initialize slidding window and control vars
3
file.get(letter) != EOF
YES
NO
window size >= k
YES
insert/update info on hashtable of frequencies
update sliding string window
Output Parameter error
Save model info to a file
4
FCM End

FCM

Example of a model file



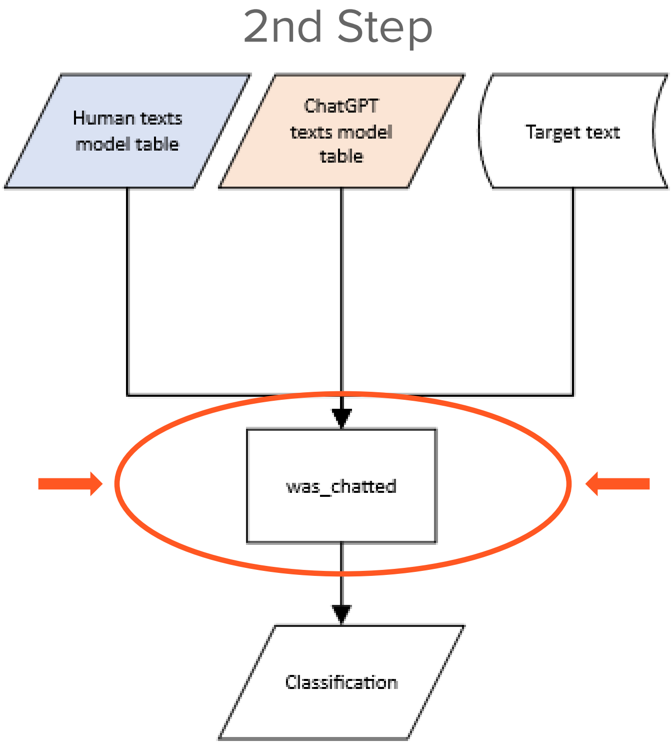
```
A                  ## Type of the model.
4                  ## Order of the model (k).
191187             ## Number of contexts in the model
p yo               ## Context
4                  ## Number of symbols that appear after the context
u                  ## Symbol that appears after the context
1658               ## Frequency that the symbol appears after the context
c                  ## Another symbol
1
b
1
w
1
TXp                ## Another context
1
t
1
ram                ## Another context
44
a
357
```

...

In this example...

- “A” stands for an AI or ChatGPT model;
- Its a order-4 finite-context model;
- Has a total of 191187 contexts

Methodology (was_chatted)



was_chatted

Goal: compress a text and classify it as written by humans or generated by AI.

Arguments:

- h - The name of the file with the human-based model.
- c - The name of the file with the ai-based model.
- t - The name of the file with the text to be classified.
- a - The alpha.
- k - The order of the model.

was_chatted

Implementation

- Reading each model into an unordered map and checking the type and order

- Reading the target file character by character in a loop performing the following actions:
  - If the character is a newline or a tab, the character is skipped.
  - If the context is the same size as the order of the models, do the following for both models:
    - Get the frequency of the character after the context.
    - Get the sum of the frequencies of every symbol that appears after the context.
    - Calculate the probability.
    - Calculate the number of bits needed to represent the probability.
    - Add to the total number of bits.
  - Update the context to have the last k characters

- Comparison of the number of bits needed to compress text in both models. The model that requires fewer bits for compression determines the classification of the text.

Datasets and Results

Results

Original Dataset :

- 'AI_Human.csv' from Kaggle
    - **text**
    - **generated**

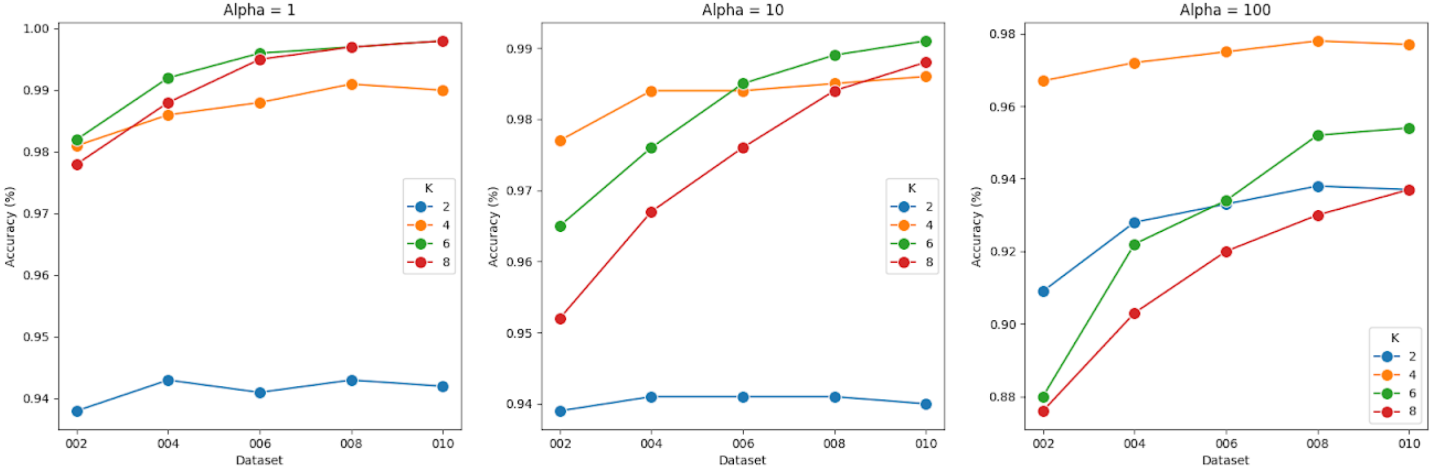
Datasets derived from the original:

- Train – 5 files with 2-10% of texts of the original
- Test – 500 texts of each type
- Small Test – 40 texts from Test dataset but with 25/50% size

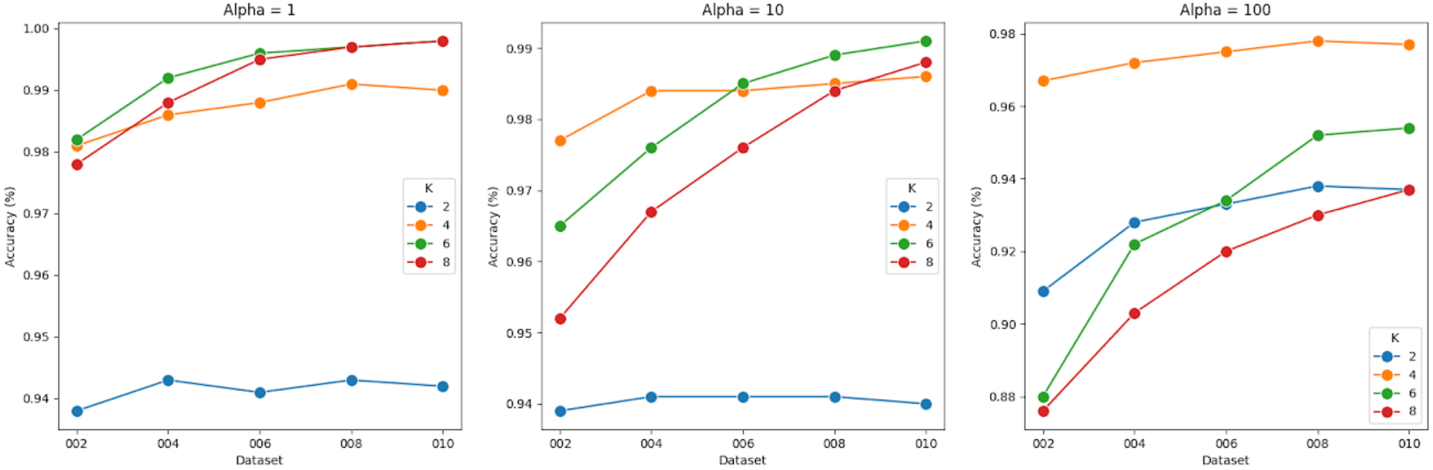
Range of input parameters:

- k - {2,4,6,8}
- alpha - {1,10,100}

Results

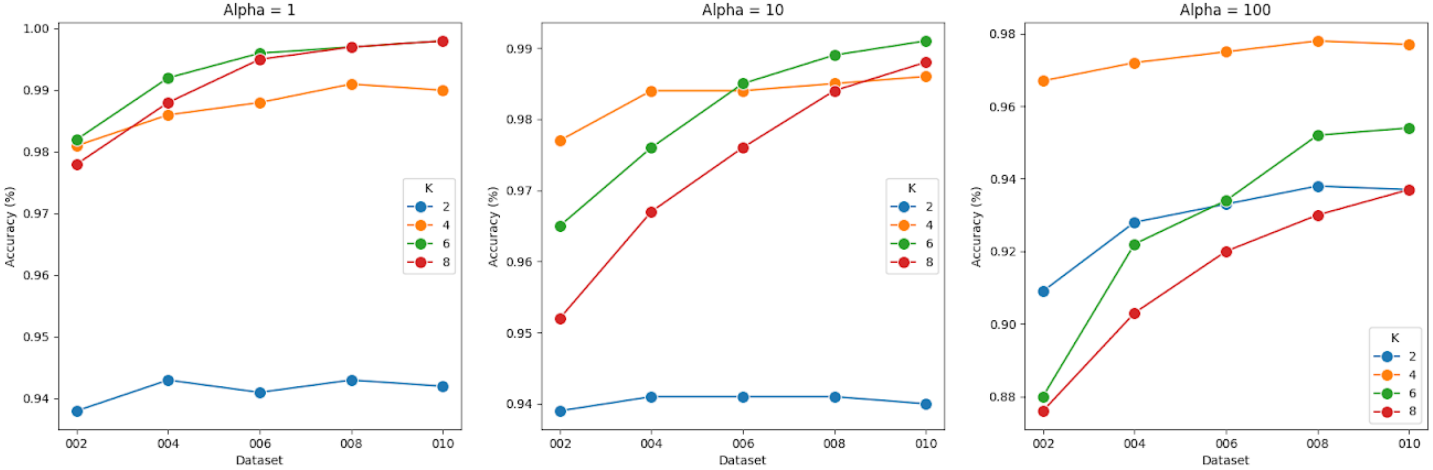

Alpha = 1
Alpha = 10
Alpha = 100
Accuracy (%)
Dataset
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.99
0.98
0.97
0.96
0.95
0.94
0.98
0.96
0.94
0.92
0.90
0.88
002
004
006
008
010
K
2
4
6
8

Results



Lower alpha => More stable

Results

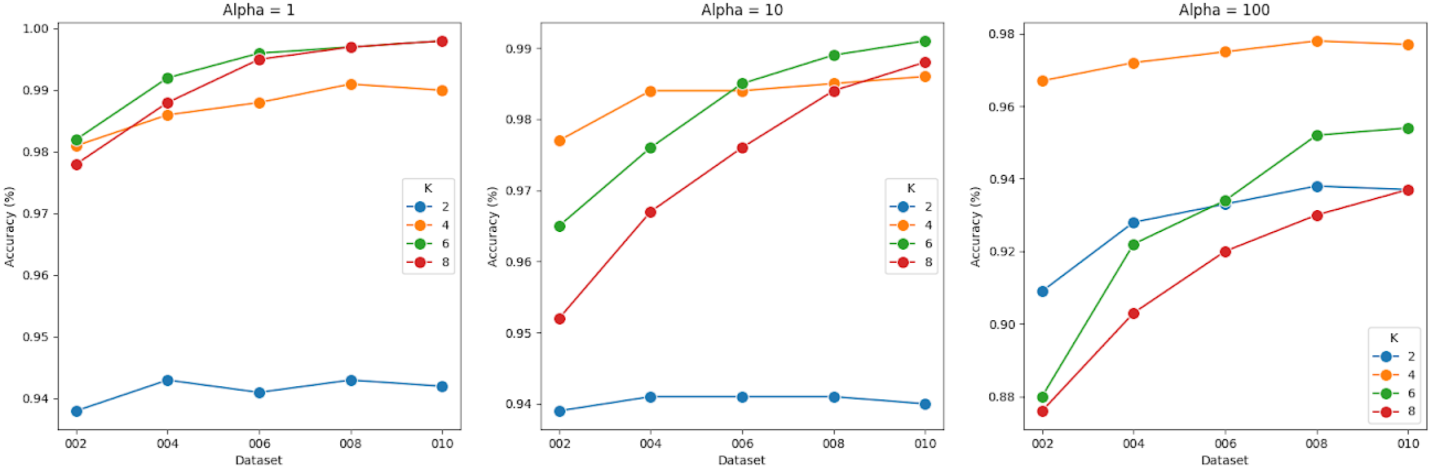


Lower k => Poor performance

However,

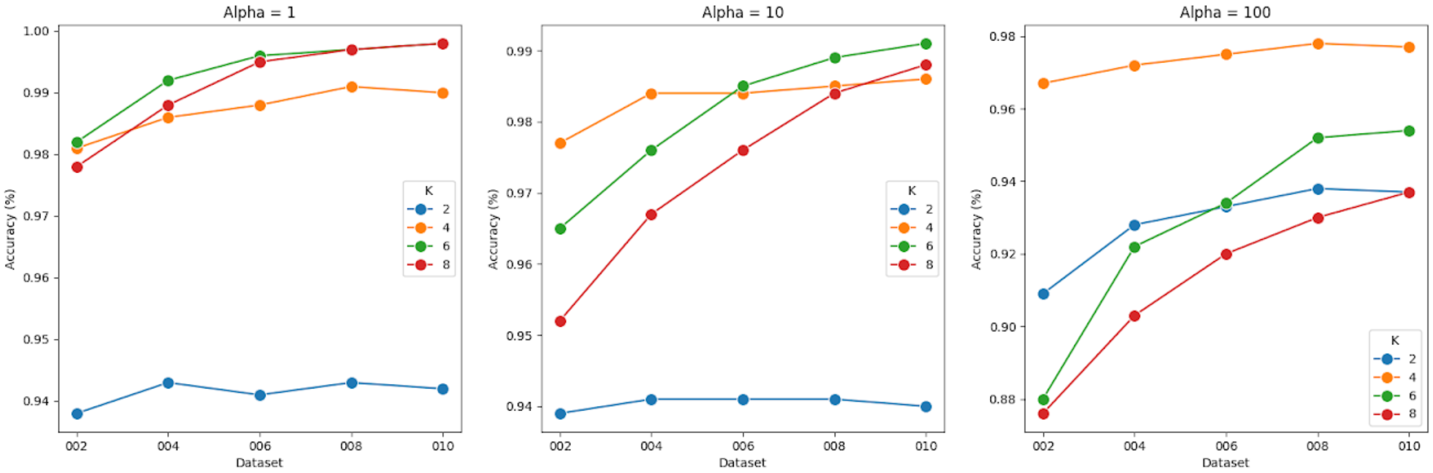
Higher alpha => Higher k => Good performance

Results



Proportion of training data impacts accuracy

Results



Best Results: Higher k, Low alpha & Rich datasets

Results – AI vs Human

Human text:

- Best results:
  - Lower alpha & Richer datasets
- Accuracy:
  - Did not fall below 88.6%

AI-generated text:

- Best results:
  - Richer datasets
- Accuracy:
  - 75.6% minimum

Note: This difference may relate to the size of the datasets, with human text datasets being larger compared to those for AI-generated texts.

Conclusion

- Data compression can differentiate texts written by humans and texts generated by AI.
- Accuracy between 88.6% and 100% in human texts and between 75.6% and 99.6% for AI-generated texts.
- More diversity in data sets improve program accuracy.
- More complex models with parameter adjustment (e.g. alpha factor), achieve better results, with a maximum global accuracy of 99%.
- The program's accuracy does not seem to depend on the size of the texts tested but we cannot say for certain due to some limitations: maximum size of texts to be analyzed due to ChatGPT restrictions, and the need for more data and tests

Future Work

- Accuracy – test with larger datasets, different types of text and wider range of input parameters.
- Impact of target text length - tests with smaller target texts.
- Simplify testing - use an array of alpha values.
- Optimizing memory usage by exploring alternative model storage methods.
- Allow the creation and use of various models. Offer customization options for different types of models.
- Evaluate the potential benefits of multithreading on program execution.