Detect AI-generated content with

Finite-context Models

Grupo 9

Daniel Ferreira	102885
Inês Castro	98384
João Teles	123456

Table of Contents

Finite-context models

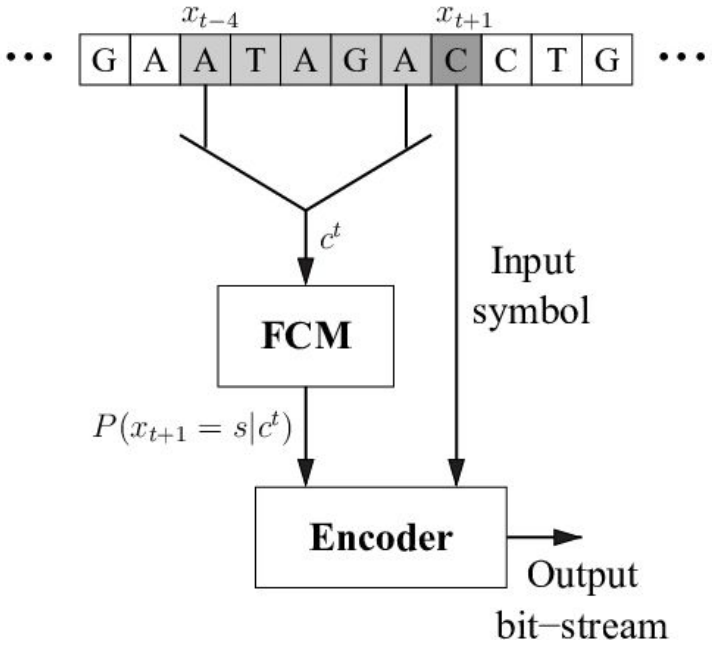


Figure 1: Finite-context mode with M=5, the probability of the next outcome, x t+1 , is conditioned by the M last outcomes.[1]

Finite-context Models

- Type of **markov chain**
- Used for *lossy data compression*
- Estimates probabilities of symbols appearing within a context
- Can we use it for **classification tasks**?

Dataset creation*




* Published on kaggle[2]

Dataset Structure

- 2 columns

Text	Label
Celebrities have always been a sub...	1
Have you ever went to someone for...	0
There are a few people who think...	1

0 Human texts 1 AI texts

Training Dataset: 693.95 MB

Testing Dataset: 193.72 MB

Data Preprocessing

- Eliminate duplicate text
- Remove missing values
- Dataset balancing based on the character count

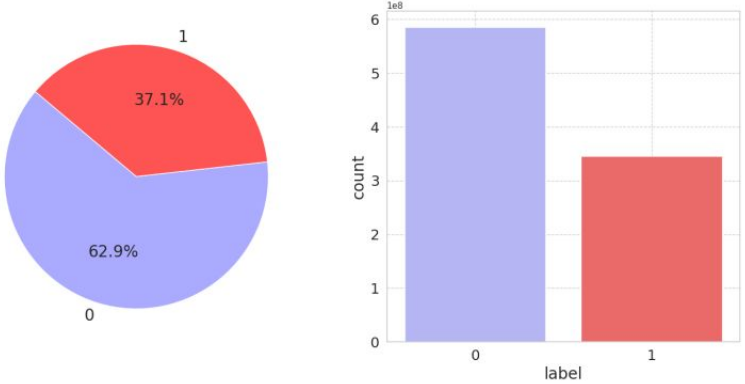


Figure 1: Unbalanced training dataset.

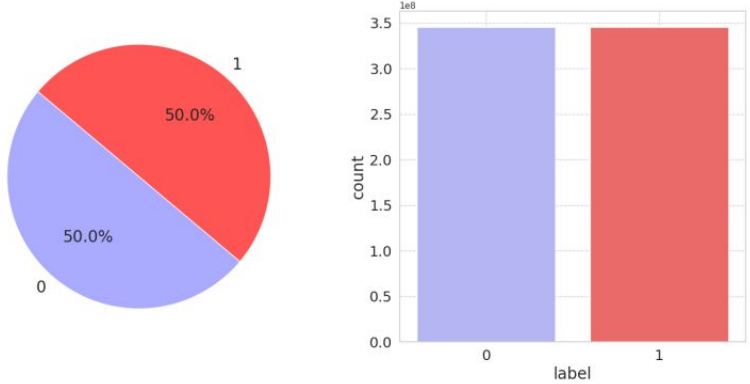


Figure 2: Balanced training dataset based on character count.

Dataset Analysis

- Based on the training set character count distribution

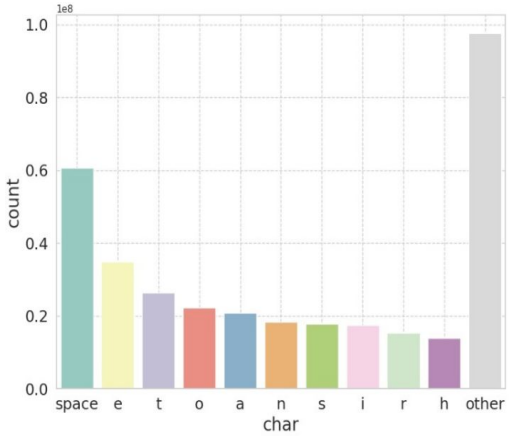


Figure 3: Human texts (label 0) letter distribution - training set.

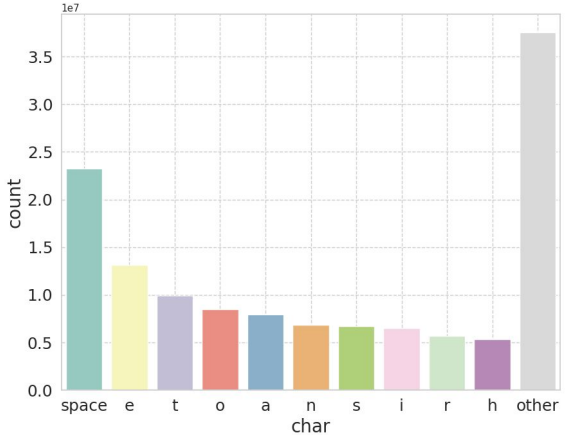


Figure 4: Human texts (label 0) letter distribution - testing set.

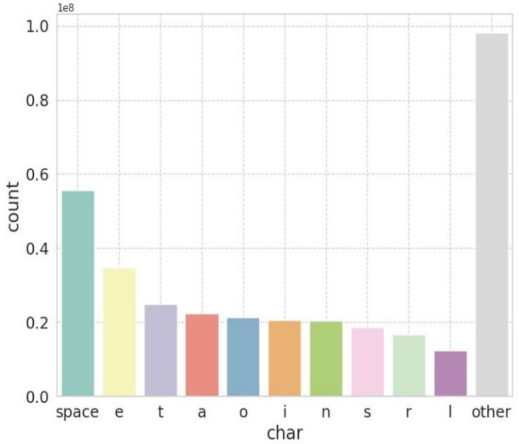


Figure 5: AI texts (label 1) letter distribution - training set.


Implementation

Model Parameters

k	Integer representing the context length or order of the model
smoothing_factor	Float value used for Laplace smoothing
alphabet	String containing the characters in the alphabet
ignore_case	Boolean flag indicating whether the model should ignore case when processing characters
scaling_factor	Integer used to scale down counts when they reach the maximum value (UINT32_MAX)
id	String identifier of the model

Model Structure

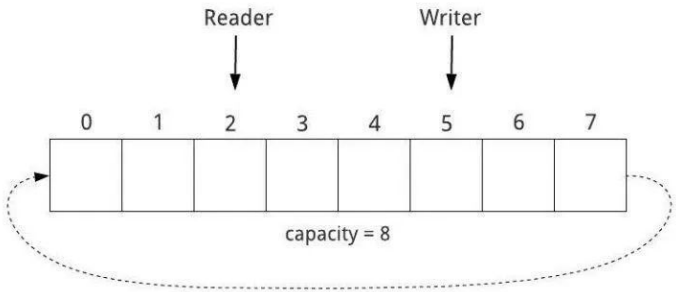
```
unordered_map<char, uint32_t> events;
uint32_t total;
```

```
unordered_map<string, EventMap> context_counts;
```

context

```
circular_buffer<char> buffer(k);
```

context



Model Persistence

(1) save(output);
Serializes model parameters and data to a binary (.bin) file.

(2) load(source);
Load the pre-trained model for prediction, inference,
or retraining on new data.

Core functionality

Training:

(1) update(source);
Updates context counts based on input text/file.

(2) increment(counts, event);
Increments event count within a context;
Uses fail-safe mechanism to prevent overflow using scaling factor.

Inference:

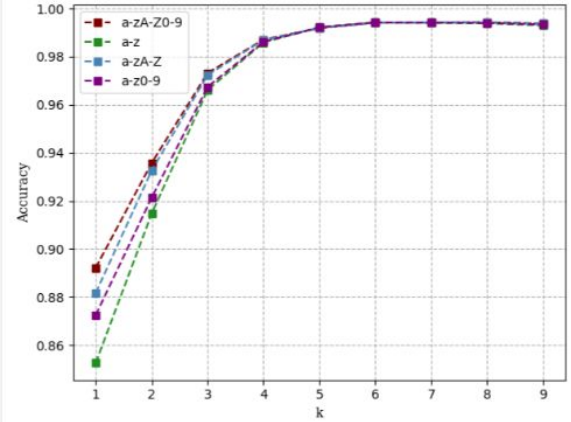
(3) estimate_bits(source);
Estimates total bits for encoding text/file.

(4) estimate_bits(context, event);
Estimates bits needed to encode a character in a context.

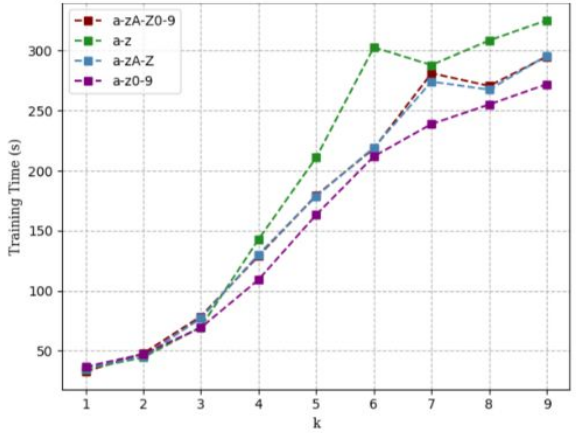
(5) probability(context, event);
Calculates probability of a character in a context.

Results

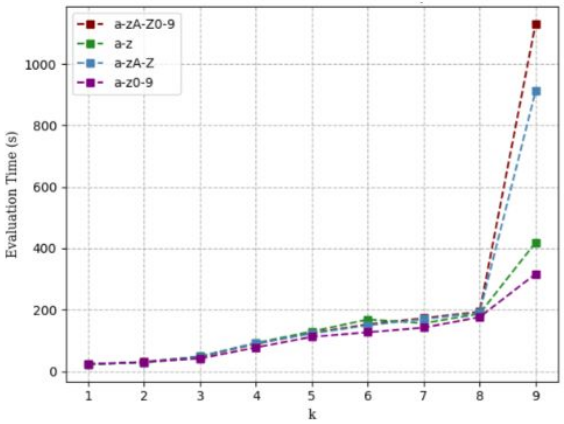
Alphabet parameter tuning



(a) k vs. Model Size (Binary File)



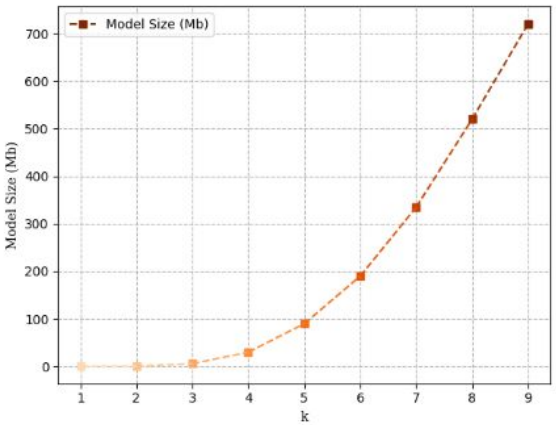
(b) k vs. Training Time



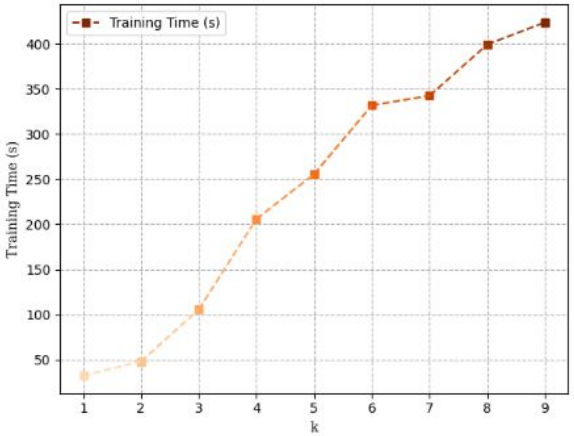
(c) k vs. Training Time

Figure 6: Comparison of the different alphabet's result

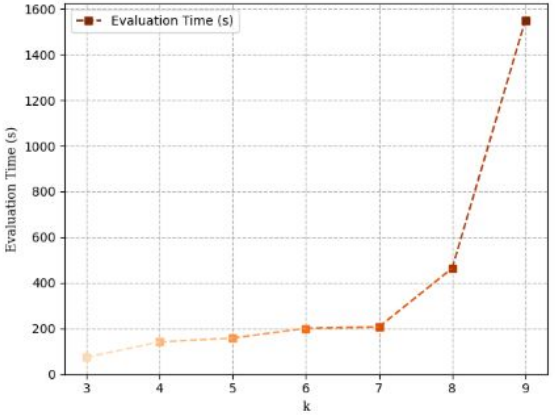
Impact of context length k on model performance



(a) Model Size (MB) as a function of k



(b) Training time (s) as a function of k



(c) Evaluation Time (s) as a function of k

Figure 7: Impact of context length k on model performance

Accuracy as a function of context length (k)

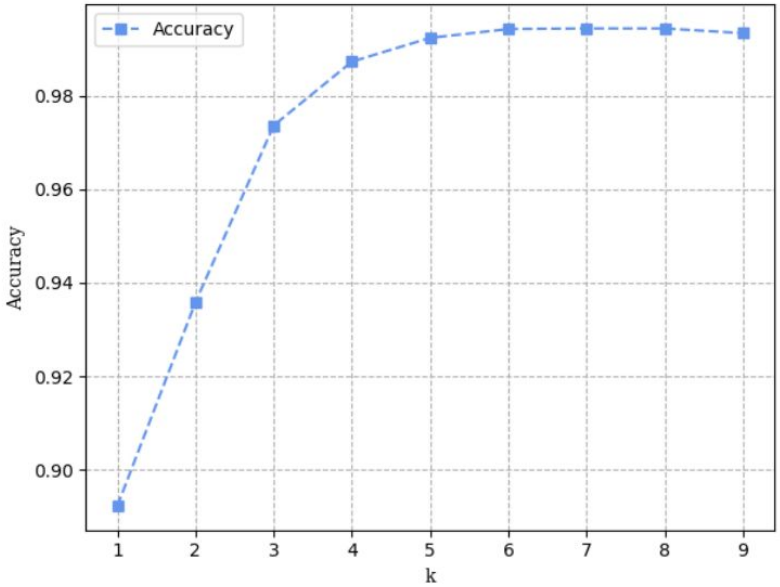


Figure 8: Accuracy as a function of context length (k)

4.9

Average size of word in English

Source:https://www.wyliecomm.com/2021/11/whats-the-best-length-of-a-word-online (*New York Times*) [3]

Accuracy as a function of target text length

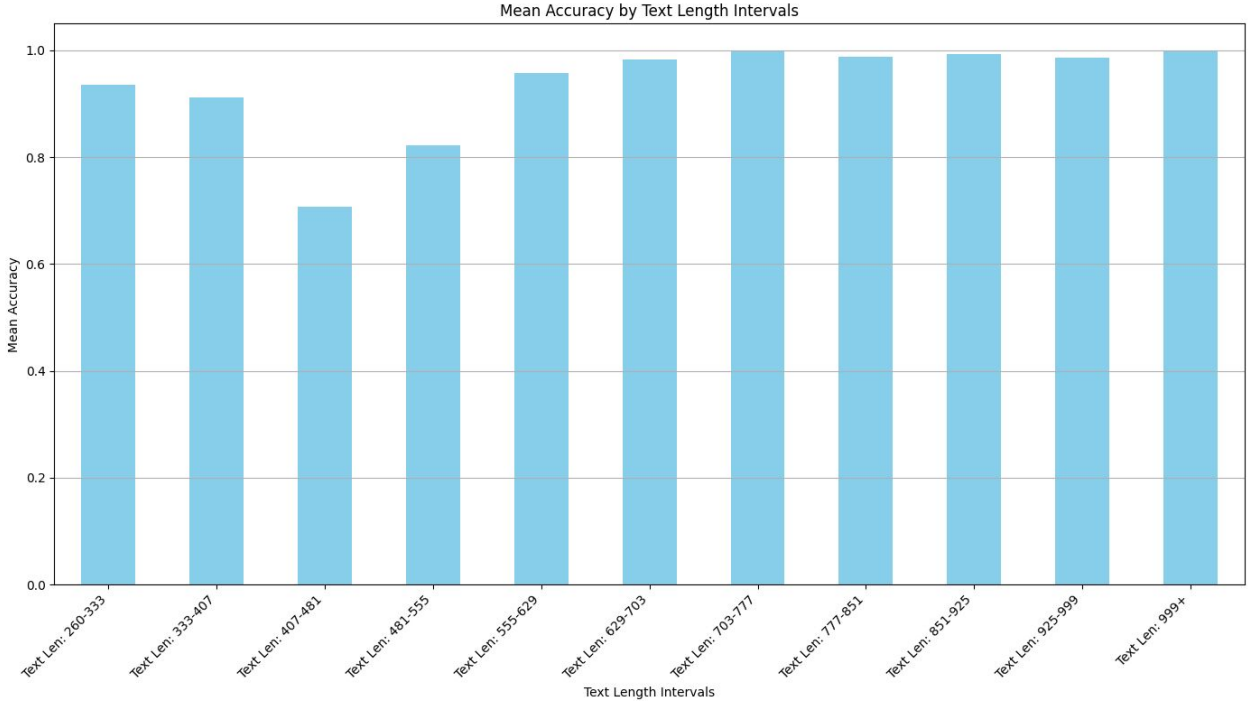


Figure 9: Accuracy as a function of target text length

Model evaluation

Approximately 50 combinations tested with:

k = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
s = [0.5, 0.75, 1, 1.25, 1.5]

Parameter	Value
`smoothing_factor`	0.5
k	7
alphabet	`[a-zA-Z0-9]`
Training time (s)	342.5
Evaluation time (s)	206.5
Average evaluation time (s)	206.5
Total Bytes (MB)	35.14
Accuracy (%)	99.43

Model's Binary File Size: ~324 MB

Table 5: Confusion Matrix for the best model.

		Predicted	
		Human	AI
Actual	Human	55,560	285
	AI	205	30,537

Table 6: Model Evaluation Metrics for the best model.

Accuracy	Recall	Precision	F1-Score
99.43	99.33	99.07	99.20

Conclusions

(1) The model demonstrates exceptional performance in distinguishing between human-written and AI-written texts. With an accuracy of 99.43%

(2) For a greater k the model gets more complex, but not necessarily more accurate

(3) Despite the accuracy, the model may be overfitted because the training and the testing dataset are very similar. It may not work as well in a real scenario

(4) Maybe data compression can be explicitly used to address classification problems, removing the need for a separate feature extraction stage

References

1. “Fig. 1. Finite-context model: the probability of the next outcome, x...," ResearchGate. (May, 2024). url: https://www.researchgate.net/figure/Finite-context-model-the-probability-of-the-next-outcome-x-t-1-is-conditioned-by-the_fig1_221907789
2. tai-dgt. Accessed 6 May 2024. May 2024. url: https://www.kaggle.com/datasets/danielnivalis/tai-daigt.
3. Ann Wylie. “What’s the Best Length of a Word Online?” In: Wylie Communications, Inc. (Jan, 2024). url: https://www.wyliecomm.com/2021/11/whats-the-best-length-of-a-word-online.