

# Compressed Context Modeling for Text Compression

M. Oğuzhan Külekci

kulekci@uekae.tubitak.gov.tr

TÜBİTAK-UEKAE

National Research Institute of Electronics&Cryptology, Turkey

IEEE Data Compression Conference  
Snowbird, UT, USA  
March 31, 2011

# Lossless Data Compression and PPM

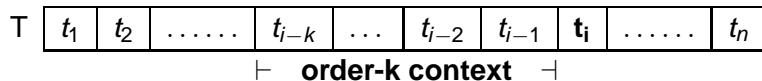
- Devise algorithms to represent input data with least number of bits.
- Prediction by partial matching (CW'84) variants are among the best *lossless* compression methods.
- Main idea: Predict next symbol based on the preceding symbols.



- Compress  $t_i$  by encoding the probability  $P(t_i | t_{i-1} \dots t_{i-k})$ .

# PPM Compression Mechanism

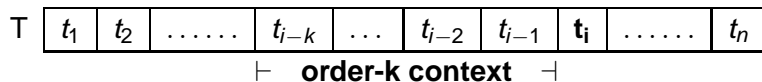
Assume  $t_1 \dots t_{i-1}$  has been encoded and  $t_i$  is to be encoded.



- 1 If  $[t_q \dots t_{q+k}] = [t_{i-k} \dots t_i]$  for  $\exists q : q < (i - k)$   
then send probability  $P(t_i|[t_{i-k} \dots t_{i-1}])$  to the encoder.

# PPM Compression Mechanism

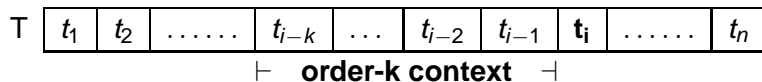
Assume  $t_1 \dots t_{i-1}$  has been encoded and  $t_i$  is to be encoded.



- 1 If  $[t_q \dots t_{q+k}] = [t_{i-k} \dots t_i]$  for  $\exists q : q < (i - k)$   
then send probability  $P(t_i | [t_{i-k} \dots t_{i-1}])$  to the encoder.
- 2 Else **emit escape symbol**,  
**change context** (*usually by decrementing  $k$* ),  
if new context is non-empty, go to previous step  
else encode actual value of  $t_i$ .

# PPM Compression Mechanism

Assume  $t_1 \dots t_{i-1}$  has been encoded and  $t_i$  is to be encoded.

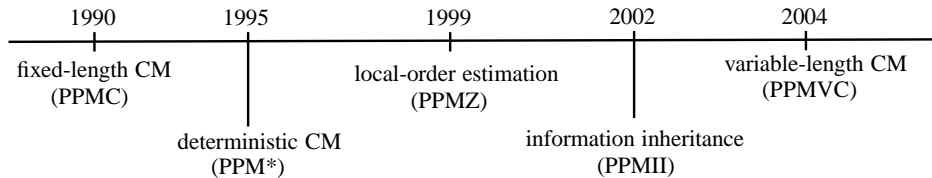


- 1 If  $[t_q \dots t_{q+k}] = [t_{i-k} \dots t_i]$  for  $\exists q : q < (i - k)$   
then send probability  $P(t_i | [t_{i-k} \dots t_{i-1}])$  to the encoder.
- 2 Else **emit escape symbol**,  
**change context** (*usually by decrementing  $k$* ),  
if new context is non-empty, go to previous step  
else encode actual value of  $t_i$ .
- 3 Continue encoding with the next character,  $i = i + 1$ .

# Improvements in PPM

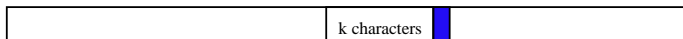
Two main points of research to improve PPM:

- 1 Escape probability handling  
PPMA, PPMB, PPMC, PPMX, PPMP
- 2 **Alternative context-modeling techniques**



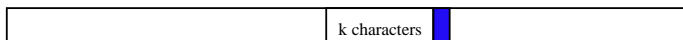
## Approaches in Context Modeling

- **Fixed-length CM:** Context length is predetermined, basic scheme, easy to implement, not very efficient

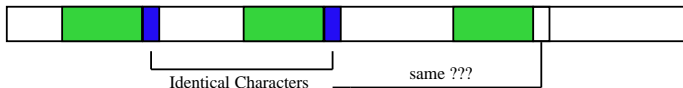


## Approaches in Context Modeling

- Fixed-length CM:** Context length is predetermined, **basic** scheme, **easy to implement**, **not very efficient**

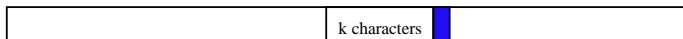


- Deterministic CM:** The context that is always followed by the same symbol, **heavy computational load**, **not that much gain**

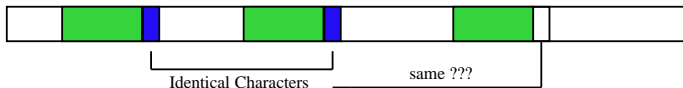


## Approaches in Context Modeling

- **Fixed-length CM:** Context length is predetermined, **basic** scheme, **easy to implement**, **not very efficient**



- **Deterministic CM:** The context that is always followed by the same symbol, **heavy computational load**, **not that much gain**



- **Variable-length CM:** Compute context length due to some heuristics, **very complicated**, **requires more memory**

## Information Content of Contexts are Variable

- The information contents carried by different contexts of a symbol are not equal.

... adequate,  $P(u|q)$  is possibly good enough.

However;

... obvious,  $P(u|o)$  ???

... obvious,  $P(u|io)$  ??

... obvious,  $P(u|vio)$  ?

- Thus, the context length should be variable!

## The Main Question

- What should be the best context length for a particular position?
- We need **enough amount of information**.

## The Main Question

- What should be the best context length for a particular position?
- We need **enough amount of information**.

### Previous definition of *context*

The context of a symbol at a particular position is its **preceding  $k$  symbols**.

### New definition of *context*

The context of a symbol at a particular position is its **preceding  $k$ -bits information**.

## The Main Question

- What should be the best context length for a particular position?
- We need **enough amount of information**.

### Previous definition of *context*

The context of a symbol at a particular position is its **preceding  $k$  symbols**.

### New definition of *context*

The context of a symbol at a particular position is its **preceding  $k$ -bits information**.

**How to compute the preceding  $k$ -bits of information for all positions in the text?**

## Main Idea

- The amount of information can be calculated by compression !
- We can measure the previous information content of  $t_i$  by compressing its preceding characters  $t_{i-1}t_{i-2} \dots$
- Something like **compression-by-compression** ?

## Compressed Context Modeling

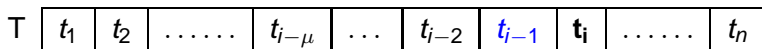
Compressed context of  $t_i$  is its preceding  $\mathcal{I}$ -bit information.



Bit-array  $B = b_1 b_2 \dots b_\ell = \text{compress}(S = s_1 s_2 \dots s_\omega)$ ,  $S$  is character array.

## Compressed Context Modeling

Compressed context of  $t_i$  is its preceding  $\mathcal{I}$ -bit information.

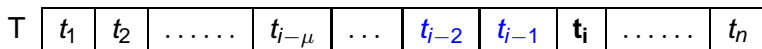


- $B = \text{compress}(t_{i-1})$ , while  $|B| < \mathcal{I}$ , continue ...

Bit-array  $B = b_1 b_2 \dots b_\ell = \text{compress}(S = s_1 s_2 \dots s_\omega)$ ,  $S$  is character array.

## Compressed Context Modeling

Compressed context of  $t_i$  is its preceding  $\mathcal{I}$ -bit information.



- $B = \text{compress}(t_{i-1})$ , while  $|B| < \mathcal{I}$ , continue ...
- $B = \text{compress}(t_{i-1}t_{i-2})$ , while  $|B| < \mathcal{I}$ , continue ...

.....

Bit-array  $B = b_1 b_2 \dots b_\ell = \text{compress}(S = s_1 s_2 \dots s_\omega)$ ,  $S$  is character array.

## Compressed Context Modeling

Compressed context of  $t_i$  is its preceding  $\mathcal{I}$ -bit information.



- $B = \text{compress}(t_{i-1})$ , while  $|B| < \mathcal{I}$ , continue ...
- $B = \text{compress}(t_{i-1}t_{i-2})$ , while  $|B| < \mathcal{I}$ , continue ...
- .....
- $B = \text{compress}(t_{i-1}t_{i-2} \dots t_{i-\mu})$ , stop when  $|B| \geq \mathcal{I}$ .

Bit-array  $B = b_1 b_2 \dots b_\ell = \text{compress}(S = s_1 s_2 \dots s_\omega)$ ,  $S$  is character array.

## Compressed Context Modeling

Compressed context of  $t_i$  is its preceding  $\mathcal{I}$ -bit information.



- $B = \text{compress}(t_{i-1})$ , while  $|B| < \mathcal{I}$ , continue ...
- $B = \text{compress}(t_{i-1}t_{i-2})$ , while  $|B| < \mathcal{I}$ , continue ...
- .....
- $B = \text{compress}(t_{i-1}t_{i-2} \dots t_{i-\mu})$ , stop when  $|B| \geq \mathcal{I}$ .
- The context is the first  $\mathcal{I}$  bits of  $B = \text{compress}(t_{i-1}t_{i-2} \dots t_{i-\mu})$ .

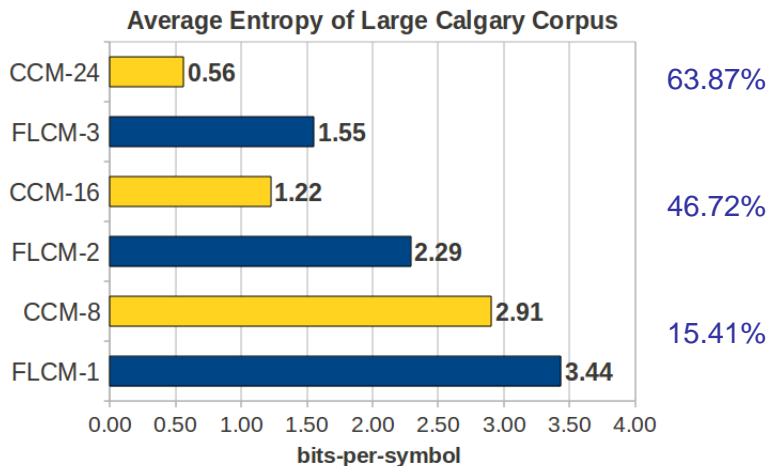
Bit-array  $B = b_1 b_2 \dots b_\ell = \text{compress}(S = s_1 s_2 \dots s_\omega)$ ,  $S$  is character array.

## Measuring the Performance

- Files in *large* Calgary corpus are modeled with compressed context modeling.
- Average empirical entropies are measured as **bits-per-character**.
- The compression function *comp()* used in CCM is a simple 0<sup>th</sup>-order static Huffman coding.
- We compare the results with fixed-length modeling for length from 1 to 10.

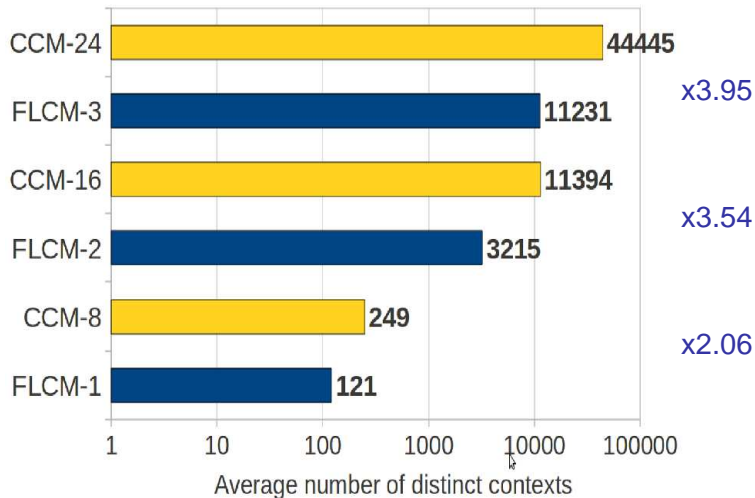
## Performance Comparison on Equal-Length Contexts

Order- $k$  **byte** fixed-length versus order-8 ·  $k$  **bit** compressed context.



## However, ...

The number of distinct contexts in CCM is much higher.



## Restricted-space CCM versus FLCM

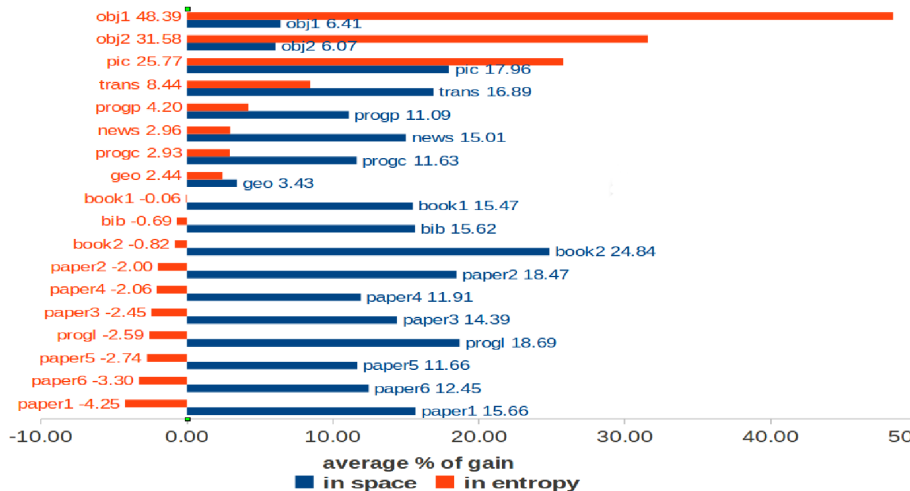
Choose largest  $\alpha$  such that the number of distinct contexts in CCM- $\alpha$  is less than that of in FLCM- $\beta$ .

Context Length (bits) (bytes)		# of distinct contexts		Avg. Empirical Entropy in bpc		Percentage of Gain in Space	Percentage of Gain in Entropy
CCM	FLCM	CCM	FLCM	CCM	FLCM		
6	1	64	95	3.61	3.64	32.63	0.9
11	2	1477	1556	2.13	2.33	5.08	8.62
15	3	4769	6155	1.47	1.40	22.52	-4.55
18	4	11324	12841	0.94	0.90	11.81	-4.84
22	5	19412	19841	0.61	0.63	2.16	2.09
24	6	20920	26074	0.56	0.44	19.77	<b>-27.74</b>

Avg.: 15.66    Avg.: -4.25

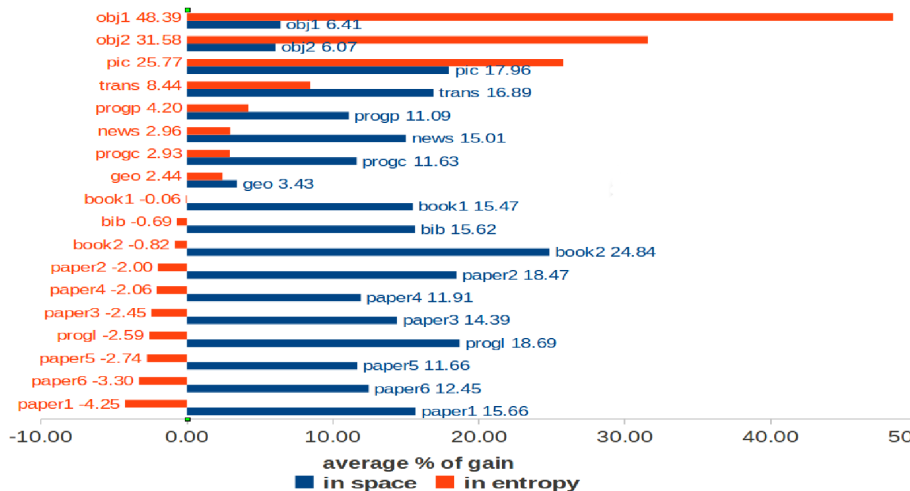
The result of space-restricted comparison on file *paper1*.

## Overall Space–Restricted Comparisons



## Overall Space–Restricted Comparisons

5.88% better empirical entropy in 13.76% less space



## Conclusions

- A new context modeling technique changing the definition of context in text compression.
- Approximately equal amount of information on encoding of each symbol!
- Enhanced empirical entropy within less space.
- Much better modeling especially on semi-structured or unstructured inputs.

## Conclusions

### Data compression performance metrics

- Compression ratio
- Compression speed
- Resource (memory) requirement

### *(possible)* Improvement in compression ratio via CCM

- Compression ratio is directly proportional to the measured empirical entropy under the used model.
- The empirical entropies measured by CCM on Calgary corpus is promising.
- Thus, PPM type compression based on CCM might have a good chance to achieve better compression ???

## Conclusions

### Data compression performance metrics

- Compression ratio
- Compression speed
- Resource (memory) requirement

### *(possible)* Improvement in compression speed via CCM

- PPM is slow due to complex data structures.
- Searching a specific context takes time (hashing helps, but does not solve the problem).
- Directly addressable context tables are possible via CCM, which eliminates the searching phase and can retrieve the statistics directly!

## Conclusions

### Data compression performance metrics

- Compression ratio
- Compression speed
- Resource (memory) requirement

### *(possible)* Improvement in memory requirement via CCM

- Space–restricted CCM has shown good performance.
- Trading space versus entropy via CCM seems advantageous, especially in limited resource environments such as the mobile phones or hand–held devices.
- CCM based compression might be a good alternative in mobile data compression.

## Future Work

Design of PPM compressors based on **compressed context modeling** to investigate possible improvements in

- compression ratio,
- speed,
- memory usage.

## Future Work

Design of PPM compressors based on **compressed context modeling** to investigate possible improvements in

- compression ratio,
- speed,
- memory usage.

**THANK YOU!**