REDUCE COMPLEXITY OF MATRIX MULTIPLICATION IN WORD PREDICTION MODEL

ChengHan Chung[1], Ayushmat Bhardwaj Soni[1], Ziya Mammadov[1]
[1]University of Tartu, Institute of Computer Science

INTRODUCTION

To predict the words, N-gram with Markov model is used in word prediction by computing the probability of a sentence or sequence of words.

N-gram

N-gram model predicts w_n based on w_{n-1}. In probability terms, $P(w_n|w_{n-1})$. In the Table.1, we can compute probibility of bigram *(n=2)* as following below:

$$P(\text{"the cat heard a mouse"}) = .6 \times .4 \times .5 \times .33 \times .5 \times .6 = 0.12$$

Table.1 Bigram model

bigram	count	unigram	count	bigram r.f.
START the	3	START	5	.6
the cat	2	the	5	.4
cat saw	2	cat	4	.5
saw the	1	saw	3	.33
the mouse	2	the	5	.4
mouse END	3	mouse	5	.6
cat heard	2	cat	4	.5
heard a	1	heard	3	.33
a mouse	2	a	4	.5
...				

Markov Chain

N-gram is a **Markov property** which the probability of future states only depends on the present state. This concept can be elegantly implemented using a Markov Chain storing the probabilities of transitioning to a next state (Shown as Fig.1).

The probabilities associated with various state changes are called transition probabilities. The process is characterized by a state space, a **transition matrix** describing the probabilities of particular transitions.

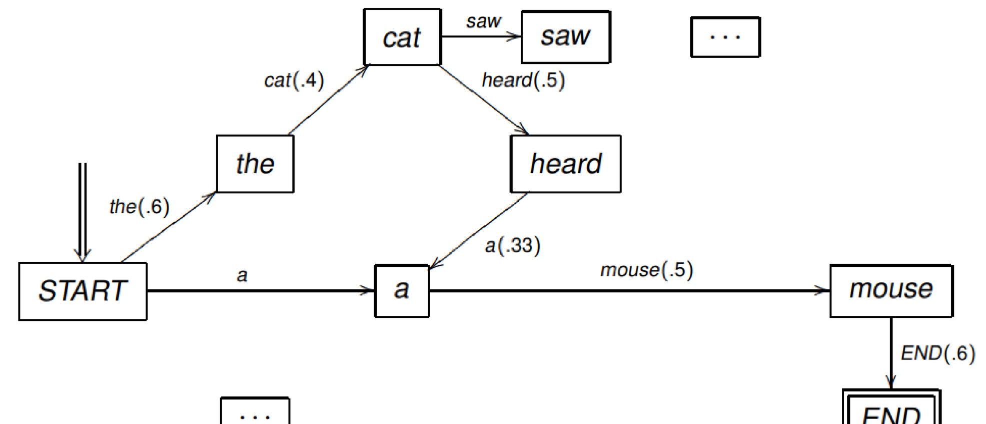

Fig.1 Bigram model

BOTTLENECK

The transition matrix is a $n \times n$ matrix (n words in a sentence or sequence), it might cause **sparse-data problem** as n increases. We predict k^{th} oreder word by **matrix multiplication**, therefore, the time complexity of matrix multiplication is $O(n^3)^k$, the time complexity will be **exponential growth** as k and n increases (Shown as Fig.2). To reduce the time complexity of matrix multiplication, we implement **Strassen algorithm** and **Compressed Sparse Row Format** in processes of word prediction.

THE SOLUTION

In the result, By **Strassen Algorithm** we can reduce complicilty of matrix multiplication to $O(N^{\log_2 7})$, By **Compressed Sparse Row Format** we can reduce complicilty of matrix multiplication to $O(mD)$.(Shown as Fig.2 and Fig.3)

Strassen Algorithm

The Strassen Algorithm is based on the concept of divide and conquer method. This method reduces the number of recursive calls to *7*. Hence , The complexity of the given expression is $T(n) = 7T(n/2) + O(n2)$, which come up to be $O(n^{\log_2 7})$.

Compressed Sparse Row Format (CSR)

The CSR represents a matrix M by three arrays (one-dimensional), 1. a array *(D)* contains nonzero values *(V)*, 2. the extents of rows *(n)*, and 3. column indices *(m)*. Hence, this format allows fast row access and matrix-vector multiplications (Mx), the operation can be at least as good as $o(m\sum_{i=1}^{m}|V_i|)$. (Shown as Fig.5)

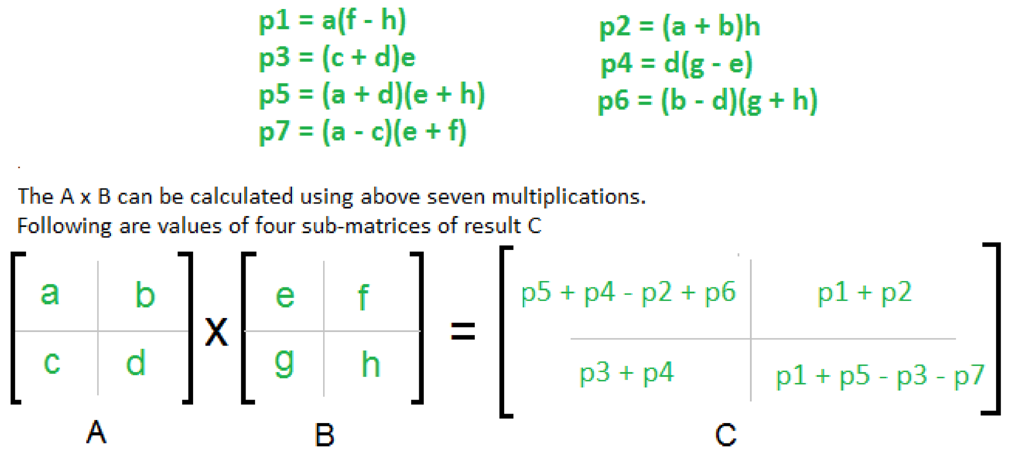

Fig.4 Strassen Algorithm

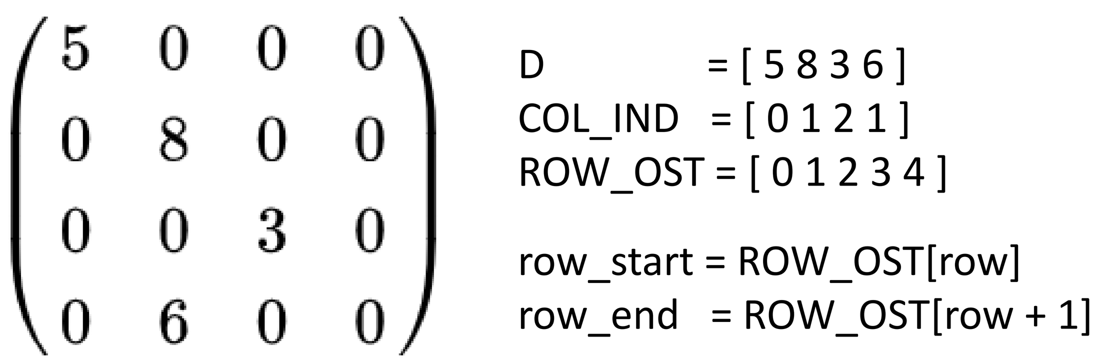

Fig.5 Compressed Sparse Row Format

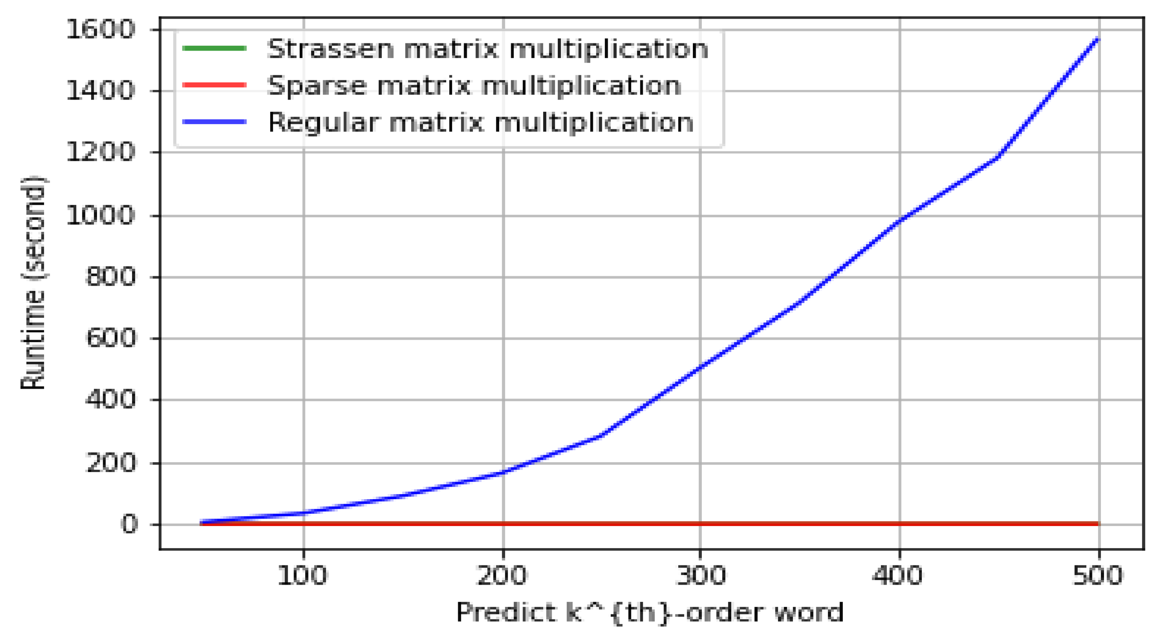

Fig.2 Runtime of matrix multiplication

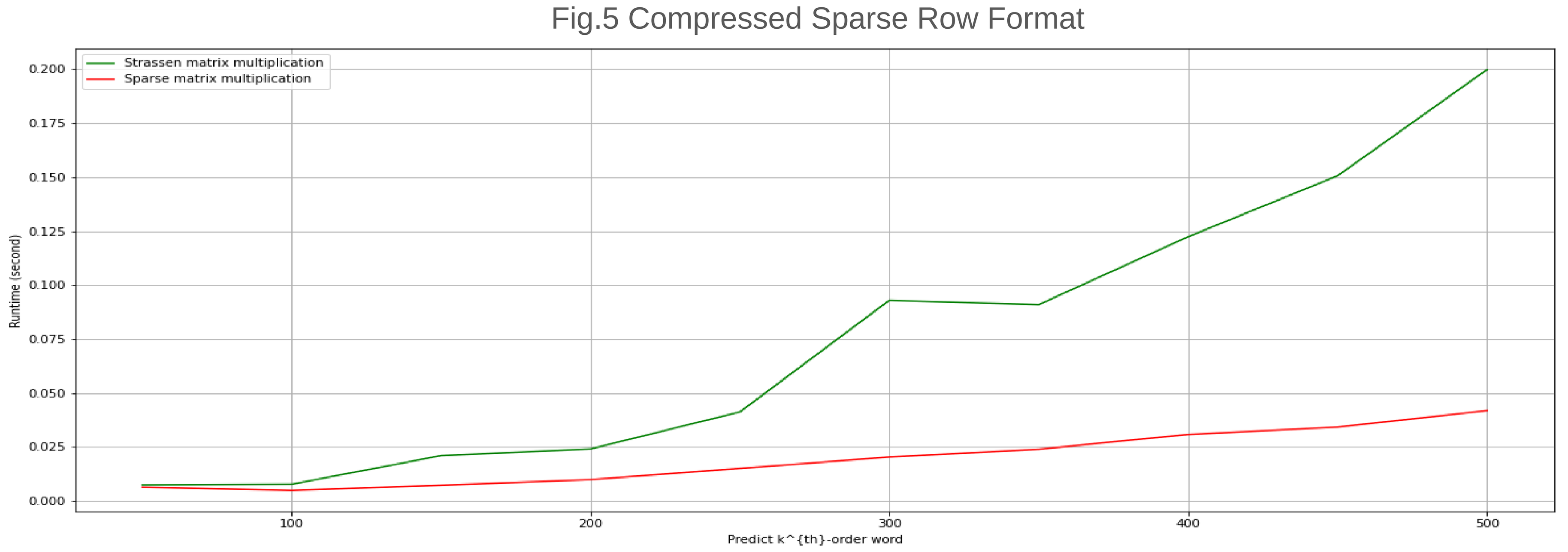

Fig.3 Runtime of Strassen Algorithm and CSR

REFERENCES

[1] Borštnik, Urban, et al. "Sparse matrix multiplication: The distributed block-compressed sparse row library." Parallel Computing 40.5-6 (2014): 47-58.
[2] Fink, Gernot A. "n-Gram Models." *Markov Models for* Pattern Recognition. Springer, London, 2014. 107-127.
[3] Meizar, Abdul, Frinto Tambunan, and Erwin Ginting. "Optimizing the complexity of time in the process of multiplying matrices in the hill cipher algorithm using the strassen algorithm." 2019 7th International Conference on Cyber and IT Service Management (CITSM). Vol. 7. IEEE, 2019.