S-Box Hashing for Text Mining

Sadi Evren SEKER
Department of Business,
Istanbul Medeniyet University
Istanbul, Turkey
academic@sadievrenseker.com

Cihan MERT
Department of Informatics
International Black Sea University
Tbilisi, Georgia
cmert@ibsu.edu.ge

Abstract— **One of the crucial points in the text mining studies is the feature hashing step. Most of the text mining studies starts with a text data source and processes a feature extraction methodology over the text. Most of the time the feature extraction method should be decided wisely, because, most of the times, it directly effects the results and performance. Another well-known approach is using any feature extraction method, together with the feature hashing. By the way, the feature extraction can be executed without worrying about the performance and the feature hashing reduces the size of the extracted feature vector.**

Today, one of the widely used hashing algorithms in text mining is the modern hashing algorithms like MD5 or SHA1, which are built over substitution permutation networks (SPN) or Fiestel Networks. The common property of most of the modern hashing algorithms is the implicitly implemented s-boxes.

One of the drawbacks of the modern hashing algorithms is the collision free purpose of the algorithm. The permutation step in most of the time is implemented for this purpose and the correlation between the input text and output bits is completely obfuscated.

This study focuses on the possible implementations of the s-boxes for the feature hashing. The purpose feature hashing in this study is reducing the feature vector, while keeping the correlation between the input text and the output bits.

Keywords— Data Mining; Feature Extraction; Feature Hashing; KNN; Hashing; Text Mining

I. INTRODUCTION

Because of the increasing demand on the big data studies, the importance of the dimension reduction on the feature vectors has an increasing impact on the contemporary studies. For example a study on social networks or blogosphere can easily end up with terabytes of data, which is a challenging issue for both computer hardware and the algorithms running on top of it.

A classical view of text mining on any data source is given in Fig. 1.

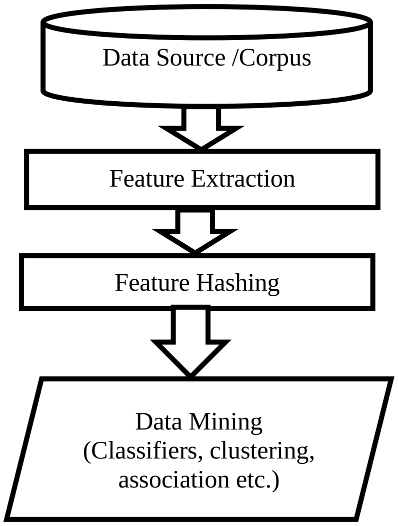

Fig. 1. Generic Deployment Diagram of Text Mining

Most of the feature hashing method is deployed after the feature extraction from the raw text data source. Also in some applications, the feature extraction method does the hashing implicitly or the feature extraction is applied after the hashing on the text data.

Besides those varieties, all of the text mining studies yield a feature vector before the data mining phase with a hashing implementation, if the hashing is applied.

Most of the famous feature hashing algorithms are the modern hashing algorithms like MD5 [1] or SHA-1 [2] in the current studies. Some natural language processing tool kits even comes with the implementation of those hashing algorithms.

In this study, we have focused on the s-boxes which almost all modern hashing algorithms uses for dimension reduction. Also the permutation phases in most of the modern hashing algorithms, aims to reduce the collision. We remove the permutation phase from the hashing algorithm and try to simplify the algorithm in the performance. Another reason for removing the permutation phase is the increasing the correlation between the input and the output of the hashing algorithm.

A background review on the hashing algorithms and the s-boxes [3] will be provided in the second section. The details of novel hashing algorithm and sample run explanations will be given in the third section. Finally the experiment details and real life data set IMDB62 [4] will be explained on the fourth section.

II. Background

The concept of Feistel Networks [3] has a major role in the feature hashing, including the text mining studies. A generic approach to the text mining is already provided in Figure 1 and from the figure, the feature hashing can be done before executing the data mining operations. Also in some studies, the hashing can be executed before the extraction phase. So in the latter approach, the feature is extracted from the hashed data source instead of extracting and hashing order.

In both of the approaches, the hashing has a major role to reduce the length of the feature vector. By the increasing importance of the big data studies, the size of the feature vectors can be considered as more important now. Besides the memory requirements to keep the information and move the data from servers, or processors working in distributed or parallel environments, also the processing speed of the data mining is closely related with the size of the feature vector.

The generic view of a SPN network is demonstrated in Figure2.

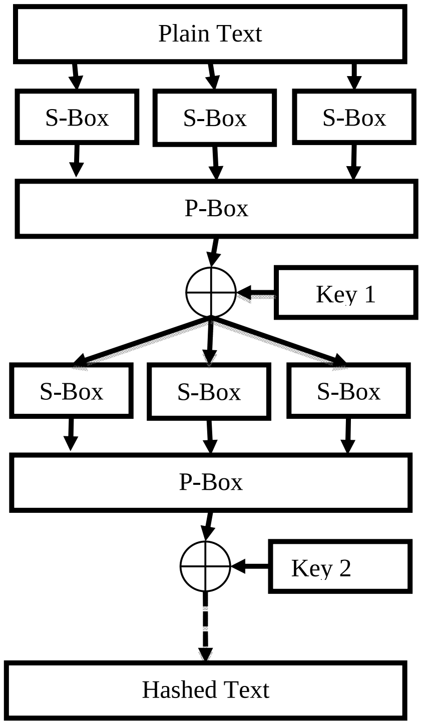

Fig. 2. Generic view of a SPN

In the SPN, an input text in plain form is mixed with the P-Boxes (permutation boxes) and reduces the size with the S-Boxes [5] (substitution boxes). For example an S-Box can reduce the number of input bits from 8 to 6 at the output. The function of hashing comes mainly in the S-boxes since a hashing function can be defined as one-way function from a bigger input domain to a smaller output range.

A major problem in reducing the size of the feature vector is the loose of some properties of the text. For example in an author attribution problem, the data set holds lots of indicators about the authors, like using a rare word more frequently. In this case such a word should be considered in a distance away from the frequently used words. Unfortunately the hashing algorithms do not deal with the distance of the input. A solution proposed in this study is keeping the substitution path from input to the output level by using only an S-box.

A sample S-Box is demonstrated in Figure 3.

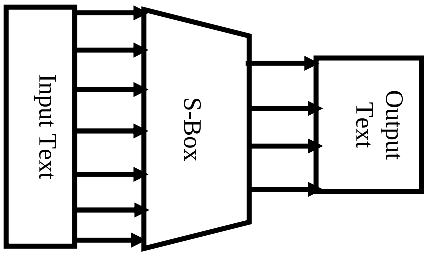

Fig. 3. Generic view of an S-Box

In an S-Box structure an input bit is mapped to an output bit. The bit reduction plays a role, since multiple input bits are connected to a single output bit.

TABLE I. Sample S-Box Table

		First 2 bits			
		00	***01***	***10***	***11***
Last 2 bits	***00***	111	110	101	100
	01	001	010	011	101
	10	101	100	001	010
	11	111	001	101	110

In Table 1, a sample S-Box is deployed with sample values. The s-box is 4 bit to 3 bit reducer in this design and an input with 4 bits will be divided into 2 groups, the first 2 bits and last 2 bits. The table will be crossed using those two values and the cell value in the cross of those values will be the output. For example an input of 1010 will be read as in Table 2.

TABLE II. SAMPLE RUN FOR INPUT 1010 ON S-BOX

		First 2 bits			
		00	***01***	***10***	***11***
Last 2 bits	***00***	111	110	101	100
	01	001	010	011	101
	10	101	100	001	010
	11	111	001	101	110

Table 2 demonstrates the crossing values of first two bits (10) and last 2 bits (10) of the input and the output is 001, which is a three bits output. Obviously the 3 bits output requires a collision and 001 is output for 0001 and 0111 inputs at the same time.

III. A NOVEL FEATURE HASHING FOR TEXT MINING

In this study, we propose a novel feature hashing, built over the s-boxes.

The hashing algorithm we propose is working on the word level and the first letter of the word is kept as unchanged. The rest of the word is passed through an s-box and the output of s-box is concatenated to the end of the first letter of the word.

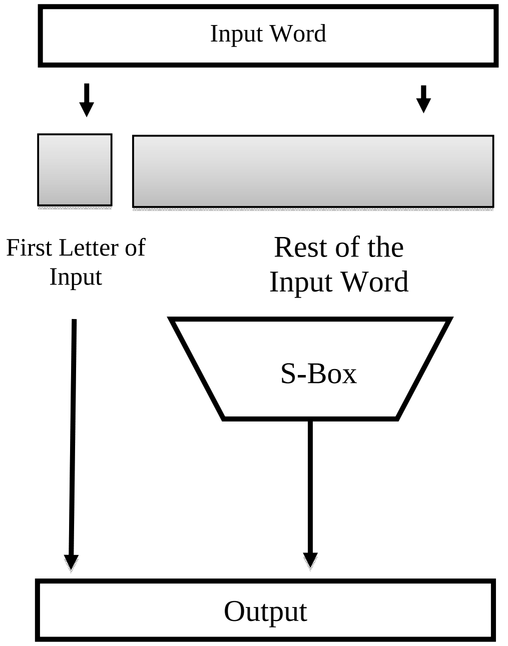

Fig. 4. Novel Feature Hashing Diagram

The execution of algorithm keeps the first letter of each input. We propose to use the hashing algorithm word by word on the text. This is not necessary in fact but gives a performance up to word count which is widely implemented on the feature hashing. The word count is between 100 and 150 thousand of words in most of the text mining studies. Depending on the size of the data source this number can be increased up to 200 thousand of words. The proposed algorithm here reduces the size to 8 bits for each input, which can be calculated as:

8 bits for the first letter + 8 bits output from the S-Box

The possible word count in the novel approach is 216 = 65 thousand possibilities besides the first letter alternatives. The number of output is fixed and cannot increase like in the word count. Besides the number is less than the possible word counts.

IV. EXPERIMENTS

This section explains the methodology of experiments run over the IMDB62 data set and the classification methods applied after the feature extraction methods. In this study two different feature hashing method is directly applied over the plain text.

i) MD5
ii) The Novel Hashing method

This study compares the conventional two hashing methods, MD5 with the novel hashing method proposed.

Finally the evaluation of feature hashing methods is applied on the author recognition via the classification algorithms, k-nearest neighborhood (KNN) [6]. The results are evaluated via the root mean square error (RMSE) [7] and relative absolute error (RAE) [7].

A. *Dataset*

We have implemented our approach onto IMDB62. Table 3 demonstrates the features of the datasets.

TABLE III. SUMMARY OF DATASET

	IMDB62
Authors	62000
Texts per Author	1000
Average number of words per entry	300
Std. Dev. of words per author	198
Number of distinct words in corpus	139.434

In the IMDB62 database, there are 62 authors with a thousand of comments for each of the authors. The database is gathered from the internet movie database[1] which is available for the authors upon request [8].

The dataset is quite well formed for the research purposes. Unfortunately in a plain approach to text mining, like word count, the hardware in the study environment would not qualify

[1] IMDB, internet movie database is a web page holding the comments and reviews of the users and freely accessible from www.imdb.com address.

the requirements for the feature extraction of all the terms in data source which is 139,434 for IMDB data set.

Memory Requirement = 139,434 words x 62,000 posts x 300 average word length x 2 bytes for each character =~4830GByte

The amount required to process the data set via the word counts requires a feature vector, allocating memory for each of the distinct words [9].

After applying the feature hashing methods, the number of bits required can be reduced to quite processable amount. For example, in the novel hashing method, we propose, the number of bits is reduced to 16.

V. Classification

The results collision rate of both hashing algorithms is given in Table IV.

TABLE IV. Hashing statistics

	MD5	**Novel Hashing**
Number of duplicates	31	21833
Number of unique values	61957	40155
Average hash per instance	1.0005	2.025477
Stdev hashper instance	0.04633	1.146775

The low collision rates for MD5 can yield a better result in the name of hashing while the novel hashing algorithm is designed to have collisions in order to see the correlation between the text and the hash result.

The success rates after the classification is given below in Table 5.

TABLE V. Hashing statistics

	MD5	**Novel Hashing**
RMSE	3647286.54	0.49
RAE	120.77	98.17
Success	0.19%	39.95%

The higher success rates are related to the higher collision rate in Table 2.

VI. CONCLUSION

This paper proposes a new hashing algorithm especially for the feature hashing over the text mining applications. Since current hashing algorithms are useful for the collision free hashing on texts, the novel approach only focus on reducing the dimension of the data set.

The experiments on hashing success, shows the novel approach has a weak effect on the coision while this weakness is getting an advantage on the text mining approach with the higher success rate for the classification.

REFERENCES

[1] R. Rivest, “The MD5 message-digest algorithm,” Internet RFC 1321, April 1992.

[2] National Institute of Standards and Technology, “Secure Hash Standard,” FIPS 186-1, US Department of Commerce, April 1995.

[3] H. Feistel, “Cryptography and computer privacy,” Scientific American, vol. 228, no. 5, pp. 15–23, 1973.

[4] Y. Seroussi, I. Zukerman, and F. Bohnert, “Collaborative inference of sentiments from texts,” In UMAP 2010: Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization, pages 195–206, Waikoloa, HI, USA, 2010

[5] K. Nyberg, "Perfect nonlinear S-boxes," Advances in Cryptology - EUROCRYPT '91: 378–386, 1991.

[6] Ibrahim Ocak, Sadi Evren Seker, 2012, "Estimation of Elastic Modulus of Intact Rocks by Artificial Neural Network", Rock Mechanics and Rock Engineering, Vol. 45, issue 6, pp. 1047-1054

[7] Ibrahim Ocak, Sadi Evren SEKER, 2013, "Calculation of surface settlements caused by EPBM tunneling using artificial neural network, SVM, and Gaussian processes", Environmental Earth Sciences, October 2013, Volume 70, Issue 3, pp 1263-1276

[8] Sadi Evren Seker, Cihan Mert, Khaled Al-Naami, Ugur Ayan, Nuri Ozalp, 2013, ”Ensemble classification over stock market time series and economy news”, Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on, pp. 272-273

[9] Sadi Evren Seker, Khaled Al-Naami, Latifur Khan, ”Author attribution on streaming data”, Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on, pp. 497-503