A Novel Feature Hashing for Text Mining

Cihan MERT*
Sadi Evren SEKER**

Abstract

This study focuses on the second group of hashing algorithms and criticizes the hashing algorithms using Feistel Network which are widely utilized in text mining studies. We propose a new approach which is mainly built on the substitution boxes (s-boxes), which is in the core of all Feistel Networks and processes the text faster than the other implementations.

Keywords: data mining, SVM, ANN, KNN, hashing, text mining

Introduction

Feature hashing has a major role in literature especially with increasing studies on big data, such as holding text as a data source. A usual way of text mining on big data mostly requires a layer of feature hashing, which reduces the size of feature vector. For example getting the word count yields hundreds of thousands of features in most of the cases and in some cases, executing some algorithms is impossible with current hardware, where parallel or distributed programming is taken into account. On the other hand taking the POS-tagging would reduce this number into features to about 50. By the feature hashing, the size of feature vector reduces reasonably and data mining processes like classification, clustering or association can run faster.

The feature hashing approaches usually can be categorized into two groups. The first group deals with natural language processing (NLP) algorithms and the mathematical hashing algorithms. While NLP algorithms tries to extract a relatively smarter hash result which represents the input characteristics at maximum, the mathematical hashing algorithms do not deal with the context or meaning of the text input and just processes the input for some binary output. For example POS-tagging approaches can carry on some features of the input to the output; on the other hand, hashing algorithms like MD5 (R. Rivest, 1992) or SHA-1 (National Institute of Standards and Technology, 1995) have no effect on input where they only worry about less collision on the output.

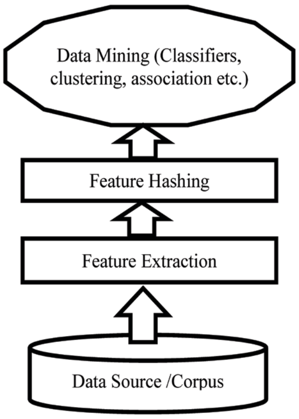

Figure 1: *Generic Deployment Diagram of Text Mining*

Feature hashing studies have a major role in text mining studies. Most of the text mining studies deal with big data “like studies on social networks or web- mining”. A generic deployment diagram of the text mining which uses feature hashing is demonstrated in *Figure1*.

The major feature hashing studies cover the well-known hashing algorithms like MD5 or SHA-1 with most built over the Feistel Network and substitution permutation network (SPN). In the literature, because of the operations held on the Feistel Networks, they are also named as in an SPN (H. Feistel, 1973); the operations of substitution and permutations also takes into account, besides the “shift operations” , “exclusive or” and “and” operations. All those operations are placed in to reduce the risk of collision in the output. The problem in a text mining operation is in reducing the number of collisions which will put two related features into far away and will also cause the reduced neighborhood for most of the data mining tools. For example in a classification algorithm like k-nearest neighborhood or a clustering algorithm like k-means, the distance between instances plays a key role for finding the related instances. Proposed in this study is using the s-boxes in order to reduce the length of the feature vector, while holding some of the input properties.

This paper gives a brief introduction to the background of the study in Section 2. After the introduction and the background, the novel feature hashing algorithm proposed in this study is introduced. Finally, the evaluation results on a sample dataset, IMDB’62, and the conclusion are provided at the end of the paper [Y. Seroussi, I. Zukerman & F. Bohnert., 2010].

II. Background

The concept of Feistel Networks has a major role in feature hashing, including text mining studies. A generic approach to text mining is already provided in Figure 1 and from the fig-

* Lecturer, Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia. E-mail: cmert@ibsu.edu.ge
** Assist. Prof., Department of Business, Istanbul Medeniyet University,Istanbul, TurkeyE-mail: academic@sadievrenseker.com

ure, 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 in reducing the length of the feature vector. The increasing importance of studies involving huge amounts of data has also increased the importance of the size of feature vectors. 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 to the size of the feature vector.

The generic view of a SPN network is demonstrated in *Figure 2*.

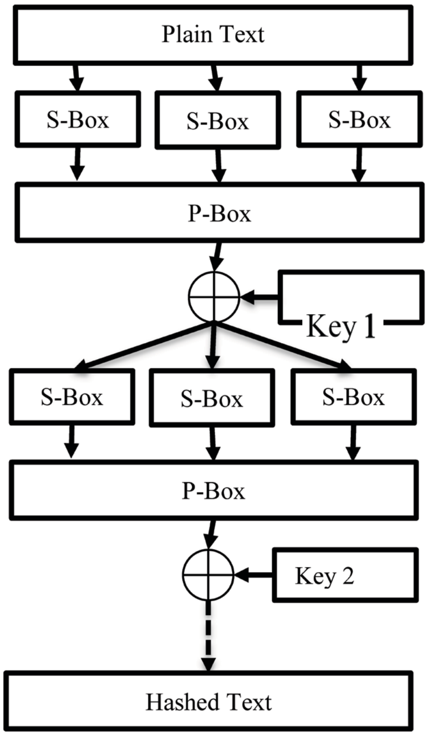

Figure 2: *Generic view of a SPN*

In the SPN, an input text in plain form is reduced in size with the S-Boxes (K. Nyberg, 1995) (substitution boxes) and mixed with the P-Boxes (permutation 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 loss of some properties of the text. For example, in an author attribution problem, the dataset holds lots of indicators about the authors, like using a rare word more frequently. In this case such a word should be considered 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 to keep the substitution path from input to the output level by using only an S-box.

A sample S-Box is demonstrated in *Figure 3*.

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.

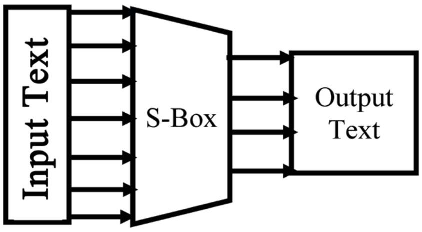

Figure 3: *Generic view of an S-Box*

III. A Novel Feature Hashing for Text Mining

In this study, we propose a new hashing network which is built from only the substitution boxes and we remove the permutation boxes from a classical SPN, since the purpose of permutation boxes is to reduce the number of collision.

The generic view of the novel hashing method proposed in this study is demonstrated in *Figure 4*.

The novel approach has a connection map from a single bit in the plain text to the hashed text. Also the keys provided on each step via a key generation algorithm play a major role in the success of the hashing method. We propose a simplified key generation algorithm as in *Figure 5*.

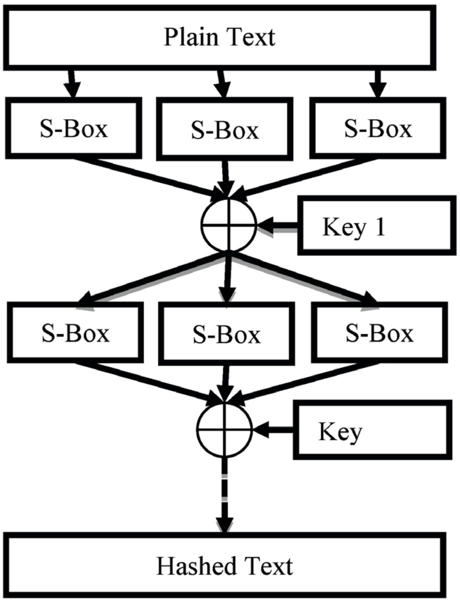

Figure 4: *Generic view of a Novel Hashing*

In Figure 5, the key generation algorithm keeps running for each step of the novel hashing method. From each step the key generation algorithm uses a shift operation (not a cyclic shift but a regular shift where the number of bits are reduced) and the key for the pass is generated. Since the number of bits from each s-box pass in the hashing algorithm is reduced, the key size is also reduced by the key generation algorithm in each step.

IV. Sample Run

This section describes a brief sample run for a minimized input string.

i) The input is divided into two parts (Left :L and Right: R)

ii) Each part (both L and R) is given to the S-Boxes as parameters.

iii) Each output from the S-Boxes is collected into a single string.

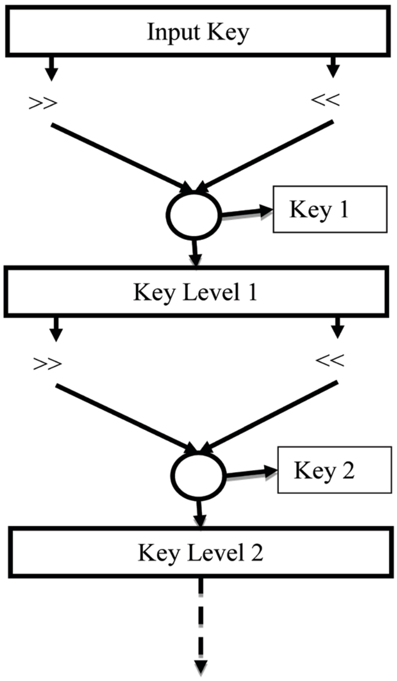

***Figure 5:** Simplified Key Generation Algorithm*

iv) The Key 1, generated from the key generation algorithm is taken into XOR operation with the collected string in step iii.

v) Result is again given as an input to the step i until the desired length of output is reached.

Also the key generation algorithm can be listed in a step by step view as below:

i) Get input key string and divide it into two parts (Left: L and Right: R)

ii) Use right shift operator on the left part and left shift operator on the right part simultaneously.

iii) Concatenate both parts and use the result as Key 1.

iv) Keep applying the steps from i to the output until the number of passes are completed.

A sample run with a numeric input is given below:

i) Let's consider the input value as 10110101

ii) Let's consider the key string value as 10111101

iii) The key is divided into two parts L_{key}: 1011 and Rkey:1101

iv) The first key is concatenation of L_{key} >> 1 = 1011>> 1 = 101 and Rkey >> 1 = 1101 >> 1 = 110, which is 101 110.

v) Let's assume the s-box result for the Linput = 1011 is 110 and the s-box result for the Rinput = 0101 is 100.

vi) The first pass result will be $L_{input}R_{input}$ XOR $L_{key}R_{key}$ which is 110 100 XOR 101 110 = 011010

vii) The output achieved at this level is a 6bits reduced version of the 8bits input. The algorithm can keep iterating until reaching the desired output size.

In this study, we have reduced the inputs into 16bits outputs. The number of inputs is not important in the implementation since the output can be limited. For example an author with a single word or thousands of words can be reduced into the same output size. Also take into consideration that the size of 16 bits is a size of two characters and the whole post of an author is reduced into the size of two characters.

Another implementation fact about the hashing algorithm is a classical block hashing problem. The last block of the input may not fit into the s-box length. In this case we use the bit padding where the first bit is 1 and the rest of bits are 0 until the length of the s-box input size is filled.

For example a system with 8 bits s-boxes can process a 13 bits input string with adding 3 bits to the end of the input. A numerical example would be an input of "1100110011001" and the bit padding would result in an input of "1100110011001100" since the last 3 bits are filled with "100".

V. Experiments

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

i) MD5
ii) The Novel Hashing method

This study compares the conventional two hashing methods, MD5 and the novel hashing method proposed. Finally the evaluation of feature hashing methods is applied to the author recognition via the classification algorithms, k-nearest neighborhood (KNN) (I. Ocak, S. E. Seker, 2012). The results are evaluated via the root mean square error (RMSE) (I. Ocak, S. E. Seker, 2013) and relative absolute error (RAE) (I. Ocak, S. E. Seker, 2013).

VI. Dataset

We have implemented our approach onto IMDB62. Table 1 demonstrates the features of the datasets.

In the IMDB62 database, there are 62 authors with a thousand comments for each of the authors. The database is gath-

***Table 1:** 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

ered from the internet movie database which is available for the authors upon request.

The dataset is quite well organized for research purposes. Unfortunately in a plain approach to text mining, like word count, the hardware in the study environment would not qualify the requirements for the feature extraction of all the terms in data source which is 139,434 for IMDB[1] dataset.

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 dataset via the word counts requires a feature vector, allocating memory for each of the distinct words. After applying the feature hashing methods, the number of bits required can be reduced to quite a processable amount. For example, in the novel hashing method, we propose, the number of bits be reduced to 16.

VII. Classification

The results of the collision rate of the both hashing algorithms are given below:

Table 2. *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 are given in *table 3*.

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

Table 3: *Error and Success rates of Classification Methods.*

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

Conclusion

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

The experiments on hashing success show the novel approach has a weak effect on the collision while this weakness is getting an advantage on the text mining approach with a 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] Yanir Seroussi, Ingrid Zukerman, and Fabian 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] Kaisa Nyberg . “Perfect nonlinear S-boxes” . Advances in Cryptology - EUROCRYPT ‘91 : 378–386, 1991.

[6] I. Ocak, S. E. SEKER (2012), Estimation of Elastic Modulus of Intact Rocks by Artificial Neural Network, Rock Mechanics and Rock Engineering, Springer, DOI: 10.1007/s00603-012-0236-z,

[7] I. Ocak, S. E. SEKER (2013), Calculation of surface settlements caused by EPBM tunneling using artificial neural network, SVM, and Gaussian processes, Environmental Earth Sciences, Springer-Verlag, DOI: 10.1007/s12665-012-2214-x,

[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.