Fast *n*-gram Assortment Construction for Filtering Hazardous Information

TADASHI YANAGIHARA,[†1] KAZUSHI IKEDA,[†1] KAZUNORI MATSUMOTO[†1]
and YASUHIRO TAKISHIMA[†1]

Document filtering systems based on pattern matching require well-chosen features to provide high accuracy. Using high level features such as bi-grams can boost accuracy, but require large amount of calculation time to find the optimal bi-gram set. In this paper, we propose a method to find an approximate optimal bi-gram set from a given set of uni-grams, while ensuring the selected assortments are independent from one another. Our method drastically reduced the amount of calculation time in comparison of using exhaustive calculation methods, while maintaining a much higher accuracy in terms of precision.

1. Introduction

The internet has now become an essential part of society, providing a platform for quick and reliable communication between users. However, the increase of the usage of the internet has also brought up new problems for the society to face, some of which are the use of the internet to spread hazardous information or promote crime acts. Internet service providers or monitoring services have introduced manpower-based monitoring solutions to detect such information, however, the increase of the variety and amount of hazardous information created each day is too fast and too costly to be solved using such methods.

In some cases, machine-based solution such as document filtering systems have been introduced to help reduce human monitoring costs. Provided with a list of features, the machine-based solution can automatically tag the data on whether the data is to contain hazardous data or not. This tag can be used to control the amount of documents which needs to be checked by human.

This machine-based solution can be treated as a type of topic detection problem. In a topic detection problem, feature sets are *n*-grams, most commonly keywords which are supposedly found in text data of a specific class. Using a feature set which is relevant to hazardous information, one can determine whether a document is hazardous or not. Therefore, we must find an effective way to provide a proper list of features for the system to use.

In the remaining part of this paper, we will discuss on further detail of this topic detection problem. In Section 2, we will define our problem to clarify what features will be necessary to solve our problem. In Section 3, we will show our proposed method, which calculates an approximate list of bi-grams constructed from a list of uni-grams, which will cut down the calculation time drastically in comparison to exhaustive methods and ensure the independency between selected features. In Section 4, we will compare our proposed method against a standard exhaustive method in terms of performance and accuracy.

†1 KDDI R&D Laboratories

2. Problem Definition

A document filtering system is a system which uses a given feature to decide whether a document is hazardous or not. The feature set set itself may be obtained manually by hand or automatically, often by observing training data to search for features which are commonly found in documents containing hazardous information.

Once the feature sets have been selected, the software filter can label documents on whether they are hazardous or not. After the documents have been labeled, the documents are handed over to human for viewing, to validate the correctness of the label. Figure 1 is an illustration of how a document filtering system is applied.

2.1 Issues in document filtering

In document filtering, there are several issues or characteristics to solve in order for such systems to be practical in reality. Such issues are being the large amounts of training data and testing data, also to mention the preference towards a higher precision than higher recall. We will discuss on each issue in more detail below.

Evaluation Data

Monitoring data from the web involves the viewing of thousands to millions of documents which are created everyday. This means the document fil-

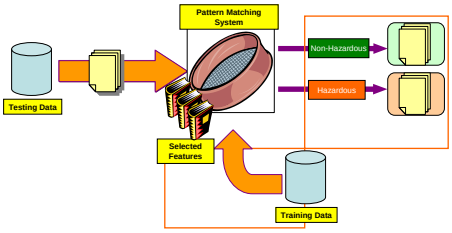

Fig. 1 System Overview

tering system must accurately label multiple documents per second. Classifers such as Naive Bayes Classifiers or Support Vector Machines have been known to be accurate and fast with separating datas into classes, however, all data must be transformed into feature vectors before evaluation, resulting in a large overhead per document. To prevent such overhead, we suggest using fast pattern matching methods provided with high level features such as bi-grams.

Training Data

Expressions found in hazardous documents change rapidly, quite often from day to day, meaning old feature sets can easily become out of date. Feature sets must be renewed oftenly, most often everyday. This means the creation of the features must be fast as well as being effective.

Precision

Since the aim of the document filtering system is drastically reducing the amount of documents which need to be monitored by human, we must consider on raising the precision. This means lowering the error on false positives. Usage of high level feature sets such as bi-grams instead of unigrams can help reduce false positives, yet they still generate many false positives. This is because most of the selected features are redundant, which contribute to over-labeling non-hazardous data as being hazardous. We propose on preventing redundant features to be considered as well when selecting key features.

2.2 Proposed Method

To solve the issues found in document filtering, we propose a method to create fast and efficient *n*-grams assortments to be used as features. Our method uses an approximate search method to search selected uni-grams in order to create efficient bi-grams, while also ensuring the independance of each selected feature. Our approximate search method requires uni-grams which have been generated from common feature selection methods[1)], and then combined with the remainder unigrams to form bi-grams. Bi-grams are then passed through feature selection methods once again to estimate which bi-gram is most relevant to deciding whether a document is hazardous or not.

We consider for feature indepencency for the following reason. When selecting uni-grams using feature selection, each feature does not consider the feature being selected before, that is, such methods select features which alone are most relevant. This would mean the selected features will show large coverance of the training data, resulting in a high recall, yet at the same time result in a high overlap of features which will lower the precision. Our proposed method ensures feature independence, by recursively removing documents containing the selected feature from the training data. In this way, we can reduce redundant features which cover the same documents.

3. Formula Description

We will be using the calculation procedure which was illustrated in reference 2). The following description is on how we applied the method onto the hazardous document problem.

3.1 Selecting Uni-grams

First, we require selected uni-grams which are highly relevant to the topic which we wish to detect. Since selecting uni-grams based on their term-frequency or document-frequency tend to create false positives, we will use model fitting methods to choose the uni-grams. For our current implementation, we have used a score calculation scheme based on Akaike Information Criteria (AIC)[3)], since this method does not need any external parameters to calculate scores.

To conduct model fitting, we first need to define the various values needed to create a 2x2 contingency table.

- n_{11} : Number of hazardous documents containing term T.
- n_{10} : Number of non-hazardous documents containing term T.
- n_{11} : Number of hazardous documents not containing term T.
- n_{11} : Number of non-hazardous documents not containing term T.

Given these variables,

- $n_{1.}$: Number of documents containing term T.
- $n_{0.}$: Number of documents not containing term T.
- $n_{.1}$: Number of hazardous documents.
- $n_{.0}$: Number of non-hazardous documents.

Finally,

- n : Total number of documents.

Shown below in table 1 is a typical 2x2 contingency table used in model fitting. In this example, we are comparing to see whether term T found in documents contribute towards the document being hazardous (C) or not hazardous($\neg C$). We first declare variables as follow:

	C	$\neg C$	
T	n_{11}	n_{10}	$n_{1.}$
$\neg T$	n_{01}	n_{00}	$n_{0.}$
	$n_{.1}$	$n_{.0}$	n

Table 1 Contingency Table for uni-grams

Next, we use the values found in the 2x2 contigency table to conduct model fitting. In the case of AIC, we test two hypothesis, being the feature found in the documents is independent or dependant to the document being hazardous or not.

To test whether the term appearance is an independent situation, we use the following formula.

$$MLL = {}_{n1.}\log_{n1.} + {}_{n0.}\log_{n0.} + {}_{(n-n1.)}\log_{(n-n1.)} + {}_{(n-n0.)}\log_{(n-n0.)} - {}_{2n}\log_{n} \quad (1)$$

$$AIC(IM) = -2 \times MLL + 2 \times 2 \quad (2)$$

To test whether the term appearance is an independent situation, we use the following formula:

$$MLL = {}_{a}\log_{n11} + {}_{n10}\log_{n10} + {}_{n01}\log_{n01} + {}_{n00}\log_{n00} - {}_{n}\log_{n} \quad (3)$$

$$AIC(DM) = -2 \times MLL + 2 \times 3 \quad (4)$$

Finally, we take the difference in the obtained scores to achieve score E. This score represents on how relevant the uni-gram is towards the topic, in this case, towards the dependence on whether the document being hazardous or not.

$$\frac{n_{11}}{n_{1.}} > \frac{n_{01}}{n_{0.}} \rightarrow E = AIC(IM) - AIC(DM) \quad (5)$$

$$\frac{n_{11}}{n_{1.}} < \frac{n_{01}}{n_{0.}} \rightarrow E = AIC(DM) - AIC(IM) \quad (6)$$

After comparing all calculated E, we select the uni-gram T with E above 0 to add to our list of unigrams.

3.2 Creating Bi-grams

Once we have been supplied with a set of unigrams, we can start creating bi-grams. We first select uni-grams with the k highest E and combine each of the selected uni-grams with all unigrams in the set to create assortment A. We will then create a new 2x2 contingency table, seperate from the one we have used to create the uni-grams. The redefined values are as follows:

- n_{11} : Number of hazardous documents containing assortment A.
- n_{10} : Number of non-hazardous documents containing assortment A.
- n_{11} : Number of hazardous documents not containing assortment A.
- n_{11} : Number of non-hazardous documents not containing assortment A.

Given these variables, the following variables can be defined.

- $n_{1.}$: Number of documents containing assortment A.
- $n_{0.}$: Number of documents not containing assortment A.

In the following 2x2 contingency table shown below in table 2, we are comparing to see whether assortment A found in the documents contribute towards the document being hazardous (C) or not hazardous($\neg C$). We first declare variables as follow:

	C	$\neg C$	
A	n_{11}	n_{10}	$n_{1.}$
$\neg A$	n_{01}	n_{00}	$n_{0.}$
	$n_{.1}$	$n_{.0}$	n

Table 2 Contingency Table for bi-grams

The calculation of each assorment(A)'s score E is conducted the exact same way as we calculated the score E of each uni-gram T. After comparing all calculated E, we select the assortment A with the highest E to add to our list of bi-grams to be used. Before we resume with the next uni-gram T to use to select bi-grams, we remove documents containing the selected assortment A and recalculate n_{11}, n_{10},... once again before moving on to the next unigram. This ensures the selected features are independent with one another to avoid redundancy.

4. Evaluation

We will evaluate our algorithm against an exhaustive bi-gram calculation method, which is a simple calculation which tests all possible combinations of uni-grams to form bi-grams, then use model fitting to add scores to all obtained bi-grams. We have implemented both methods using GNU C compiler (gcc) and ran tests using a Dual Pentium Xeon with 32GB RAM Linux machines.

4.1 Data

For evaluation data, we have acquired 2 month of blog data which was available to public during December 1st, 2008 to January 31st, 2009. All data was categorized by human on whether the blog data contained hazardous information or not. The breakdown statistics of the acquired data on both hazardous data (C) and non-hazardous data ($\neg C$) are shown in table 3.

	C	$\neg C$	total
Dec 2008	173,465	1,321,127	1,494,592
Jan 2009	141,858	1,944,858	2,086,716

Table 3 Breakdown of Hazardous and Non-Hazardous Data

4.2 Calculation Time and Memory Usage

To evaluate the efficiency of both methods, we have measured the amount of calculation time and memory usage required to calculate the optimal bi-grams. For our proposed method, we used $k = 1000$ for evaluation.

Calculation Time

The calculation time required to select the optimal bi-grams are shown in table 4. We have found our proposed method to be much faster, up to 11% of an exhaustive approach. When using 2 month data, the calculation time took almost 17 hours, where as the proposed method took less than 2 hours.

	exhaustive	proposal
1 month data	34497 sec	3850 sec
2 month data	60942 sec	6939 sec

Table 4 Calculation time

Memory Usage

We show in table 5 the maximum memory usage of each method. The exhaustive method required 16.6 GB with the 1 month data and 27.05 GB with the 2 month data. With the proposed method, the maximum memory usage was reduced to 0.65 GB with 1 month data, while the 2 month data required 1.11GB. This shows our proposed method requires 2.4 to 4.1% of the memory which was required in an exhaustive method.

	exhaustive	proposal
1 month data	16.62 GB	0.65 GB
2 month data	27.05 GB	1.11 GB

Table 5 Memory usage for selecting feature set

4.3 n-fold Cross Validation

To evaluate the accuracy of each method, we conducted an n-fold cross validation test using the 2 month data set. The data was separated into small data sets based on the day the data was created. We then randomly selected one of the small data sets to use as testing data, and the remaining data sets as training data. This test was repeated 5 times.

Figure 2 shows the precision and recall of an exhausted method and our proposed method. We took the number of documents which were labeled in all 5 tests and used the average numbers to calculate the precision and recall. In total, an exhaustive method ended with a precision of 35.4% and recall of 90%, whereas our proposal method ended with a precision of 98.02% and a recall of 85.3%.

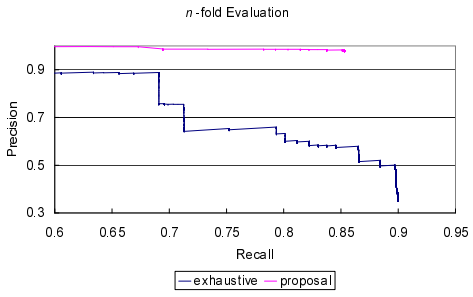

Fig. 2 n-fold evaluation of exhaustive and proposed method

5. Discussion

From the results, we find bi-grams to be highly accurate to find hazardous information, in fact much higher than other topic detection problems. One issue to cinsider is the amount of duplicate entries found in the data set. For example, 18% of the

blog data were duplicate copies of another blog data found in the same data set. We believe there are even more near-duplicate copies in the same data set. Such blogs are likely to be ”spam blogs”, duplicate text data including hyperlinks. We wish to investigate more on the actual amount of spam blogs and non-spam blogs which were labeled hazardous and measure the accuracy of each blog data.

Conclusion

Our contribution in this paper as follows. We have defined the problem of providing high level feature sets such as bi-grams to document filtering systems. We have illustrated a fast and efficient method to aquire bi-grams, while ensuring the independence of the extracted features. Our experiments showed our method could find optimal bi-grams in 10% less computation time and 2.4 to 4.1% memory usage of the exhaustive method. In the *n*-fold evaluation test, our method had 63% better precision than using a simple exhaustive bi-gram method.

Acknowledgments We thank Shigeyuki Akiba, Shuichi Matsumoto, and Fumiaki Sugaya from KDDI R&D Laboratories Incorporated for their comments. Part of this work was conducted on research aid of ‘Detection of Illegal and Harmful Information on the Internet’ by the National Institute of Information and Communications Technology (NiCT).

References

1) C. Manning, P. Raghavan, H. Schutze, ”Introduction to Information Retrieval”, Cambridge University Press, 2008
2) T. Yanagihara, K. Matsumoto, C. Ono, Y. Takishima. “Applying *n*-gram Assortment Methods for Topic Determination”, In 7th Information Science and Technology Forum (FIT2008), Vol. D, pp. 59-61, 2008
3) Kazunori Matsumoto and Kazuo Hashimoto. ”Schema Design for Causal Law Mining from Incomplete Database.” In Discovery Science, Second International Conference, Vol. Lecture Notes in Computer Science 1721 Springer, pp. 92-102, 1999. Massachusetts University: ”Using Bigrams in Text Categorization”