Hot Topic Detection in Chinese Web Forum Using Statistics Approach

Xiaoyu Li, Guanzhong Dai, Shuang Lai, Hang Dai
College of Automation
Northwestern Polytechnical University
Xian, China

***Abstract*—In this paper we propose a statistics approach for hot topic detection in Chinese web forum. In order to solve the fundamental obstacles of Chinese web data mining, such as new words, nonstandard syntax and Chinese word segmentation, we present the longest common segmented consecutive subsequence (LCSCS) and other techniques. The algorithm can run even without prior knowledge. Our experiments show the satisfying results both in performance and quality.**

Keywords-hot topic detection; Chinese web forum; web data mining

I. INTRODUCTION

Topic Detection is one of the basic tasks of Topic Detection and Tracking (TDT). Quite a lot of techniques have been developed since the TDT emergence[1], which was sponsored by DARPA and run by NIST, started in 1997. Some methods[2], such as Self-organization Neural Network[3], BBN[4], have achieved fine results in dealing with newspaper documents in both English and Chinese[5, 6].

However, little work has been done in processing web forum documents, especially those written in Chinese. There are several basic obstacles. First, the syntax is usually ignored in web forums when people write their posts. There are many non-standard or even mistaken expressions in posts. Second, large numbers of new words, which have not been listed in the dictionary, are used in forums. And the numbers of this kind of words grow everyday. Third, there is no natural delimiter between Chinese words, which can be made up by one or several Chinese characters. It is difficult to segment words from a Chinese character sequence.

In this paper, we propose a statistic approach for detecting hot topics in a Chinese web forum. This method computes the common parts of every two posts, and counts the repeated parts for building hot topics. No dictionary or word segmentation is needed. And the algorithm can run even without prior knowledge, which can be collected automatically.

This paper is organized into three main sections. First, we give the basic definitions and presumptions of our study. The second section contains the specific techniques we use to perform Chinese web forum hot topic detection. Finally, we present our research result and analysis in the third section.

II. DEFINITIONS AND PRESUMPTIONS

TDT defines "topic" to mean a specific event or activity plus related events or activities4. So the "hot topic" can be considered as those topics arousing intense interest, excitement, or controversy. We presume "hot topic" in web forum has following characters: "Hot topic" shall be discussed in a mass of posts, "Hot topic" can be obtained from the titles of posts, "Hot topic" can be expressed in character sequence extracted from posts titles.

We hold this kind of belief: if something is really a hot topic in a web forum, there must be a lot of people writing posts to discuss it. And most of these people will refer to this topic in their post titles. This frequently used character sequence in titles just indicates the hot topic in this web forum. The rationality of our presumptions is supported by our experiment results.

As we mentioned above, the essential part of our hot topic detection technique is to find the common part of the titles. There are many methods to define the common part of titles. The most widely used concept is the longest common subsequence (LCS). However, it could lead to absurd results when applied in Chinese, because single Chinese characters generally can not express a complete concept.

So we present the longest common segmented consecutive subsequence (LCSCS) to solve this problem.

Definition 1. Given a sequence $X=\{x_1,x_2,\cdots,x_n\}$, another sequence $Y=\{y_1,y_2,\cdots,y_m\}$ is a segmented consecutive subsequence of X if there exists a strictly increasing sequence of $\{i_1,i_2,\cdots,i_m\}$ of indices of X such that for all $j=1,2,\cdots,m$, we have $x_{i_j}=y_j$, and $i_j=i_{j-1}+1$, or $i_j=i_{j+1}-1$. Given two sequences X and Y, a third sequence Z is a common segmented consecutive subsequence of X and Y if it is a segmented consecutive subsequence of both X and Y. Z is the longest common segmented consecutive subsequence (LCSCS) of X and Y if there is no other common segmented consecutive subsequence of X and Y longer than Z.

By using LCSCS, we achieved much more rational result than LCS.

III. HOT TOPIC DETECTION TECHNIQUES

1) The Algorithm for LCSCS problem

The definition of LCSCS is quite similar to LCS except it has one more restriction. So the algorithm for LCSCS can be implemented based on the one for LCS. Wagner7 introduced the dynamic programming algorithm to solve LCS problem, whose time complexity is O(mn). Better performance has been achieved by using other algorithms8, 9. However in our application, the length of post titles is not long, so the difference is not distinct. The algorithm in Figure 1 computes the length of the longest common segmented consecutive subsequence (LLCSCS) of two given string x and y, based on Wagner's one.

```
LLCSCS (x, y)  ▷  m = |x|, n = |y|, T = [t_{i,j}]_{(m+1)×(n+1)}
1   begin
2     t_{i,j} ← 0
3     α ← FALSE , β ← FALSE
4     for i = 1 to m do
5       β = FALSE
6       for j = 1 to n do
7         β ← (x_i = y_j)
8         if (x_{i-1} = y_{j-1}) ∧ (α ∨ β) then
9           t_{i,j} ← t_{i-1,j-1} + 1
10          β ← α ∨ β
11        else if t_{i-1,j} ≥ t_{i,j-1} then
12          t_{i,j} ← t_{i-1,j}
13        else t_{i,j} ← t_{i,j-1}
14      α ← β
15    return t_{m,n}
16  end
```

Figure 1. The LLCSCS algorithm

```
LCSCS (x, y)
▷  m = |x|, n = |y|, l = LCSCS(x, y) AND k = |l| = t_{m,n}
1   begin
2     while k > 0
3       if t_{i,j} = t_{i-1,j} then i ← i − 1
4       else if t_{i,j} = t_{i,j-1}, then j ← j − 1
5       else
6         k ← k − 1
7         l_k ← x_{i-1}
8         i ← i − 1
9         j ← j − 1
10    return l
11  end
```

Figure 2. The LCSCS Algorithm

By inspection, we see the time complexity to compute the LCSCS of two titles x and y is $O(|x|\times|y|)$. Because the title length in a forum is always limited, the time can be taken as a constant.

2) Finding LCSCS in Large Numbers of Titles

Given N titles, we define something is a hot topic if it is referred in more than K titles. The most direct way is to compute the LCSCS of every combination of K titles. However its time complexity is C_N^K and not practical in use. We solve this problem in another way. First the LCSCS of every title pair is computed, and then those LCSCS' which repeat in more than K titles are selected. So the time complexity is limited to C_N^2.

Strictly speaking, the results of the two methods are different, especially when the LCSCS' of title pairs is long, for the LCSCS of multiple titles is a subsequence of the LCSCS' of title pairs.

However the experiments show that more than 80% of the LCSCS' of title pairs contain only two characters. Most of them are quite short. So the results of the two methods are similar, and our method is much faster. To get the more accurate results, we recount the selected LCSCS' in all titles one by one. It spends $O(N)$ time because the number of LCSCS' which are chosen in previous step is quite small.

3) Filtering Writing Mistakes

One of the basic obstacles of web data-mining is there are usually some people making writing mistakes in their posts. It may be a Chinese character with the similar pronunciation or writing method. This kind of mistakes can bring negative effects to our hot topic finding task.

We solve this problem in the following way. It is easy to see that if topic A repeats X times in all titles, B repeats Y times, and B is a subsequence of A, the actual repeating times of topic B is Y-X. So first we sort all the selected LCSCS' by length, from short to long. Then we check each pair of LCSCS' with different length. If the shorter one is a subsequence of the longer one, its repeat times will decrease. After this operation, all of the LCSCS' which repeat less than K times will be filtered. Here, we presume that most of the people can write their topic correct.

4) Stop-Words List and Assistant-Words List

Many meaningless LCSCS' are still left after previous steps. Most of them are interjections, conjunction, adverbs, and pronouns. We build a small stop-words list to filter these words. Some words which are frequently used in forums are also included in stop-words list.

Another situation may happen. Imaging a forum that talks about the relationship between China and US, it is not surprising that the words "China" and "US" will appear frequently. This kind of words contains some meaning but little new information. We put these LCSCS', which appear frequently almost every day, into the assistant-words list. These words can be collected by the computer automatically.

5) Association Analysis and Topic Net

After filtering, only a small number of LCSCS' are left for building hot topics. Let L to be a set which contains all of these LCSCS'. Taking every title as a transaction, and every LCSCS in L as an item, association rule mining can be done within these data.

In fact, the L is already the frequent 1-item set with minimal support K/N. We can find the large item set using the classic algorithms10. However our experiments show that there is no frequent item set which contains 3 or more items, unless we set the threshold to be unreasonably small. Generally speaking, 2-item set is enough. The association rules can be generated with the given minimal confidence. Almost no rules between 2-item sets and little between 1-item set and 2-item set are found in our experiments. So we simply get the frequent 2-item sets at last.

A topic net can be built if the elements of all frequent 2-item sets are linked. Those directly linked LCSCS', which refer to some specific event, person and related ones, is the hot topic in this forum.

B. *Experiments and Analysis*

We do our experiments with two data sets which are extracted from Qiang-Guo Forum (www.qglt.com). One data set contains 116,431 posts, which were written from Dec 11th, 2002 to Sep 10th, 2003; the other contains 1,802,548 posts, which were written from Dec 2nd, 2002 to 10th, 2003.

1) *Performance Test*

The computer used to do the experiments is a PC with 2.8GHz CPU and 2048M DDRIII Memory. The software is written in Java. The operating system is Windows 2003 Server. Multi-thread technique is employed in the algorithm implementaion.

First, the time complexity is tested. We randomly choose some continuous posts and to see how long it will take to get the hot topics. Second the space complexity is estimated. The most space-consuming step is to store all the LCSCS' of title pairs. We count the number of LCSCS' with nonzero length. The result is shown at table 1.

TABLE I. PERFORMANCE TEST

N (the amount of titles)	500	1000	1500	2000	2500	3000
T (time, in seconds)	0.7	2.5	6.0	10.4	17.2	26.3
M (nonzero LCSCS)	7892	39318	79149	132302	205313	291688
$coup(N) = M/C_N^2$	0.0633	0.0787	0.0704	0.0662	0.0657	0.0648

The experiments show that the time and space complexity of our hot topic finding algorithm is $O(n^2)$. One notable thing is that the ratio between the amount of nonzero LCSCS' and the title pairs vibrates only in a small range. This ratio can describe the coupling degree of topics within posts. We assume this ratio is nearly a constant in the same forum, and the experiments with different data sets confirm our thoughts. The possible reason is that a mature web forum has a relatively stable user group.

2) *Hot Topic Detection*

We choose all the posts, which were written from Dec 23rd, 2002 to Dec 29th, 2002 to do our experiments. Total 1908 posts are selected. The result is shown at figure 3. The points in it present the LCSCS'. A LCSCS is an "assistant word" if it is written in "[ ]". The number written in a "<>" which follows the Chinese characters is the repeating time of this LCSCS. The threshold of the minimal support is set to 0.01.

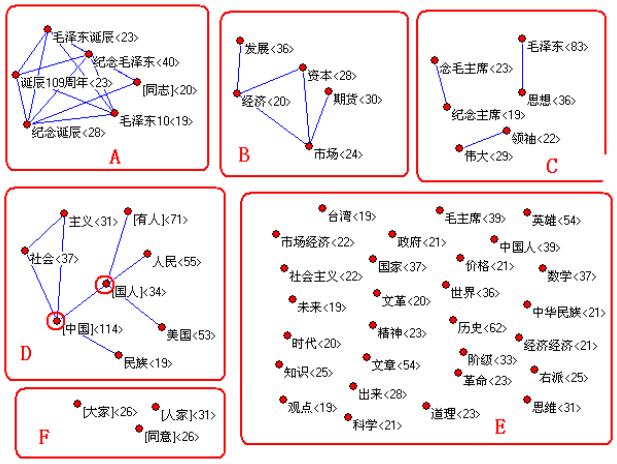

Figure 3. Topic Net of Posts

One of the hot topics, the birthday of Mao Zhe-Dong (Dec 26th), is detected by our algorithm. The related events are connected in region A. Another hot topic which talks about economy, market and capital is shown in region B. Region C contains some topics made by two LCSCS'. Region D cannot be taken as one topic for its hub points, which are circled, are assistant words. So it must be divided into several topics. Region E contains lots of isolated topics. The LCSCS' in region F can notbe taken as topics for they are assistant words.

Similar experiments have been done and the hot topics in the forum such as SARS, Iraq War, are all detected by our Algorithm.

IV. CONCLUSION AND FUTURE WORK

In this paper, we present a statistic approach for hot topic detection in Chinese web forum. We introduce LCSCS and other techniques to overcome the basic obstacles of Chinese web data-mining: new words, non-standard syntax and Chinese word segmentation. The time and space complexity of the algorithm is $O(n^2)$. Satisfied results have been achieved in our experiments. This method has been used in our study of hot topic detection, user interest discovery and common interests

finding within a virtual community. It should be noted that this algorithm is based on our presumptions described in the second section. It is possible that some web forums are not satisfied with these presumptions and cannot be dealt with this algorithm. And some meaningless sequences may be taken as hot topics mistakenly. However, in most cases, this algorithm, which uses purely statistic approach, can work fast and accurately.

In future work, we would like to investigate more ways to build topics based on the founded LCSCS', for the topic net is not understandable enough for ordinary user. The public opinions about hot topics would also be concerned. And the topic tracking is also included in the future plans.

REFERENCES

[1] The web site of TDT. http://www.nist.gov/speech/tests/tdt/.

[2] J. He, A. Tan, and C. Tan, "A Comparative Study on Chinese Text Categorization Methods", In Proceedings of PRICAI'2000 International Workshop on Text and Web Mining, 2000, pp. 24-35..

[3] K. Rajaraman and Ah-Hwee Tan. "Topic detection, tracking and trend analysis using self-organizing neural networks", In Proceedings of PAKDD'2001, 2001, pp.102-107,

[4] T. Leek, H. Jin, S. Sista, and R. Schwartz. "The BBN crosslingual topic detection and tracking system". In Working Notes of the Third Topic Detection and Tracking Workshop, Feb. 2000.

[5] Charles L. Wayne. "Topic detection and tracking in English and Chinese", In Proceedings of the fifth international workshop on on information retrieval with Asian languages, Nov. 2000, pp.165-172,

[6] C.L. Wayne, "Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation". In Proceedings of the Language Resources and Evaluation Conference (LREC), 2000.

[7] Wagner, R.A., Fischer, M.J. "The String-to-string Correction Problem". Journal of the ACM 21, 1974, pp.168-173,

[8] M. Paterson and V. Danc'ik. "Longest common subsequences", In Proc. 19th International Symposium on Mathematical Foundations of Computer Science, 1994, pp.127-142,

[9] Daniel Kunkle, Empirical complexities of longest common subsequence algorithms. 2002. http:// www.redfish.com/dkunkle/mypapers/lcs.pdf.

[10] J. Hipp, U. Guntzer, and G. Nakaeizadeh, "Algorithms for Association Rule Mining - A General Survey and Comparison". In Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000