Exploration of the Weibo Data Mining Strategy Based on an AI Inference Engine

Gong Jun
Ningbo Dahongying University, Ningbo, Zhejiang, 315175, China;
*gongjun0824@qq.com

Abstract
Based on an artificial intelligence (AI) inference engine, an automatic learning and inference system can be built up to realize data mining of Weibo (a Chinese version of Twitter). This paper first conducts a demand analysis of the automatic learning and inference system, and then studies the process design, layer design and function design of the automatic learning and inference system. After that, the method to realize the automatic learning and reasoning system is explored. Test results suggest that the Weibo data mining strategy based on the AI inference engine can achieve favorable outcomes. ALRS can help realize basic functions of the AI inference engine, dig tendencies of Weibo users, and recommend friends. Moreover, it can increase work efficiency and convenience of backend operation maintenance personnel.

Key words: Artificial Intelligence (AI) Inference Engine, Data Mining Technology, Automatic Learning and Inference System, ALRS

1. INTRODUCTION

In a wide range of Internet applications, various platforms have launched their own Weibo accounts, such as Sina Weibo, Tencent Weibo and Sohu Weibo. All these Weibo platforms have become important media for Chinese netizens to share their personal ideas. Currently, the data mining technology has been widely used in Weibo platforms. Data mining is generally realized in five steps, namely proposal of questions, collection of data, data mining, result evaluation, and model discovery.

The automatic learning and inference system can be built based on artificial intelligence (AI) extraction of inference rules, operation interface of some third-party spaces and uncertainty inference operation. The system can be used to achieve automatic attenuation or gain adjustment of the operation process. In this way, users can get better use experience; the backend personnel maintenance cost can be reduced; and services can be provided on a more accurate basis. The automatic learning and inference system is mainly targeted at two kinds of users—ordinary Weibo users and backend administrators. Application of the system can satiate the demand of one-off entering of inference data and multiple uses of data. The inference data form the basis of the system. The operation maintenance personnel should build the knowledge system, such as facts and classes. Then, the results of model building should be entered in the system. Relevant knowledge will be absorbed by the Arithmetic Logic Register Stack (ALRS), and the database saving can be accomplished. The facts and classes show favorable transparency and publicity to ordinary Weibo users. After type-in of knowledge, operation maintenance personnel can finish a series of inference process construction under the current conditions of facts, namely the inference tree. In the backend control panel, application demands of ordinary Weibo users can be fully reflected. The inference system can also provide an interface for the learning system so that the learning system can meet the requirement of inferring requirements based on the knowledge system. Besides, in order to facilitate model verification for operation maintenance personnel and make inference tree reasonability checking easier, the graph inference and operation interface has been introduced to the inference system (Arevalillo-Herráez, Arnau and Marco-Giménez, 2013).

In terms of performance demands, the lowest configurations for the automatic learning and inference system include the display card (SVGA display adapter), internal storage (10GB), processor (Intel Pentium 166MX), and hard disc (20GB). Thus, it can be seen that there is a high requirement of the hardware configurations. Since the configurations here are the lowest, necessary improvements should be made of the processor, internal storage and hard disc to facilitate system operation. Since the system adaption of Windows 10 is not perfect, the operation system chooses Windows 7/XP, learning machine Linux, Developer Express Inc. Net200 module, Netron module, Spring.Net 1.2.0 framework package, Microsoft.Net Framework 2.0 or of a higher version.

In terms of function demands, operation maintenance personnel are mainly responsible for building the inference tree and entering the knowledge. Web users give feedback about the inference system processing results by adding or deleting the intelligence tags or recommending tags worthy of attention. The collective inference programs constitute the inference engine. It is necessary to make use of the built-up inference tree.
Then, the inference algorithm can be adopted to infer user entry, and obtain user tendency. The learning engine is mainly responsible for Weibo word segmentation, inference distribution and summary of inference results.

2. DESIGN OF THE AUTOMATIC LEARNING AND INFERENCE SYSTEM

2.1 General design

(1) Basic functions
Demand analysis shows the automatic learning and inference system mainly have five basic functions, including inference preparation, inference tree management, knowledge management, automatic summary, summary of inference results, and recommending of friends. To be specific, inference preparation is to obtain the latest words target users tend to type in using the hot word system and based on records of users’ hot words. Inference tree management is to simplify operations for the backend operation and maintenance personnel. The already built-up facts can finish building of the inference tree using static semantic symbols in the visual interface. Knowledge management allows backend operation maintenance personnel to type in knowledge. Automatic inference can provide convenience for verification by backend operation maintenance personnel. Inference system can increase conciseness of the operation interface for maintenance personnel. It is similar to spreadsheet. Summary of inference results together with recommendation of friends can jointly respond to inference results, and help users search the target result set (Acampora, Pedrycz and Vasilakos, 2014; Zhao, Pan and Li, 2017).

(2) Design layer
The ALRS can be divided into the inference sub-system and the learning sub-system, and has the functions of inference preparation, inference tree management, knowledge management, inference and friend recommendation. In inference tree management, it consists of the newly-built inference tree and maintenance of the inference tree. In knowledge management, there are two parts—classes and facts.

3. Design flow
In the ALRS flow, the hot-word system will statistically analyze the latest hot words searched by users. If the tags of users are more than three, the inference will be given up automatically. In other words, ALRS serves users with no more than three tags only. The qualified users will be added into the inference set. Then, the inference work can be accomplished. Later, inference results are summarized to recommend friends to users or enter a new period or let the inference set automatically update based on user feedback.

2.2 Detailed design

(1) Framework model
The automatic learning and inference system adopts the MVC framework model. Figure 1 presents a traditional MVC framework diagram.

Figure 1. Standard MVC framework diagram

As one observes in Figure 1, the framework of the automatic learning and inference system features a prevailing three-layer framework, which consists of the representation layer, service logic layer, and data analysis layer. Then, the ALRS data access layer structure is introduced. The focus is on the design of various parts in data access, including DAO, Service, Beam, etc. Finally, the service logic algorithm and the work flow design plan are introduced (Chen, Li and Liu, 2014; Storm, Gieren and Fouqué, 2005).

Under general conditions, the data access layer will be divided into the data persistence layer and the service layer during design of the system structure. In order to realize this, it is necessary to maintain the loose coupling of layer items. The Hibernate and Spring framework can be used to realize the learning sub-system. The Spring framework is shown in Figure 2.
Based on the NHibernate, application programs and Spring.Net can be developed on the Net platform. This can help build the business structure. Hibernate and Spring, NHibernate and Spring.Net are combined. The database is packaged to process relevant database operations (Sanfirov, Stepanov and Kin, 2013). The complex database operation targets in the program codes and the embedded SQL operation language which are used can be adopted to simplify the development programs and the program test process. Since cross-platform interaction between Net and Java is necessary to the automatic learning and inference system, database selection is important. In the current stage, MySQL still has wide applications in the Internet. Therefore, MySQL is the database selected in this paper.

(2) Inference tree management

Based on knowledge base management, inference tree management allows backend operation maintenance personnel to manage and enter in the inference tree. This is a basic part of the automatic learning and inference system. The type-in function of inference tree management can help formulate relevant rules for inference tree management. It is quite convenient for the backend operation maintenance personnel to finish the inference tree building, inference tree checking, inference tree deletion, management rules and inference tree alteration. Meanwhile, the inference tree management is made up of the inference tree maintenance and the newly-built inference tree. The former includes rule management, inference tree checking, inference tree alteration, and inference tree deletion.

(3) User inference

User inference is a function developed by backend operation maintenance personnel for automatic inference of ordinary Weibo users. After building the inference tree, backend operation maintenance personnel can make use of the inference function in the inference part to calculate and verify the inference tree. Then, the inference tree can be adjusted. In the automatic learning and inference tree, the major service targets are the ordinary Weibo users. The system backend can be used to achieve automatic invoking. The participants include the MySQL database management system and the automatic learning and inference system.

(4) Summary

The user responses and inference results, after being summarized, constitute the post-inference summary. After inference, the user potential tendency table and the newly-increased user tendency table can be inferred.
ALRS can help match users with inference results in the table. Other uses meeting three requirements are recommended as friends to users (Lim and Chan, 2015). If users accept the recommendation, it means the tendency and potential tendency are acceptable to users. ALRS can increase accuracy of the inference tree based on relevant evidence, thus allowing the inference tree to obtain gain. The gain is generated by updating the inference tree information in the database and cache. The gain size can be adjusted. In the automatic inference and learning system, the gain size by default is 5%. If the confidence level is 1 or higher than 1, the automatic learning and inference system will ignore the follow-up gain request, and users will delete the tags or reject the recommendation. This means the tendency and potential tendency are not acceptable to users. ALRS can find the inference tree based on relevant evidence, and finish attenuation. Attenuation is realized by updating the inference tree information in the cache and database. Similarly, the attenuation size can be adjusted. In the automatic learning and inference system, the initial attenuation size is set to be 5%. If the confidence level is 0 or smaller than 0, the follow-up attenuation request is ignored. These methods used can realize automatic maintenance and adaption of the inference tree (Kittang, Iversen and Fossum, 2014; Shi, Deng and Shi, 2015).

3. REALIZATION OF THE AUTOMATIC LEARNING AND INFERENCE SYSTEM

(1) Function realization

The third-party user space provided by Netron and DevExpress can help realize multiple functions. For example, GraphControl class provided by Netron can realize the function of drawing panel. This allows graphic establishment of the inference tree by the system. Shape class provided by Netron can realize node functions of the inference tree. In the system, the nodes of the inference tree are connected with demands based on the interconnected points in the space. SimpleButton, ComboBoxEdit and other classes provided by DeyExpress can realize the function of interface control.

![Figure 4. A DAG for Course Priority](image)

In realizing knowledge management, new root classes or sub-classes can be built or deleted. In the root classes and sub-classes, inclusion means expansion of the simple class-facts relationship. In other words, the subordination system of more facts and more layers of classes can be formed. In fact, management, new facts are established, and the already existing operations can be deleted. During the management process, it is necessary to choose the class to which the facts belong. The fact class should be typed in. Establishment of class and fact management interface is similar to each other. In every fact, relevant information includes class ID, ID, notes and name.

In realizing user inference, one can adopt the inference operation part for example. The part consists of three links. First, one should enter the inference operation page to finish the inference tree selection. Second, the drop-down box items provided by the system can be used to endow the leaf nodes in the list with the fact confidence level. Third, click on the button of starting the interference, and the system can start the automatic inference. The users can then obtain the feedback.

Performance test

(1) Overview of environments
The software environment consists of the operation system (Windows 7), Microsoft .NET Framework 2.0, MySQL 5.0, JDK1.6, Developer Express Inc. .NET2005 modules, Spring.NET 1.2.0 framework package and Netron modules. The Netron modules are shown in Figure 3.

![Figure 3](image3.png)

Figure 3. An Improved DAG of Knowledge Points

As shown in Fig. 5, the Netron-server contains agents, plugins, message queue, etc. In the hardware environment, its internal storage is 2GB. The Intel Core i5 processor, 20GB hard sic, and NVIDIA display adapter are adopted.

(2) Result analysis
Multiple design plans can be adopted verify the performance of AILRS.
Plan 1: Memcached is not used to process all user tag guesses. The test results are shown in Table 1.

<table>
<thead>
<tr>
<th>Operation day</th>
<th>ALRS</th>
<th>Word segmentation quantity</th>
<th>Number of tags</th>
<th>Number of friends recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28020</td>
<td>112892</td>
<td>88923</td>
<td>149390</td>
</tr>
</tbody>
</table>

![Table 1](image4.png)

Table 1. Performance test results of Plan 1

Combining Table 1, it can be seen that the system load will achieve the maximum under the condition. During data checking and data update, the database should be processed.
Plan 2: Memcached is not used, and user tag guesses with less than three tags are processed. The test results are shown in Table 2.

<table>
<thead>
<tr>
<th>Operation days</th>
<th>ALRS</th>
<th>Word segmentation quantity</th>
<th>Number of tags</th>
<th>Number of friends recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27684</td>
<td>112791</td>
<td>85362</td>
<td>158723</td>
</tr>
</tbody>
</table>

![Table 2](image5.png)

Table 2. Performance test results of Plan 2

Figure 6. The impact of performance improvement before and after using multi source data mining.

Combining Table 1, it can be seen that the system load will achieve the maximum under the condition. During data checking and data update, the database should be processed.
Plan 2: Memcached is not used, and user tag guesses with less than three tags are processed. The test results are shown in Table 2.
Figure 7. The influence of performance improvement before and after using multi source data mining.

As one observes in Table 2, the plan improves the new user service efficiency. Reduction of the service coverage can increase the accuracy. Meanwhile, ALRS performance is indirectly improved.

Plan 3: Memcached is used, and all user tag guesses are processed. The test results are shown in Table 3.

Table 3. Performance test results of Plan 3

<table>
<thead>
<tr>
<th>Operation days</th>
<th>ALRS</th>
<th>Word segmentation quantity</th>
<th>Number of tags</th>
<th>Number of friends recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52893</td>
<td>113921</td>
<td>168232</td>
<td>293020</td>
</tr>
</tbody>
</table>

Figure 8. Memcached bifurcation diagram

Table 3 shows that, compared with Plan 1, Plan 3 adopts memcached for ALRS, which helps finish cache of users and inference tree. Consequently, the performance of the whole automatic learning and inference system is improved.

Plan 4: Memcached is used, and the user tag guesses with less than three tags are processed. The performance test results are shown in Table 4.

Table 4. Performance test results of Plan 4

<table>
<thead>
<tr>
<th>Operation days</th>
<th>ALRS</th>
<th>Word segmentation quantity</th>
<th>Number of tags</th>
<th>Number of friends recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58489</td>
<td>119432</td>
<td>173821</td>
<td>338293</td>
</tr>
</tbody>
</table>

Figure 9. Logistic mapping results
As one notices in Table 4, after use of memcached, the physical performance of the system is improved, indirectly leading to improvement of the system service efficiency. Combining these four plans, it can be seen that use of memcached can effectively improve the system operation efficiency. In order to verify the conclusion, Plan 4 is verified again on a holiday. The results obtained are shown in Table 5.

Table 5. New performance test results of Plan 4

<table>
<thead>
<tr>
<th>Operation days</th>
<th>ALRS</th>
<th>Word segmentation quantity</th>
<th>Number of tags</th>
<th>Number of friends recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59491</td>
<td>12941</td>
<td>18921</td>
<td>358514</td>
</tr>
</tbody>
</table>

Figure 10. Analysis results of 2D Logistic mapping sensitivity; Fig. a shows the results of simulation 1; and Fig. b shows the results of simulation 2.

Table 5 can verify the test results of Plan 4. When Plan 4 is adopted, the system can recommend relevant friends for new users, and relevant functions designed by the system can be realized. In realizing the inference tree management, one should type in the name of the inference tree. This can allow ordinary Weibo users and backend operation maintenance personnel to conduct sensational distinguishing of different inference trees. Then, the inference nodes should be added to facilitate checking of the backend operation maintenance personnel. According to the classes selected, the automatic learning and inference system can input the factors under the class into the database. In the interface, the existing nodes are deleted. If the existing nodes should be deleted, click on the right of the mouse to realize deletion. After the fact nodes are added, the relationship among fact nodes can be added or be deleted in the interface.

4. CONCLUSIONS

Based on the performance comparison of the above four plans, it is found out that use of memcached and shrinking of the tag scope can achieve favorable effects, and improve the user loyalty. After accomplishment of the inference tree construction, click to save the data. In the database, relevant information, including the inference tree, nodes, node relationship and knowledge bank, can be saved. If there are mistakes in the database, the wrong information will be returned by the system. If the information is successfully added to the rule base, the successfully added information will be returned. ALRS can help realize basic functions of the AI inference engine, dig tendencies of Weibo users, and recommend friends. Moreover, it can increase work efficiency and convenience of backend operation maintenance personnel.

REFERENCES:


