

〔非公開〕

TR-C-0122

User Interest Acquisition

リヤン マッシュー 西本 一志
Matthew LEUNG Kazushi NISHIMOTO

1 9 9 5 5 . 3 1

A T R 通信システム研究所

User Interest Acquisition

ユーザの興味の獲得

Matthew Leung (リヤン マット)
Kazushi Nishimoto (西本 一志)

Artificial Intelligence Department
知能処理研究室
ATR Communication Systems Research Laboratories
(株) エイ・ティ・アール通信システム研究所

May 31, 1995

Contents

1	Introduction	4
1.1	The Problem	4
1.2	Overview	4
2	Keyword Vector Approach	5
2.1	Introduction	5
2.2	Implementation	5
2.2.1	System Files	5
2.2.2	Representation	8
2.2.3	Initialization of User Interest Vector	9
2.2.4	Updating User Interest Vector	9
2.3	Discussion	10
2.4	Conclusion	10
3	Neural Network Approach	11
3.1	Introduction	11
3.2	Implementation	11
3.2.1	System Files	11
3.2.2	Neural Network	13
3.2.3	The Graphical User Interface (GUI)	13
3.2.4	The Flow of The Program	15
3.3	Experiments and Results	16
3.3.1	Expected Results	17
3.3.2	Experimental Results	17
3.3.3	Reasons of Unstable Performance	17
3.3.4	Proposals of Improvement	19
3.3.5	Questionnaires	19
3.4	Conclusions	20

List of Figures

2.1	The graphical user interface of Electronic Tokyo Tourist Information Center	6
2.2	An example of <i>.attr</i> file	7
2.3	An example of <i>.place</i> file	8
2.4	A keyword vector.	8
2.5	The user interest vector	9
3.1	A sample ClariNet news article	12
3.2	Graphical User Interface of UIAN	14
3.3	The flow chart of UIAN	16
3.4	Geometrical illustration of the neural network in a simplified case, with only two input nodes, i.e. input patterns can be varied in two dimensional vector space. The circle and cross represent input patterns of an interesting article and an uninteresting article respectively. The task of the neural network is to adjust its weights until it separates the interesting article input pattern from the uninteresting one. (a)The two patterns are relatively far away. There are more "ways", represented by straight lines between the two patterns, to separate the two patterns. Therefore, the associates are easier to be learned by the neural network. (b) The two patterns are close together. There are fewer "ways" to separate the two patterns, i.e. harder to learn	17
3.5	The flow chart of selecting sample articles of the first training session.	18
3.6	Processing and scoring an article	19
3.7	Experimental results	21
3.8	The questionnaire after the experiments	22

Abstract

One goal of our work group is to develop a virtual agent, which participates in meetings to supports creative concept formation of the real participants.

A number of human creativity supporting systems acquire user interest in the first place and then provide relevant or partially relevant information according to the user interest in order to support human creativity process. A common way to acquire user interest is by entering a few keywords. Better and more generic ways to acquire user interest is needed to improve human creativity systems.

In this report, two approaches to acquire user interest are proposed, namely a keyword vector approach and a neural network approach. The former approach assumes both the user interest and the available information can be represented by sets of keywords. The latter approach assumes the user interest can be acquired in the weights of a neural network and available information can be represented by sets of keywords.

The experimental results of the two approaches are positive. The results show they probably able to acquire the user interest and maybe even user implicit interest.

Chapter 1

Introduction

1.1 The Problem

The purpose of this project is to support human creativity of the participants in the Real Sensation Virtual Space Teleconferencing System [1] of Communication Systems Research Laboratory (CSRL), a division of Advanced Telecommunications Research International Institute (ATR), Japan. One goal of our work group, Intelligent Database Group, is to develop a virtual agent, which participates in meetings to supports creative concept formation of the real participants.

Meeting is an important part of office workers, they spend 30 to 70 percent of their time in meetings according to statistics [2]. A significant portion of meeting time is spent on creative concept formation process, e.g. meetings for business tactics and new style sport cars of the next century. Ironically, with the wide spread of computer technology today, various kinds of computerized teleconferencing systems have been built to solve the distance constrains of people. However, rarely any of the systems support creative concept formation process. If human creativity can be supported by computers, significant time can be saved for other important subjects.

A number of human creativity supporting systems, e.g. The Metaphor Machine [3] and The Outsider System [4], acquire user interest in the first place and then provide relevant or partially relevant information according to the user interest in order to support human creativity process. A common way to acquire user interest is by entering a few keywords. A few keywords may not good enough to represent user interest, but entering a lot of keywords is time consuming. Meanings of some keywords depend on context and a small set of keywords may not be able to represent some complicated interests. In addition, the interest of the user changes from time to time, an automated user interest acquisition system is desired.

Better and more generic ways to acquire user interest is therefore needed to improve human creativity supporting systems. Besides, if we have such a generic module that could acquire user interest, we can plug the module into any databases to get desired information.

In this report, we discuss two approaches to acquire user interest. Both of the approaches assume the attributes of all the available information in the database can be represented by a set of keywords. The first approach, a keyword vector approach, represents user interest in a keyword vector, while the second approach, a neural network approach, represents user interest in the weights of a neural network.

1.2 Overview

The rest of this report is organized as follows. In chapter 2, a keyword vector approach, we discuss the theory of the approach, and an implementation, Electronic Tokyo Tourist Information Center, of the algorithm. The aim of the program is to suggest a visiting order of tourist spots in Tokyo to the user according to his/her interest.

In chapter 3, we discuss another approach, a neural network approach. The detail of the algorithm and an implementation will be discussed. The implementation is a USENET news reader which retrieves "interesting" news according to the user interest. Real user test was done on the system. The details and results of the experiments are at the end of this chapter.

Chapter 2

Keyword Vector Approach

2.1 Introduction

In this approach, one important assumption is both user interest and information in the database can also be represented by a set of keywords. Those keyword sets can be treated as vectors, so called keyword vectors, in a multi-dimensional vector space. The dimension of the vector space depends on the number of keyword used.

A keyword vector is used to represent the user interest and every piece of information in the database can be also represented by its own keyword vector. For those pieces of information lie near to the user interest vector are the ones that the user probably interested in.

The user interest can be acquired by adjusting the user interest keyword vector according to the feedback from the user. If the user is interested in a piece of information, the user interest vector should be moved towards the keyword vector of the piece of interesting information. On the contrary, if the user is not interested in the piece of information, the user interest vector should be moved away from the keyword vector of the piece of uninteresting information.

In a word, keyword vectors of interesting information attract user interest keyword vector, while keyword vectors of uninteresting information repel user interest keyword vector.

2.2 Implementation

An implementation of the approach is Electronic Tokyo Tourist Information Center. Figure 2.1 is the Graphical User Interface (GUI) of the program. It contains pictures, local maps and brief descriptions of tourist spots in Tokyo. It is designed to suggest a visiting order of the tourist spots in Tokyo according to the user interest.

2.2.1 System Files

Two system files are maintained manually by the system builder. They are stored in the same directory as the program executable. One file, *.attr* file, contains all the available keywords which are used to describe every single tourist spot, while the other file *.place* file stores the descriptions of all the available tourist spot with the keywords in *.attr* file.

The details of the two file are written in the following two sections. In both of the file, a comment line starts with a “#” character and words are case sensitive, e.g. “A” does not equal to “a”.

.attr File – Attribute File

The *.attr* file contains a list of main attributes and their corresponding sub-attributes. Only the attributes in this file can be used to describe the tourist spots in *.place* file.

All the main attributes are headed by a character “A”, a space character and a main attribute name. A carriage return character should be at the end of the main attribute name. All the sub-attributes of a given main attribute should be written after the main attribute line but before the next main attribute. Sub-attributes are headed by a “SA” string and then a space character. A return character should be entered right after each sub-attributes. Figure 2.2 is an example of the *.attr* file.

.place File – Tourist Spot Description File



Figure 2.1: The graphical user interface of Electronic Tokyo Tourist Information Center


```
# .attr file of the Electronic Tokyo Tourist Information Center
```

```
A business-and-finance
```

```
SA bank
```

```
SA money-exchange
```

```
SA insurance-company
```

```
SA stock-market
```

```
A food
```

```
SA Japanese
```

```
SA French
```

```
SA Chinese
```

```
A sightseeing
```

```
SA palace
```

```
SA park
```

```
SA zoo
```

```
SA temple
```

```
A accommodations
```

```
SA hotel
```

```
SA youth_hostel
```

```
SA ryoukan
```

```
SA minshuku
```

```
SA capsule_hotel
```

```
...
```

```
...
```

```
...
```

Figure 2.2: An example of *.attr* file

```

# .place file of the Electronic Tokyo Tourist Information Center

# Shinjuku -- attributes of Shinjuku
P Shinjuku
A accommodations 1
SA hotel 1

A shopping 1

# Ueno -- attributes of Ueno
P Ueno
A sightseeing 1
SA park 1
SA zoo 1

# Shibuya -- attributes of Shibuya
P Shibuya
A food 1
SA French 1

...
...
...
END

```

Figure 2.3: An example of *.place* file

The *.place* file contains the descriptions of all the available tourist spots with the attributes in the *.attr* file. The syntax is very similar to the one in *.attr* file. For every tourist spot, the description starts with a character "P", a space character, the name of the tourist spot and a carriage return character. The attributes of the tourist spot follow this line and before the name of the next tourist spot.

Only the main attributes and sub-attributes in *.attr* file can be used to describe the properties of tourist spots. The assignment of the attributes are done by the system builder manually. The syntax is headed with an "A" and "SA" for main attribute and sub-attribute respectively, follows by a space character, the name of the main or sub attribute, a space character, the weight of the attribute and a carriage return character. In this implementation, weight is either assigned to 1 or 0, i.e. a tourist spot either has the attribute or not. Main attributes should be placed after a tourist spot name and sub-attributes should be placed after a main attribute. An "END" string should be put at the end of the file. Figure 2.3 is an example of *.place* file.

2.2.2 Representation

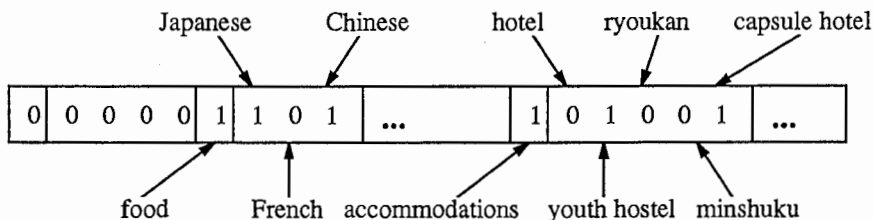


Figure 2.4: A keyword vector.

From the two system files, *.attr* and *.place*, we have the descriptions of tourist spots in sets of keywords, which are separate into main attributes and corresponding sub-attributes. A set of keywords with weights can be represented as a $N \times 1$ matrix, or a vector, where N is the number of the total main and sub-attributes. A vector entry stores the corresponding keyword weight. Therefore, we can treat the user interest and every tourist spot as vectors in a N dimensional vector space.

Figure 2.4 is an example of a keyword vector. Only two main attributes are shown in the figure, i.e. food and accommodations. Japanese food, French food and Chinese food are sub-attributes of food, while hotel,

youth hostel, ryokan, minshuku and capsule hotel are sub-attributes of accommodations. The vector entry of a sub-attribute can be set to a non-zero value only if the corresponding main attribute vector entry is non-zero.

Keyword vectors is used to represent both the user interest and tourist spots attributes. In case of representing user interest, the vector in figure 2.4 can be interpreted as part of the user interests is in food, especially in Japanese and Chinese food, and accommodations, especially in youth hostel and capsule hotel. In case of representing a tourist spot, the vector can be interpreted as part of the tourist spot attributes is food, especially Japanese and Chinese food, and accommodations, especially youth hostel and capsule hotel.

2.2.3 Initialization of User Interest Vector

At the beginning of the program, the list of main attributes in *.attr* file are shown to the user. The user clicks on those attributes interest him/her. Once the user then clicks on a main attribute, the sub-attributes of the main attribute are shown to the user. The user clicks on interesting sub-attributes. A size N , where N is the number of the total number of main and sub-attributes, vector can be generated. The user selected attributes corresponding entries are set to 1, while the rest of the entries are set to 0. An initial user interest vector is built.

According to the initial user interest keyword vector, a list of touring spots is shown. The display order of the tourist spots is according to the distance between the tourist spot vectors and the user interest vector. The tourist spot with the keyword vector nearest to the user interest vector is displayed at the top of the list, while the tourist spot with the keyword vector farthest from the user vector is displayed at the bottom. The equation of distance between two vectors \mathbf{X} and \mathbf{Y} is:

$$distance(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

2.2.4 Updating User Interest Vector

When the user clicks on the name of a tourist spot, a picture, a local map and a brief description of the tourist spot will be shown to the user. The user can evaluate a tourist spot by sliding the evaluation switch, in the range of $[-10, 10]$. The evaluation number shows whether the user interested in the tourist spot or not. Evaluation number 10 means the current tourist spot is very interesting, while -10 means it is not the type of place the user wants to visit.

The user vector is adjusted according to the feedback value given by the user. The idea is to move the user interest vector close to interesting tourist spot vectors and away from those uninteresting ones. In this implementation, equation 2.2 is used to adjust the user interest vector.

$$\mathbf{X}_u = (\mathbf{X}_e - \mathbf{X})(fb)(0.1) + \mathbf{X}_u \quad (2.2)$$

where \mathbf{X} is:

$$\mathbf{X} = \begin{cases} \mathbf{X}_u & \text{if } fb > 0 \\ \mathbf{X}_c & \text{otherwise} \end{cases}$$

and \mathbf{X}_u is the user interest keyword vector. \mathbf{X}_e is the keyword vector of the tourist spot under evaluation. \mathbf{X}_c is the center vector of all known positive feedback vectors. fb is the feedback value given by the user in the range of $[-10, 10]$.

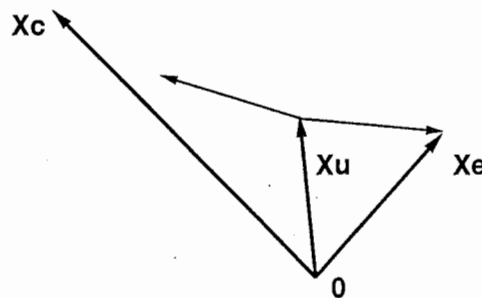


Figure 2.5: The user interest vector

If the feedback value is positive, the user interest vector moves towards the evaluating tourist spot vector. Otherwise, the user interest vector moves from the current position towards the center of all positive feedbacked tourist spot vectors. Figure 2.5 shows the geometrical interpretation of equation 2.2.

2.3 Discussion

There are two extreme ways to update the user interest vector in this design.

1. The first way is to update the main attribute part and the sub-attribute part of the user interest vector separately. The user interest vector can be broken down into a sub-vector of main attribute and sub-vectors of sub-attributes. These sub-vectors can be updated by using equation 2.2 individually.
2. The second way is to update the whole user interest vector in a vector space formed by all the main and sub-attributes and assume no dependency between main and sub-attributes. The user can set sub-attributes value to non-zero even the corresponding main attributes vector entries are zero, i.e. there is no difference between main and sub-attributes.

The update method of the user interest vector of the current implementation is between method 1 and 2. The user interest vector is updated in the vector space formed by only main attributes with non zero vector entries and the corresponding sub-attributes. The main and sub-attributes are dependent because sub-attributes values cannot be set to non-zero if the corresponding main attributes vector entries are zero. Which is the best way to update the user interest is to be determined by further experiments.

2.4 Conclusion

Simple experiments have done on the system. The results showed the system could almost obtain user interest and could provide tourist spot information according to the user interest.

Moreover, from the first few tourist spots of the final suggested visiting order list, sometimes we found some common attributes that were not selected by the user in the initialization stage. This fact suggest that the system might has the ability to acquire user implicit interests too.

However, the assignment of the tourist spot attributes, the input of pictures and local maps are all done by the system builder manually. The size of the database is limited and not big enough, only the information of ten tourist spots is available, for extensive experiments. A bigger database, better and efficient ways to assign attributes to the pieces of information are needed to perform more detail experiments.

Chapter 3

Neural Network Approach

3.1 Introduction

This approach is to teach a neural network to identify pieces of user interested information from uninteresting ones. The two main assumptions of the approach are every available piece of information can be represented by a set of keywords, while the user interest can be represented in the weights of a neural network.

First of all, we need a set of representing keywords, say N keywords, which are sufficient to represent the attributes of every single piece of available information. By fixing the order of the representing keywords, each piece of information can be represented by varying the weights of the representing keywords according to the attributes of the piece of information. Therefore, every piece of information can be treated as a vector, we refer this as a keyword vector in this report, in a N dimensional vector space.

Since neural networks are good tools to recognize patterns, we use a neural network to recognize the keyword vector pattern of the user interested information. In other words, the task of the neural network is to divide pieces of information into two groups, namely interesting and uninteresting information groups. If the neural network is able to learn the task, it means the user interest can be acquired in the weights of the neural network.

3.2 Implementation

A USENET news reader, which is called User Interest Acquisition News Reader (UIAN), is implemented. It retrieves information from the USENET database according to the user interest. The news reader has a neural network which works as a filter to filter out uninteresting articles and let interesting ones get through.

The USENET news system is an efficient way to distribute various types of data, information and bulletins to thousands of users through out the world [5]. We collectively call such items of information as "news" and we group "news" with similar topics into groups, which are called "newsgroups".

ClariNet news is electronic newspaper in USENET format. It provides live news, computer industry news, columns and features, financial information, stock quotes like a real newspaper we get from newsstands. Figure 3.1 is a sample news article of ClariNet news.

In this experimental phase, only ClariNet news is provided to the user. It has a number of reasons. Firstly, the topics of these news articles are relatively obvious, while anybody can post no central topic news to other newsgroups. Secondly, UIAN is a keyword based system. It is sensitive to the correctness of spellings and most of the words in these commercial electronic news articles are probably correct. Due to these reasons, only ClariNet news articles are restricted to be retrieved.

3.2.1 System Files

A couple of system files are maintained for keeping track of the status of the trained neural network and available newsgroups. Each user has his/her own set of system files. All the system files are stored in the ".uia" directory under the user home directory. The descriptions of the system files are written below.

newsrsrc.uia file

This file contains a list of available newsgroup names, the corresponding subscription status and a list of read article id's for each newsgroup.

Path: atr-sw!news.nig.ac.jp!looking!bass!clarinews
Approved: khaight@clarinet.com
From: C-reuters@clarinet.com (Reuters)
Newsgroups: clari.world.asia.japan,clari.biz.industry.automotive,clari.biz.world_trade,clari.news.usa
Distribution: clari.reuters
Subject: Auto talks can restart if U.S. shifts stance, Japan says
Copyright: 1995 by Reuters, R
Message-ID: <Rtrade-japan-carsUR5a2_5y8@clarinet.com>
References: <Rtrade-japan-carsUR16f_5y8@clarinet.com> <Rtrade-japanUR353_5y7@clarinet.com> <Rtrade-ja
X-Supersedes: <Rtrade-japan-carsUR684_5y8@clarinet.com>
Date: Mon, 8 May 95 6:50:21 PDT
Expires: Mon, 15 May 95 6:50:21 PDT
ACategory: financial
Slugword: TRADE-JAPAN-CARS
Priority: regular
ANPA: Wc: 293/0; Id: a0664; Src: reut; Sel: reuff; Adate: 05-08-N/A; Ver: 0/3
Lines: 32
Xref: atr-sw clari.world.asia.japan:9630 clari.biz.industry.automotive:4998 clari.biz.world_trade:721

TOKYO (Reuter) - Japan said Monday it was willing to resume talks with the United States over trade in cars and car parts if Washington stopped pressing the issue of voluntary Japanese purchases of foreign auto parts.

"It is possible to settle the issue if the U.S. side does not seek the voluntary plan, and (in that case) we would be ready to continue with the talks," government spokesman Koza Igarashi told reporters.

The biggest sticking point in the dispute, which threatens to escalate into a bitter trade conflict, has been a U.S. demand that Japanese car manufacturers renew so-called voluntary purchase agreements, which spell out how they plan to boost purchases of U.S.-made parts.

Igarashi also reiterated Japan's vow to take the automotive trade dispute with the United States to the World Trade Organization if Washington goes ahead with a threat to impose sanctions on Tokyo.

After talks last week failed to resolve the auto dispute, President Clinton was advised by top aides Sunday to hit Japan with record-breaking import tariffs and other tough trade measures in a move to force Japan to open its car market to more foreign competition.

Officials in Washington said Clinton, alarmed by a \$66 billion trade deficit with Japan, was leaning toward taking strong action, with a decision expected by mid-week.

In Tokyo Monday, Prime Minister Tomiichi Murayama discussed the issue with his cabinet ministers, including Trade Minister Ryutaro Hashimoto and Foreign Minister Yohei Kono.

At the meeting, they agreed that Murayama's cabinet would stick to the basic position that Tokyo would bring the case to the world trade watchdog if the U.S. announces punitive measures, Kyodo news agency said.

Figure 3.1: A sample ClariNet news article

keyword.lt file

keyword.lt file contains a list of representing keywords, five hundred in total. This set of keywords is used to describe all news articles. In ideal case, we can use all the nouns of an article to represent the attributes of it. However, one input node is correspond to one keyword in this design. It is not practical to have billions of input nodes in order to describe a news article with all the nouns it has. Therefore, a fixed number of keywords, five hundred keywords in this implementation, are used. For those nouns in articles appear in the representing keyword list, we refer them as keywords.

weight.nn file

It stores the state of the neural network's weights from the last training session. When the program starts, this file is loaded to the neural network.

\$USER.fb file

It is a log of the user sessions. The name of this file depends on the login name of the user, which is stored in the environment variable \$USER. It keeps track of what newsgroup does the user read and the number of positive and negative feedbacks.

3.2.2 Neural Network

A neural network is used to acquire user interest. Back-propagation learning method and incremental update mode are used. The configuration of the neural network has three layers. Input layer contains five hundred input nodes, one hidden layer with one hundred fifty hidden units and one output unit at output layer.

All the nodes are fully interconnected to the adjacent layer. Each node takes a weighted sum of its inputs, plus a bias weight and passes it through the sigmoid transfer function:

$$F(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

and the output value in the range of (0,1).

A neural network package, The Aspirin/MIGRAINES Neural Network Software, is used. For more detail, please refer to the User's Manual of the package [6].

3.2.3 The Graphical User Interface (GUI)

Figure 3.2 is the appearance of the graphical user interface of UIAN. Brief descriptions of the GUI is written as follows.

Icon Window

The small Icon window is at the top left corner of UIAN. It displays different icons according to the state of the program. When UIAN is being trained, a punching fist is displayed. When it is retrieving news articles, a rotating clock-like icon is displayed.

Listing Window

It displays a newsgroup list. It also displays a news article list when a newsgroup is selected.

News Window

It displays the currently selected news article. The headers of news articles are striped except the subject and the date of the news articles.

Feedback Buttons

There are four feedback buttons under the news window. Two bigger buttons on two sides and two smaller ones right next to them. Every feedback button has either a happy face or a sad face. If the user finds the current article very interesting, the user can click on the far right big happy face. Otherwise, if the user finds the current article very uninteresting, the user can click on the far left feedback button, which has a big sad face on it.



Figure 3.2: Graphical User Interface of UIAN

“Quit” Button

It is on the top right hand corner of the GUI. The user can quit the program by pressing this button. This button is only enabled when the listing window displays newsgroup list.

“News Group List” Button

If the user is reading articles of a newsgroup and presses this button, the listing window will show the newsgroup list.

“Filter ON/OFF” Button

This button enables/disables the information filtering process. If the filter is off, UIAN becomes a normal news reader. The button can be pressed only when a news group list is showed in the listing window.

“Keywords Input” Button

The user can enter keywords which s/he likes. The program selects relevant news articles according to the keyword match score, if the “Filter ON/OFF” button is on.

Each keywords should be separated by a space. A “!” right in front of a keyword means the keyword is unwanted. For example, the input string “Japan yen U.S. !politics” tells the program to search for articles with keywords “Japan”, “yen” and “U.S.” in them but not the keyword “politics”.

“Get New News” Button

This button is used to get new arrived news if the news server has received new news articles. Otherwise, older unread articles will be processed and displayed to the user.

“Alias” Button

The user can alias more than one news group names to an user defined name. The user defined newsgroup can be treated as a single newsgroup.

The procedure to alias a few newsgroup names is simple. First of all, the user clicks on the “Alias” button. A dialog window pops up and prompts the user to enter an aliased name. Then, the user can click “OK” in the dialog window. The cursor changes to a finger cursor. The user can click on the desired newsgroups and finish the process by clicking on the “Alias” button again. The user defined newsgroup name should be displayed in the newsgroup list.

“Subscribe” Button

The button is used to display all the available newsgroups and to change the subscription status of them. When a news group list is showed in the listing window, the user can click on the “Subscribe” button and then click on the desired news groups to toggle their subscription status. The process ends by pressing the button again.

3.2.4 The Flow of The Program

Figure 3.3 is a simplified flow chart of UIAN. First of all, It checks for the existence of the *keyword.lt* file, which contains five hundred representing keywords and locates in the *.uia* directory, which is under the user home directory.

If the file does not exist, the first training session will begin. Some sample articles are given to the user from the user selected newsgroup(s). The user has to evaluate all the sample articles by pressing one of the feedback buttons. Then, the neural network has the first training session with the user provided feedback numbers.

However, if the file exists, a list of subscribed newsgroup names will be displayed. The user can click on “Filter ON/OFF” button, “Keyword Input” button and “Alias” button before selecting a newsgroup.

UIAN processes a batch of the latest unread articles at a time. If the filter is on, only those articles with high score are displayed to the user. Otherwise, all articles are displayed. The user are highly recommended to evaluate the displayed articles in order to give more examples to the neural network. Otherwise, if the user does not input any feedback, the neural network has no way to learn interesting news articles keyword patterns.

When the user finished reading the current newsgroup, s/he can click on “News Group List” button to return to the newsgroup list. The button also triggers the training of the neural network with the feedback of the current user session.

If the user wants to quit the program, s/he can click on “QUIT” button. UIAN will update the *newsrsrc.uia* file and *weight.nn* file.

The First Training Session

During the first training session, the representing keyword list is built and a few sample articles is selected. The user has to select a few interesting newsgroups in the beginning of the session. A total of forty news articles are taken from these newsgroups as sample news articles and the representing keywords are taken from the nouns of these sample articles. The user reads the sample articles and evaluates them by clicking on one of the feedback buttons according to how interesting the articles are. The feedback values will be stored and the neural network will be trained with the feedback values in a batch after the user finish evaluating all the sample articles.

In order to accelerate the learning process, the concept of “shaping” [7] is used to select sample articles. Shaping is a schedule of learning that begins with presenting associations that are simple to learn, progressively more difficult associations are used in the training. The neural network can be viewed as learning multi-dimensional surface that partitions the input vector space [8]. It is relatively easier to learn the associations if the inputs in the vector space separate as far as possible, please refer to figure 3.4 for more details.

Figure 3.5 is the flow chart of selecting sample articles of the first training session. The aim of the algorithm is to find sparsely distributed articles in the vector space as the sample articles of the first training session. So the learning process can be accelerated.

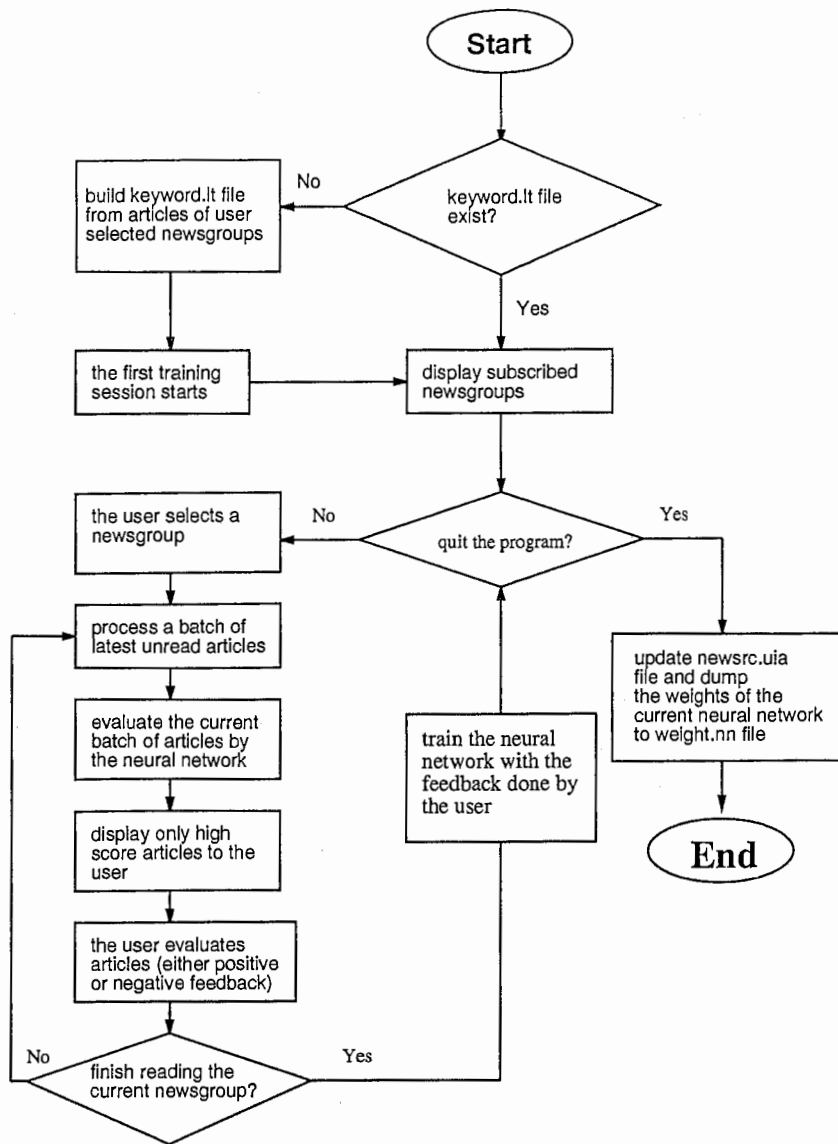


Figure 3.3: The flow chart of UIAN

Processing and Scoring an Article

The processing and scoring of an article is described as follows, please refer to figure 3.6. First of all, all the nouns of an article are extracted but only those can be found in the representing keyword set can be used to represent the attributes of the article.

The input vector has five hundred entries. One entry correspond to one keyword. Each entry is the relative keyword count of the corresponding keyword. The relative keyword count of a keyword w is given by:

$$relative\ keyword\ count(w) = \frac{keyword\ count\ of\ the\ current\ article(w)}{keyword\ count\ of\ the\ batch\ of\ articles(w)} \quad (3.2)$$

The relative keyword count is in the range of $[0,1]$. It measures the relative importance of the current article with the other articles in the current batch with respect to a keyword.

The input vector is fed into the input layer of the neural network and the output of the neural network is the score of the current article, in the range of $(0,1)$.

3.3 Experiments and Results

Four users were given access to the software and have tried UIAN for a week on a daily basis. The newsgroup names and the number of feedback, both positive and negative, were measured. After the one week experiment

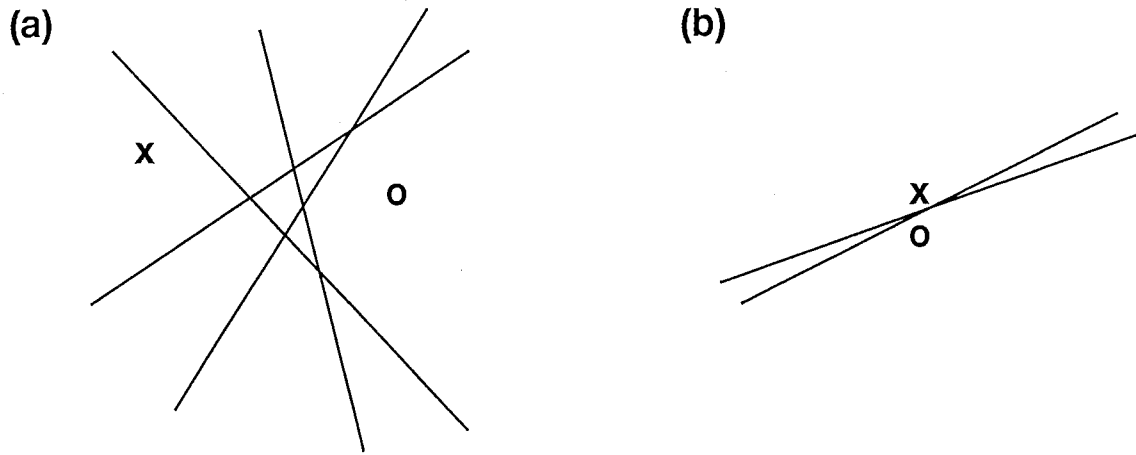


Figure 3.4: Geometrical illustration of the neural network in a simplified case, with only two input nodes, i.e. input patterns can be varied in two dimensional vector space. The circle and cross represent input patterns of an interesting article and an uninteresting article respectively. The task of the neural network is to adjust its weights until it separates the interesting article input pattern from the uninteresting one. (a) The two patterns are relatively far away. There are more “ways”, represented by straight lines between the two patterns, to separate the two patterns. Therefore, the associates are easier to be learned by the neural network. (b) The two patterns are close together. There are fewer “ways” to separate the two patterns, i.e. harder to learn

period, the users were requested to fill out a questionnaire. The purpose was to get suggestions and subjective results from the users.

3.3.1 Expected Results

We expect the news reader to learn the interest of the user and provide more and more interesting news articles to the user over time. The user do not required to evaluate every single news articles but s/he is encouraged to evaluate as many news articles as possible. It is because the more examples presented to the neural network the quicker and more accurate the neural network can learn the user interest. The expected results are the number of net positive feedback, i.e. number of positive feedback subtracts number of negative feedback, should be a non-negative number and increase over time.

3.3.2 Experimental Results

Figure 3.7 was the experimental results. The number of positive feedback (marked by o's), negative feedback (marked by x's) and net positive feedback (marked by *'s) of all the users were plotted against every user sessions. An user session started from the user selected a newsgroup and ended with closing it. The neural network got trainings after a newsgroup was closed.

We concerned more on the number of net positive feedback over time. The experimental results showed the net positive feedback of almost every user sessions were above zero, i.e. non-negative.

The non-negative net positive feedback numbers were not probably because of more interesting news articles in the user selected newsgroups. The first data point in figure 3.7 showed the number of feedback during the first training session. The sample articles were taken from user selected newsgroup but three out of four users inputted more negative feedback than positive feedback during the first training session. This showed the non-negative net positive feedback number was probably not the reason of the user selected relatively interesting newsgroups.

However, the performance was not stable because the number of net positive feedback fluctuated over time. We will discuss some possible reasons lead to unstable results and proposals of improvement.

3.3.3 Reasons of Unstable Performance

- Only a fixed number of five hundred keywords was used to represent every articles. This led to the following problems.
 - The representing keyword set might not general enough to represent news articles in different news-groups.

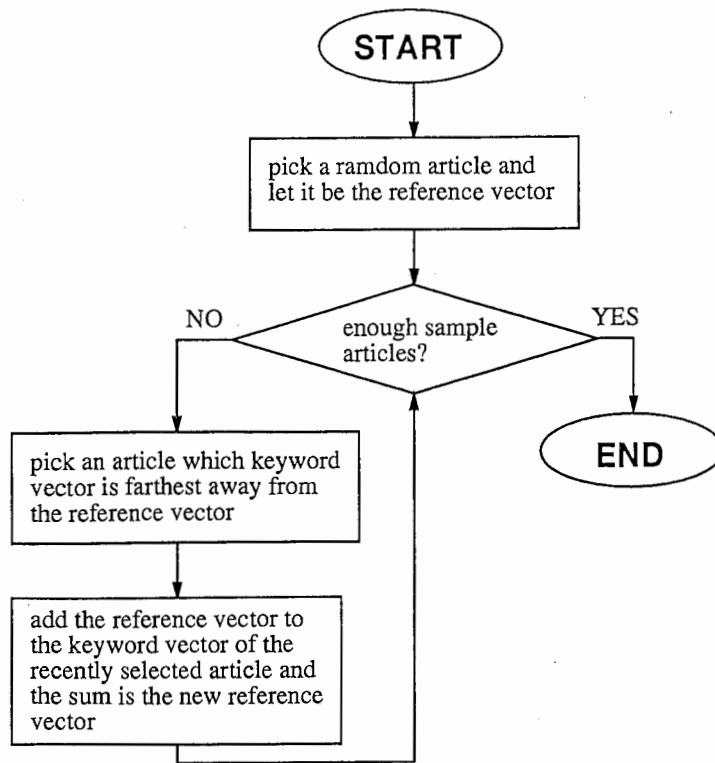


Figure 3.5: The flow chart of selecting sample articles of the first training session.

- Information lost due to only those nouns which were in the representing keyword set could be used to represent the attributes of an article. Not all the nouns were used to represent the attributes of an article.
- It increased the chance of having conflict input-output pairs, i.e. same input but different target outputs. If an interesting article and an uninteresting article contained no nouns in the representing keyword set, the vector entries of both of the input vectors would be all zero. However, the user would input high feedback value for the interesting one but low feedback value for the uninteresting one. In case of conflict input-output pairs were in the training set, they slowed down the neural network learning process.
- The input representation might be too hard to be learned by the neural network. Input representation is a crucial part of neural network learning. A neural network may not be able to learn certain task because of difficult input representation. For example, to train a neural network to distinguish between odd and even numbers. If the input representation is in binary, the neural network learns the task easily by focusing on the last digit of the input. However, if base three representation is used, it is a harder job for the neural network [9].
- A big neural network was used, number of input and output nodes were five hundred and one respectively. Therefore, a lot of parameters had to be adjusted in order to learn the patterns of interesting articles. That slowed down the learning process.
- The network got not enough training.
 - The training set of the first training session was too small, only forty sample articles. Of course we could show the user as many news articles as there were available. However, it would be a tough job for the user to read thousands of articles and evaluate them one by one. Besides, the purpose of UIAN is to ease the load of the user by providing high proportion of interesting news articles but not make the user shoulder a heavier load.
 - The training loops were too few. The neural network got training right after each user session. In order to compromise performance and time, training would be stopped after certain number of training loops even the neural network was not converging.

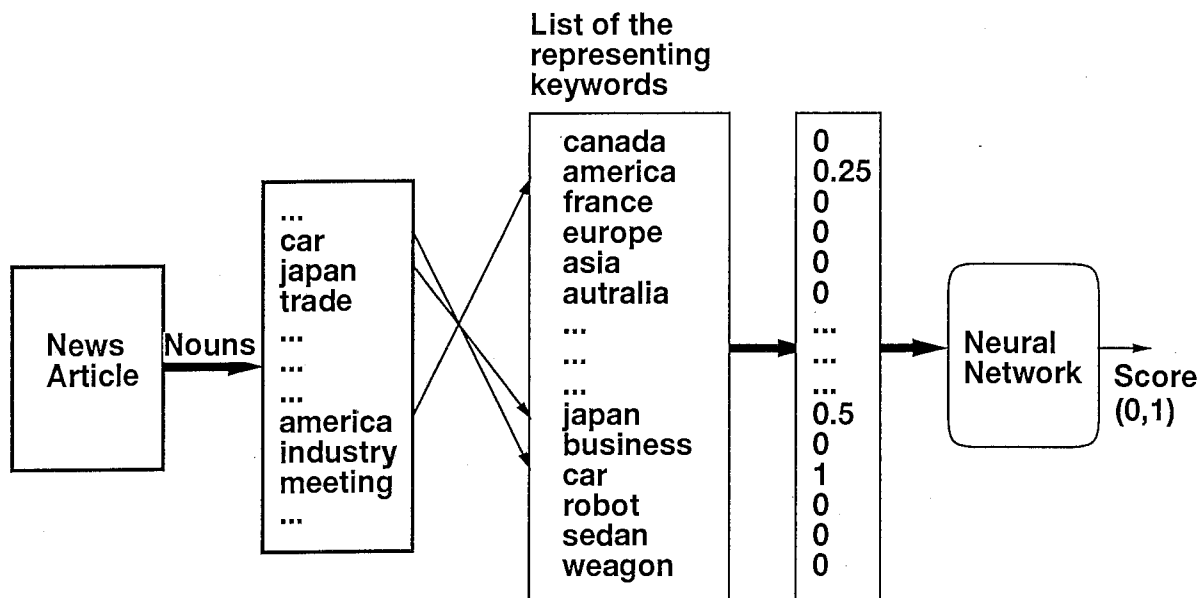


Figure 3.6: Processing and scoring an article

- The experiment term was too short. The experiment period was only five days because of time constrain. Besides, there were technical problems on the first day and not every user could start the experiments on the first day. The users needed time to get use to UIAN. Further, one goal of UIAN is to be used daily or regularly by the user and to learn the user interest while providing articles to the user. The performance of a longer experiment term is desired.

3.3.4 Proposals of Improvement

- To increase the number of significant keywords gradually in the representing keyword list. This can be done by checking the weight.nn file after every user session and find out which input nodes contribute significant influence to the output. The corresponding insignificant keywords are dropped and replaced by new keywords. Then, a training session is needed for the new representing keyword list.
- To search for a better input representation that is easier to be learned by a neural network, i.e. preprocess the input vectors. For example, to reduce keywords into different categories and reduce the size of the input vector.
- To optimize the number of nodes of the neural network by using some network constructing algorithms, e.g. upstart algorithm and tiling algorithm [9].

3.3.5 Questionnaires

Besides objective data collecting, a questionnaire was also distributed to the users to collect subjective comments on UIAN. Figure 3.8 is a copy of it. The questions in the questionnaire could be divided into three categories. The first category was about the background of the user, whether they read USENET news often before the experiments and would language be a problem to distinguish interesting articles from uninteresting ones. The second category of the questions was mainly about the user interface of UIAN, whether the interface easy and intuitive enough to use. The third category was about the performance of providing interesting articles to the user.

Two users out of four were native English speaker. The two non-native English speakers claimed their English level was good enough to distinguish interesting articles from uninteresting ones. All of them read USENET news regularly but not ClariNet newsgroups before the experiments.

The responds about the GUI were quite positive. Most of the users felt the GUI was simple and intuitive to use for basic tasks, e.g. read articles and providing feedbacks. One area needs more work is to implement more news reader fundamental functions like post news and store news to specified files. An interesting suggestion from an user is not only high score articles shows to the user, but also provides low score articles in another window. This function provides heterogeneous information and may support human divergent thinking process too, please refer to chapter 1.

All of the users thought the experiment period was too short, only five days. Some of them could not start to access the software on the first day due to some technical problems. Further, not all of them used UIAN everyday namely due to forgetfulness and business trips. They wanted more time to train the neural networks and see the effect of well-trained ones.

3.4 Conclusions

There is a possibility that UIAN provided news articles to the user according to his/her interest. The reason is the number of net positive feedback is almost always above zero in the experimental results. It is probably not because more interesting articles in the user selected newsgroups. Experimental results showed that even in user selected newsgroups, uninteresting articles portion may be bigger than interesting portion.

From the questionnaire, the users give quite encourage responds to the system and the approach seems to point to the right direction. Some areas need to be work on more. In user interface level, more fundamental news reader functions should be implemented. In theory level, the learning of the user interest by a neural network can be accelerated by some methods mentioned in section 3.3.4.

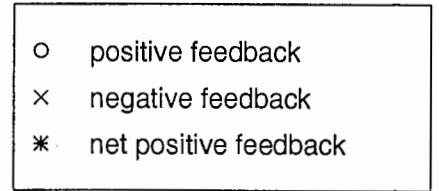
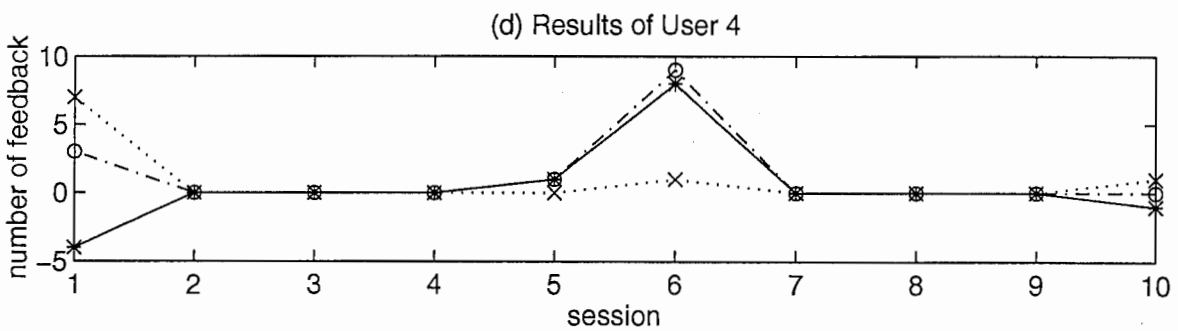
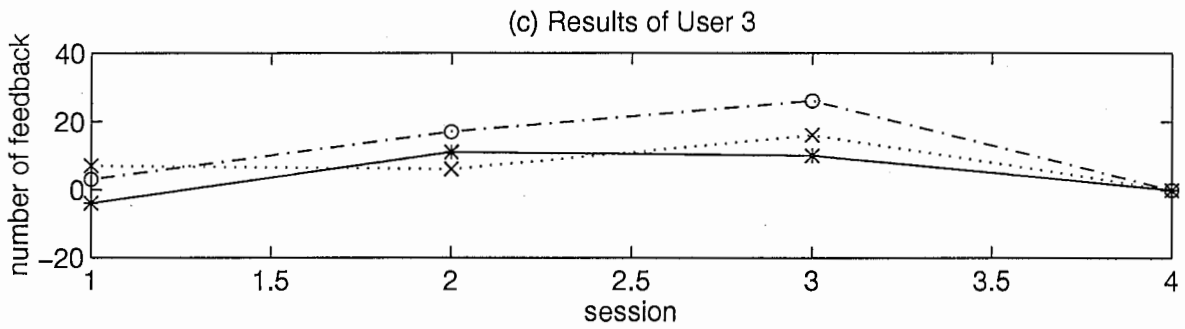
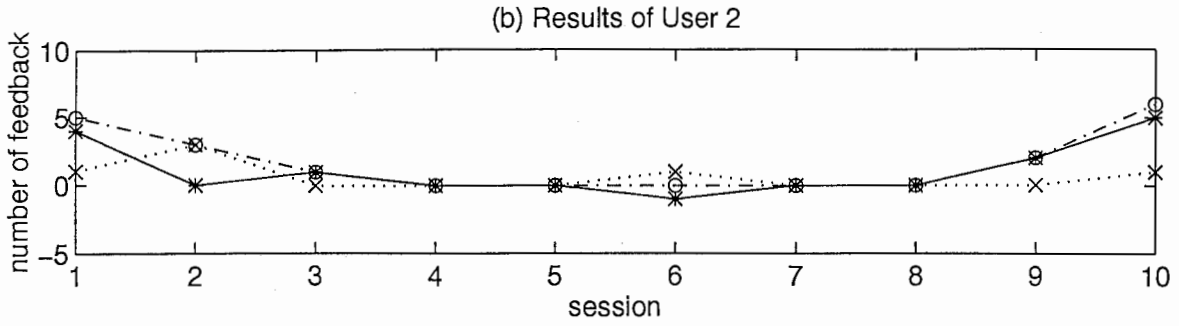
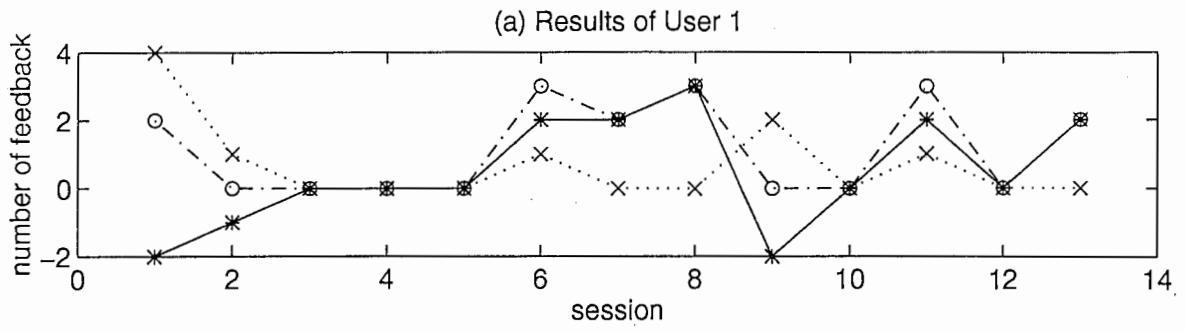


Figure 3.7: Experimental results

This is the questionnaire of the experiment of UIA news reader.

User background:

1. Did you often read USENET news before this experiment?
2. Is English your native language? If no, does it affect you how to determine an article interesting or not?

Please answer the following questions on a scale of 1 to 5. You are also highly encouraged to write short description about your answer.

3. Do you think the user interface is intuitive and easy to use?
4. Do you think the news reader provides high proportion of interesting articles over time?
5. Did the news reader often provide you articles in expected categories?
6. Did the news reader provide you articles in unexpected categories, but you found the articles interesting?
7. Did your news interests change over time?
8. If you answer yes in the last question. Could the news reader keep providing interesting articles in spite of the changes?
9. Overall comment.

Figure 3.8: The questionnaire after the experiments

Acknowledgements

I would like to thank Dr. K. Habara, Executive Vice President of ATR International and Chairman of the Board of ATR Communication Systems Research Laboratories, and Dr. N. Terashima, President of ATR Communication Systems Research Laboratories, for inviting me to ATR for eight months.

I would like to express my sincere gratitude to Mr. F. Kishino, Head of the Artificial Intelligence Department, who welcomed me in to his department, gave me the opportunity to attend Co-op Japan Program Symposium at Canadian Embassy in Tokyo.

I would like to thank my supervisor Mr. Kazushi NISHIMOTO, Researcher, for his guidance and support of my work, Mr. O. SATO for his help of coding part of the system prototypes, and those who took part in my experiments and gave me valuable advice.

Finally, I would like to thank the members of the Planning Division and the Planning Section of ATR Communication Systems Research Laboratories for having organized such a wonderful stay in Japan for me.

Bibliography

- [1] Fumio Kishino. Cooperative work environment using multi-model interface in virtual space teleconferencing. Technical report, Advance Telecommunicates Research, Communication Systems Research Laboratories, Japan.
- [2] R.R. Panko. Office work. *Office Technology and People 2*, pp. 205-238.
- [3] Young, Lawrence F. The metaphor machine: A database method for creativity support. Technical report, University of Cincinnati.
- [4] 西本 一志 Kazushi NISHIMOTO. 連想記憶を用いた異質性を含む情報の抽出手法の検討. Technical report, (株) エイ・ティ・アール通信システム研究所 ATR Communication Systems Research Laboratories, 四月 April 1995.
- [5] Kantor, Brian and Lapsley, Phil. Network news transfer protocol: A proposed standard for the stream-based transmission of news. February 1986.
- [6] Russell R. Leighton. *The Aspirin/MIGRAINES Neural Network Software User's Manual*. MITRE Corporation, release v6.0 edition, October 1992.
- [7] Wieland, Alexis and Leighton, Russell. Shaping schedules as a method for accelerated learning. Technical report, The MITRE Corporation.
- [8] Wieland, A. and Leighton, R. Geometric interpretation of neural network capabilities. IEEE International Conference on Neural Networks, 1987.
- [9] Hertz, John, Krogh, Anders, and Palmer, Richard G. *Introduction to The Theory of Neural Computation*. Addison-Wesley Publishing Company, 1991.