WEB APPLICATIONS

WEB APPLICATIONS

WEB APPLICATIONS

A Web-based Intelligent System for the Daya Bay Contingency Plan in Hong Kong

James Liu, Raymond Lee and Jane You

Department of Computing, Hong Kong Polytechnic University
Hung Hom, Hong Kong
{csnkliu, csstlee, csyjia}@comp.polyu.edu.hk

Abstract

Decision making is particularly important for emergency managers as they often need to make quick and high quality decisions under stress based on scratch and inadequate information; and to follow expert knowledge or past experience. The potential release of radioactive material from the Guangdong Nuclear Power Station (GNPS) at Daya Bay, though is highly unlikely, could perhaps be the most dreaded disaster which would cause drastic damages to lives and properties. The Government of the Hong Kong Special Administrative Region Government (HKSAR) has therefore in 1990 completed the Daya Bay Contingency Plan (DBCP) to prepare for such disasters. To supplement the experts in assisting disaster managers with a useful tool to make better quality decisions based on well-structured, accurate, sufficient expert knowledge, a prototype expert system has been developed to cover two major areas of the plan, namely: (A) Determination of activation level of the DBCP and provision of an action checklist and (B) recommendation on counter-measures.

1. Introduction

Decision making is often a challenging job for disaster managers as they often need to make quick and high quality decisions under stress based on scratch, inadequate, unstructured information [Turban, 1993]. The seriousness of consequences in the event of an accident in GNPS resulting release of radioactive materials impose great pressure on disaster managers in the Government of HKSAR [1999]. Unlike some large countries such as Canada, Russia, France, British etc which have long history of developing nuclear industry and some have experienced different scale of accidents e.g. the well-known Chernobyl accident (1986) and the Three Mile Island accident (1978), HK has little experience and relative low public awareness in nuclear industry. Even though we have a comprehensive contingency plan with detailed rules and procedures in hardcopy format and human experts, it is difficult and time-consuming for disaster managers to retrieve, study and organise such information and expert knowledge to cope with real emergencies under great pressure.

2. The Daya Bay Contingency Plan

The GNPS at Daya Bay is located at about 24 km from the northeast coast of HK. It began commercial operation in 1994. The pressurized water reactors used in the plant adopt a successful French reactor design and the safety review has been conducted by the International Atomic Energy Agency (IAEA) which found to be in accordance with international standards and operated to international practices. Nevertheless, as part of the emergency planning system, the Government of HKSAR has prepared a comprehensive contingency plan [HKSAR, 1999]. The main purpose of emergency planning is to ensure that proper and prompt actions are taken in an accident to protect the health and safety of the general public.

3. An Intelligent Expert System

According to Martin [1988] and Medsker [1994], Expert System (ES) "Reproduced the reasoning process a human decision maker would go through in reaching a decision, diagnosing a problem, or suggesting a course of action". The components of an ES include: (a) knowledge base which contains heuristic and judgmental programs; (b) inference engine consisting of the reasoning logic; and (c) user interface where the user supplies data and receives an answer, and often with the reasoning supporting the answer. EXSYS Professional (Version 5.1.4) by the EXSYS Inc. (note) has been selected to build the prototype because it is user-friendly shell and easy to understand and use. Domain knowledge is captured in a set of rules entered in the system's knowledge base. The system along with other information contained in the working memory to solve a problem. Active research of ES on process planning and scheduling can be found in Kingston et al. [1997], Boutilier et al. [1997] and Liu [2001].

4. Development of DBCP ADVISOR

4.1 Overall System Architecture

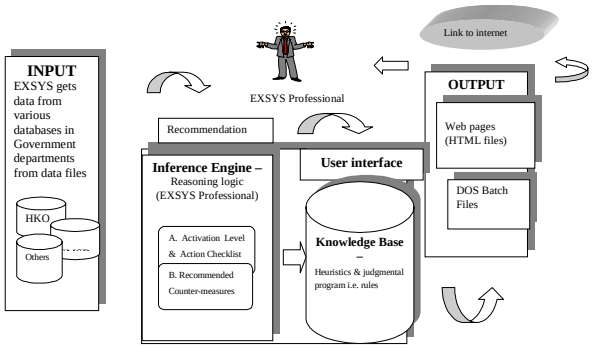

Figure 1 – Overall System Architecture

Note: EXSYS Professional is a product of the EXSYS Inc. founded in 1983 (renamed as MultiLogic afterwards) in USA .The company is one of the longest lived Expert System companies in the market.

4.2 Design Structure

The system is divided into two parts " (A) activation level and action checklist; and (B) recommended counter-measures. An overview of the design structure of the prototype is as follows:

Activation level of DBCP and provision of an action checklist

The system will guide managers through the stipulated procedures under the existing DBCP according to different sources of notification.

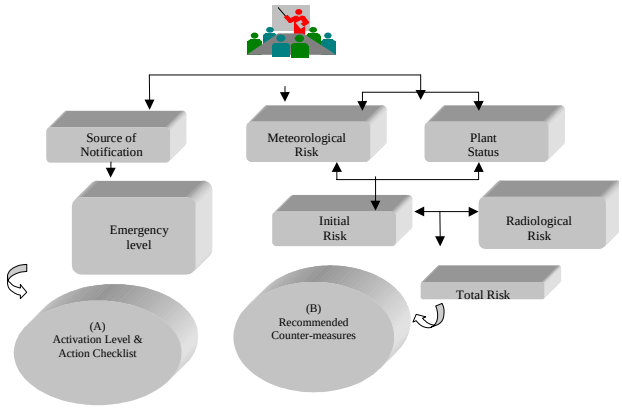

Figure 2 – Overview of System Structure

Source of Notification

There are six major sources of notification under the DBCP. They are: (1) from the Guangdong authority, (2) from the Hong Kong Nuclear Investment Company (HKNIC), (3) from the International Atomic Energy Agency (IAEA), (4) Hong Kong Observatory (HKO)'s Radiation Monitory Network (RMN), (5) Water Supplies Department's Water Contamination Monitoring System (WCMS), and (6) any other possible sources. For example, when the source of notification is from the Guangdong authority, the system will ask the disaster managers to supply information on whether it is a request for assistance and advise appropriate action. Expert rule involved will look like this:

For example :

IF the notification message from the Chinese authority received at HKO AND request for assistance is yes
THEN HKO refers message to Security Bureau (SB)

Emergency level of accident

After confirming the notification message, the system will guide the disaster managers to clarify the emergency level of the accident in order to determine the activation level of the DBCP. The management of the GNPS has adopted the IAEA's four-category system for classifying nuclear power emergencies: *(i) Emergency standby (ii) Plant emergency (iii) Site emergency and (iv) Off-site emergency.*

Activation level of DBCP and Action Checklist

Under the DBCP, according to information on the emergency level and other relevant information, the SB will have to determine the final activation level of on recommendation by operational departments. There are 4 levels of activation, namely: *(1) Observation level (2) Ready level (3) Partial activation level and (4) Full activation level*, which is corresponding to the emergency level as illustrated in Figure 3.

On recommendation of the activation level of the DBCP, DBCP ADVISOR will provide an action checklist as follows:

For Example:
IF the notification message from the Chinese authority received at HKO
AND request for assistance is no
AND the notified emergency level is off-site emergency
THEN HKO alerts SB - Confidence=1
AND recommends full activation of DBCP - Confidence=1
AND HKO initiates cascade calls according to activation level decided by SB - Confidence=1
AND HKO consults DH and advises EMSD immediately with respect to Ping Chau/Mirs Bay, monitoring centres and border controls - Confidence=1

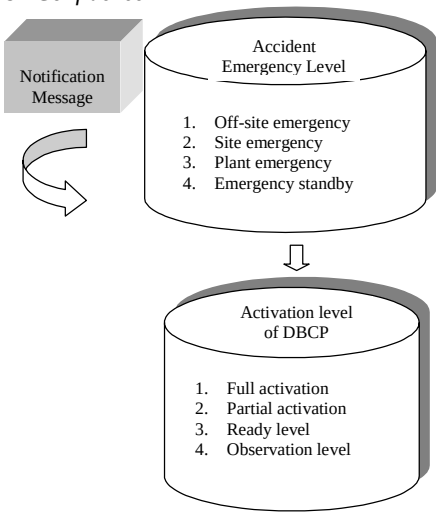

Figure 3 – Determination of Activation Level of DBCP

Recommendation on Counter-measures During Emergencies in GNPS

This part of the system advises disaster managers of the appropriate counter-measures. It is a very complicated

process in determining what are the appropriate counter-measures to be taken during different levels of emergency in GNPS. It involves expert advice after numerous discussion on various risk factors among technical departments.

Initially, experts have to assess the **(A) INITIAL RISK** of Hong Kong (HK) by referencing to the **(a) meteorological risk** and the **(b) plant status**; and finally to determine the **TOTAL RISK** when assessment results of **(B) RADIOLOGICAL RISK** is available at a later stage. Depending on the degree on total risk to HK, experts will recommend corresponding counter-measures**.**

(A) INITIAL RISK - At the early stage of activation of the DBCP, initial risk is assessed by looking at **(a) meteorological risk** and **(b) plant risk** from different sources.

(a) Meteorological Risk Level (MRL) - Two main attributes: (i) plume track coming from a particular quadrant of GNPS and (ii) plume arrival time will be considered. *Plume* is the released radioactive fission products carried by wind which behave in a way similar to a cloud of smoke dispersing into the atmosphere and depositing some of its content on the ground. The concentration of radioactive materials in the plume decreases with distance from the site.

(i) Plume track coming from a particular quadrant of GNPS – It is measured in terms of degrees within 0-360 degrees (0 degree=due North). *S*ince GNPS is located at the North-East (NE) direction from HK, wind blowing from SW (i.e. 180 to 270 degrees) will bring the plume to HK which is very unfavourable to HK. Though plumes towards NW and SE quadrants of GNPS are not directly threatening HK, there is the possibility of change in direction, so it is still unfavourable to HK. If the plume is towards NE quadrant of GNPS which will blow the radioactive materials away from HK towards the opposite direction, it will therefore be favourable to HK. HKO will get meteorological information from the Chinese authority during emergencies which will be verified against measurement by HKO.

(ii) Plume arrival time - Plume movement indicates whether sufficient tie is available for preparation and implementation of counter-measures. It is measured in terms of minutes, according to the HKO's guideline, plume arrived within 120 minutes will threaten HK because it takes at least two hours to evacuate residents/visitors from the Ping Chau region and clearance of vessels from Mirs bay; and carry out other counter-measures. HKO will get information on plume movement from the Chinese authority during emergencies. The Accident Consequence Assessment System (ACAS) which is a computerized system operated by HKO will base on meteorological data and radiological information, models the transport and dispersion rates of the released radioactive materials.

Fuzzy Logic - The process to derive the MRL involves expert advice on plume track direction and plume arrival time using linguistic terms such as very favourable, very fast, which are descriptive in nature. The resulted MRL is also description in linguistic terms: high/medium/low. EXSYS Professional can handle inexact reasoning by fuzzy logic which deals with uncertain knowledge. Fuzzy logic is primarily concerned with quantifying and reasoning about vague or fuzzy terms, called fuzzy variables, that appear in our natural language. It provides a rigorous mathematical method to handle parameters that are defined subjectively. It does this by associating membership grades to different values of such parameters and carrying out calculations on those membership grades (Altrock, 1995).

Table 1 – Linguistic Terms of GNPS Quadrant Which the Plume Track Comes From

Class	Description
(1) > 170 AND <280	*Very Unfavourable* - since GNPS is located at the North-East direction from HK, plume comes from the South-West quadrant of GNPS (i.e. 180 to 270 degrees) means towards HK.
(2) >=80 AND <=190 OR >=260 AND <=360	*Unfavourable* – though plume moving North-West and South-East quadrants of GNPS are not directly threatening HK, there is the possibility of change in direction. So, it is still unfavourable.
(3) > 0 AND <100	*Favourable* – Plume moving North-East quadrant of GNPS will be away from HK, so it is favourable to HK.

The fuzzy inference can identify the rules that apply to the current situation and can compute the values of the output linguistic variable. We first establish rules with linguistic terms according to expert advice, quadrant of GNPS which the plume track is coming from and plume arrival time are associated with the output variable i.e. MRL in linguistic terms (high/medium/low) as shown in the Table 1-3.

Table 2 – Linguistic Terms of Plume Arrival Time

Class	Description
(1) <70 min.	*Very Fast* – Plume moving quickly and will arrive HK within one hour implying very little time to arrange counter-measures.
(2) >= 50 AND <=130 min.	*Fast* – Plume moving quickly and will arrive HK within two hours implying the arrangement of counter-measures must be taken timely.
(3) >110 min.	*Slow* – Plume moving relatively slow with sufficient time to prepare for /implementing counter-measures.

Table 3 – Determination of MRL

GNPS Quadrant	Plume Arrival Time	Meteorological Risk Level (MRL)
Very unfavourable	Very Fast OR Fast	High
Very unfavourable	Slow	Medium
Unfavourable	Very Fast OR Fast	Medium
Unfavourable	Slow	Medium
Favourable	Very Fast	High
Favourable	Fast	Medium
Favourable	Slow	Low

At the end of fuzzy inference, the result for *MRL* is given as

a linguistic variable. The system will translate the resulted *MRL* into mathematical values i.e. defuzzification. EXSYS Professional follows the following procedures to determine the *FUZZY MRL (defuzzified)* :

*[FUZZY MRL] = # Σ [weighting factor assigned to each possible result of MRL (i.e. high/medium/low) * corresponding confidence level of rules]*

(b) Plant Status (PS) – In order to derive the ***INITIAL RISK,*** we have also to derive the level of plant risk from the plant status at GNPS. It is measured by the level of defence-in-depth degradation which is classified into three levels with reference to the International Nuclear Event Scale. The higher the level of defence-in-depth degradation, the lower the safety level of the plant, hence the higher the risk.

Table 4– Classification of the Plant Status

P S	Defence-in-depth Degradation
Poor	Serious Incident – near accident with no safety layers remaining.
Bad	Incident – incidents with significant failures in safety provisions.
Fair	Anomaly – anomaly beyond the authorized operation regime.

In the system, the weighting assigned for different status of the plant to indicate its relative importance are set as below:

Table 5 - Relative Weighting of the Plant Status

PS	Relative Weighting
Poor	5
Bad	3
Fair	1

Taking into account the fuzzy scores and weighting assigned by experts of **(a) MRL** and **(b) PS**, the **INITIAL RISK SCORE** can be derived as shown below:

Table 6 – Example on Determination of Initial Risk Score

Factors	Fuzzy score (Si)	Weighting (Wi)	Initial Risk Score (0-10) (Si x Wi)
MRL	*0.7	0.3	2.1
P S	*0.6	0.7	4.2
∑ Wi		1	
∑ (Si x Wi)			6.3

***** Fuzzy score is derived by system on MRL & Relative Weighting of Specific PS assigned by experts

***** Weighting is assigned by experts for MRL & PS

* Assumed that the fuzzy MRL is 3.5 which is normalized to be 0.7 and plant status is bad (i.e. relative weighting=3) which is normalized to be 0.6.

Table *7* - The Classification of Environmental Gamma Dose-rate

Class (in mSv)	Description
X< 5	The level of X will not impose any harmful effect on public health.
5 <= X <= 50	The level of X will impose certain harmful effect on public health, sheltering is recommended.
X > 50	The level of X will impose serious harmful effect on public health, evacuation is recommended.

RADIOLOGICAL RISK – it is assessed by three main attributes: (i) environmental gamma dose-rate; (ii) radionuclide concentration activity in food; and (iii) radionuclide concentration activity in water.

Taking into account the three attributes, the **(B) RADIOLOGICAL RISK** can be derived as below:

Table 8– Determination of Radiological Risk

Environmental Gamma Dose-rate (X)	Radionuclide Concentration Activity in Food (Y)	Radionuclide Concentration Activity in Water (Z)	Radiological Risk
X > 50	Yes OR No	Yes OR No	High
5 =< X =< 50	Yes	Yes OR No	High
5 =< X =< 50	No	Yes	High
5 =< X =< 50	No	No	Medium
5 < X	Yes	Yes	High
5 < X	Yes	No	Medium
5 < X	No	Yes	Medium
5 < X	No	No	Low

In the system, the relative weighting assigned for different levels of radiological risk are set as below:

Table 9 – Relative Weighting of the Radiological Risk

Radiological Risk	Relative Weighting
High	5
Medium	3
Low	1

Taking into account the fuzzy scores and weighting assigned by experts of ***(A) INITIAL RISK SCORE*** and the ***(B) RADIOLOGICAL RISK SCORE,*** the ***TOTAL RISK SCORE*** can be derived. An example is shown below:

Table 10 – Example on Determination of Total Risk Score

Factors	Scores (St)	Weighting (Wt)	Total Risk Score (0-100) (St x Wt)
Initial Risk Score (IRS) ∑ (Si x Wi)	*0.63	40	25.2
Radiological Risk Score (RRS)	0.6	60	36
∑ Wt		100	
∑ (St x Wt)			61.2

* Weighting factor refers to the weighting on IRS & RRS assigned by experts. Assumed that the Radiological Risk is medium (i.e. relative weighting =3) which is normalized to be 0.6.

4.3 User Interface

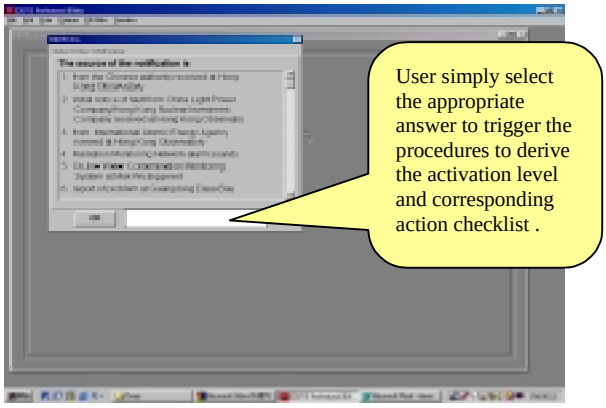

Figure 6 – Question Screen on Part A

Activation Level of DBCP

Question Screens - When user starts the EXSYS program on "Activation Level and Action Checklist", DBCP ADVISOR will pop-up a series of screens to ask for the required information in multiple choice format as follows:

Advisory Screen – At the end, ***DBCP ADVISOR*** will pop-up an advisory screen with recommended actions as below:

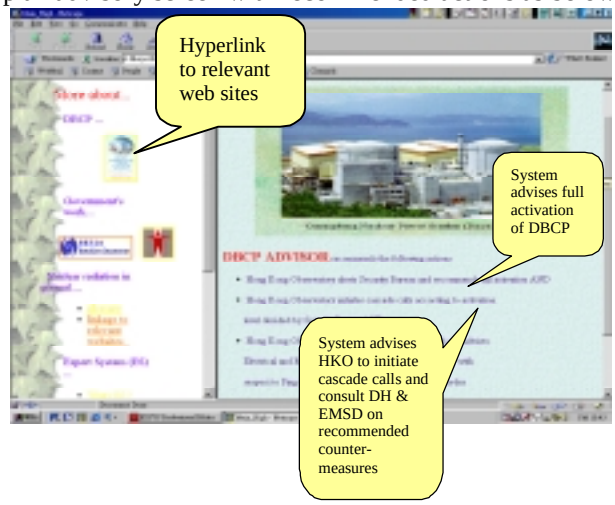

Figure 7 – Advisory Screen on Part A

Recommended Counter-measures

Data File – When the user starts the EXSYS program on "Recommended Counter-measures", ***DBCP ADVISOR*** will get information on relevant parameters to derive the **TOTAL RISK LEVEL** by calling from external databases in relevant government departments through .DAT files. An example is as follows:

Table 11 – Parameters in the Data Files

GNPS Quadrant	280 degrees
Plume arrival time	200 minutes
Defence-in-depth Degradation	2 – significant failure of safety provision
Environmental Gamma Dose-rate (X)	2 (i.e. 5<=X<=50)
Radionuclide concentration activity (RCA) in food	1 (i.e. RCA in food exceeds the intervention level = YES)
Radionuclide concentration activity (RAC) in water	2 (i.e. RCA in water exceeds the intervention level = NO)

Advisory Screen – At the end, ***DBCP ADVISOR*** will derive the TOTAL RISK SCORE (TRS). An example is as follows:

[Initial Risk level(IRS)]* .2*40 + [Radiological Risk Score (RRS)*.2*60] = 3*.2*40 + 5 *.2*60 = 84

Relevant Rule will finally be fired to advise user of the appropriate counter measures as follows:

IF [IRS].2*40 + [RRS] *.2*.60 >67*
THEN RUN (trial6.bat /B)
AND Immediate evacuation of residents/visitors from Ping Chau and clearance of vessels from Mirs Bay area if sufficient time is available OR temporary sheltering if running out of time
AND Radiological screening at the Border Controls Points for returnees from area within 20 Km of the GNPS be implemented
AND Monitoring of foodstuff and livestock as well as water be initiated AND
AND All Monitoring Centres be opened for evacuees from Ping Chau and Mirs Bay, and for the returnees via Lo Wu, Man Kam To and Sha Tau Kok. Provided that these centres are unlikely affected by the plume
AND Distribution of iodinetablets when situation required

5. Knowledge Verification and System Evaluation

5.1 Knowledge Verification

Knowledge must be verified for "redundancy, inconsistency, incompleteness, circularity, and other errors" [Tepandi, 1991, quoted in H. O. Nourse *et.al.*, 1994]. Verification has also been characterised as the process of determining "if the system was built right " [O'Keefe *et. al.*, 1987 quoted in H. O. Nourse *et.al.*, 1994]. While T. J. O'Leary [1990] suggested that verification is the " authentication that the formulated problem contains the actual problem in its entirety and is sufficiently well structured to permit the derivation of a sufficiently credible solution".

EXSYS has a self-contained verification function which help identify utilisation (frequency of usage) of rules, qualifiers, variables and choices, and which of them have never been used during the verification. This function are frequently performed during the development of the prototype until no error is identified and rules which have never been used are picked up for testing by case simulation. Besides, testing cases have been conducted: 6 cases for Part A and 10 cases for Part B. These are critical cases and testing results are quite satisfactory.

5.2 System Evaluation

T. L. O'Leary *et. al.* [1990] defined evaluation as "the process of examining an expert system's ability to solve real-world problems in a particular problem domain. Evaluation focuses on the expert system and the real world."

Validation - According to T. J. O'Leary *et. al.* [1990], Turing test is to test the system's ability to supply response comparable to an expert's decision [Wallace, 1985]. In **Part B**, data from a total of four important exercises were used for validation. The experts considered that the recommendation by DBCP ADVISOR were very close to the human experts' recommendations.

User evaluation – A user survey has also been conducted in the form of a user survey to measures (1) usefulness (2) logic (3) system design (4) user-friendliness and (4) potential further development of the system. Twelve colleagues in the Emergency Support Unit of the SB and HKO participated in the survey. Results of the survey indicated that DBCP ADVISOR is well received. The mean

scores of colleagues strongly agree (36.4%) and agree (50.6%) in terms of the system's usefulness, system logic, design, user-friendliness as well as the potential for further development are very satisfactory.

6. Limitations

The main difficulty involved is that the process to arrive at final consensus on recommended counter-measures is a very complicated process involving much discussion and exchange of view and opinion among many experts in different fields. The prototype can only base on limited scope of knowledge provided by the nuclear plant experts in a short period of time and numerous determining factors that leading to the recommendations have been simplified. In real situation, sometimes experts have to make decision base on personal experience and intuitive judgement.

7. Recommendation and Further Study

Given the contributions of the prototype and positive user feedback, the usefulness of the system can be further enhanced by developing it into a full version, of course enriching the knowledge-base is required. As the system has very high compatibility with other external programs and can accept inputs from various kinds of database files, it can be further enhanced through system integration. On the output end, the system's capability to link to the Internet provides great flexibility to enrich the system with supplementary information and explanation to the users. Nevertheless the system currently lacks learning capability, further study to develop a hybrid system with the introduction of the Neuro Network technique, or to build up historical databases from previous exercises may enable learning through case-base reasoning [Zhu and Yang, 1999; Eyke Hullermeier, 1999], data mining approaches [Liu and Sin, 2000] and other intelligent business advisory systems [Liu, 2001; Liu et al, 1998; Liu and Jane, 2000]. Finally, further study on the feasibility to apply the system to other contingency plans such as the plans on natural disaster and aircraft crash etc. will also generate benefits in ensuring security in the society.

References

[Boutilier *et al.*, 1997] C. Boutilier, R. Brafman, C. Geib. Processes: Towards a Synthesis of Classical and Decision Theoretic Planning. *Proceedings of IJCAI' 97*, pages 1156-1162, 1997.

[Constantin, 1995] Constantin von Altrock. *Fuzzy Logic and NeuroFuzzy Applications Explained*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[Hullermeier, 1999] E. Hullermeier. Toward a Probabilistic Formalization of Case-Based Inference. *Proceedings of IJCAI' 997*, 1999.

[Kingston *et al.*, 1997] J. Kingston, A. Griffith and T. Lydiard. Multi-Perspective Modelling of the Air Compaign Planning process. *Proceedings of IJCAI' 97*, pages 668-677, 1997.

[Liu, 2001] Liu, N.K. An intelligent business advisory system for stock investment. *Encyclopedia of Microcomputers*, 27: 161-184, 2001.

[Liu and Jane, 2000] James N.K. Liu and Jane You. An Agent-based intelligent system for E-commerce applications. *Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in virtual organizations and E-commerce (MAMA'2000)*, December 11-13, 2000, Wollongong, Australia.

[Liu and Sin, 2000] Liu, J.N.K. and Sin, D.K.Y. A data mining approach for maintenance scheduling. *International Journal of Engineering Intelligent Systems*, 8(2): 119-126, 2000.

[Liu *et al*., 1998] Liu, J., Mak, T., Tang, B, Ma, K. and Tsang, Z. An intelligent Job Counseling System. Hing-Yan Lee and Hiroshi Motoda, Eds., Springer-Verlag, *Lecture Notes in Artificial Intelligence* 1531: 518-529, 1998.

[Martin, 1988] Martin James. *Building Expert Systems : A Tutorial*. Englewood Cliffs, N. J. :Prentice Hall, 1988.

[Medsker, 1994] Medsker Larry. *Design and Development of Expert Systems and Neural Networks*. New York: Macmillan; Toronto: Maxwell Macmillan Canada; New York: Maxwell Macmillan International, 1994.

[Nourse, 1994] Nourse Hugh O., Watson Hugh J., Bostrom Robert P., and Gatewood Robert D. Evaluation of a Generic Expert system for Corporate Real Estate Disposition. *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences*. Edited by Nunamaker Jay F. Jr. and Sprague Ralph H., Jr., Vol III, IEEE Computer Society Press, 1994.

[Leary, 1990] O'Leary Timothy J., Goul Michael, Moffitt Kathleen E., Radwan A. Essam Validating Expert Systems. *IEEE Expert*, 8 (3): pages55-58, 1990.

[HKSAR, 1999] The Government of Hong Kong Special Administrative Region Daya Bay Contingency Plan, 1999.

[Turban, 1993] Turban Efraim. *Decision Support and Expert Systems: Management Support Systems*. New York: Macmillan; Toronto: Maxwell Macmillan Canada; New York: Maxwell Macmillan International, 1993.

[Wallace, 1985] Wallace William A. and De Balogh Frank. Decision Support Systems for Disaster Management. *Public Administration Review*, (Special Issue), 45: 134-146, Jan.1985.

[Zhu and Yang, 1999] J. Zhu and Q. Yang. Remembering to Add: Competence-preserving Case-Addition Policies for Case Base Maintenance. *Proceedings of IJCAI' 99*, 1999.

ExpertClerk: Navigating Shoppers' Buying Process with the Combination of Asking and Proposing

Hideo Shimazu
NEC Corporation
8916-47 Takayama, Ikoma, Nara, Japan
shimazu@ccm.cl.nec.co.jp

Abstract

This paper analyzes conversation models of human salesclerks interacting with customers. The goal of a salesclerk is to effectively match a customer's buying points and a product's selling points. To achieve this, the salesclerk alternates among asking questions, proposing sample goods, and observing the customer's responses. Based on this analysis, we developed ExpertClerk, an agent system that imitates a human salesclerk and navigates Web shoppers in merchandise databases. In the system, a character agent talks with a shopper in a natural language and consolidates the shopper's request by narrowing down a list of many matching goods by asking effective questions using entropy (Navigation by Asking). Then, it shows *three* contrasting samples with explanations of their selling points (Navigation by Proposing). This cycle is repeated until the shopper finds an appropriate good. Evaluations show that the combination of Navigation by asking and Navigation by proposing works as most effectively as human salesclerks.

1 Introduction

This paper describes a salesclerk conversation model that interacts with customers. It is used in e-tailer Web sites to help customers easily find, compare, and decide on appropriate goods from among a lot of merchandise goods.

Specifically, we have implemented ExpertClerk as an agent system that imitates a human salesclerk and navigates Web shoppers in merchandise databases. In the system, a character agent talks with a shopper in a natural language and consolidates the shopper's request by narrowing down a list of many matching goods by asking effective questions using entropy (*Navigation by Asking*). Then, ExpertClerk shows *three* contrasting samples with explanations of their selling points (*Navigation by Proposing*). This cycle is repeated until the shopper finds an appropriate good.

The motivation behind this research was two-fold. First, we have developed several knowledge retrieval systems for customer support using conversational case-based reasoning techniques [Aha & Breslow, 1997]. Recently, requests have been increasing by internal and external customers to apply these techniques into e-tailer Web sites to help shoppers find and decide on goods in large-scale merchandise databases. This prompted us to develop prototype systems and show them to our clients. However, their responses were not good. This is because the system designs were based on conversations between customers of products and customer support agents. Most complaints centered around the conversations in the shops being different from those in customer supports. Other complaints were heard about the knowledge retrieval systems not having a different conversation structure and control.

Second, the Patricia Seybold Group reported the top 10 e-tailer ranking of the 1999 pre-Christmas season [Seybold, 1999], and Lands' End (landsend.com) earned the number-one spot. According to their survey, it was the company's superb customer service that made it first. The Web site provides the ability to press a button and talk to a customer service representative from the Web site via live chat or phone for shopping help, professional advice, or gift suggestions. We were surprised that even the top e-tailer site relies on the conversation skills of human representatives.

We surveyed various conversation skill's manuals for human salesclerks. The survey taught us that salesclerks effectively match customers' buying points and products' selling points. Today's Web shop sites including Lands' End do not have such human-like salesclerk-customer conversational interfaces. However, ExpertClerk was designed and developed to imitate the conversation techniques of actual salesclerks.

2 Conversations between Salesclerks and Shoppers

In a conversation between a shopper and a salesclerk, the shopper plays the role of a decision-maker and the salesclerk plays the role of an adviser. The conversation is a clarifying process of the subconscious desire of the decision-maker. The adviser is a catalyst that promotes the shopper's decision-making. The following sequence shows a typical conversation between a shopper and a salesclerk (S: Shopper, C: Clerk).

> S: Please show me that blouse.
> C: Ok. We have a similar color that may also suit you. And, this is a blouse of the same type but with a different design.

S: Well, the design is fine, but the neck looks very tight.
C: How about this one if you prefer a loose neckband? That one also has a loose neckband and an interesting design.
S: I like this one. How much is it?
C: It is $200.
S: Wow. $200 is a bit too expensive.
C: How about this one? It has a similar design and color, but the price is only $88. The material is polyester.
S: Ok, I'll take it.
C: Thank you.

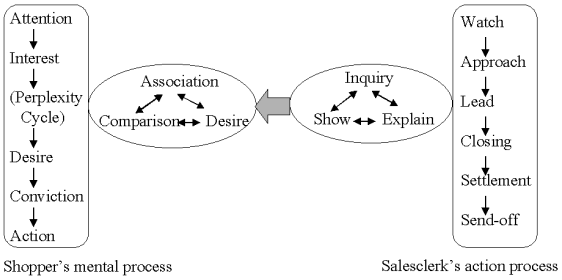

Figure 1: AIDCA model

Figure 1 show a typical mental process of shoppers, called AIDCA model [Shigeta, 1995]. A shopper first turns his/her *attention* towards a specific good and becomes *interested* in the good. The shopper then starts to *desire* the good and becomes perplexed about the decision, i.e., in the *perplexity cycle* of *desire*, *comparison* and *association*. The shopper eventually decides to buy the good with *conviction*, and moves to the purchasing *action*.

A salesclerk follows the shopper's AIDCA process to promote each step. The salesclerk *watches* for a chance and *approaches* the new shopper. The approach must be done at the appropriate time. If it is too soon, the shopper guards against the salesclerk. If it is too late, the salesclerk is regarded as a dull clerk. When the conversation starts, the salesclerk asks a few questions to identify the shopper's necessary conditions (buying points). Then, the salesclerk picks a few sample goods and shows them to the shopper and explains their selling points.

We analyzed a real conversation corpus between shoppers and salesclerks and interviewed senior salesclerks. In order to understand a customer's buying points, a salesclerk alternates between asking questions and proposing sample goods. Many salesclerks said the best approach is to present *three* sample goods at the same time. According to these salesclerks, shoppers become frustrated if they are shown only one or two samples. On the other hand, if four or more samples are presented, they become puzzled as to which to choose. Each of these three samples must have clear selling points that differentiate itself from the other two. The salesclerk explains the selling points, like for example, "This is twice as expensive as those because it is made of silk and the other two are made of polyester". While hearing the explanations, the shopper can more easily exclude one of the three proposed goods with a specific reason, like "this one is too dark for me compared with the other two". The salesclerk excludes it and chooses a new one satisfying the shopper's comment. The salesclerk repeats picking sample goods, explaining their selling points, and modifying his/her sample picking strategy by observing the customer's responses.

It is also very important that the salesclerk presents appropriate sample goods. If a proposed sample is opposite to a shopper's preference, the shopper may feel distrust towards the opinion of the salesclerk. In order to avoid this, the salesclerk must first ask a few effective questions to infer the shopper's preference.

3 Design Decision

3.1 Modeling a salesclerk's action sequence

ExpertClerk was designed to model the typical action sequence of human salesclerks. It has the following steps:

1. Approach: ExpertClerk approaches a shopper.
2. Navigate by asking appropriate questions: ExpertClerk finds a set of goods matching the shopper's request. If too many matching goods exist, ExpertClerk narrows them down by asking a few questions using entropy which effectively discriminates the shopper's intention.
3. Navigate by proposing three sample goods: If the shopper can not identify his/her own intention, ExpertClerk shows three contrasting samples with explanations of their selling points.
4. Observe: ExpertClerk observes the shopper's reactions on likes/dislikes and why.
5. Repair: ExpertClerk modifies the proposal to fit the shopper's responses.

The process repeats until the shopper finds an appropriate good. The key to ExpertClerk effectiveness is its combination of Navigation by asking and Navigation by proposing. The following sections explain these features.

3.2 Merchandise database

ExpertClerk is a front-end system for a merchandise database. We assume that merchandise records are represented as a flat record of n-ary fields, and stored in a table in a commercial RDBMS. Each field stores an attribute of merchandise, such as price, color, material, the country of origin, brand, and so on. For each record attribute, a conceptual hierarchy is defined by domain experts.

This hierarchy represents a classification and is defined as a discrimination tree. Each node in the hierarchy is a question node that subdivides the set of nodes stored underneath it. Each child node represents a different answer to the question posed by its parent. Each leaf node refers to a set of merchandise records which satisfy the above discrimination conditions. For example, the "price" attribute may have a question "Which price range are you interested in?". The possible answers may be divided into "Very expensive", "Expensive", "Reasonable", "Fairly cheap", or "Very cheap". They can be

subdivided into more detailed categories. "Very expensive" may be divided into "Over $10,000?" and "Between $1,000 and $9,999?". Each merchandise record is linked from several different leaf nodes of different hierarchies.

These questions and possible answers are used to discriminate merchandise records. The discrimination capability of each question is different. It depends on the set of remaining merchandise records. For example, "Which price range are you interested in?" is useful if the remaining goods have various price ranges. However, the same question is meaningless if all of the remaining goods are over $10,000.

3.3 Navigation by asking

Navigation by asking calculates the information gain of possible questions and sorts them according to their statistical efficiency. It works as follows:

(1) Extracting questions. Within each question hierarchy having leaf-nodes pointing to merchandise records, the paths from each of these leaf-nodes are traversed in the direction of the root, and the first question node reached that is common to all of these leaves is extracted as a *question* node. Then, the nodes on the level just below this question node are extracted as its *answer* nodes.

(2) Calculating the information amount of question nodes. Questions are determined from their calculated information amount. The algorithm is based on ID3's [Quinlan, 1986] information gain approach. Let C = $\{r_1, r_2, ..., r_k\}$ be the set of retrieved question nodes and let h_j be the retrieval counts of $r_j (1 \leq j \leq k)$. Then, the occurrence probability p_j of r_j is calculated by Equation (1) . The entropy $M(C)$ of C is given by Equation (2). When C is divided into subsets C_1, C_2, ..., C_n by answer nodes a_1, a_2, ... a_n of question node a, the expected information $B(C, a)$ is given by Equation (3). The information $gain(C, a)$ gained by question node a is given by Equation (4). Dividing C into subsets by using question node a which maximizes $gain(C, a)$ should narrow down the number of result sets efficiently.

$$p_j = \frac{h_j}{\sum_{i=1}^{k} h_i} \quad (1)$$

$$M(C) = -\sum_{j=1}^{k} p_j \log_2 p_j \quad (2)$$

$$B(C, a) = \sum_{i=1}^{n} \frac{\mid C_i \mid}{\mid C \mid} M(C_i) \quad (3)$$

$$gain(C, a) = M(C) - B(C, a) \quad (4)$$

3.4 Navigation by proposing

It is very annoying for shoppers to be asked many questions. After merchandise records are narrowed down to a pre-defined threshold number after several questions, ExpertClerk changes its conversation mode from Navigation by asking to Navigation by proposing. In the navigation by proposing mode, three goods as the most contrasting among remaining goods are selected. They are selected by the following algorithm.

- The first sample good (1st-SG) is the goods record closest to the center point of the set. Its selling points directly reflect the customer's request.
- The second sample good (2nd-SG) is the goods record positioned most distantly from the center point of the set. The selling points of 2nd-SG clearly differentiate itself from 1st-SG.
- The third sample goods (3rd-SG) is the goods record positioned most distantly from 2nd-SG. 3rd-SG has selling points which differentiate itself from 1st-SG and 2nd-SG.

Let G = $\{g_1, g_2, ..., g_k\}$ be the set of retrieved goods records and let $g_i = \{f_{i1}, f_{i2}, ...f_{in}\}$ be the set of attribute values. Then the median g_{med} of G is calculated by Equation (5) and the standard deviation of the ith attribute value set $g_{dev(i)}$ is calculated by Equation (6)

$$g_{med} = (\frac{1}{k}\sum_{j=1}^{k} f_{j1}, \frac{1}{k}\sum_{j=1}^{k} f_{j2}, ..., \frac{1}{k}\sum_{j=1}^{k} f_{jn}) \quad (5)$$

$$(k-1)g_{dev(i)}^2 = (\sum_{j=1}^{k} g_j - g_{med(i)})^2 \quad (6)$$

The distance (D) between g_{med} and a sample g_i is given by Equation (7).

$$D(g_{med}, g_i) = \frac{\sum_{j=1}^{n} W_j \times d(g_{med(j)}, g_{i(j)})}{\sum_{j=1}^{n} W_j} \quad (7)$$

where $g_{med(j)}$ is the j-th attribute of g_{med}, $g_{i(j)}$ is the j-th attribute of g_i, and W_j is the j-th attribute weight.

1st-SG is a record whose distance from g_{med} $D(g_{med}, g_i)$ is the shortest among $g_j (1 \leq j \leq k)$. 2nd-SG (g_{2nd-sg}) is a record whose distance from g_{med} $D(g_{med}, g_i)$ is the largest among $g_j (1 \leq j \leq k)$. 3rd-SG is a record whose distance from g_{2nd-sg} $D(g_{2nd-sg}, g_i)$ is the largest among $g_j (1 \leq j \leq k)$. Figure 2 shows the dot frequency diagram of a set of retrieved goods records. Three points represent the median and range of the set. 1st-SG is located at the center of the diagram, 2nd-SG is located at the right edge of the diagram, and 3rd-SG is at the left edge of the diagram. A salesclerk's proposing strategy is modified by the shopper's response to the proposal. A shopper's response is typically expressed with "I don't like this sample because of this attribute value A. I want the attribute value to be more/less than this value V". The salesclerk's next proposal reflects this response.

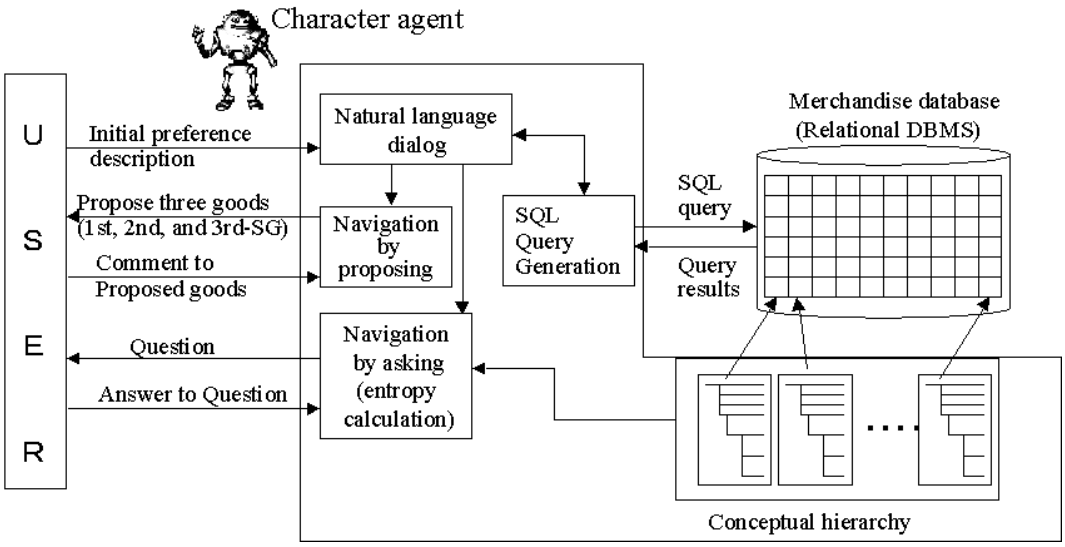

Figure 3: The ExpertClerk System

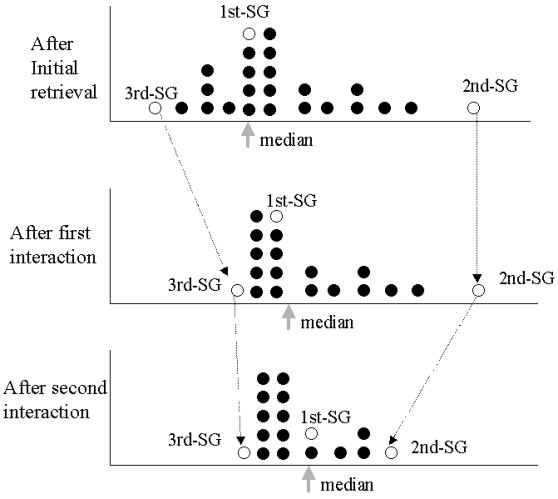

Figure 2: Narrowing down records by manipulating three samples

4 The ExpertClerk System

Figure 3 shows the ExpertClerk system structure. The user interaction module uses a natural language dialog system [Shimazu et al., 1992] with a character animation. It translates a user's request into a corresponding SQL query and issues the query to a backward relational DBMS. The module is integrated with the navigation by proposing module and the navigation by asking module and calls these modules when either asking questions to a user or proposing three samples to the user. The navigation by asking module refers to conceptual hierarchies each of which corresponds to each attribute of the backward relational DBMS.

The selection strategy of the two navigation modules is defined and can be changed by a few parameters. A system administrator defines the parameters, such as "If the number of matching records becomes less than seven, change to Navigation by proposing" or "Use both narrowing modules at random". Figure 4 shows the screen image of ExpertClerk.

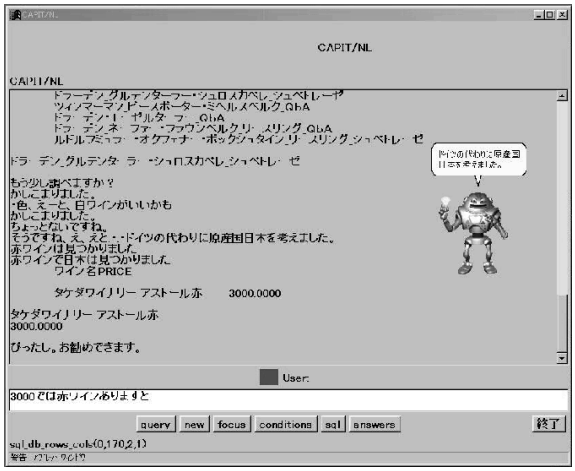

Figure 4: The ExpertClerk screen image

5 Evaluation

We evaluated the efficiency and precision of ExpertClerk. We used a small wine database of 150 wine records. This database is used to recommend wines meeting a user's specific conditions. All of the wine records have the same structure and each record has ten attribute values. For each attribute, a similarity measure and weight are defined.

Since many of the records in the database have unique solutions, we could not evaluate the retrieval precision based on the correctness of the retrieved merchandise record. Instead, we tested ExpertClerk using Aha and Breslow's *leave-one-in*

testing methodology [Aha & Breslow, 1997] in which each merchandise record is used once as a test record, but without removing it from the merchandise record database during testing. The *leave-one-in* method randomly selects a test record from the database. A conversation starts by activating retrieval methods in ExpertClerk. A user interacts with ExpertClerk by entering a query or excluding one of three sample wine records proposed by ExpertClerk. The conversation ends after the interaction exceeds *m* times or after retrieved records are narrowed down to the test record. The efficiency is measured by the degree of narrowing down after *n* ($n \leq m$) interactions. The precision is measured by examining whether the retrieved records match the test record.

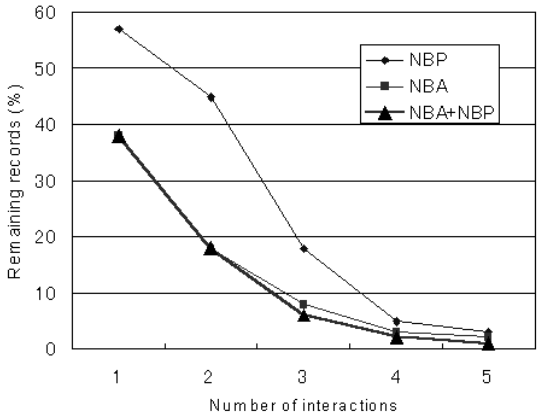

Figure 5: Degree of narrowing down records at each interaction

Figure 5 shows how effectively records are narrowed down per interaction by *Navigation by asking* (NBA), *Navigation by proposing* (NBP) and their combination (NBA+NBP) . The first and second interaction of NBA+NBP is NBA and the following interactions are NBP.

NBA narrows down possible records to between one third and one half at each interaction, but it can narrow down little at the third, fourth, or fifth interaction. NBP with no customer input can not narrow down records effectively. After a few interactions with a user's response, however, it can work well and the remaining records can decrease rapidly. NBA+NBP takes the best parts of NBA and NBP. At early interactions, NBA works better than NBP. At later interactions, NBP works better than NBA. In addition, because Navigation by proposing sounds more gentle than Navigation by asking as the style of conversation, the combination of NBA and NBP is the best among the three.

Figure 6 shows how precisely records are narrowed down per interaction. The figure shows the similarity between a test record and each retrieved record set (1st-SG, 2nd-SG, and 3rd-SG) per interaction. The higher border shows the best similarity values for a test record among 1st-SG, 2nd-SG, and 3rd-SG and the lower border shows the worst similarity values for a test record among them. The dashed line shows the similarity value between a test record and the center of each retrieved record set. After the first and second interactions, the precision value of three samples are refined, and at the

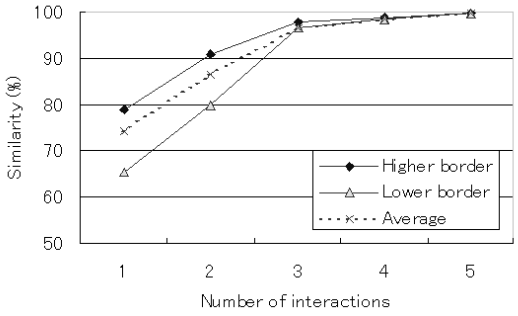

Figure 6: Precision during narrowing down records at each interaction

third interaction, the three samples shown to the customer are nearly appropriate with higher than 95% accuracy.

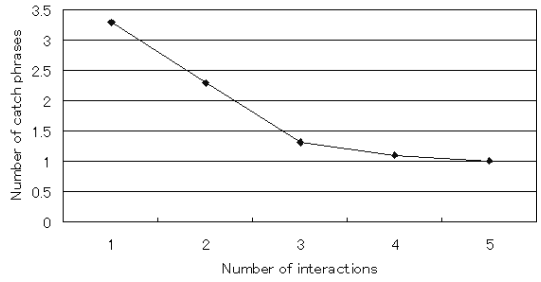

Figure 7: Generated selling points at each interaction

Figure 7 shows how many selling points can be presented to a customer per interaction. At early interactions, each of 1st-SG, 2nd-SG, and 3rd-SG has its own selling points. The number of selling point decreases at later interactions because the retrieved records eventually becomes homogeneous and the three samples also become similar.

6 Related research

Several research projects have focused on user-interface issues of conversational case-based reasoning (CBR). Our research was inspired by FindMe [Hammond et al., 1996; Burke et al., 1997] and its enhanced version The Wasabi Personal Shopper (WPS) [Burke et al., 1999]. FindMe combines instance-based browsing and tweaking by difference. The system shows a user an example of retrieved results. By critiquing the example, the user then indicates his/her preferences and controls the system's retrieval strategies. NaCoDAE [Aha et al., 1998] has a dialogue-inference mechanism to increase the conversational efficiency. It infers the details of a user's problem from his/her incomplete text description by using model-based reasoning. It is especially useful in front office systems like help-desk or sales-support systems. Wilke, Lenz and Wess [Wilke et al., 1998] surveyed various research projects that applied CBR into sales support solutions on the Internet. They pointed out the importance of ne-

gotiation, a process in which two parties bargain resources for an intended gain. They classify negotiation into competitive negotiation and cooperative negotiation. The former is used in various Internet auction sites. ExpertClerk can be categorized under the latter.

The incremental query modification technique of Navigation by proposing is influenced by the relevance feedback technique used in the SMART system [Salton, 1983]. Relevance feedback incrementally shifts the query vector to the center of a related document cluster. The navigation by asking module is based on ID3 [Quinlan, 1986] and was first described in our previous work, ExpertGuide [Shimazu et al., 2001]. Selling points finding can be regarded as a salient feature extraction problem and is discussed in many CBR research projects [Kolodner, 1993]. KASBAH [Chavez & Maes, 1996] is an intelligent agent with negotiation capabilities and has a similar idea to selling point. Negotiation rules are defined for each attribute of goods and activated during every negotiation step.

Several CBR projects have applied CBR techniques into commercial RDBMS [Shimazu et al., 1993; Watson & Gardingen, 1999; Burke et al., 1999]. It is important to design an architecture in a simple and scalable manner if it is to be used in large-scale Web shopping sites.

7 Conclusion

This paper analyzed conversation models of salesclerks interacting with customers. A salesclerk alternates among asking questions, proposing sample goods, and observing the customer's responses. Based on this analysis, we developed ExpertClerk, an agent system that imitates a human salesclerk and navigates Web shoppers in merchandise databases. In the system, a character agent talks with a shopper in a natural language and consolidates the shopper's request by narrowing down a list of many matching goods by asking effective questions using entropy (Navigation by asking). Then, it shows three contrasting samples with explanations of their selling points (Navigation by proposing). This cycle is repeated until the shopper finds an appropriate good.

Evaluations showed that the combination of the two navigation modes works most effectively as human salesclerks do. Finally, although the human salesclerks we surveyed insisted that it is the best approach to propose three goods at the same time, we have not examined any subjective evaluation on this. It is a future issue to justify the ExpertClerk conversation model.

References

[Aha & Breslow, 1997] Aha, D.W. and Breslow, L.A.: 1997, Refining conversational case libraries, *Proceedings of the Second International Conference on Case-Based Reasoning*, pp. 267 – 278.

[Aha et al., 1998] Aha, D.W., Maney, T., and Breslow, L.A.: 1998, Supporting dialogue inferencing in conversational case-based reasoning, *Proceedings of the 4th European Workshop on Case-Based Reasoning*, pp. 267 – 278.

[Burke et al., 1997] Burke, R.D., Hammond, K.J., and Young, B.C.: 1997, The FindMe approach to assisted browsing, *Journal of IEEE Expert*, **Vol. 12, 4**, pp. 32 – 40.

[Burke et al., 1999] Burke, R.: 1999, The Wasabi Personal Shopper: A Case-Based Recommender System. Proceedings of the Seventeenth National Conference on Artificial Intelligence, Menlo Park, CA: AAAI Press.

[Chavez & Maes, 1996] Chavez, A. and Maes, P. : 1996, Kasbah: An Agent Marketplace for Buying and Selling Goods. Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology, London.

[Hammond et al., 1996] Hammond, K.J., Burke, R., and Schumitt, K.: 1996, A case-based approach to knowledge navigation, in: Leake, D.B. (Eds.): *Case-Based Reasoning Experiences, Lessons, & Future Directions*, pp. 125 – 136, Menlo Park, Calif.: AAAI Press.

[Kolodner, 1993] Kolodner, J.: 1993, *Case-Based Reasoning*, San Francisco, Calif.: Morgan Kaufmann.

[Quinlan, 1986] Quinlan, J.R.: 1986, Induction of decision trees. *Journal of Machine Learning*, **Vol. 1**, pp 81 – 106.

[Seybold, 1999] Seybold, P.B. and Miller, J.: 1999, Top 10 Pre-Holiday E-Tailing Picks, Customers.com / Perspective, www.customers.com

[Salton, 1983] Salton, G.: 1983, An Introduction to Modern Information Retrieval New York: McGraw-Hill

[Shigeta, 1995] Shigeta, T.: 1995, 30 lessons on how to be an excellent salesclerk. (in Japanese) Keirin Shobo: Tokyo.

[Shimazu et al., 1992] Shimazu, H., Arita, S., and Takashima, Y.: 1992, Design Tool Combining Keyword Analyzer and Case-based Parser for Developing Natural Language Database Interfaces *Proceedings of the Fourteenth International Conference on Computational Linguistics*.

[Shimazu et al., 1993] Shimazu, H., Kitano, H., and Shibata, A.: 1993, Retrieving cases from relational data base: Another stride towards corporate-wide case-based systems, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 909 – 914.

[Shimazu et al., 2001] Shimazu, H., Shibata, A., and Nihei, K.: 2001, ExpertGuide: A Conversational Case-Based Reasoning Tool for Developing Mentors in Knowledge Spaces. *Applied Intelligence*, 14, pp 33 – 48, Kluwer Academic Publishers, January 2001.

[Watson & Gardingen, 1999] Watson, I. and Gardingen, D.: 1999, A Distributed Case-Based Reasoning Application for Engineering Sales Support, *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann.

[Wilke et al., 1998] Wilke, W., Lenz, and M., Wess, S.: 1998, Intelligent Sales Support with CBR, in: Lenz, M., Bartsch-Spörl, B., Burkhard, H-D., and Wess S. (Eds.): *Case-Based Reasoning Technology From Foundations to Applications*, pp. 91 – 113, Lecture Notes in Artificial Intelligence 1400, Springer.

WEB APPLICATIONS

WEB SEARCH

Preference-Based Configuration of Web Page Content*

Carmel Domshlak Ronen I. Brafman Solomon E. Shimony
Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
{dcarmel,brafman,shimony}@cs.bgu.ac.il

Abstract

We present a new approach for personalized presentation of web-page content. This approach is based on preference-based constrained optimization techniques rooted in qualitative decision-theory. In our approach, web-page personalization is viewed as a configuration problem whose goal is to determine the optimal presentation of a web-page while taking into account the preferences of the web author, layout constraints, and viewer interaction with the browser. The preferences of the web-page author are represented by a CP-network, a graphical, qualitative preference model developed in [Boutilier *et al.*, 1999]. The layout constraints are represented as geometric constraints. We discuss the theoretical basis of this approach and its implementation within the CPML system.

1 Introduction

An important goal for web-page design is to provide viewer-oriented personalization of web-page content. Designers may strive to condition web-page content and appearance on the current preferences of the viewer, and probably on some underlying structure of the web-page content. For example, when the user is viewing an article about the consequences of a traffic accident in an on-line newspaper, the designer may wish to present a Volvo ad, as well.

Much of the content personalization literature focuses on learning user profiles. Although this technique is useful, it generally suffers from low availability, and tends to address only long-term user preferences. These schemes are thus applicable only to frequent viewers, that are, in addition, amiable to having information about their behavior managed by an external agent.

We propose a new model for representing web-page content. This model is unique in two ways. First, it emphasizes the role of the author in the process, viewing her as a content expert whose taste is an important factor in how the web-page will be presented. The resulting model exhibits dynamic response to user preferences, but does not require learning long-term user profiles. Second, to accomplish this behavior, we use well-founded tools for preferences elicitation and preference optimization grounded in qualitative decision theory. These tools help the designer structure her preferences over web-page content off-line, in an intuitive manner, and support fast algorithms for optimal configuration determination.

A web-page is typically composed of several components. The information content of each component can be either presented or not. In our model, the first step is for the web-page designer to express her preferences regarding the presentation of the web-page content. For example, the author may prefer some material to be presented if and only if some other material is not presented. This is done in an intuitive yet expressive manner using the CP-network (short for *ceteris paribus* networks) [Boutilier *et al.*, 1999] which are an intuitive, qualitative, graphical model of preferences that captures statements of conditional preferential independence. The description of these preferences, as captured by the CP-net, becomes a *static* part of the web document, and sets the parameters of its initial presentation. Then, for each particular session, the actual presentation changes *dynamically* based on the user's actual choices. These choices exhibit the user's content preferences. They are monitored and reasoned about during each session. No long-term learning of a user profile is required, although it can be supported.

Using this approach, we achieve content personalization through dynamic preference-based reconfiguration of the web-page. Whenever new user input is obtained (e.g., a click indicating his desire to view some item), the configuration algorithm attempts to determine the best presentation of all web-page components with respect to the web-page designer's preferences that *satisfies* the user's viewing choices. This process is based on an algorithm for constrained optimization in the context of a CP-network.

Determining only preferred content is generally insufficient – the chosen configuration should be presentable w.r.t. the layout constraints of the viewer's browser. Likewise, a web-page designer may wish to specify expectations of the exact appearance of the document, in addition to preferences about content. Declarative specification of the desired layout of a web-page are well known: cascading style sheets were introduced as a part of the HTML 4.0 standard [Lie and Bos,

*Partially supported by an infrastructure grant from the Israeli Science Ministry, and by the Paul Ivanier Center for Robotics and Production Management.

1997], constraint style sheets were proposed in [Borning *et al.*, 2000].[1] We extend our content personalization approach to handle different layout constraints on web-page rendering, providing the viewer with a preferentially optimal feasible presentation of the web-page components.

We implement our approach in the framework of the Ceteris Paribus Markup Language (CPML) system, which consist of an authoring tool for the preference-based web-pages, and a corresponding viewing tool, which is implemented as a browser plugin.

The paper is organized as follows: Section 2 presents the framework of web-page preference-based configuration in the context of qualitative decision making. Section 3 discusses relevant preference representation issues, and describes the CP-network model. Section 4 describes the basic architecture and implementation of CPML. Section 5 illustrates the approach by example. Section 6 extends CPML by integrating layout constraints into the web-page optimization process.

2 Configuration and Qualitative DT

Any web-page can be considered as a set of components $C_1, \ldots, C_n$. Each component is associated with its content. For example, the content of a component may be a block of text, an image, etc. In our work, each component may be either presented to the viewer or hidden, and these options for C_i are denoted by c_i, c_i'. However, the model can be expanded to handle more options for components' content presentation.

The web-page components define a configuration space $\mathcal{C} = \{c_1, c_1'\}, \times \cdots \times \{c_n, c_n'\}$. Each element σ in this space is a possible configuration of the web-page content. Our task will be to determine the preferentially optimal configurations, and to present one of them to the viewer. In terms of decision theory, the set of components of web-page is a set of features, the optional presentations of component's content are the values of the corresponding feature, and configurations are outcomes, over which a preference ranking can be defined.

First we define a preference order $\succeq$ over the configuration space: $\sigma_1 \succeq \sigma_2$ means that the decision maker views configuration σ_1 as equal or more preferred than σ_2. This preference ranking is a total preorder, and, of course, it will be different for different decision makers. Given a preference order $\succeq$ over the configuration space, an *optimal configuration* is any $\sigma \in \mathcal{C}$ such that $\sigma \succeq \sigma'$ for any $\sigma' \in \mathcal{C}$.

The preference order reflects the preferences of a decision maker. The typical decision maker in preference-based product configuration is the consumer. However, in our application the role of the decision maker is relegated to another actor – the web author. The author is the content expert, and she is likely to have considerable knowledge about appropriate content combinations. We would like the web-page to reflect her expertise as much as possible.

During the creation of the web-page, the designer describes her expectations regarding content presentation. Therefore, the preference order $\succeq$ represents the static subjective preferences of the web-page designer, not of its viewer. Thus, preference elicitation is performed on the web-page designer off-line once for all subsequent accesses to the created web-page. The dynamic nature of the web-page stems from the interaction between the statically defined author preferences and the constantly changing content constraints imposed by recent viewer choices.

The choice of a representation model for the preferences is crucial in any preference driven configuration process. In particular, one can adopt either a quantitative, utility-theoretic, model, or a qualitative model. We believe that qualitative models can form a good basis for the automated product configuration in general, and for the web-page configuration in particular. For a comprehensive overview of the field of qualitative decision theory see [Doyle and Thomason, 1999]. The main advantages of qualitative decision theory tools, as opposed to traditional decision theory, are compactness, intuitiveness – which can considerably reduce the preference elicitation burden – and potentially reduced computational effort. Thus, in our domain, designers are likely to find (quantitative) utility assessment of content configurations unintuitive. Yet, the web-page designer is likely to be able to (qualitatively) compare and to rank alternative content presentations.

3 Preference Representation

Although a qualitative representation of preferences is typically simpler to represent and manipulate, the preference elicitation stage can still be quite complicated. To perform real-life preference-based configuration, we must represent user preferences in a compact, yet expressive manner. Even relatively small web-pages may consist of numerous components, making an explicit (exponential size) ranking table for all alternative content configurations impractical. Likewise, we wish to capture conditional preference dependencies between different components. For example, the designer may prefer to expose a short IJCAI call for papers (CFP) if the viewer explicitly examined the content of an article about a new book on AI, but not to expose the CFP otherwise.

An appropriate representation of preferences is insufficient on its own; we need to be able to reason about them efficiently. Reasoning tasks include finding preferentially optimal configurations, comparing two configurations. Likewise, the representation model should be intuitive in order to simplify the web-page designer's task.

Because of these requirements, in our work we decided to exploit the advantages of the CP-network model developed in [Boutilier *et al.*, 1999]. This is an intuitive, qualitative, graphical model, that represents statements of conditional preference under a *ceteris paribus* (all else equal) assumption. In terms of our domain, this conditional *ceteris paribus* semantics requires the web-page designer to specify, for any specific component C_i of interest, content presentation of which other components $\Pi(C_i)$ can impact her preferences over the presentation options of C_i. For each content configuration π of $\Pi(C_i)$, the designer must specify her preference ordering over the presentation options of C_i given π. For example, suppose that the designer determines that $\Pi(C_i) = \{C_j, C_k\}$ and that c_i is preferred to c_i' given c_j and c_k' *all else being equal*. This means that given any two configurations that agree on all components other than C_i and in which C_j is

[1] For a discussion on using constraints in user interfaces and interactive systems we refer to [Borning *et al.*, 2000].

IN and C_k is OUT, the configuration in which C_i is OUT is preferred to the configuration in which C_i is IN.

CP-networks bear a surface similarity to Bayesian networks: their representation is by a directed acyclic graph, with additional information - for each possible state of each node's predecessors. A node in the graph stands for a feature, which here represents a component C_i of the web-page. The immediate predecessors of C_i in the graph are associated with $\Pi(C_i)$. Formally, if $\overline{C_i} = \{C_1, \ldots, C_n\} \setminus \{C_i, \Pi(C_i)\}$ then C_i and $\overline{C_i}$ are *conditionally preferentially independent* given $\Pi(C_i)$. This standard notion of multi-attribute utility theory can be defined as follows: Let X, Y, and Z be non-empty sets that form a partition of feature set F. X and Y are conditionally preferentially independent given Z, if for each assignment z on Z and for each x_1, x_2, y_1, y_2 we have that

$$x_1 y_1 z \succeq x_2 y_1 z \text{ iff } \mathrm{x_1 y_2 z} \succeq \mathrm{x_2 y_2 z}.$$

Each node of the CP-net contains a table, which describes the preferences about the values of the corresponding feature C_i given all possible combinations of $\Pi(C_i)$.

The acyclicity requirement is necessary, as introduction of cycles may cause an inconsistent preference structure. An effective algorithm for detecting such a problem does not yet exist, and we believe that the problem is NP-hard.

An example CP-network with the corresponding preference table is shown in Figure 1. We see that the designer specifies unconditional preference for presenting the content of component C_1 (denoted in figure by $c_1 \succ c_1'$). However, if C_1 content is presented and C_2 content is not, then the designer prefers not to expose the content of C_3 (denoted by $(c_1 \wedge c_2') : c_3' \succ c_3$).

In our domain, the interactive construction of a CP-network for web-page's content consists of two stages: for each component C_i of interest, asking the web-page designer to identify $\Pi(C_i)$, and to specify the C_i's preference table. Note that each complete assignment for the nodes of the network represents a configuration of the web-page content.

Suppose that there are no constraints on web-page rendering. Then all content configurations are feasible. In this case, given a CP-network representation of the preferences, finding preferentially optimal configuration is straightforward: Traverse the nodes of the network according to a topological ordering and set the value of the processed node to its preferred value, given the (already fixed) values of its predecessors. The more complicated case of constrained configuration space $\mathcal{C}$ will be discussed later in this paper.

4 CPML System Description

This section presents a framework for preference-based web-page configuration, and shows how decision theoretic tools provide a basis for this application. Past work has dealt with general preference-based configuration (e.g., [Boutilier *et al.*, 1997; D'Ambrosio and Birmingham, 1995], but its adaptation to information personalization has not been explored. A prototype system (CPML) has been developed at our university, and here we describe some of its implementation issues.

The process of a partly unsupervised, preference-based web-page content configuration can be divided into 3 stages:

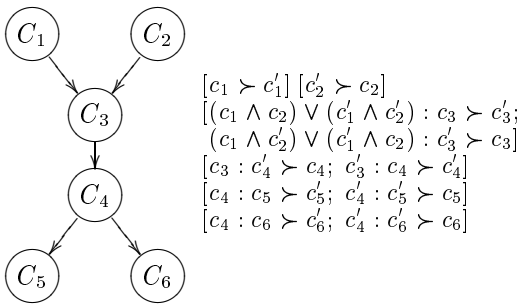

Figure 1: An example CP-network

1. Identify the components on whose content presentation the designer has a preference.
2. Apply interactive elicitation of the designer's preferences for the presentation of the selected components.
3. Reason about the specified preferences during viewer interaction with the web-page.

The CPML prototype system for this process consists of two modules - the *authoring tool*, and the *viewing tool*. The central part of the authoring tool is a specification of the CP-network for the created web-page (steps 1 and 2). Given such a specification, the viewing tool is responsible for reasoning about the preferences, i.e. for an optimal content reconfiguration after an interaction of the viewer with the web-page. Likewise, the authoring tool allows an optional specification of layout constraints on the web-page rendering. The nature of these constraints and their integration into the process of content optimization is described in details in Section 6. The outline of the system is presented in Figure 2.

First, the designer creates an HTML web-page. This task is standard for most web-page authoring tools. However, HTML document component contents and their representation are indivisible. Therefore, during step 1 (see below), the specified HTML components are automatically wrapped by javascripts [Rule and Rule, 1998]. This will allow presentation of components in different manners. Recall that this

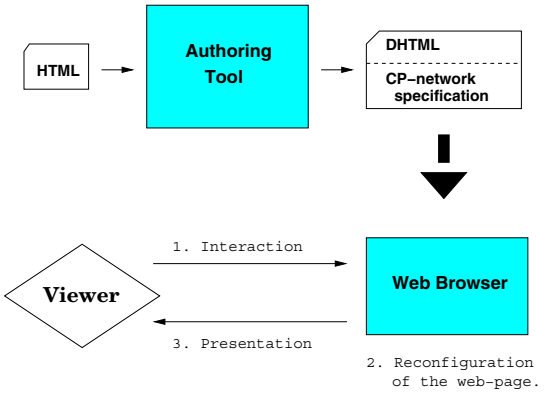

Figure 2: Framework of the CPML system

prototype system is restricted to only two options: presenting, or hiding, the component contents. The whole process of HTML expansion is completely transparent to the designer.

Next, step 2 in our approach is the definition of a CP-network over the specified components. For this purpose, an editor for CP-networks is integrated with the authoring tool. The result of preference-based web-page design, is a document specifying both what to present and how to present.

In turn, the viewing tool on the client side consists of a standard browser expanded by a CPML agent, implemented by a browser-compatible plugin. Accessing a preference-based web document in our system results in shipping the web-page to the browser and the embedded CP-network to the agent. Upon downloading the document, the agent sets all the components to their values in the optimal configuration, given no evidences on the viewer's current interests. From this point the agent acts in event-driven fashion, and waits for a viewer interaction with the browser. The user may either choose to expose or hide a component. The agent is responsible for reacting to viewer actions by reconfiguring the content presentation of the web-page. More specifically, the new content presentation should be a Pareto-optimal configuration of the web-page that contains (or omits) the item selected by the user for presentation (or for omission, respectively).

Determining the best configuration is done by the procedure presented in Figure 3. The simplicity of this algorithm

Procedure Reconfiguration
$\mathcal{G}$ - CP-network of the web page's components
$\mathcal{E}$ - Queue of the recent events (component, value)

Let e_i be the recent event: an interaction of the viewer with component C_i (setting on/off). Then

1. Switch the value of C_i and add e_i into $\mathcal{E}$. If $\mathcal{E}$ is bigger than some threshold - remove the oldest event from $\mathcal{E}$.
2. Project $\mathcal{E}$ on $\mathcal{G}$ (specify values of recently observed components w.r.t. $\mathcal{E}$).
3. Traverse $\mathcal{G}$ in a topological order and set each unspecified component to its most preferred value w.r.t. to the values of its predecessors in $\mathcal{G}$.

Figure 3: Preference-based Reconfiguration

stems from an important property that emerges from the semantics of CP-networks: parent preferences are more important than children preferences. This sanctions a fast top-down traversal process for generating optimal configurations.

As the CPML prototype is restricted to components with only two presentation options, in step 1 of Reconfiguration, we *switch* the value of the observed component, and then add it to the list $\mathcal{E}$ of the recent viewer's choices. The maximal size of $\mathcal{E}$ is specified by the designer of the web page and is passed as a part of the document. In step 2, we specify the values of the recently observed components as constraints on the required optimal configuration of the web-page. This is done in order to personalize the content to the viewer by reflecting her recent choices in content configuration. Finally, in step 3, given the constrained components, we determine the optimal configuration of the web-page. Due to the semantics of the CP-network model, this process can be performed in time linear in the number of components.

5 Example

We illustrate the process through the following example. The designed web-page consists of seven components: four short articles, and three commercials. The articles are about the ongoing elections (`Elections`), a traffic accident (`Traffic Accident`), a new car airbag (`New Airbag`), and the results of the recent NBA games (`NBA`). The commercials are for New-York Times magazine (`NY Times`), Volvo cars (`Volvo`), and Nike shoes (`Nike`). After specification of the web-page content, the designer expresses her preferences about the content presentation:

1. By default, presenting the central article `Elections` (C_1) is preferred to hiding it (i.e. C_1 **on** is preferred to C_1 **off**. For the secondary article `Traffic Accident`, C_2 **off** is preferred to **on**.
2. Article `New Airbag` (C_3): C_3 **on** is preferred only if `Traffic Accident` is **on** and `Elections` is **off**.
3. Article `NBA` (C_4) **on** is preferred only if `Traffic Accident` is not presented.
4. Commercial `NY Times` (C_5) is preferred only if both `Elections` and `Traffic Accident` are presented.
5. Commercial `Volvo` (C_6) **on** is preferred if either `New Airbag` or `Traffic Accident` are presented.
6. The commercial `Nike` (C_7) **on** is preferred only if `NBA` is **on**.

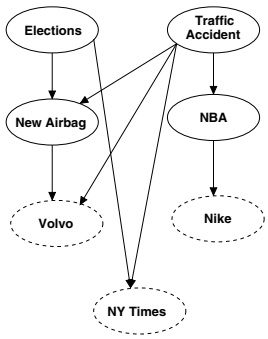

Figure 4: Example CP-network

The corresponding graph of the CP-network is presented on Figure 4 and the CP-tables are as follows:

$$
\begin{array}{lcl}
C_1 & \Rightarrow & c_1 \succ c_1' \\
C_2 & \Rightarrow & c_2' \succ c_2 \\
C_3 & \Rightarrow & (c_1' \wedge c_2) \rightarrow c_3 \succ c_3';\ \ (c_1 \vee c_2') \rightarrow c_3' \succ c_3 \\
C_4 & \Rightarrow & c_2 \rightarrow c_4' \succ c_4;\ \ c_2' \rightarrow c_4 \succ c_4' \\
C_5 & \Rightarrow & (c_1 \wedge c_2) \rightarrow c_5 \succ c_5';\ \ (c_1' \vee c_2') \rightarrow c_5' \succ c_5 \\
C_6 & \Rightarrow & (c_2 \vee c_3) \rightarrow c_6 \succ c_6';\ \ (c_2' \wedge c_3') \rightarrow c_6' \succ c_6 \\
C_7 & \Rightarrow & c_4 \rightarrow c_7 \succ c_7';\ \ c_4' \rightarrow c_7' \succ c_7
\end{array}
$$

Upon downloading this web document, the initial presentation of its content, depicted in Figure 5(a), is determined

by the Reconfiguration procedure. In the figure shaded nodes stand for presented components, and the rest for hidden components. Suppose that the viewer clicks on the link to the `Traffic Accident` article. Recall that such an interaction involves not just the presentation of the directly selected component, but the entire web-page presentation is reconsidered. The result of the reconfiguration is shown in Figure 5(b). Now, suppose that the viewer consequently clicks on the `Elections` component, trying to express that she is not interested in this topic[2]. If the event queue was specified by the designer to contain only one, most-recent event, then previous interaction with the `Traffic Accident` component is removed from the queue, and the result of the subsequent reconfiguration is presented in Figure 5(c). Otherwise, reconfiguration is performed w.r.t. the last two interactions (Figure 5(d)).

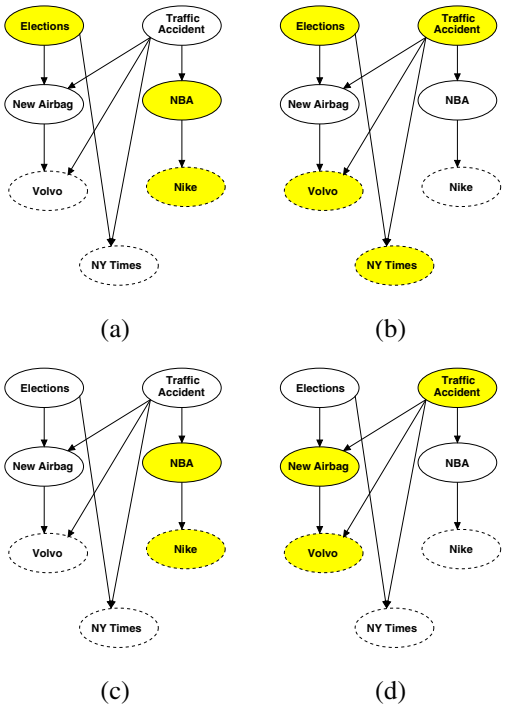

Figure 5: Reconfigurations of the content

6 Adding Layout Constraints

As mentioned above, the designer may specify expectations about layout of the presented components, as well as preferences about content presentation. The designer is allowed to specify layout constraints on web-page components, and we integrate these constraints into our preference-based optimization. We begin by discussing adopted layout constraints issues, followed by the modified reconfiguration process.

[2]Obviously, clicking on a component as expression of loss of interest is just one type of possible interaction - other kinds of interaction can be considered.

Using the CPML authoring tool, a designer can specify different constraints on the components of a web-page. Each component C_i is considered to be rendered in a rectangular area R_i, with neither its size nor its actual placement specified *a priori*. In turn, the designer is able to restrict the size and the form of each R_i, and to specify their relative placement and alignment. All these constraints are expressed as linear equalities and inequalities over the coordinates of the corners of the R_is. The required solution for them should minimize an objective function that expresses the size of the rendering area[3]. This problem is an instance of linear programming, and thus can be solved in polynomial time.

Unfortunately, the designer may specify only a few layout constraints, thus we expect our problems to be underconstrained. In conjunction with the frequently used objective function that attempts to minimize the rendering area of the web-page, this will result in a spatial overlap of some of the components, an unacceptable situation in our domain. General overlap prevention for the unconstrained rectangular areas cannot be expressed as a linear program. We express the non-overlap constraint as a set of linear inequalities over integers as follows. For each pair of rectangles R_1 and R_2 of undetermined relative position, denote their spatial extents by $\{x_1^l, x_1^r, y_1^t, y_1^b\}$ and $\{x_2^l, x_2^r, y_2^t, y_2^b\}$ respectively. Here l, r, t and b stand for left, right, top and bottom respectively. The non-overlap of R_1 and R_2 is expressed by the disjunction:

$$x_1^l - x_2^r \geq 0 \bigvee x_2^l - x_1^r \geq 0 \bigvee y_1^b - y_2^t \geq 0 \bigvee y_2^b - y_1^t \geq 0$$

We rewrite this disjunction as the following set of inequalities, by introducing four new integer variables:

$$\begin{aligned} &x_1^l - x_2^r \geq -K_1 \cdot M \qquad x_2^l - x_1^r \geq -K_2 \cdot M \\ &y_1^b - y_2^t \geq -K_3 \cdot M \qquad y_2^b - y_1^t \geq -K_4 \cdot M \\ &K_1 + K_2 + K_3 + K_4 \leq 3 \end{aligned}$$

where M is a "sufficiently large" constant, and $K_i \in \{0, 1\}$ for $1 \leq i \leq 4$. These inequalities are added to other linear constraints specified by the web-page designer, and solved by a general purpose ILP solver. The additional constraints make the general layout optimization problem NP-hard. However, we do not expect that to cause intractability in practice, due to the usually restricted number of components in a web-page. During the preliminary evaluation of the CPML prototype, no significant response time degradation was observed.

As with CP-networks creation, our authoring tool provides a simple user interface that supports icon-based layout constraint specification. The web-page designer is not aware of the underlying linear expressions, which are generated automatically. These linear expressions automatically become a part of the created web document, together with the actual web-page and the CP-network's description. Upon document download, these embedded constraints are passed to the CPML agent (together with the embedded CP-network), and become a part of the optimization process.

During user interaction, optimal layout w.r.t. the objective function is chosen for the configuration determined by the Reconfiguration procedure. The obvious question now is: what

[3]Other objective functions are also supported.

happens if the constraint system is unsatisfiable for a given, or even maximally possible, width and height of the viewer's browser? The first possibility is to use scroll bars. The second option is to try to compromise a little in the preferential optimality of the presented content. If we could determine a content presentation, which is both close to optimal w.r.t. the preferences of the designer and the viewer, and feasible w.r.t. the layout constraints, it may be a better solution.

One of the properties of the CP-network model is that given an outcome σ on can easily determine a set of outcomes $\{\sigma_1, \ldots, \sigma_m\}$ such that, for $1 \leq i \leq m$, $\sigma \succ \sigma_i$ and there is no other outcome σ' such that $\sigma \succ \sigma' \succ \sigma_i$. In other words, $\{\sigma_1, \ldots, \sigma_m\}$ is a set of *all* outcomes which are less preferred than σ but with a minimal loss of preference. The corresponding *worsening search* procedure is presented in [Boutilier *et al.*, 1999]. Informally, each outcome σ_i is reached by changing value of a single feature f in σ, such that $\sigma(f) \succ \sigma_i(f)$ given the same values of $\Pi(f)$ in σ and σ_i.

Procedure ConstrainedReconfiguration (e_i)
$\mathcal{G}$ - CP-network of the web page's components
$\mathcal{E}$ - Queue of the recent events (component, value)
$\mathcal{L}$ - Set of layout constraints

1. σ = Reconfiguration
2. If an l = ILPsolve ($\mathcal{L}$, σ) exist, then present σ according to the layout specification l. Otherwise:
   (a) Using restricted worsening search starting from σ, find the frontier of preferentially Pareto-optimal configurations $\mathcal{C}_o$ consistent with $\mathcal{E}$ that satisfy $\mathcal{L}$.
   (b) If $\mathcal{C}_o = \emptyset$ then remove the oldest event from $\mathcal{E}$ and continue from 1. Otherwise, present a $\sigma' \in \mathcal{C}_o$.

Figure 6: Constrained reconfiguration

The extended version of the Reconfiguration procedure is shown in Figure 6. Note that the number of preferentially optimal, feasible configurations may be exponential in the number of features, thus in step 2a we explicitly restrict ourself in the worsening search. Consequently, if such a compromise in the designer's preferences does not result in a feasible configuration, we can allow ourself less attention to the preferences of the viewer. This is done in step 2b by explicitly ignoring some of the oldest viewer's choices. Naturally, all these parameters of the reconfiguration process can be specified by the web-page designer.

7 Summary and Future Work

A framework for preference-based configuration of the web page content was presented. Our approach is based on qualitative decision theory, and in particular on the CP-network graphical model for preference representation. The choice of qualitative, well-founded tools leads to an application that is both intuitive and fast. The configuration process is performed according to the preferences of the web author, as well as to the current interests of the viewer. Preferences of the web-page author are encoded as a CP-network. Given this CP-network and the recent actions of the viewer, the optimal configuration of the web page content is determined. Likewise, geometric constraints on web-page layout are integrated within the optimization process, and different aspects of this integrated optimization were discussed. A prototype system was implemented, consisting of an authoring tool on the web-page author's side, and a decision making agent on the client side.

In future work, we plan to deal with a number of issues. First, preferred presentation of a web component may depend on the presentation of its neighbors, whose preferred presentation depend on its presentation, in turn. The resulting cyclic preference graphs are not well understood. The CP-network model presented in [Boutilier *et al.*, 1999] requires an acyclic graph, as the consistency of cyclic preference graphs is not guaranteed and depends on actual values in the preference tables. Thus, we plan to investigate the computational aspects of consistency verification in cyclic preference graphs.

Second, we wish to exploit our domain-specific knowledge in order to achieve an efficient ILP solver for our specific domain, in a manner similar to [Badros and Borning, 1998]. Likewise, we will consider translating some geometric constraints into actual constraints on the content presentation options of the web components.

Finally, specifying the behavior of intelligent user interfaces and autonomous multimedia objects is another potentially important application of qualitative preference models. Here as well, qualitative constraint-based optimization techniques seem useful.

References

[Badros and Borning, 1998] Greg J. Badros and Alan Borning. The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation. Technical Report 98-06-04, University of Washington, 1998.

[Borning *et al.*, 2000] Alan Borning, Richard Lin, and Kim Marriott. Constraint-Based Document Layout for the Web. *ACM Multimedia Systems Journal*, 8(3):177–189, 2000.

[Boutilier *et al.*, 1997] C. Boutilier, R. Brafman, C. Geib, and D. Poole. A Constraint-Based Approach to Preference Elicitation and Decision Making. In *AAAI Spring Symposium on Qualitative Decision Theory*, Stanford, 1997.

[Boutilier *et al.*, 1999] C. Boutilier, R. Brafman, H. Hoos, and D. Poole. Reasoning with Conditional Ceteris Paribus Preference Statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, 1999.

[D'Ambrosio and Birmingham, 1995] Joseph D'Ambrosio and William Birmingham. Preference-Directed Design. *Journal of Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, 9:219–230, 1995.

[Doyle and Thomason, 1999] J. Doyle and R.H. Thomason. Background to Qualitative Decision Theory. *AI Magazine*, 20(2):55–68, 1999.

[Lie and Bos, 1997] H.W. Lie and B. Bos. *Cascading Style Sheets*. Addison-Wesley, 1997.

[Rule and Rule, 1998] Jeff Rule and Jeffrey S. Rule. *Dynamic Html: The Html Developer's Guide*. Addison-Wesley, November 1998.

Keyword Spices: A New Method for Building Domain-Specific Web Search Engines

Satoshi OYAMA, Takashi KOKUBO* and **Toru ISHIDA**
Department of Social Informatics
Kyoto University, Kyoto 606-8501, Japan
{oyama, t-kokubo, ishida}@kuis.kyoto-u.ac.jp

Teruhiro YAMADA†
Laboratories of Image Information Science and Technology
yamateru@kuis.kyoto-u.ac.jp

Yasuhiko KITAMURA
Department of Information and Communication Engineering
Osaka City University, Osaka 558-8585, Japan
kitamura@info.eng.osaka-cu.ac.jp

Abstract

This paper presents a new method for building domain-specific web search engines. Previous methods eliminate irrelevant documents from the pages accessed using heuristics based on human knowledge about the domain in question. Accordingly, they are hard to build and can not be applied to other domains. The keyword spice method, in contrast, improves search performance by adding domain-specific keywords, called keyword spices, to the user's input query; the modified query is then forwarded to a general-purpose search engine. Keyword spices can be effectively discovered automatically from web documents allowing us to build high quality domain-specific search engines in various domains without requiring the collection of heuristic knowledge. We describe a machine learning algorithm, which is a type of decision-tree learning algorithm, that can extract keyword spices. To demonstrate the value of the proposed approach, we conduct experiments in the domain of cooking. The results confirm the excellent performance of our method in terms of both precision and recall.

1 Introduction

The expansion of the Internet and the number of its users has raised many new problems in information retrieval and artificial intelligence. Gathering information from the web is a difficult task for a novice user even if he uses a search engine. The user must have experience and skill to find the relevant pages from the large number of documents returned, which often cover a wide variety of topics. One solution is to build a domain-specific search engine[McCallum *et al.*, 1999]; an engine that returns only those web pages relevant to the topic in question.

This paper proposes a new method for building domain-specific search engines automatically that it is based on applying machine learning technologies to determine keyword occurrence in web documents.

When one of the authors used a popular Japanese search engine (Goo[1]) to find some beef recipes, he input the obvious keyword *gyuniku* (beef), but only 15 of the top 25 returned pages (60%) pertained to recipes. He hit on the idea of adding another keyword *shio* (salt) to the query, at which point all but one of the returned pages(96%) contained recipes. Surprised at this enhancement, he used the same approach for other ingredients such as pork and chicken... the same improvement in search performance was seen. This indicated the possibility of making a domain-specific search engine simply by adding a few keywords to the user's query and forwarding the modified query to a general-purpose search engine. Our *keyword spice* method is a generalization of this finding.

Several research papers have described domain-specific web search services. A straightforward approach to building a domain-specific web search engine is to make indices to domain documents by running web-crawling spiders that collect only relevant pages. Cora[2][McCallum *et al.*, 1999] is a domain-specific search engine for computer science research papers. Its web-crawling spiders effectively explore the web by using reinforcement learning techniques. SPIRAL[Cohen, 1998] or WebKB[Craven *et al.*, 1998] also use crawlers. These systems offer sophisticated search functions because they establish their own local databases and can apply various machine learning or knowledge representation techniques to the data. Unfortunately, domains such as personal home-

*Presently with NTT Docomo, Inc.
†Presently with SANYO Electric Co.,Ltd.

[1] http://www.goo.ne.jp
[2] http://cora.whizbang.com/

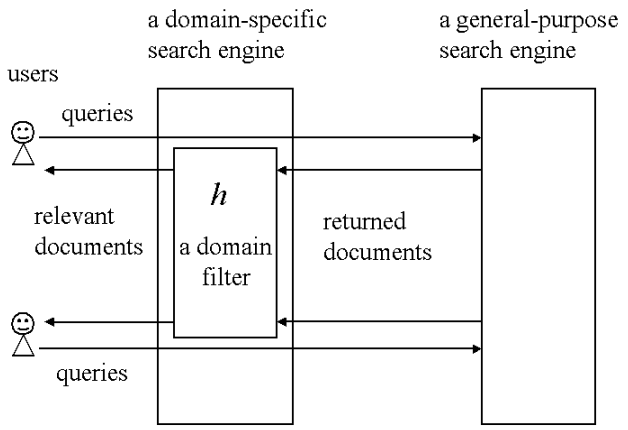

Figure 1: Filtering model for building domain-specific web search engines

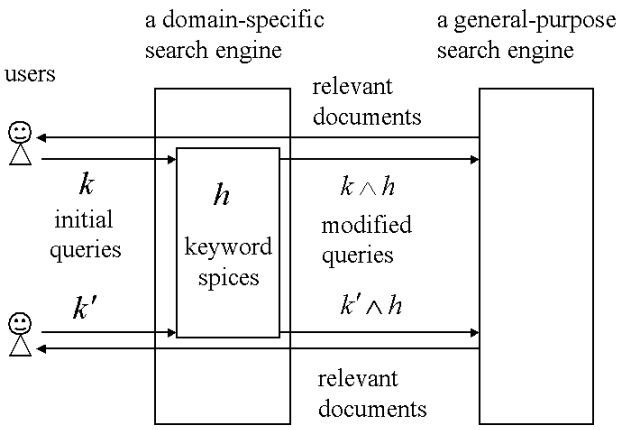

Figure 2: The keyword spice model of building domain-specific web search engines

pages or cooking pages, which are dispersed across many web sites, are not well handled by spiders since the time and network bandwidth consumed are excessive. Accordingly, such types of systems are suitable only for those domains that have few web sites.

Reusing the large indices of general-purpose search engines to build domain-specific ones is a clever idea[Etzioni, 1996]. For example, Ahoy![3] [Shakes *et al.*, 1997] is a search engine specialized for finding personal homepages. It forwards the user's query to general-purpose search engines and sifts out irrelevant documents from the returned ones to increase precision by domain-specific filters. We call this the *filtering model* for building domain-specific search engines (Figure 1). Ahoy! has a learning mechanism to assess the patterns of relevant URLs from previous successful searches, but overall accuracy basically depends on human knowledge.

One solution to the above problem is to make domain filters automatically from sample documents. Automatic text filtering, which classifies documents into relevant and non-relevant ones, has been a major research topic in both information retrieval[Baeza-Yates and Ribeiro-Neto, 1999] and machine learning[Mitchell, 1997].

We can use various machine learning algorithms to find such filters if the training examples, which consist of documents randomly sampled from the web together with their manual classification, are available. Unfortunately, making such training examples is the real barrier because the web is very large, and randomly sampling the web will provide only a small likelihood of encountering the domain in question. In fact, most studies on text classification have been applied to e-mail, net news, or web documents at limited sites where the ratio of positive examples is rather high. Thus previous methods of text classification cannot be directly applied to the problem of building domain-specific web search engines.

The keyword-spice method considers only those web pages

[3]http://ahoy.cs.washington.edu:6060/

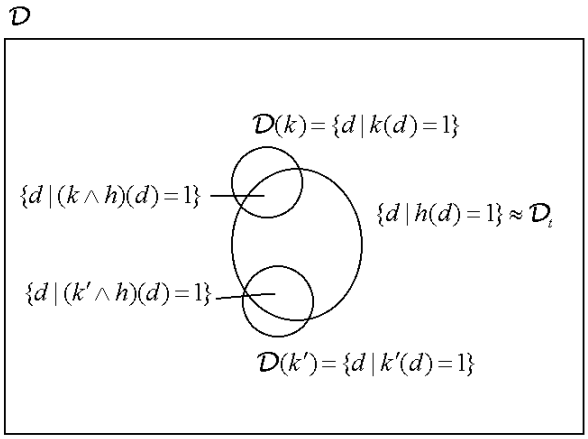

Figure 3: Sampling with input keywords to increase the ratio of positive examples

that contain the user's input query keyword, not all web pages. This eliminates the problem of finding positive examples and enables us to make domain-specific search engines at low cost.

The remainder of this paper is organized as follows: Section 2 presents the idea of building domain-specific search engines using keyword spices. Section 3 describes a machine learning algorithm for discovering keyword spices. Section 4 evaluates our method and our conclusions are given in Section 5.

2 The keyword spice model of building domain-specific web search engines

Here we introduce some notations to define the machine learning problem. We let $\mathcal{D}$ denote the set of all web documents;

$\mathcal{D}_t$ denotes the set of documents relevant to a certain domain. The target function (an ideal domain filter) that correctly classifies any document $d \in \mathcal{D}$ is given as

$$f(d) = \begin{cases} 1 & \text{if } d \in \mathcal{D}_t \\ 0 & \text{otherwise} \end{cases}$$

We let $\mathcal{K}$ be the set of all keywords in the domain and let $\mathcal{H}$ be the hypothesis space composed of all Boolean expressions where any keyword $k \in \mathcal{K}$ is regarded as a Boolean variable. We adopt the Boolean hypothesis space because most commercial search engines can accept queries written in Boolean expressions.

A Boolean expression of keywords can be regarded as a function from $\mathcal{D}$ to $\{0, 1\}$ when we assign 1(true) to a keyword (Boolean variable) if the keyword is contained in the document and 0(false) otherwise. In the filtering model, the problem of building a domain filter is equal to finding hypothesis h that minimizes the error rate

$$\frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \delta(h(d), f(d))$$

Note: quantity $\delta(h(d), f(d))$ is 1 if $h(d) \neq f(d)$, 0 otherwise.

The keyword spice model does not filter documents returned by a general-purpose search engine. Instead, it extends the user's input query with a domain-specific Boolean expression (keyword spice), which better classifies the domain documents, and passes the extended query to a general-purpose search engine (Figure 2). This model is just the reverse of the filtering model.

Our method is based on the idea that when we build a domain-specific web search engine, we need consider only those web pages that contain the user's input query keywords; not all web pages.

As described in Figure 3, the scope of sampling is reduced from set $\mathcal{D}$, all web documents, to $\mathcal{D}(k)$, the set of web pages that contain input keyword k; this increases the ratio of positive examples $\{d|(k \wedge h)(d) = 1\}$. This idea makes it easier to create training sets and it becomes possible to build a domain filter, which is not possible with random sampling.

By using domain filter h, we modify the user's input query k to $k \wedge h$, so the returned documents contain k and are included in the domain. In short, h is the *keyword spice* for the domain.

3 Algorithm for extracting Keyword Spices

3.1 Identifying Keyword Spices

It is rather easy to find good keyword spices for any input keyword k (for example "beef"). The problem is to find that the keyword spices that provide enough generalization to handle all future user keywords.

We let $p(k)$ denote the probability of that a user will input keyword k to a domain-specific search engine. Then

$$\sum_{k \in \mathcal{K}} p(k) \sum_{d \in \mathcal{D}(k)} \frac{1}{|\mathcal{D}(k)|} \delta((k \wedge h)(d), f(d))$$

is the expectation of the error rate when users try to locate domain documents using this system.

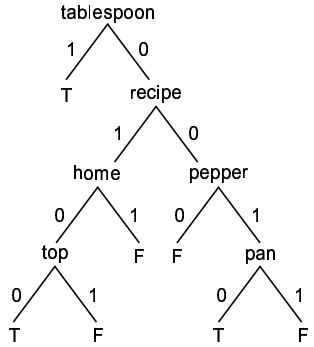

Figure 4: An example of decision tree that classifies documents

tablespoon
$\vee$ $(\neg$tablespoon $\wedge$ recipe $\wedge \neg$home $\wedge \neg$top$)$
$\vee$ $(\neg$tablespoon $\wedge \neg$recipe $\wedge$ pepper $\wedge \neg$pan$)$

Figure 5: An example of Boolean expression converted from the tree in Figure 4

The Boolean expression that minimizes the above expectation value is the most effective keyword spice. It would be best to make training examples using $p(k)$ but we do not know $p(k)$ beforehand. Obviously, we have to start with some reasonable value of $p(k)$, and modify the value as statistics on input keywords are collected.

In this paper, we choose several input keyword candidates in the cooking domain. We assume that all candidates have the same probability of occurrence and collect the same number of documents for each keyword as described in Section 4. We then split the examples into two disjoint subsets, the training set $\mathcal{D}_{training}$ (used for identifying initial keyword spices), and the validation set $\mathcal{D}_{validation}$ to simplify the keyword spices described in Section 3.2.

We apply a decision tree learning algorithm to discover keyword spices because it is easy to convert a tree into Boolean expressions, which are accepted by most commercial search engines. In this decision tree learning step, each keyword is used as an attribute whose value is 1(when the document contains this keyword) or 0(otherwise). Figure 4 shows an example of simple decision tree that classifies documents.

The node indicates attribute, the value of branch indicates the value of the attribute, and the leaf indicates the class. In order to classify a document, we start at the root of the tree, examine whether the document contains the attribute (keyword) or not and take the corresponding branch. The process continues until it reaches a leaf and the document is asserted to belong to the class corresponding to the value of the leaf. This tree classifies web documents into T (domain documents) and F (the others), and the web document, for example, that does not include "tablespoon", does "recipe", does

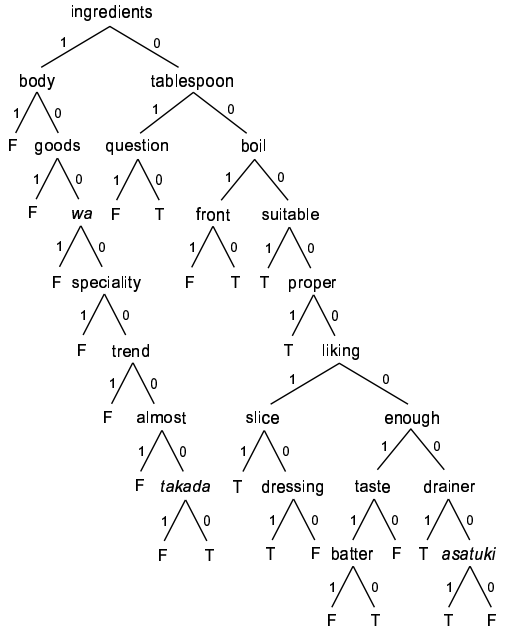

Figure 6: A decision tree induced from web documents

not "home", and does not "top" belongs to class T.

We make the initial decision tree using an information gain measure[Quinlan, 1986] for greedy search without using any pruning technique. In our real case, the number of attributes (keywords) is large enough (several thousands) to make a tree that can correctly classify all examples in the training set $\mathcal{D}_{training}$. Then for each path in the induced tree that ends in a positive result, we make a Boolean expression that conjoins all keywords (a keyword is treated as a positive literal when its value is 1 and a negative literal otherwise) on the path. Our aim is to make a Boolean expression query that specifies the domain documents and that can be entered into search engines; accordingly, we consider only positive paths.

We make a Boolean expression h by making a disjunction of all these conjunctions (i.e. we make a disjunctive normal form of a Boolean expression). This is the initial form of keyword spices. Figure 5 provides an example of a Boolean expression converted from the tree in Figure 4.

3.2 Simplifying Keyword Spices

Figure 6 shows a decision tree induced from collected web document in the experiments described in the next section[4]. Decision trees usually grow very large which triggers the over-fitting problem. Furthermore, too-complex queries cannot be accepted by commercial search engines and so we have to simplify the induced Boolean expression. We developed a two-stage simplification algorithm (described below) that is like rule post-pruning[Quinlan, 1993].

1. For each conjunction c in h we remove keywords (Boolean literals) from c to simplify it.
2. We remove conjunctions from disjunctive normal form h to simplify it.

In information retrieval research, we normally use precision and recall for query evaluation. Precision is the ratio of number of relevant documents to the number of returned documents and recall is the ratio of the number of relevant documents returned to the number of relevant documents in existence.

In this section, precision P and recall R are defined over validation set $\mathcal{D}_{validation}$ as follows:

$$P = \frac{|\mathcal{D}_{domain} \cap \mathcal{D}_{Boolean}|}{|\mathcal{D}_{Boolean}|}$$

$$R = \frac{|\mathcal{D}_{domain} \cap \mathcal{D}_{Boolean}|}{|\mathcal{D}_{domain}|}$$

where $\mathcal{D}_{domain}$ is the set of relevant documents classified by humans and $\mathcal{D}_{Boolean}$ is the set of documents that the Boolean expression identifies as being relevant in the validation set.

In our case, we use the harmonic mean of precision P and recall R[Shaw Jr. *et al.*, 1997]

$$F = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

as the criterion for removal. The harmonic mean weights low values more heavily than high values. High values of F occur only when both precision P and recall R are high. So if we simplify keyword spices in the way that results in high value of F, we can obtain the keyword spices that are well-balanced in terms of precision and recall.

In the first stage of simplification we treat each conjunction as if it is an independent Boolean expression. We calculate the conjunction's harmonic mean of recall and precision over the validation set. For each conjunction, we remove the keyword (Boolean literal) if it results in the maximum improvement in this harmonic mean and repeat this process until there is no keyword that can be removed without decreasing the harmonic mean.

When we remove a keyword from conjunction recall either increases or remains unchanged. Before the simplification, each conjunction usually yields high precision and low recall. Accordingly, we can remove the keyword that results in improvement in recall in exchange for some decrease in precision, because the harmonic mean weights lower recall values more heavily. The removal of the keywords from the conjunction by the harmonic mean may appear to cause some problems. If the initial conjunction contains only a few relevant documents, the algorithm makes conjunctions that contain very large numbers of irrelevant documents. However, we can remove the conjunction from the keyword spices by the algorithm for simplifying a disjunction as is described below.

In the second stage of simplification, we try to remove conjunctions from the disjunctive normal form h to simplify the keyword spices. We remove the conjunctions so as to maximize the increase in harmonic mean F. We repeat this process until there is no conjunction that can be removed without decreasing the harmonic mean F.

[4]The original keywords are Japanese.

0. Generate input keywords according to some estimate of distribution $p(k)$ and collect web pages that contain keyword k and classify them into positive and negative examples by hand.
1. Split the examples into two disjoint subsets, the training set, $\mathcal{D}_{training}$ (for generating the initial decision tree) and the validation set, $\mathcal{D}_{validation}$ (for simplifying the tree).
2. Make the initial decision tree from $\mathcal{D}_{training}$ using an information gain measure without any pruning technique.
3. Convert the tree so learned into a set of positive conjunctions by creating one conjunction for each path from the root node to each leaf node: this classifies positive examples.
4. Make a disjunctive normal form of Boolean expression h by making a disjunction of all positive conjunctions.
5. **For each** conjunction c **in** h **do**

   Repeat

   - Remove the keyword (Boolean literal) from the conjunction c that results in the maximum increase in the harmonic mean

   $$F_c = \frac{2}{\frac{1}{R_c} + \frac{1}{P_c}}$$

   of precision measure P_c and recall measure R_c of c over the validation set.

   Until there is no keyword that can be removed without decreasing F_c.

   End
6. **Repeat**

   - Remove the conjunctive component from the disjunctive normal form h that results in the maximum increase in the harmonic mean

   $$F_h = \frac{2}{\frac{1}{R_h} + \frac{1}{P_h}}$$

   of precision measure P_h and recall measure R_h of h over the validation set.

   Until there is no conjunction that can be removed without decreasing F_h.

   Return h

Figure 7: The keyword spice extraction algorithm

After the first stage of simplification, each conjunction is generalized and changed to cover many examples. As a result, the recall of h becomes rather high, but some conjunctions may cover many irrelevant documents. We can remove the conjunctions that cause the large improvement in the precision with a slight reduction in recall. Those components that cover many irrelevant documents are removed in this stage, because the other conjunctions cover most of the relevant documents and the removal of the defective conjunctions does not cause a large reduction in recall. This yields simple keyword spices composed of a few conjunctions.

After the above simplification processes h is returned as the keyword spices for this domain. Our algorithm for extracting keyword spices is summarized in Figure 7.

Table 1: Collected web documents in the cooking domain

Keyword	relevant	irrelevant	total
beef	47	153	200
chicken	88	112	200
paprika	79	121	200
potato	49	151	200
pumpkin	42	158	200
radish	64	136	200
salmon	15	185	200
tofu	45	155	200
tomato	33	167	200
whitefish	103	97	200
Total	565	1435	2000

Table 2: Pruning results

		Trials				
		1	2	3	4	5
Initial	conjunctions	10	15	13	15	10
	keywords	65	89	76	87	62
Step 5	conjunctions	10	15	13	15	10
	keywords	17	32	26	34	19
Step 6	onjunctions	2	2	2	2	2
	keywords	4	3	4	4	4

4 Evaluation in the Cooking Domain

4.1 Experimental Settings

As described in the previous section, we gathered two thousand sample pages of the cooking domain that contained human-entered keywords in Japanese: *gyuniku* (beef), *toriniku* (chicken), *piman* (paprika), *jagaimo* (potato), *kabocha* (pumpkin), *daikon* (radish), *sake* (salmon), *tofu* (tofu), *tomato* (tomato), and *shiromizakana* (whitefish). We used a Japanese general-purpose search engine Goo to find and download web pages containing the above input keywords. We collected two hundred sample pages for each initial keyword. We examined the pages collected and classified them as either relevant or irrelevant by hand (Table 1).

In splitting the collected documents into the training set and validation set, we paid no attention to which keywords were input. Thus each set was randomly composed of documents containing the input keywords. We performed 5 trials in which the sample pages were split randomly in this fashion.

Table 2 shows the pruning results after each step. In the early steps, induced trees are very large and after translating trees to conjunctions, we have more than 10 conjunctions; the number of keywords in these conjunctions exceeded 62. This number is too large to permit entry into commercial search engines. After step 5 the number of keywords was reduced to one third. Step 6 removed redundant conjunctions and keyword number was reduced again to 3 to 4. This number of keywords can be accepted by commercial search engines.

Different trials yielded different keyword spices. Figure 8

$$(\text{ingredients} \wedge \neg\text{speciality} \wedge \neg\text{goods})$$
$$\vee \quad \text{tablespoon}$$

Figure 8: Extracted keyword spices

Table 3: Average precision of the queries over the index of a general-purpose search engine

Query	The input query	The query with keyword spices
pork	0.271	0.995
spinach	0.205	0.979
shrimp	0.063	0.986

Table 4: Estimated recall of the queries with keyword spices over the index of a general-purpose search engine

Query	$Reldoc_{index}$	$Reldoc_{spice}$	Estimated recall
pork	10728	10084	0.940
spinach	4744	4126	0.870
shrimp	5868	5728	0.976

shows, as an example, the keyword spices discovered in the first trial. We used these keyword spices in subsequent experiments.

To conduct realistic tests with external commercial search engines, we choose the keywords of *butaniku* (pork), *horenso* (spinach) and *ebi* (shrimp) which were not used to generate the keyword spices.

4.2 Precision

Figure 9 compares the precision values for the queries containing only keywords and the queries with keyword spices for the three input keywords. We checked up to the top 1000 pages as ranked by the search engine Goo. In general, as the number of pages viewed increases, the precision with query-only input decreases, while the precision of queries with keyword spices stays high. Table 3 lists the average precision of the top 1000 returned results. Precision is higher than 97% for all queries.

4.3 Estimated Recall

It is easy to achieve high precision if we do not address recall, but keeping both high is rather difficult. The recall of a query is much harder to calculate than the precision because $\mathcal{D}_t$, the set of all relevant documents in the web, is unknown. We estimated $\mathcal{D}_t$ from the results returned from a general-purpose search engine. Most search engines show the total number of documents that matched the query. We can calculate the estimated number of relevant documents in the search engine's index ($Reldoc_{index}$) by using the average precision of the query for the top 1000 returned documents.

$Reldoc_{index} \simeq$

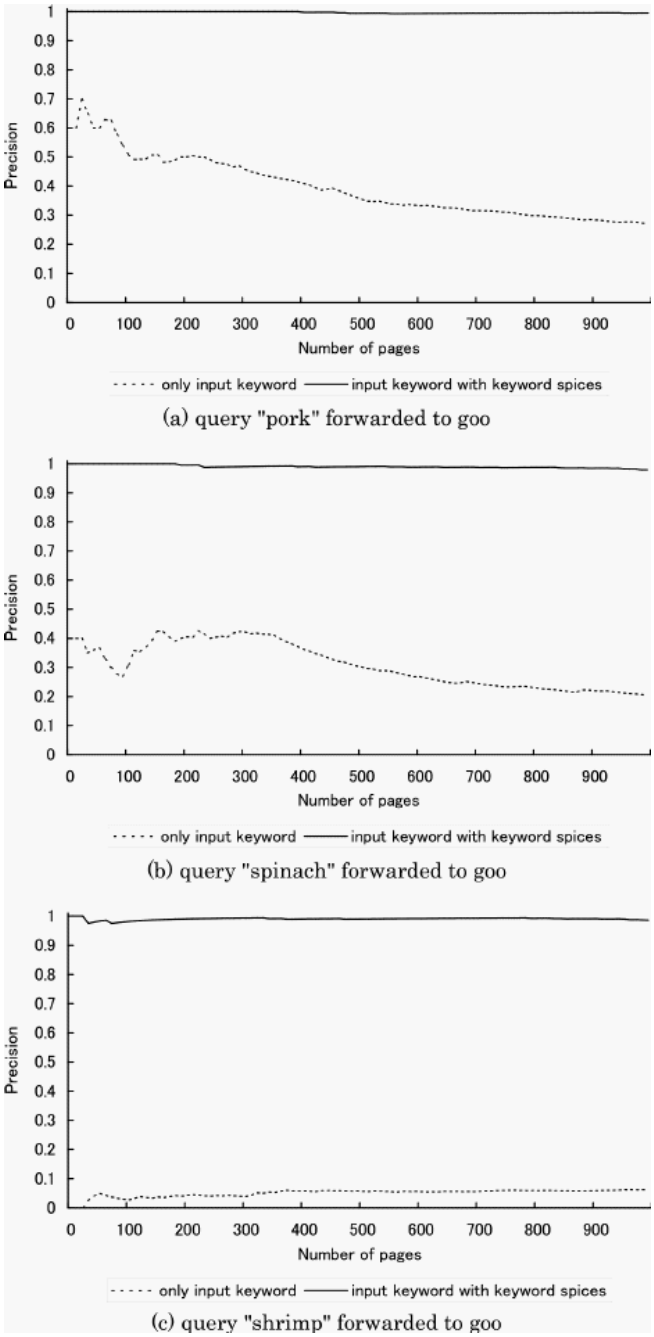

Figure 9: Precision of queries forwarded to a general-purpose search engine

(The number of document found with the input query)
$\times$(Average precision of the input query)

The number of relevant documents found with the spice-extended query can be calculated in the same way.

$Reldoc_{spice} \simeq$
(The number of document found with the query with keyword spices)

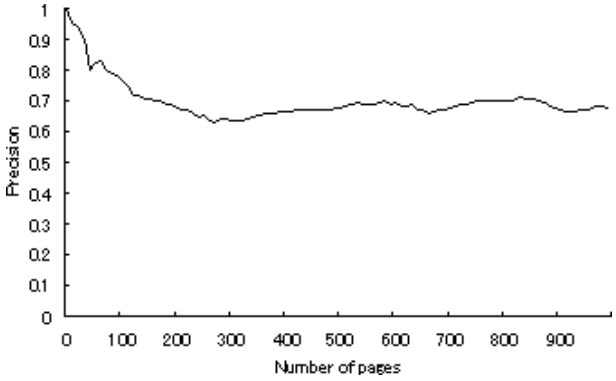

Figure 10: Precision of the query "pork AND salt" forwarded to Goo

$\times$(Average precision of the query with keyword spices)

It is reasonable to use $Reldoc_{index}$ because we have no consistent way of finding web pages that are not linked to any general-purpose search engine. We estimate the recall of a spice-extended query as follows

$$R \simeq \frac{Reldoc_{spice}}{Reldoc_{index}}$$

Table 4 shows the estimated recall values of different spice-extended queries over the index of Goo. The high value of recall (higher than 87%) indicates that our method filters out only non-relevant documents and does not drop any useful information in the search process.

To compare these results with the example in the Introduction, Figure 10 shows the results of submitting the query "pork AND salt" to Goo. The average precision and estimated recall for the top 1000 returned documents are 0.674 and 0.871, respectively. This shows that our systematic method yields a great improvement in search performance.

5 Conclusion

We have proposed a novel method for domain specific web searches that is based on the idea of keyword spices; Boolean expressions that are added to the user's input query to improve the search performance of commercial search engines. This method allows us to build domain-specific search engines without any domain heuristics. We described a practical learning algorithm to extract powerful but comprehensive keyword spices. This algorithm turns complicated initial decision trees to small Boolean expressions that can be accepted by search engines. Our experiments with an external general-purpose search engine yielded good results. For two different keywords in the field of cooking, precision was higher than 97%. High estimated recall(higher than 87%) over the search engine's index was also confirmed.

We used the domain of cooking as an example, and we are now developing search services for other domains such as restaurant pages and personal homepages.

In this paper, we used input keywords selected by humans to make training examples. To be more comprehensive, we need some criteria with which input keywords can be selected. As discussed in Section 3, it is sufficient to make examples based on the distribution of user's input query $p(k)$. We are planning to open our recipe search system to the public through the web and we will obtain the value of $p(k)$ afterwards. In future work we will study how the input keywords used to form the training examples affect the performance of the system.

Acknowledgments

This research was partially supported by Laboratories of Image Information Science and Technology and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research(A), 11358004, 1999.

References

[Baeza-Yates and Ribeiro-Neto, 1999] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[Cohen, 1998] William W. Cohen. A web-based information system that reasons with structured collections of text. In *Agents'98*, pages 116–123, 1998.

[Craven *et al.*, 1998] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to extract symbolic knowledge from the world wide web. In *AAAI-98*, pages 509–516, 1998.

[Etzioni, 1996] Oren Etzioni. Moving up the information food chain: Deploying softbots on the world wide web. In *AAAI-96*, pages 1322–1326, 1996.

[McCallum *et al.*, 1999] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. A machine learning approach to building domain-specific search engines. In *IJCAI-99*, pages 662–667, 1999.

[Mitchell, 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[Quinlan, 1986] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Quinlan, 1993] J. Ross Quinlan. *C4.5:Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Shakes *et al.*, 1997] Jonathan Shakes, Marc Langheinrich, and Oren Etzioni. Dynamic reference sifting: a case study in the homepage domain. In *Proceedings of the 6th International World Wide Web Conference(WWW6)*, pages 189–200, 1997.

[Shaw Jr. *et al.*, 1997] W. M. Shaw Jr., Robert Burgin, and Patrick Howell. Performance standards and evaluations in ir test collections: Cluster-based retrieval models. *Information Processing & Management*, 33(1):1–14, 1997.