INTERACTIVE SPOKEN CONTENT RETRIEVAL BY EXTENDED QUERY MODEL AND CONTINUOUS STATE SPACE MARKOV DECISION PROCESS

Tsung-Hsien Wen, Hung-yi Lee, Pei-hao Su,and Lin-Shan Lee

National Taiwan University, Taipei, Taiwan

r00921033@ntu.edu.tw, lslee@gate.sinica.edu.tw

ABSTRACT

Interactive retrieval is important for spoken content because the retrieved spoken items are not only difficult to be shown on the screen but also scanned and selected by the user, in addition to the speech recognition uncertainty. The user cannot playback and go through all the retrieved items to find out what he is looking for. Markov Decision Process (MDP) was used in a previous work to help the system take different actions to interact with the user based on an estimated retrieval performance, but the MDP state was represented by the less precise quantized retrieval performance metric. In this paper, we consider the retrieval performance metric as a continuous state variable in MDP and optimize the MDP by fitted value iteration (FVI). We also use query expansion with the language modeling retrieval framework to produce the next set of retrieval results. Improved performance was found in the preliminary experiments.

Index Terms— Markov Decision Process, MDP, Fitted Value Iteration, Interactive Retrieval, Language Model Retrieval.

1. INTRODUCTION

Interactive Information Retrieval (IIR) [1, 2] uses interactive user interface to help the user transmit better information to the machine regarding what he is looking for, and help the machine better clarify the user's needs. "Dialogue Navigator for Kyoto City" used a Bayes risk-based dialogue manager to offer an efficient interaction interface for the user to find information about Kyoto city [3, 4]. "MIT MovieBrowser" adopted Conditional Random Field (CRF) for natural language inquiry understanding and managed to build an organic spoken language movie search system [5, 6]. Such systems usually have the content to be retrieved in text form stored in a structured or semi-structured database and make more efforts on transforming user's natural language queries into semantic slots for subsequent database search.

Interactive retrieval is specially important for spoken content, not only because recognition errors produce high degree of uncertainty for the content and the subword-based technologies usually lead to relatively high recall rates, but because the spoken content is difficult to be shown on the screen and difficult to be scanned and selected by the user. The user cannot simply playback and go through all the retrieved items and then choose the ones he is looking for. Markov decision process (MDP) was used to help the user select key terms in the IIR process of a broadcast news browser [7, 8]. But when the retrieved results are poor, the user still needs to take long time to go through the long list of irrelevant key terms before finding the results are unsatisfactory. A different approach was then proposed recently, in which the machine can take different types of actions depending on the estimated quality of the present retrieval results also based on MDP [9]. However, in this approach the MDP state representation by quantized retrieval quality metric was not precise enough, and the Vector Space Model retrieval framework used was also less effective and reliable.

In this paper, we propose to use continuous state space representation on MDP modeling for Interactive Spoken Content Retrieval realized with query expansion for the language modeling retrieval framework. The continuous state space MDP trained with fitted value iteration (FVI) optimizes the policy to select the best system action at each iteration based on a set of pre-defined rewards, while the expanded query model produces the next set of retrieval results. Improved performance was found in the preliminary experiments.

2. PROPOSED APPROACH

The proposed framework is depicted in Fig. 1. The language modeling retrieval module is on the top. The MDP with continuous state space for dialogue manager is in the middle, while MDP training with fitted value iteration is at the bottom. More details are given below.

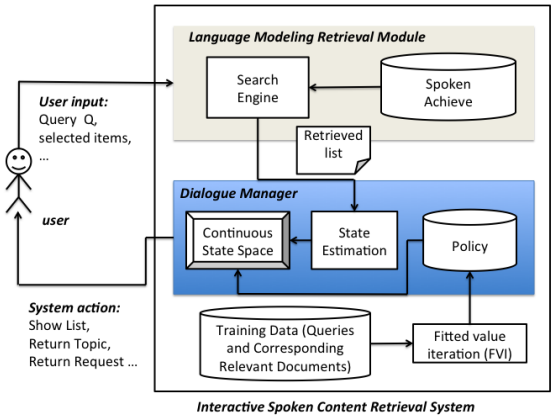

Fig. 1: *Block diagram of the proposed system.*

2.1. Language Modeling Retrieval and Query Expansion

2.1.1. Language Modeling Retrieval Framework

The basic idea of language modeling retrieval framework is that the query Q and the document d can be represented as a query language model θ_Q and a document language model θ_d respectively, and the relevance score function $S(Q, d)$ used for ranking documents d with respect to query Q is simply based on the KL divergence between θ_Q and θ_d [10, 11]:

$$S(Q, d) = -KL(\theta_Q||\theta_d) \tag{1}$$

But since there may be negative feedback from the user, a negative model θ_N can be added to collect negative information, and the (1) can be rewritten as [12, 13]

$$S(Q, d) = -KL(\theta_Q||\theta_d) + \beta \cdot KL(\theta_N||\theta_d), \tag{2}$$

where β is a weight parameter. The document model θ_d is usually estimated by smoothing the empirical document model $\hat{\theta_d}$ with the background model θ_B:

$$P(w|\theta_d) = \alpha_d P(w|\hat{\theta_d}) + (1-\alpha_d)P(w|\theta_B), \quad (3)$$

where $P(w|\hat{\theta_d}) = N(w,d)/\|d\|$, $N(w,d)$ is the count of word w in d, $\|d\|$ the length of d, $P(w|\theta_B) = \sum_{d\in C} N(w,d)/\sum_{d\in C}\|d\|$, C the document archive, and $\alpha_d = \frac{\|d\|}{\|d\|+L}$ a document dependent interpolation weight with a parameter L. So the problem is reduced to estimate the models θ_Q and θ_N. More details about estimating θ_d for spoken documents models are left out here [11, 14].

2.1.2. Query-regularized Mixture Model for Query Expansion

Through user feedback, the machine accumulates relevant and irrelevant information during the interations. Such information is used to estimate a new query model θ'_Q and negative model θ_N respectively [15]. We adopt the query-regularized mixture model [14, 16] previously proposed for pseudo-relevance feedback to estimate the new query model θ'_Q for the interactive retrieval task here. The model assumes that the words in the set of relevant documents R obtained from user feedback are either query-related words or general words, with a document-dependent ratio between the two. These document-dependent ratios and which words are query-related are actually unknown, but can be estimated from the relevant document set R. Suppose d is a document in R, with the assumption that the words in d are either query-related or general, the interpolated language model θ_d in (3) should be close to an estimated model θ'_d which is the interpolation of the new query model θ'_Q to be estimated (for query-related words) and the background language model θ_B (for general words) with a document-dependent weight γ_d.

$$P(w|\theta'_d) = \gamma_d P(w|\theta'_Q) + (1-\gamma_d)P(w|\theta_B), \quad (4)$$

where γ_d is the document-dependent interpolation weight for document d, which is to be estimated too. As a consequence, the new query model θ'_Q minimizing (5) below is taken as the query model to be used in (2) above:

$$F(\theta'_Q, \{\alpha_d\}_{d\in R}) = \sum_{d\in R} KL(\theta_d|\theta'_d) + \mu KL(\theta_t|\theta'_Q), \quad (5)$$

where the first term on the right hand side implies the sum of the KL divergence between each document model θ_d and the interpolated model θ'_d in (4) for all documents d in R should be minimized. However, the new query model θ'_Q thus obtained may be just for the common content of the documents in R, not necessarily query-related. This is why the second term on the right hand side of (5) is added, in which θ'_Q is "regularized" by a prior key term model θ_t estimated from a relevant key term set R'. Initially R' contains just all the terms in the query, and $P(w|\theta_t) = N(w,R')/\|R'\|$. R' may grow when more key terms are added through user feedback. Because the value of (5) will be larger for model θ'_Q far from θ_t, so the model θ'_Q estimated via minimizing (5) would not be totally drifted away by the documents in R because a new query model θ'_Q similar to the prior key term model θ_t is preferred. μ in (5) is a parameter controlling the influence of the second term. Although these formulations follow the previous work [14, 16] for pseudo relevance feedback, here the purpose is to organize the relevant documents d in R and relevant key terms in R' obtained through user feedback for query expansion. During the interactive retrieval process, both the relevant document set R and the relevant key term set R' may grow gradually when more positive information becomes available, as will be discussed in more detail in Sec 2.2.2. Also, exactly the same procedure is used for estimating the negative model θ_N to be used in (2), in which we also maintain an irrelevant document set I and an irrelevant key term set I', both of them are growing, and obtain θ_N by minimizing an expression very similar to (5).

2.2. MDP Framework

A Markov Decision Process (MDP) [17, 18, 19] is defined as a tuple $\{S, A, T, R, \gamma\}$, where S is the set of states, A the set of actions, R the reward function, T the transition probabilities and γ the discount factor. A mapping from a state $s \in S$ to an action $a \in A$, or action selection at each state, is a policy π. Given a policy π, the value of the Q-function ($Q^\pi : S \times A \to \mathbb{R}$) is defined as the expected discount sum of all rewards that can be received by an agent over an infinite state transition path starting from state s taking action $\pi(s)$: $Q^\pi(s,a) = E[\sum_{k=0}^{\infty}\gamma^k r_k | s_k = s, a_k = a, k = 0, 1, 2...]$, where r_k is the reward received from the action a_k taken at state s_k, where k is the sequence index for states and actions. The optimal policy maximizes the value of each state-action pair: $\pi^*(s) = \arg\max_{a\in A} Q^*(s,a)$, where

$$Q^*(s,a) = E_{s'|s,a}[R(s,a,s') + \gamma \max_{b\in A} Q^*(s',b)] \quad (6)$$

and $R(s,a,s')$ is the reward for taking action a at state s and transit to state s'. (6) is known as the Bellman optimality equation [20]. So finding an optimal policy is equivalent to find the optimal Q-function, which can be solved iteratively known as *Value Iteration*.

2.2.1. States

For the task here, the system is to select actions based on its confidence about the quality of the current retrieval result. So a selected evaluation metric for retrieval (e.g. Mean average precision (MAP)) is taken as the state variable. We take this selected evaluation metric as a continuous variable directly without quantization [9] and explore the continuous state space approach for MDP. In reality, the retrieval result quality can be judged only by the user, so the system never knows the true state it is in. The way to estimate the state variable will be discussed in Sec 2.4. Furthermore, we maintain a policy for each state and action time k because the action needed is different for earlier and later steps of interaction.

2.2.2. Actions

Five actions defined in this work are presented below. At each state and action time k the retrieval module offers a list of relevant documents ranked by $S^k(Q,d)$. The system action can be simply (a) Show list : The system shows the retrieved results ranked by $S^k(Q,d)$ to the user and ends the retrieval session. If this action is not selected, the system interacts further with the user with one of the following actions in order to collect relevant/irrelevant information and updates the relevance score of each document to $S^{k+1}(Q,d)$ by reestimating θ'_Q and θ_N mentioned above.

(b) Return Document: The system returns the current retrieved list ranked in decreasingly $S^k(Q,d)$, and asks the user to view the document list from the top and select a relevant document as the feedback. The returned document is then added to the relevant document set R as in Sec 2.1.2 for query expansion.

(c) Return Key Term: the system asks the user whether a term t^* is relevant or not,

$$t^* = \arg\max_t \sum_{q\in\theta'^{(k)}_Q} P(q|\theta'^{(k)}_Q) \times Jaccard(q,t) \quad (7)$$

where $\theta'^{(k)}_Q$ is the new query model as defined in Sec 2.1.2 at time k and the summation is over the top M terms q selected based on this

model, and $Jaccard(q,t) = \|D_q \cap D_t\|/\|D_q \cup D_t\|$ is the Jaccard coefficient of co-occurrence between two terms q and t [21]. D_q and D_t are the set of all documents in the archive containing q and t respectively. So we use the top M terms q in the model as references to identify t^*. t^* is then added to the relevant key term set R' or the irrelevant key term set I', depending on the user's reply.

(d) Return Request: the user is asked to provide an additional query term $\hat{t}$, which is then added to the relevant key term set R'.

(e) Return Topic: The system returns a list of topics inferred via some latent topic models [22, 23, 24, 25]. Each latent topic is shown to the user by the top-N words with the highest probability given the topic. The user then selects a relevant topic. The complete word distribution given the selected topic is then treated as a document with length equal to the average document length in the archive to be added to the relevant document set R.

Note that all these feedback actions are for collecting extra information from the user and obtaining the expanded query θ'_Q accordingly via the sets R' (for *(c) Return Key Term* and *(d) Return Request*) and R (for *(b) Return Document* and *(e) Return Topic*).

2.2.3. Reward and Return

A retrieval session starts from state s_0 with actions selected by a policy π, $\Gamma_\pi(s_0) = \{s_0, s_1, ...s_K\}$, where s_K is the final state. For each action $a_k = \pi(s_k)$ taken at state s_k, the system obtains a reward $r_k = C(a_k)$ which is defined for all actions a. In the proposed approach here, negative rewards or costs are assigned to actions *(b) Return Document*, *(c) Return Key Term*, *(d) Return Request* and *(e) Return Topic* since they all involve efforts from the user. The last action of a session $\pi(s_K)$ is always *(e) Show List*, for which a positive reward is received, which is the improvement in retrieval evaluation metric via the whole interaction process and can be written as $r_K = \tau[E(s_K) - E(s_0)]$, where $E(s)$ is the retrieval metric at state s. τ is the trade-off parameter between user effort and retrieval quality, with a smaller τ indicating the system prefers to minimize the user effort than maximize the retrieval quality. The Return of the system for the entire retrieval session $\Gamma_\pi(s_0)$ is then $G = \sum_{k=0}^{K} r_k$.

2.3. Fitted Value Iteration

We use linear parameterization [26, 27, 28] for representing a continuous Q-function. Given a set of basis functions $\{\phi_m(s,a)\}_{1\le m\le M}$, we represent the Q-function as a linear combination of those basis functions by parameters $\underline{\rho} \in \mathbb{R}^M$, $Q_{\underline{\rho}_i}(s,a) = \sum_m \rho_m \phi_m(s,a) = \underline{\rho}^T \underline{\phi}(s,a)$ where $\underline{\rho}$, $\underline{\phi}(s,a)$ are in vector form. The goal is to compute a good approximation $Q_{\underline{\rho}_i}(s,a)$ for $Q_i^*(s,a)$, and fitted value iteration (FVI) [26, 29, 30] offers an approximate iterative solution as shown at the lower right corner of Fig 1. We first take the *sampled* Bellman optimality function [30] as the right hand side of (8) serving as the *sampled* version of (6) handling the problem of unknown transition probabilities here,

$$D(Q(s_i,a_i)) = r_i + \gamma \max_{a\in A} Q(s'_i, a) \tag{8}$$

for a sampled transition (s_i, a_i, r_i, s'_i), where $D(\cdot)$ is the sampling operator. Since the sampled function in (8) can not necessarily fit in the space expanded by $\underline{\phi}(s_i,a_i)$, the next representing parameter vector $\underline{\rho}_i$ given the present one $\underline{\rho}_{i-1}$ can be estimated by the general fitted-Q algorithm [31], which is equivalent to solving the following least-square optimization problem:

$$\underline{\rho}_i = \arg\min_{\underline{\rho}\in\mathbb{R}^K} \sum_{j=1}^{N} \left(D(Q_{\underline{\rho}_{i-1}}(s_j,a_j)) - Q_{\underline{\rho}}(s_j,a_j)\right)^2 + \frac{\eta}{2}\|\underline{\rho}\|^2 \tag{9}$$

given a set of training examples $\{(s_j, a_j, r_j, s'_j)_{1\le j\le N}\}$. The second term in (9) is the regularization term for preventing overfitting [32, 33], where η is the parameter to control the influence of regularization. Note that (9) here is exactly the same form as a regularized linear regression problem with a closed form solution. Started with an initial parameter vector $\underline{\rho}_0$ chosen, the iterations should be stopped when some criterion is met.

2.4. State Estimation

At time k within the retrieval session, the system needs to estimate the quality of the present retrieval results ranked by $S^k(Q,d)$, which is the middle part of Fig 1. For this purpose, we collect different pre-retrieval and post-retrieval predictors extracted from the current retrieval context to form a feature vector f_k. These predictors include query length, the current time index k, the indices of the historical actions, clarity score [34], query scope [34], the simplified query clarity score (SCS) [34], ambiguity score [35], similarity between the query and the collection [36], weighted information gain (WIG) [37], query feedback [37], and the top-N similarity scores among the retrieved list as well as the mean and variance. Given a set of training examples $\{(E_j, f_j)_{1\le j\le L}\}$ where E_j is the selected evaluation metric for f_j, we then fit another regularized linear regression to those data points by minimizing least-square errors.

3. EXPERIMENTS

3.1. Experimental Setup

We used a broadcast news corpus in Mandarin Chinese recorded from radio or TV stations in Taipei from 2001 to 2003 as the spoken document archive to be retrieved. There was a total of 5047 news stories, with a total length of 198 hours. In order to evaluate the performance of the proposed approaches with respect to different recognition conditions, we used two different recognition conditions for generating the lattices for the spoken archive. For *Doc (I)*, we used a tri-gram language model trained on 39M words of Yahoo news, and a set of acoustic models with 64 Gaussian mixtures per state and 3 states per model trained on a corpus of 24.5 hours of broadcast news different from the archive tested here. The acoustic features used were MFCC with cepstral mean and variance normalization (CMVN) applied. The one-best character accuracy for the archive was 54.43%. For *Doc (II)*, we cascaded Perceptual Linear Predictive (PLP) features and phone posterior probabilities estimated by a Multilayer Perceptron (MLP) trained from 10 hours of broadcast news different from those tested in a Tandem system. A tri-gram language model trained on 98.5M words of news from several sources, and a set of acoustic models with 48 Gaussian mixtures per state and 3 states per model trained on the 24.5 hours of broadcast news were used. The one-best character accuracy was 62.13%.

Mean Average Precision (MAP) was selected as our retrieval evaluation metric. The costs of actions were set empirically considering the burden caused by each action given to the user. We used a set of gaussians as the basis functions $\phi_m(s,a)$ for the Q-function [28]. The number of gaussians, means, and variances were tuned by a development set. 163 text queries and their relevant spoken documents (not necessarily including the queries) were provided by 22 graduate students. The number of relevant documents for each query ranged from 1 to 50 with an average of 19.5, and the query length ranged from 1 to 4 Chinese words with an average of 2.7 characters. We generated simulated users with the following behavior for training the MDP. When the system took the action *(b) Return Document*, the simulated user viewed the list from the top and chooses the first relevant document. When action *(c) Return Key Term* was taken, the simulated user replied "YES" if the key term

Table 1: MAP and Return for different policies under different recognition accuracies evaluated on both 1-best and lattices.

Policy		*Doc(I) : 1-best*		*Doc(II) : 1-best*		*Doc(I) : Lattice*		*Doc(II) : Lattice*	
		MAP	Return	MAP	Return	MAP	Return	MAP	Return
Baseline	(1) First-pass	0.4521	–	0.4950	–	0.4577	–	0.5044	–
	(2) *Return Doc.*	0.5205	38.32	0.5484	28.31	0.5343	46.59	0.5618	27.40
	(3) *Return Key Term*	0.4475	-19.02	0.4854	-19.66	0.4480	-19.75	0.4931	-21.33
	(4) *Return Request*	0.4704	-31.78	0.4907	-54.32	0.4906	-17.16	0.4993	-55.11
	(5) *Return Topic*	0.4766	4.44	0.5074	-7.66	0.4957	17.91	0.5437	19.24
Oracle	(6) Discrete	0.5796	93.81	0.6178	91.96	0.5926	110.04	0.6409	107.73
	(7) Continuous	0.5839	99.29	0.6231	99.91	0.5921	102.67	0.6400	108.58
Estimated	**(8) Discrete**	**0.5354**	**61.63**	**0.5889**	**72.95**	**0.5491**	**68.12**	**0.6166**	**91.96**
	(9) Continuous	**0.5398**	**67.07**	**0.5964**	**81.38**	**0.5626**	**84.54**	**0.6204**	**96.15**

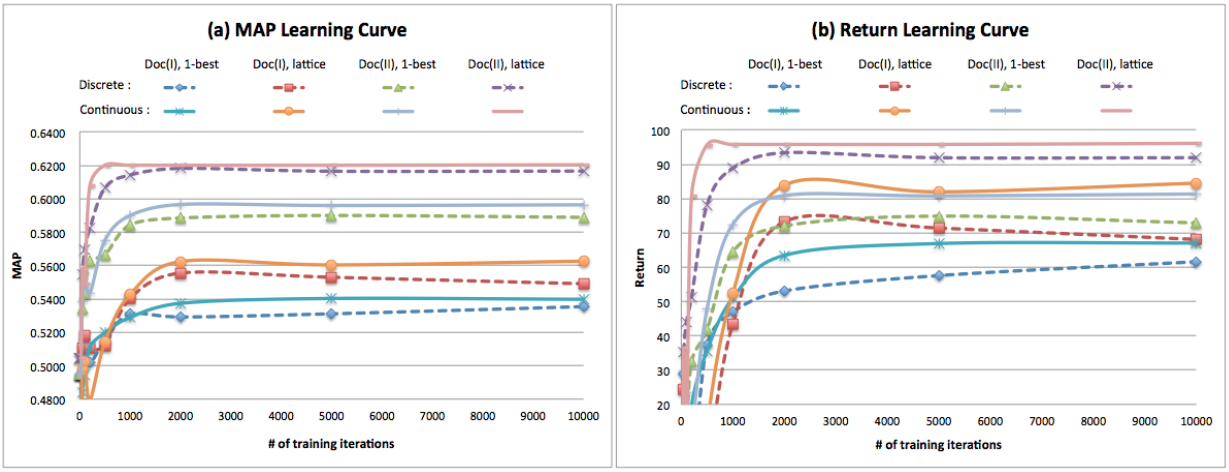

Fig. 2: *(a) Mean Average Precision (MAP) and (b) Return for the proposed approach with estimated states for discrete/continuous state space MDP for the two recognition conditions Doc (I) and (II), 1-best and lattices, under different number of training iterations.*

appeared in more than 50% of the relevant documents and "No" otherwise. In response to *(d) Return Request*, the simulated user entered a term $t^* = \arg\max_t \sum_{d \in \mathcal{R}} f(d,t) ln(1 + idf(t))$ as the feedback, where $f(d,t)$ is the term frequency of term t in document d, $idf(t)$ the inverse document frequency of term t, and R the relevant document set. For the action *(e) Return Topic*, the simulated user randomly returned one of the relevant topics manually labeled (by graduate students). 10-fold cross validation was performed in all experiments, that is, for each trial, 8 out of 10 query folds were used for training, another 1 for parameter tuning, and the remaining 1 for testing.

3.2. Experimental Results

Table 1 shows the results in MAP and Return evaluated on the transcriptions *Doc(I)* and *(II)* obtained with two different conditions as mentioned above for either 1-best results (left half) and lattices (right half). Rows (1)-(5) are baselines, with row (1) for the first-pass results without any interaction, and rows (2)(3)(4)(5) respectively for the system taking a fixed action (*Return Document*, *Return Keyterm*, *Return Request*, and *Return Topic*) for n-times and then *Show list*, where the value of n was tuned to give the best Return. Rows (6)(7)(8)(9) are results for the proposed interactive MDP framework. The *Oracle* section (6)(7) are the results assuming the state or MAP value was precisely known to the system, considered as the upper bound for the approach. Whereas the *Estimated* section (8)(9) are the results with estimated states. In both cases, *Discrete* is for discrete state space by quantizing the MAP into ten levels and trained with a standard reinforcement learning algorithm [9], and *Continuous* for the continuous state space modeling proposed here. We can find out that although conducting a fixed feedback action didn't necessarily guarantee improvements for all queries (rows (2)(3)(4)(5) vs (1)), the proposed approach for choosing actions given states did offer benefits (rows (6)(7)(8)(9) vs (1)(2)(3)(4)(5)). Also, estimated states certainly perform worse than known states (rows (6)(7) vs (8)(9)), and continuous state space was always better than the discrete counterpart (rows (7) vs (6), (9) vs (8)). The above trends are consistent for *Doc(I)* and *(II)*, 1-best or lattices.

Fig. 2 (a) and (b) respectively show the learning curves of MAP and Return for the proposed approach with estimated states compared to discrete state space MDP under different training iterations. In spite of some jitters in the early phase of training, both the MAP and Return grew gradually and then saturated during learning. We also find that for both *Doc(I)* and *(II)*, 1-best or lattice, the continuous state MDP always outperformed discrete one beginning around 2000 iterations.

4. CONCLUSION

Due to the recognition uncertainty and browsing difficulty for spoken content, interactive spoken content retrieval is important. In this paper, we propose to model the interactive spoken content retrieval as a Markov Decision Process (MDP) with a policy optimized by fitted value iteration (FVI) over a continuous state space. A language modeling retrieval engine is also implemented for better retrieval performance including query expansion based on user feedback.

5. REFERENCES

[1] David Robins, "Interactive information retrieval: Context and basic notions," *Informing Science Journal*, 2000.

[2] Ian Ruthven, "Interactive information retrieval," *Annual Review of Information Science and Technology*, 2008.

[3] Teruhisa Misu and Tatsuya Kawahara, "Bayes risk-based dialogue management for document retrieval system with speech interface," *Speech Commun.*, January 2010.

[4] Teruhisa Misu and Tatsuya Kawahara, "Speech-based interactive information guidance system using question-answering technique," in *ICASSP*, 2007.

[5] Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian McGraw, and Jim Glass, "A conversational movie search system based on conditional random field," in *Interspeech*, 2012.

[6] Ian McGraw, Scott Cyphers, Panupong Pasupat, Jingjing Liu, and Jim Glass, "Automating crowd-supervised learning for spoken language systems," in *Interspeech*, 2012.

[7] Yi-Chen Pan, Hung-Yi Lee, and Lin-Shan Lee, "Interactive spoken document retrieval with suggested key terms ranked by a markov decision process," 2012, Audio, Speech, and Language Processing.

[8] Yi-cheng Pan and Lin-shan Lee, "Type-II dialogue systems for information access from unstructured knowledge sources," ASRU, 2007.

[9] Tsung-Hsien Wen, Hung-Yi Lee, and Lin-Shan Lee, "Interactive spoken content retrieval with different types of actions optimized by a markov decision process," in *Interspeech*, 2012.

[10] John Lafferty and Chengxiang Zhai, "Document language models, query models, and risk minimization for information retrieval," 2001, SIGIR '01, ACM.

[11] Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tu Ng, "Statistical lattice-based spoken document retrieval," *ACM Trans. Inf. Syst.*, 2010.

[12] Xuanhui Wang, Hui Fang, and ChengXiang Zhai, "A study of methods for negative relevance feedback," 2008, SIGIR '08, ACM.

[13] Maryam Karimzadehgan and ChengXiang Zhai, "Improving retrieval accuracy of difficult queries through generalizing negative document language models," 2011, CIKM '11, ACM.

[14] Hung-Yi Lee, Tsung-Hsien Wen, and Lin-Shan Lee, "Improved semantic retrieval of spoken content by language models enhanced with acoustic similarity graph," in *SLT*, 2012.

[15] Chengxiang Zhai and John Lafferty, "Model-based feedback in the language modeling approach to information retrieval," 2001, CIKM '01, ACM.

[16] Tao Tao and ChengXiang Zhai, "Regularized estimation of mixture models for robust pseudo-relevance feedback," in *SIGIR'06*, 2006.

[17] Richard Bellman, "Dynamic programming," 1957.

[18] Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction," *Cambridge Journal*, 1999.

[19] Richard Bellman, "A Markovian Decision Process," *Indiana Univ. Math. J.*, vol. 6, 1957.

[20] Stuart Dreyfus, "Richard bellman on the birth of dynamic programming," *Oper. Res.*, Jan. 2002.

[21] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to Data Mining, (First Edition)*, Addison-Wesley Longman Publishing Co., Inc., 2005.

[22] Thomas Hofmann, "Probabilistic latent semantic indexing," in *ACM SIGIR*, 1999.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, 2003.

[24] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. 2004, UAI '04, AUAI Press.

[25] David M. Blei and John D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. 2006, ICML '06, ACM.

[26] Richard Bellman and Sherman Dreyfus, "Functional approximation and dynamic programming.," *Mathematical Tables and Other Aids to Computation*, 1959.

[27] Sebastian Thrun and Anton Schwartz, "Issues in using function approximation for reinforcement learning," in *In Proceedings of the Fourth Connectionist Models Summer School*. 1993, Erlbaum.

[28] Senthilkumar Chandramohan, Matthieu Geist, and Olivier Pietquin, "Optimizing spoken dialogue management from data corpora with fitted value iteration," 2010.

[29] Rémi Munos and Csaba Szepesvári, "Finite-time bounds for fitted value iteration," *Journal of Machine Learning Research*, 2008.

[30] Csaba Szepesvári and Rémi Munos, "Finite time bounds for sampling based fitted value iteration," 2005, ICML '05, ACM.

[31] Andrs Antos, Rmi Munos, and Csaba Szepesvari, "Fitted q-iteration in continuous action-space mdps," 2007, NIPS.

[32] Amir massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvari, and Shie Mannor, "Regularized fitted q-iteration for planning in continuous-space markovian decision problems," in *American Control Conference, 2009. ACC '09.*, 2009.

[33] Amirmassoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvri, and Shie Mannor, "Regularized fitted q-iteration: Application to planning," in *Recent Advances in Reinforcement Learning*, Sertan Girgin, Manuel Loth, Rmi Munos, Philippe Preux, and Daniil Ryabko, Eds., Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008.

[34] Ben He and Iadh Ounis, "Query performance prediction," *Inf. Syst.*, 2006.

[35] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft, "Predicting query performance," 2002, SIGIR '02, ACM.

[36] Ying Zhao, Falk Scholer, and Yohannes Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in *Advances in Information Retrieval*. 2008.

[37] Yun Zhou and W. Bruce Croft, "Query performance prediction in web search environments," 2007, SIGIR '07, ACM.