INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Interactive Spoken Content Retrieval by Deep Reinforcement Learning

Yen-Chen Wu[1], *Tzu-Hsiang Lin*[2], *Yang-De Chen*[1], *Hung-Yi Lee*[1], *Lin-Shan Lee*[1]

[1]Graduate Institute of Communication Engineering, National Taiwan University
[2]Electrical Engineering, National Taiwan University

{yenchen.wu.wyc, iammrhelo, yangdechen0108, tlkagkb93901106}@gmail.com, lslee@gate.sinica.edu.tw

Abstract

User-machine interaction is important for spoken content retrieval. For text content retrieval, the user can easily scan through and select on a list of retrieved item. This is impossible for spoken content retrieval, because the retrieved items are difficult to show on screen. Besides, due to the high degree of uncertainty for speech recognition, the retrieval results can be very noisy. One way to counter such difficulties is through user-machine interaction. The machine can take different actions to interact with the user to obtain better retrieval results before showing to the user. The suitable actions depend on the retrieval status, for example requesting for extra information from the user, returning a list of topics for user to select, etc. In our previous work, some hand-crafted states estimated from the present retrieval results are used to determine the proper actions. In this paper, we propose to use Deep-Q-Learning techniques instead to determine the machine actions for interactive spoken content retrieval. Deep-Q-Learning bypasses the need for estimation of the hand-crafted states, and directly determine the best action base on the present retrieval status even without any human knowledge. It is shown to achieve significantly better performance compared with the previous hand-crafted states.

Index Terms: Interactive Retrieval, Deep-Q-Network, End-to-End, Language Model Retrieval

1. Introduction

Interactive Information Retrieval (IIR)[1, 2] enhances a retrieval system by incorporating the user-system interaction into the retrieval process. The eventual goal of IIR is to guide the users to smoothly find out the desired information through the interactive process[3, 4]. Previous interactive systems, for example, the city guide [5, 6] or the movie browser [7, 8], usually had the content to be retrieved in text form stored in a semi-structured database, and made major efforts on transforming users natural language queries into semantic slots for subsequent database search. Here we focus on interactive retrieval of multimedia[9, 10] or spoken content[11], which is radically different form text.

There are several reasons which make user-system interaction important for spoken content retrieval. Firstly, speech recognition is still far from being perfect, and it inevitably produces errors which make the retrieval results uncontrollable. Moreover, the subword-based technologies are widely used in spoken content retrieval to deal with the OOV issue. These techniques result in higher recall rates, but also lead to lower precision rates. So very often many retrieved results are completely wrong. Also, it's difficult to show the retrieved multimedia or spoken information items on the screen, and it's hard for the users to scan through the retrieved results on the screen[12]. IIR is effective in updating the user instructions to boost the retrieval performance.

In several previous works [13, 14, 15], Markov Decision Process (MDP) is used to model such spoken content IIR. An earlier attempt is to let the user select among a list of retrieved key terms. However, this approach turned out to be inefficient for users. A different approach was then proposed, in which the system has more actions to choose from and decides the most suitable action based on the present status including the present retrieved results, the number of interaction turns, and so on. Some successful results have been achieved by first estimating some human-defined indicators from a set of features as the states and then selecting the actions based on the estimated states [14, 15]. However, such hand-crafted states may be inadequate for the purpose. And in those approaches, the state estimation and action selection are modeled as two cascading blocks and trained independently. Without jointly estimating the whole process, both of them can be sub-optimal.

In this paper, we propose to use deep reinforcement learning in IIR for spoken content. Deep reinforcement learning has recently achieved great renown and success, and deep-Q-Network (DQN) serves as a capable solution to learn from very raw inputs [16, 17]. In IIR with this approach, DQN can take the features originally used for state estimation as the input, and decide the actions directly. Two major contributions were made in this paper. First, we show that the hand-crafted states used previously, like average precision, cannot represent the retrieval status very well, and cascading the estimation of such states with action selection based on such states is definitely not optimal. This is why the DQN proposed here, which is an end-to-end approach, achieved remarkable improvements. Secondly, through utilizing DQN, we show that even using the raw feature–the retrieval scores from the search engine–as the DQN input outperformed the previous approach with hand-crafted states.

2. Proposed Approach

The framework for the proposed approach is depicted in Fig. 1. At the left hand side, the user first enters a query q into the system. With the user query q or plus other feedback information from the user during the interaction, the retrieval module (in the upper middle) described in Section 2.1 will generate a list of retrieved result.

The system takes actions based on the retrieved results to interact with the user as discussed in Section 2.2. A set of features extracted from the retrieved results (at the upper right corner), and the dialogue manager (in the lower middle) determining the action based on the extracted feature is introduced in Section 2.3 and 2.4 respectively. In the dialogue manager, there are paths (A) and (B) for determine the actions. Path (A) (in Section 2.4.1) is the previous approach with estimated states, while path (B) (in Section 2.4.2) uses the deep reinforcement learning proposed in this paper.

http://dx.doi.org/10.21437/Interspeech.2016-1237

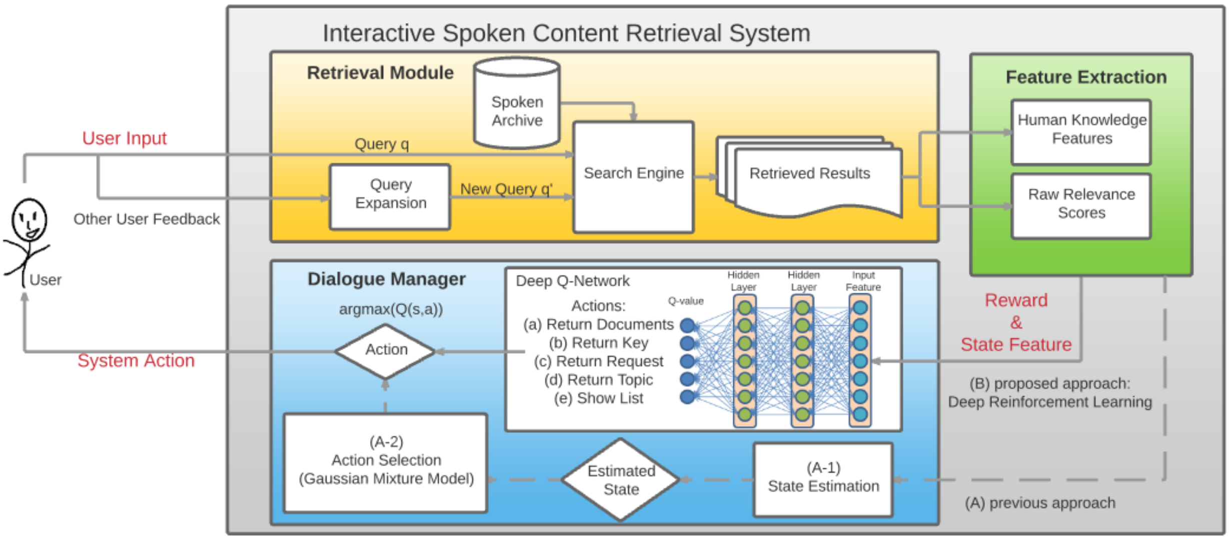

Figure 1: *Block diagram of the proposed approach compared to the previous approach in Path (A) (dash line)*

2.1. Retrieval Module

2.1.1. Language Modeling Retrieval Module

The basic idea of a language model based retrieval framework is to represent the query q and a document d both as language models θ_q and θ_d. More details about estimating θ_d for spoken documents are left out here [18, 19]. The relevance score $S(q,d)$ for the given query q and a document d, which is used to rank the documents d during the retrieval process, is evaluated based on the KL divergence between θ_q and θ_d, or $S(q,d) = -KL(\theta_q \| \theta_d)$. Furthermore, the user may designate a set of terms to be irrelevant to what he/she is seeking for, which can be modeled as a negative information model θ_N. Thus, the complete relevance score $S(q,d)$ considers both the query model θ_q and the negative information model θ_N as below.

$$S(q,d) = -[KL(\theta_q \| \theta_d) - \beta KL(\theta_N \| \theta_d)], \qquad (1)$$

where β as an adjustable parameter [19, 20].

2.1.2. Query-Regularized Mixture Model for Query Expansion

After receiving the feedback from the user during the iteration, the system generates a new query model θ'_q. We adopt the query-regularized mixture model [20, 21, 22] previously proposed for pseudo-relevance feedback to estimate the new query models θ'_q. However, in this query expansion process, the new query model may be tainted by unrelated information in the pseudo-relevant documents. We therefore regularize the query models using a key term set, which is initially composed of the original query and can be expanded throughout the interactive session.

2.2. Actions to be taken by the system

In order to help the user offer useful information, with which the system can retrieve documents better matched to the user's goal, five actions are defined for user-system interaction as presented below.

(a) *Return Documents*: The dialogue manager returns the current list of retrieved results ranked by $S^k(q,d)$ at the present time k in decreasing order and asks the user to select a relevant document.

(b) *Return Key Term*: the dialogue manager asks the user whether a key term t^* is relevant.

(c) *Return Request*: the dialogue manager asks the user to provide an additional query term $\hat{t}$.

(d) *Return Topic*: The dialogue manager returns a list of topics generated with latent topic models [23, 24, 25, 26] and asks the user to select one.

(e) *Show list*: The dialogue manager shows the retrieved results ranked by $S^k(q,d)$ to the user and ends the interactive session.

With actions (a),(b),(c),(d) the system receives extra information from the user and generates a new query q' accordingly for the next step retrieval. Action (e) ends the interactive session and shows the retrieved results to the user.

2.3. Feature Extraction

Sets of features describing the characteristics or present status of the retrieved results from the search engine in Section 2.1 is extracted, based on which the proper actions are selected in Section 2.4. Two sets of features are tested here:

- **Human Knowledge Feature**: A set of features handcrafted based on human knowledge is extracted. These features were used in the previous works [15]. Examples for these features include clarity score [27], query scope [27], the simplified query clarity score (SCS) [27], ambiguity score [28], similarity between the query and the collection [29], weighted information gain (WIG) [30] and query feedback [30].
- **Raw Relevance Scores**: Considering the power of deep learning, the dialogue manager may be able to make decisions simply based on the raw relevance scores of retrieved items without any human knowledge. Here the relevance scores of the top-N documents in the retrieved results are taken as the features with N-dimensions.

2.4. Dialogue Manager

The dialogue manager is based on Markov Decision Process (MDP). MDP [31] is defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $\mathcal{T}(s'|s,a)$ is the transition probability of ending up in state s' when executing action a in state s, $\mathcal{R}$ is the reward function, 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 ($Q^\pi : S \times A \rightarrow R$) is the estimation of the expected discounted sum of all rewards that can be received over an infinite state transition path starting from state s taking action $\pi(s)$: $Q(s,a) = E[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, a_0 = a]$, where r_k is the reward received from the action a_k taken at state s_k, and k is the sequence index for states and actions. The optimal policy maximizes the value of each state-action pair: $\pi^*(s) = arg \max_{a \in \mathcal{A}} Q^*(s,a)$, so finding an optimal policy is equivalent to finding the optimal Q-function.

The reward of a_k,the action taken at state s_k, is defined as

$$r_k = -C_k + \tau[E(s_k) - E(s_{k-1})]. \quad (2)$$

$E(s)$ is some retrieval quantity metric at the state s and C_k is the estimated effort by the user to perform the action a_k. τ is a trade-off parameter between user effort and the retrieval quality improvement.

2.4.1. Previous Approach: Estimating the hand-crafted state

The previous approach [14, 15] for the dialogue management is path (A) in Figure 1. The underlying assumption behind this approach is that the proper choice of the action can be made by considering some evaluation metric for the retrieved results, which is the average precision (AP) here. This assumption leads to a two-stage process, shown as blocks A-1 and A-2, in the dialogue manager of Fig.1. Block A-1 is the state estimation. It takes the feature set extracted from the retrieved results in Section 2.3 as the input, and estimates the AP for them, taken as the state s in the Q-function for action selection. Block A-2 is for action decision. It uses fitted value iteration (FVI) [32] to train a Gaussian mixture model (GMM) to approximate the Q-value function $Q(s,a)$ for each action a, and the system takes the action a with the maximum $Q(s,a)$.

This approach have several weaknesses. First, the evaluation metric AP is not necessary a good representation for states. AP simply indicates how well the retrieved results are. Empirical results have shown that even with the same AP, the optimal actions can be different. However, it's not easy to come up with better state definition with human knowledge. Next, it's not able to fix the error margin for relatively weak state estimation, because the state estimation (block A-1) and action selection (block A-2) are separately trained rather than considered jointly.

2.4.2. Proposed Approach: Deep Reinforcement Learning

The proposed approach used Deep-Q-Network (DQN) to do deep reinforcement learning of Q-function. DQN is able to overcome the problems of the previous approach mentioned above at least to some degree. As can be seem in path (B) of Fig 1, the DQN directly generates the proper action from the input features through the hidden layers. In this way the error propagation for the two cascaded stages (block A-1 and block A-2 in path (A)) is eliminated, and the machine automatically learns from the features extracted from the retrieve module including the human knowledge features and the raw relevance scores of the retrieved results.

The DQN is a deep neural network (DNN) [33, 34] with parameters θ to estimate the state-action value function $Q(s,a;\theta)$[1]. The input of the DQN is the features extracted in Section 2.3, while its output dimension is the same as the number of possible actions a, and the output is the state-action value $Q(s,a;\theta)$ for each action a in the action set $\mathcal{A}$. The DQN is trained by iteratively updating the parameters θ. With parameters obtained at the i-th iteration, denoted as θ^i, θ can be learned by minimizing the following the loss function $L_i(\theta^i)$ in (3) by gradient descent.

$$L_i(\theta^i) = \mathop{\mathbb{E}}_{s,a,r,s' \sim \mathcal{U}(\mathcal{D})} [(\hat{y}_i - Q(s,a;\theta^i))^2]. \quad (3)$$

$\mathcal{D} = \{e_1, e_2, ..., e_t, ...e_L\}$ includes experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ (taking action a_t at state s_t obtaining reward r_t and reaching state s_{t+1} at the next time step) is a dataset collected from many retrieval episodes to be used for training. The expression $s, a, r, s' \sim \mathcal{U}(\mathcal{D})$ in (3) means, instead of using the current experience as prescribed by the standard temporal-difference learning, the network is trained by sampling mini-batches of experiences from $\mathcal{D}$ uniformly at random. This method is referred to as experience replay, which is a key ingredient behind the success of DQN. In this way the efficiency in using the training data can be improve through reuse of the experience samples in multiple updates, and the correlation among the samples used in the update can be reduced through the uniform sampling from the replay buffer [35, 17].

$\hat{y}_i$ in (3) is defined as below:

$$\hat{y}_i = r + \gamma \cdot \max_{a' \in \mathcal{A}} Q(s', a'; \theta^-) \quad (4)$$

where θ^- represents the parameters of a fixed and separate target network, which is taken here as the parameters obtained several iterations before. Freezing the parameters of the target network $Q(s', a'; \theta^-)$ for a fixed number of iterations while updating the online network $Q(s,a;\theta^i)$ is another key innovation for the success of DQN [35, 17], which improved the stability of the algorithm.

3. Experiments

3.1. Experiment Setting

We used a broadcast news corpus in Mandarin Chinese recorded from radio or TV stations in Taipei from 2001 to 2003 as the target document archive to be retrieved. There was a total of 5047 news documents with a total length of 198 hours. We used one-best transcriptions and lattices for the spoken archive. We used a tri-gram language model trained on 39M words of Yahoo news. 163 text queries and their relevant spoken documents (not necessarily including the query terms) were provided by 22 graduate students. We used DQN with two and four hidden layers of 1024 nodes and relu as the activation function [36]. The DQN framework is modified from an open-source code [2]. Mean Average Precision (MAP) was selected as our retrieval evaluation metric. The costs of actions were set empirically considering the extra burden caused when the user provides feedback. And 10-fold cross validation was performed in all experiments.

We generated simulated users with the following behavior for training the dialogue manager.

- (a) *Return Documents*: The simulated user viewed the list from the top and chooses the first relevant document.
- (b) *Return Key Term*: Replied ”YES” if the key term appeared in more than 50% of the relevant documents and ”NO” otherwise.
- (c) *Return Request*: Entered a key term based on Tf-idf (term frequency-inverse document frequency).
- (d) *Return Topic*: Randomly returned one of the relevant topics manually labeled by graduate students.

[1] Because the state-action value function $Q(s,a)$ here depends on the deep neural network parameters θ, the function should be written as $Q(s,a;\theta)$.

[2] https://github.com/spragunr/deep_q_rl

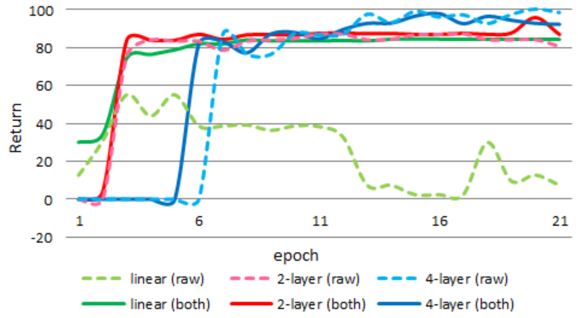

Figure 2: *Learning curves with either raw relevance scores alone (for top-100 retrieved items) (dotted curves) or with human knowledge features in addition (solid curves) with different DQN depths over lattice transcriptions.*

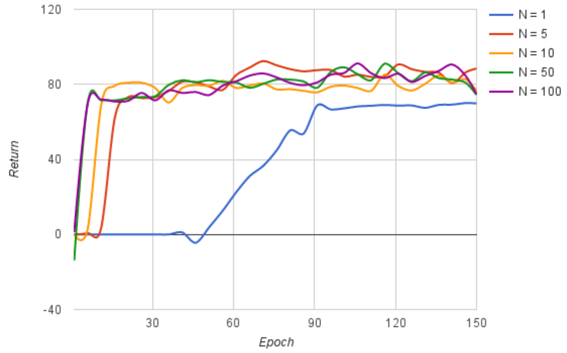

Figure 3: *Learning curves of DQN with 2 layers using different sizes of relevance scores. N is the number of the top-N retrieved items which raw relevance scores were used.*

3.2. Result and Discussion

Table 1 shows the results in MAP and Return $R = \sum_{k=0}^{T} r_k$ achieved on spoken content retrieval module based on either one-best transcriptions (left half) or lattices (right half).

- (a) Baselines: (a-1) are the first-pass results without any interaction, and their returns can be regarded as 0. (a-2) show the results for taking random actions, repeating 1000 times to estimate the MAP and expected return, and (a-3) are the results of the previous work with state estimation.

- (b) DQN with different features: (b-1) are the results when only the raw relevance scores were used. And (b-2) combine the human knowledge features and the raw relevance scores in (b-1).

- (c) Oracle: Obtained by brute-force search over every action sequence whose length was less than 5, and picked the action sequence with the highest return. It is considered as the upper bound for the finite interactive scenario.

From (a-2), we notice that taking actions randomly is not a good way to improve the retrieval process: although the system gains extra information after every interaction, inefficient requests from the system may also burden the user and yield poor returns. The previous hand-crafted state approach in (a-3) obtained much better performance than random actions in (a-2) in terms of both MAP and return. Surprisingly, the proposed approach in (b-1) using DQN end-to-end reinforcement learning only with raw relevance scores without any human knowledge easily outperformed the results using plenty of human knowledge by estimating hand-crafted states in (a-3). This shows that the end-to-end neural network can properly estimate useful action selection indicators implicitly in its hidden layers directly from raw retrieval scores. Finally, using both the raw relevance scores and human knowledge features together in (b-2) achieved the best results.

Table 1: *MAP and Return for (a) previous approach, (b) proposed approach and (c) upper bound evaluated on both one-best transcriptions and lattices.*

Approaches	one-best		lattices	
	MAP	Return	MAP	Return
(a-1) First-pass	0.4521	-	0.4577	-
(a-2) Random Actions	0.4553	-61.7	0.4117	-111.21
(a-3) Hand-crafted States	0.5398	67.07	0.5626	84.54
(b-1) Raw Feature	0.5619	89.27	0.5847	105.03
(b-2) Selected Feature+(b-1)	0.5691	95.72	0.5907	110.90
(c) Upper Bound (Oracle)	0.6554	164.94	0.6639	168.75

To further explore DQN's ability on selecting actions, Figure 2 shows the learning curves of the final return for the proposed approach with different neural network depth using different sets of features: raw relevance scores for the top 100 retrieved items (dotted curves) and plus the human knowledge features (solid curves). Colors green, red, blue represents the curves using linear model (neural network without hidden layers), 2 and 4 layers, respectively. We observe that the linear model can be easily trained when the two sets of features are jointly used. But when solely using raw relevance scores, it diverged and didn't work (dotted-green). This shows that the mapping from raw relevance scores to proper actions is complicated and cannot be modeled without any hidden layers. The red and blue curves for 2 and 4 hidden layers showed that deeper model usually yielded better performance in most cases, especially when using raw relevance scores exclusively. But with a deeper model, it took more epochs to converge. This figure shows the results experimented on the lattice transcription, whereas the results on one-best transcription were similar.

Since the results using raw relevance scores alone is comparable to those using both sets of features jointly with DQN, we conducted an extra experiment using raw relevance scores alone but with different sizes on the one-best transcription. Over DQN of 2 layers, Figure 3 shows the learning curves with the raw relevance scores alone, where N is the number of the retrieved items which relevance scores were used. We observe that DQN could learn from only 1 relevance score, though it gave a relatively poor performance, while all others (5,10,50,100) converged to some good return.

4. Conclusion

Due to the high degree of uncertainty in speech and the difficulty of showing on screen to users, user-system interaction is highly desired for spoken content retrieval. In this paper, we utilize Deep-Q-network(DQN) to learn better state-action values without estimating the hand-crafted states which were used previously. This end-to-end learning is able offer overall optimization for user-system interaction and produce significant improvements on return. We further found that even with raw relevance scores alone without any human knowledge, we achieved very good performance. We hope the results seen here can light up the future directions of DQN in interactive spoken language systems.

5. References

[1] D. Robins, "Interactive information retrieval: Context and basic notions," *Informing Science*, vol. 3, no. 2, pp. 57–62, 2000.

[2] I. Ruthven, "Interactive information retrieval," *Annual review of information science and technology*, vol. 42, no. 1, pp. 43–91, 2008.

[3] J. Luo, S. Zhang, and H. Yang, "Win-win search: Dual-agent stochastic game in session search," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 587–596.

[4] X. Jin, M. Sloan, and J. Wang, "Interactive exploratory search for multi page search results," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 655–666.

[5] T. Misu and T. Kawahara, "Bayes risk-based dialogue management for document retrieval system with speech interface," *Speech Communication*, vol. 52, no. 1, pp. 61–71, 2010.

[6] ——, "Speech-based interactive information guidance system using question-answering technique," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–145.

[7] I. McGraw, S. Cyphers, P. Pasupat, J. Liu, and J. R. Glass, "Automating crowd-supervised learning for spoken language systems." in *INTERSPEECH*, 2012, pp. 2474–2477.

[8] J. Liu, S. Cyphers, P. Pasupat, I. McGraw, and J. R. Glass, "A conversational movie search system based on conditional random fields." in *INTERSPEECH*, 2012, pp. 2454–2457.

[9] D. L. Patton, P. R. Ashe, and J. A. Manico, "Interactive image storage, indexing and retrieval system," Jun. 18 2002, uS Patent 6,408,301.

[10] S.-B. Cho, "Emotional image and musical information retrieval with interactive genetic algorithm," *Proceedings of the IEEE*, vol. 92, no. 4, pp. 702–711, 2004.

[11] J. S. Garofolo, C. G. Auzanne, and E. M. Voorhees, "The trec spoken document retrieval track: A success story," in *Content-Based Multimedia Information Access-Volume 1*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000, pp. 1–20.

[12] Y. Zhang and C. Zhai, "Information retrieval as card playing: A formal model for optimizing interactive retrieval interface," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 685–694.

[13] Y.-C. Pan, H.-Y. Lee, and L.-S. Lee, "Interactive spoken document retrieval with suggested key terms ranked by a markov decision process," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 632–645, 2012.

[14] T.-H. Wen, H.-Y. Lee, and L.-S. Lee, "Interactive spoken content retrieval with different types of actions optimized by a markov decision process." in *INTERSPEECH*, 2012, pp. 2458–2461.

[15] T.-H. Wen, H.-y. Lee, P.-h. Su, and L.-S. Lee, "Interactive spoken content retrieval by extended query model and continuous state space markov decision process," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8510–8514.

[16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[18] J. Lafferty and C. Zhai, "Document language models, query models, and risk minimization for information retrieval," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 111–119.

[19] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "Statistical lattice-based spoken document retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 1, p. 2, 2010.

[20] H.-Y. Lee, T.-H. Wen, and L.-S. Lee, "Improved semantic retrieval of spoken content by language models enhanced with acoustic similarity graph," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 182–187.

[21] C. Zhai and J. Lafferty, "Model-based feedback in the language modeling approach to information retrieval," in *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001, pp. 403–410.

[22] T. Tao and C. Zhai, "Regularized estimation of mixture models for robust pseudo-relevance feedback," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 162–169.

[23] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[25] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[26] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494.

[27] B. He and I. Ounis, "Query performance prediction," *Information Systems*, vol. 31, no. 7, pp. 585–594, 2006.

[28] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 299–306.

[29] Y. Zhao, F. Scholer, and Y. Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in *Advances in Information Retrieval*. Springer, 2008, pp. 52–64.

[30] Y. Zhou and W. B. Croft, "Query performance prediction in web search environments," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 543–550.

[31] R. Bellman, "A markovian decision process," DTIC Document, Tech. Rep., 1957.

[32] S. Chandramohan, M. Geist, and O. Pietquin, "Optimizing spoken dialogue management from data corpora with fitted value iteration."

[33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[34] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[35] L.-J. Lin, "Reinforcement learning for robots using neural networks," DTIC Document, Tech. Rep., 1993.

[36] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013, p. 1.