TYPE-II DIALOGUE SYSTEMS FOR INFORMATION ACCESS FROM UNSTRUCTURED KNOWLEDGE SOURCES

Yi-cheng Pan and Lin-shan Lee

Graduate Institute of Computer Science and Information Engineering, National Taiwan University

{thomas,lslee}@speech.ee.ntu.edu.tw

ABSTRACT

In this paper, we present a new formulation and a new framework for a new type of dialogue system, referred to as the type-II dialogue systems in this paper. The distinct feature of such dialogue systems is their tasks of information access from unstructured knowledge sources, or the lack of a well-organized back-end database offering the information for the user. Typical example tasks of this type of dialogue systems include retrieval, browsing and question answering. The mainstream dialogue systems with a well-organized back-end database are then referred to as type-I dialogue systems here in the paper. The functionalities of each module in such type-II dialogue systems are analyzed, presented, and compared with the respective modules in type-I dialogue systems. A preliminary type-II dialogue system recently developed in National Taiwan University is also presented at the end as a typical example.

Index Terms— Dialogue System, Information Access

1. INTRODUCTION

In recent decades, we have witnessed the wide application of many successful spoken dialogue systems with various application tasks and different capabilities [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. Typical application tasks include travel information services, customer inquiry services, car navigation and so on. In most cases, the research issues are primarily focused on spoken language understanding (SLU) and dialogue modeling. Spoken language understanding very often tries to transform the input speech utterance into a proper form of dialogue act with its parameter set, which can be readily mapped to a proper query to a well-organized back-end database for the desired information. Very often this is an SQL query for a relational database. The approaches of SLU have evolved from knowledge-based to statistical to alleviate the load of grammar development and maintenance [5, 6, 8, 10]. On the other hand, the progress of dialogue modeling has also been improved from rule-based methods to machine learning approaches including reinforcement learning, Markov decision processes (MDP) and the partially observable Markov decision process (POMDP) [12, 13, 11].

In recent years, in addition to the mainstream dialogue systems as mentioned above, a new type of dialogue systems also actively emerged [15, 16, 17, 18, 19]. The distinct feature for this type of dialogue systems is that instead of with a well-organized or relational database at the back-end, the knowledge source to be explored by the dialogue process is usually an unstructured archive of documents, in either text or multimedia form. Typical examples of application tasks of this type of dialogue systems include lecture or meeting minutes retrieval, manual queries, question answering, and news story access. Under such environment, we are actually faced with many new challenges. First, without a well-organized database at the back-end, the goal of SLU becomes difficult to define. Semantic slots and frames may still be useful, but they are not transformed into an SQL query. Secondly, much wider spectrum and even unknown scope and scale of the back-end knowledge source also imply a much higher degree of variations in the user input utterances, which usually include very short queries, homonyms words, polysemous words, and even OOV words. Different from the mainstream dialogue systems, for which the SLU component relies heavily on the ASR output words, here the user's intention may be difficult to correctly identify even with correct ASR output words (e.q. very short queries), not mentioning that ASR can never recognize OOV words. Thirdly, different from the mainstream dialogue systems in which the user is usually very clear about what kind of information is available and accessible; here the user is usually not aware of the content and structure of the back-end knowledge source. As a result efficient interaction and proper guidance by the system during the dialogue process become necessary. Finally, the back-end knowledge very often includes multimedia or spoken documents. Therefore the system outputs are usually difficult to be explained in short speech utterances or shown on the screen, and difficult to be browsed by the user, and the problem becomes even worse when the user tries to access the information via small hand-held clients with very small screen. Therefore the system output presented to the user in a more compact, comprehensive, and structural way becomes an important requirement for efficient interaction.

To address all the issues mentioned above, in this paper we propose a new framework and a new formulation for the new type of dialogue systems, which is referred to as the type-II dialogue systems, while the mainstream dialogue systems with a well-organized back-end relational database are referred to as the type-I dialogue systems. Analytical formulation of the functionalities of all modules in the type-II dialogue systems are presented in this framework and efforts are made such that most existing systems of this type can be properly analyzed with this framework. Each module in the well-known type-I dialogue systems is also compared with the corresponding modules in the type-II dialogue systems in detail. Below a brief summary of type-I dialogue systems is given in Sec. 2, followed by the formulation of type-II dialogue systems in Sec. 3. A preliminary type-II dialogue system developed at National Taiwan University is then summarized in Sec. 4 as a typical example, followed by the conclusion in Sec. 5.

2. BRIEF SUMMARY OF TYPE-I DIALOGUE SYSTEMS

A *type-I dialogue system* is a dialogue system with a well-organized database/knowledge-source behind. We may depict the structure diagram of *type-I dialogue system* as in Fig. 1 [1, 3]. There are two main blocks of functionalities in the system, spoken language understanding (SLU) and dialogue modeling as shown in the figure. The output generator also shown in Fig. 1 usually includes natural

language generation followed by text-to-speech synthesis, but will not be discussed further here due to space limitation. In the following we give brief descriptions about the two parts.

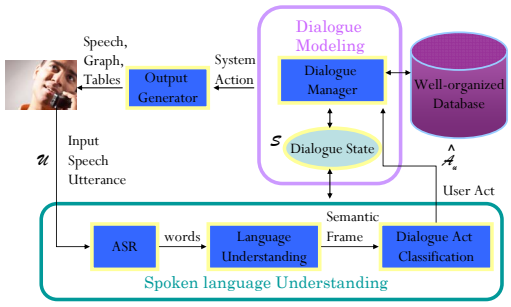

Fig. 1. General form of a *type-I dialogue system*.

2.1. Spoken Language Understanding (SLU)

This is the block at the bottom of Fig. 1. The goal is to convert the user's input speech utterance $\mathcal{U}$ on the left of Fig. 1 into a proper user act $\hat{\mathcal{A}}_u$ on the right of the figure, given the current internal dialogue state $\mathcal{S}$ at the middle of the figure, usually formulated as [3]

$$\begin{aligned}\hat{\mathcal{A}}_u &= \text{argmax}_{\mathcal{A}_u} P(\mathcal{A}_u|\mathcal{U},\mathcal{S}) \\ &= \text{argmax}_{\mathcal{A}_u} \{P(\mathcal{U}|\mathcal{A}_u,\mathcal{S})P(\mathcal{A}_u|\mathcal{S})\}. \end{aligned} \quad (1)$$

By introducing a latent variable $\mathcal{W}$, the possible word string carried by $\mathcal{U}$, with some assumptions and Viterbi approximation, we may have [3]

$$\begin{aligned}\hat{\mathcal{A}}_u = \text{argmax}_{\mathcal{A}_u} \{&\max_{\mathcal{W}} \{P(\mathcal{U}|\mathcal{W}) \\ &\cdot P(\mathcal{W}|\mathcal{S}) \cdot P(\mathcal{A}_u|\mathcal{W},\mathcal{S})\}\}. \end{aligned} \quad (2)$$

For sub-optimal solution, we may decompose Equ. (2) and first obtain $\hat{\mathcal{W}}$, the estimate of $\mathcal{W}$, by

$$\hat{\mathcal{W}} = \text{argmax}_{\mathcal{W}} \{P(\mathcal{U}|\mathcal{W}) \cdot P(\mathcal{W}|\mathcal{S}))\}, \quad (3)$$

and then in turn have $\hat{\mathcal{A}}_u$ based on $\hat{\mathcal{W}}$ [3],

$$\hat{\mathcal{A}}_u = \text{argmax}_{\mathcal{A}_u} \left\{P(\mathcal{A}_u|\hat{\mathcal{W}},\mathcal{S})\right\}. \quad (4)$$

Similarly, by introducing another latent variable $\mathcal{C}$, the possible sequence of semantic slots for the word string $\mathcal{W}$, and with similar assumption and approximation as above we may solve Equ. (4) as [5]

$$\hat{\mathcal{C}} = \text{argmax}_{\mathcal{C}} \left\{P(\mathcal{C}|\hat{\mathcal{W}},\mathcal{S})\right\}. \quad (5)$$

$$\hat{\mathcal{A}}_u = \text{argmax}_{\mathcal{A}_u} \left\{P(\mathcal{A}_u|\hat{\mathcal{C}},\mathcal{S})\right\}. \quad (6)$$

Equ. (5) alone is an important research issue and many works have been done, very often considered as the core of SLU. Equ. (5) can also be rewritten as [5]

$$\hat{\mathcal{C}} = \text{argmax}_{\mathcal{C}} \left\{P(\hat{\mathcal{W}}|\mathcal{C}) \cdot P(\mathcal{C}|\mathcal{S})\right\}, \quad (7)$$

where $P(\mathcal{C}|\mathcal{S})$ is the semantic model and $P(\hat{\mathcal{W}}|\mathcal{C})$ is the lexical model, similar to the language model and acoustic model counterparts in ASR and therefore we may use the traditional HMMs for Equ. (7).

In HMMs, however, the Markov assumption makes the adjacent words only loosely coupled and complicated nested structure is not handled. This is why the hierarchical or multi-step models were also proposed to extend the finite state transition network structure into a recursive transition network to support context-free languages [9, 8, 5].

2.2. Dialogue Modeling

Internal states $\mathcal{S}$ at the middle of Fig. 1 are usually used to handle the dialogue discourse and manage the possible actions taken by the machine [1]. Although the state transition and machine action can be rule-based, with growing scale and wider application domains, statistical approaches turned out to be more attractive for easier system development and maintenance [1].

The dialogue process has been popularly modeled as a Markov Decision Process (MDP) [20, 21], that is, (1) the current state transition from $\mathcal{S}^{t-1}$ to $\mathcal{S}^t$ depends only on the last state $\mathcal{S}^{t-1}$, the last machine action $\mathcal{A}_m^{t-1}$, and (2) the current machine action $\mathcal{A}_m$ depends only on the current state. To be explicit, we model the state transition probability as $P(\mathcal{S}^t|\mathcal{S}^{t-1},\mathcal{A}_m^{t-1})$ and the policy $\pi(\mathcal{S},\mathcal{A}_m)$ determines the probability of taking the machine action $\mathcal{A}_m$ when in state $\mathcal{S}$. In this way, the proper action taken by the system in each state can be learned by reinforcement learning, in which a reward function $r^{t+1} = r(\mathcal{S}^t,\mathcal{A}_m^t)$ is defined and the goal is to maximize the *total reward* R_0, where $R_t = \sum_{\tau=t}^{T} r^\tau$ and T is the time when the dialogue is finished.

In maximizing the *total reward*, the *value function for policy* π is usually introduced:

$$V_\pi(\mathcal{S}) = E_\pi \left\{R_t|\mathcal{S}^t = \mathcal{S}\right\}, \quad (8)$$

which gives the expected total reward after time t given the state $\mathcal{S}$ at time t by following the policy π defined for every state. We can then decompose $V_\pi(\mathcal{S})$ by different machine actions $\mathcal{A}_m^t$, that is the *action-value function for policy* π:

$$Q_\pi(\mathcal{S},\mathcal{A}) = E_\pi \left\{R_t|\mathcal{S}^t = \mathcal{S},\mathcal{A}_m^t = \mathcal{A}\right\}, \quad (9)$$

which gives the expected total rewards after time t given state $\mathcal{S}$ and taking machine action $\mathcal{A}$ while following π thereafter. We can then describe the *optimal value function* as

$$V^*(\mathcal{S}) = \max_\pi V_\pi(\mathcal{S}),$$

and also the *optimal action-value function*

$$Q^*(\mathcal{S},\mathcal{A}) = \max_\pi Q_\pi(\mathcal{S},\mathcal{A}).$$

By the Bellman equation, we then have

$$\begin{aligned}V^*(\mathcal{S}) = \max_{\mathcal{A}} \Big\{&r(\mathcal{S},\mathcal{A}) + \\ &\sum_{\mathcal{S}'} \left\{P(\mathcal{S}^t = \mathcal{S}'|\mathcal{S}^{t-1} = \mathcal{S},\mathcal{A}_m^{t-1})V^*(\mathcal{S}')\right\}\Big\}, \end{aligned} \quad (10)$$

and also

$$\begin{aligned}Q^*(\mathcal{S},\mathcal{A}) = &r(\mathcal{S},\mathcal{A}) + \\ &\sum_{\mathcal{S}'} \left\{P(\mathcal{S}^t = \mathcal{S}'|\mathcal{S}^{t-1} = \mathcal{S},\mathcal{A}_m^{t-1})\max_{\mathcal{A}'} Q^*(\mathcal{S}',\mathcal{A}')\right\}. \end{aligned} \quad (11)$$

If the state transition probability $P(\mathcal{S}^t = \mathcal{S}'|\mathcal{S}^{t-1} = \mathcal{S},\mathcal{A}_m^{t-1})$ is known, Equ. (11) can be solved by dynamic programming. If the state transition probability is not known, methods based on sampling actual

dialogues can be used. For example, in *on-policy temporal difference learning* [20], after each visit from $(\mathcal{S}, \mathcal{A})$ to $(\mathcal{S}', \mathcal{A}')$, that is, in state $\mathcal{S}$ and taking action $\mathcal{A}$, we reach state $\mathcal{S}'$ and take another action $\mathcal{A}'$, we may update the *optimal action-value function* $Q(\mathcal{S}', \mathcal{A}')$ as

$$Q^*(\mathcal{S}, \mathcal{A}) = Q^*(\mathcal{S}, \mathcal{A}) + \alpha[r(\mathcal{S}, \mathcal{A}) + Q^*(\mathcal{S}, \mathcal{A}) - Q^*(\mathcal{S}', \mathcal{A}')], \tag{12}$$

where α determines the learning rate. In both cases, once we can find $Q^*(\mathcal{S}, \mathcal{A})$, the optimal policy can be found by

$$\pi(\mathcal{S}) = \text{argmax}_{\mathcal{A}} Q^*(\mathcal{S}, \mathcal{A}).$$

In the case of *on-policy temporal difference learning*, in order to visit each possible $(\mathcal{S}, \mathcal{A})$, most of the time we choose the optimal action according to $\text{argmax}_{\mathcal{A}} Q^*(\mathcal{S}, \mathcal{A})$, but occasionally (with probability ϵ) we also randomly choose other actions, and this is called ϵ-greedy method.

Above is the basic formulation for modeling the dialogue procedure by an MDP [20]. But in realistic cases with uncertain inputs, at each time we are not sure about the exact state we are in. To handle such uncertainties, another modeling approach, *Partially Observable Markov Decision Process* (POMDP) is formulated. The basic idea of POMDP is to keep a distribution over a set of states instead of only one, which is the belief state. In this way it is possible to model the best actions for each possible distribution [11].

3. TYPE-II DIALOGUE SYSTEMS

From Fig. 1, it can be found that for *type-I dialogue systems*, after the user produces his utterances, the system tries to interpret the utterances in form of semantic frame, whose format, structure and possible content are actually determined during system development in order to be consistent with the back-end well-organized database, for the ease of accessing the database, for example by SQL queries. For *type-II dialogue systems*, however, very often we do not have a well-organized database at the back-end, thus the semantic frame/slot is difficult to define. Instead, we need to access the information from some unstructured knowledge sources in the World Wide Web, which includes huge quantities of multi-media data in addition to text information for browsing, retrieval, or question answering purposes. It is infeasible or even impossible to structure all these data into a well-organized database. So when the well-organized database as in Fig. 1 is missing, and the formulation of *type-I dialogue systems* in the above becomes inadequate.

People may wonder that, retrieval and question answering systems have been well developed for a long time, and why we need at this moment the somewhat different concept of *type-II dialogue systems* for an existing problem. The important point here is that the conventional approaches and frameworks for text/spoken document retrieval and question answering are actually not adequate without dialogue functionalities due to the several real scenarios below. First, the user's query during retrieval is usually very short which inevitably includes ambiguity and therefore results in too many outputs. For example, given the query "*George Bush*", the user may be interested in the *Iraq* or *China* issue. The system can tell the difference only with following-up interactions, or dialogue functionalities. Secondly, OOV word problem for spoken document retrieval (SDR) is often handled by subword-based indexing and retrieval techniques, but such techniques also naturally lead to many irrelevant retrieved documents and thus low precision. Following-up interactions or dialogues are therefore very helpful for the user to identify and select the desired information. Thirdly, the gap between the system and the user in such scenarios is usually huge. It is difficult for the user to formulate his queries precisely to retrieve efficiently. The user knows what he needs, but not how to translate it into a good query; while the system knows exactly which query leads to which set of documents, but needs a good mechanism to probe the user's needs. As a result, a series of follow-up questions and interactions is certainly very helpful. Dialogues are certainly a good solution to this problem.

For the issues discussed above, we propose here the concept of *type-II dialogue systems* for information access from unstructured knowledge sources. Such dialogue systems are certainly qualified to the called *dialogue systems* but they are quite different from the mainstream dialogue systems as we summarized in Sec. 2 due to the absence of a well-organized database. Such dialogue systems are thus referred to as *type-II dialogue systems* in this paper, while those summarized in Sec. 2 are referred to as *type-I dialogue systems*.

The block diagram of the proposed *type-II dialogue systems* is shown in Fig. 2. This block diagram is primarily for information retrieval task. Minor modifications may be needed for other tasks such as question-answering. There are three major building blocks in Fig. 2: spoken language based information access, dialogue modeling, and multi-modal user interface. They will be discussed in detail in the following sections. The comparison of each module in the *type-I dialogue systems* with the corresponding module in the *type-II dialogue systems* in Sec. 2 will be discussed.

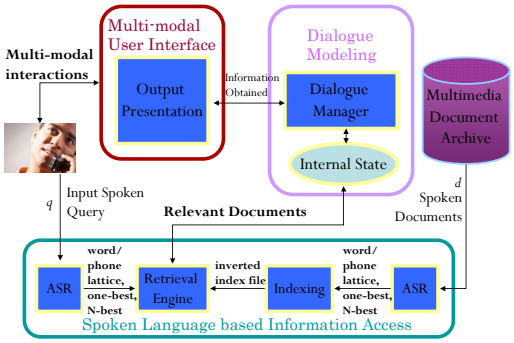

Fig. 2. Structure of a *type-II dialogue system*.

3.1. Spoken Language based Information Access

This is the block at the bottom of Fig. 2, corresponding to the block of SLU in Fig. 1. In *type-I dialogue systems*, SLU is to *understand* the user's utterance in terms of the format of semantic frames, which is tightly coupled with the back-end database. In other words, the goal of *understanding* is to access the back-end database efficiently, for example, with SQL queries. In *type-II dialogue systems*, however, the back-end multimedia document archive no longer has clear metadata tags and rigid structure. We are not able to access the database via tags (the semantic slots), but have to rely on other linguistic features. Such issue has been extensively investigated in the area of Spoken Document Retrieval (SDR). Multimedia content very often carries speech information, so they can be similarly accessed by the associated spoken documents. Note that although Spoken Document Retrieval (SDR) is quite different from SLU in Fig. 1, the basic concept is similar in some sense. We try to perform better matching between the input spoken query with the information in the back-end document archive. Below we present this block of Fig. 2 primarily as an SDR task. Extension to other related tasks are natural.

While the well-organized database in type-I dialogues can be accessed efficiently by SQL queries with clear tags, the spoken docu-

ments in type-II dialogues are conventionally accessed by the words in the query and the documents. The problem considered here can then be formulated as ranking the documents $d \in \mathcal{D}$ (here in this paper by document we mean either a document or other proper elements for retrieval, such as a segment of information significantly shorter than a document; $\mathcal{D}$ is the entire archive) given the user's input spoken query q according to $P(d|q)$. With an approach similar to Equ. (1) to (2), by introducing a latent variable $\mathcal{W}$, possible word sequence for q, and with the Viterbi approximation we may have

$$P(d|q) \approx \max_{\mathcal{W}} \{P(d|\mathcal{W}) \cdot P(\mathcal{W}|q)\}. \quad (13)$$

Similarly Equ.(13) can be solved in a sub-optimal way by first finding the optimal word sequence $\hat{\mathcal{W}}$ according to $P(\mathcal{W}|q)$, and then $P(d|q)$ can be approximated with $P(d|\hat{\mathcal{W}})$. Actually if the user's input query is in textual form, which is the scenario for many SDR tasks, the exact word sequence $\mathcal{W}$ is known and the term $P(d|q)$ in the above can be simply replaced by $p(d|\mathcal{W})$. For spoken queries it is for sure that information in addition to the one-best transcription $\hat{\mathcal{W}}$, such as word/phone lattices, are helpful. This is equally true for the spoken documents d with unknown transcriptions, and is why lattices for both q and d are shown in Fig. 2. All the above formulation based on Equ. (13) carries some concept similar to that of SLU as summarized in Sec. 2.1. Note that there are also other very successful models for SDR, for example the vector space model, which may not be well represented by the probabilistic formulation of $P(d|q)$ here.

In several works it is assumed that $P(d|\mathcal{W})$ is proportional to $P(\mathcal{W}|d)$, or how often the word string $\mathcal{W}$ occurs in d. This is a reasonable assumption, with which the original problem can be reduced to ranking the posterior probabilities $P(\mathcal{W}|d)$. An efficient way for indexing $\mathcal{W}$ given the lattices generated from d was also proposed [22], which can be considered as a direct inversion of the whole lattice. In this way the posterior probabilities can be produced in a precise way, although high storage space is required.

Some other approaches were then developed recently to use the lattice information in an approximate but space-efficient way. An efficient approach was proposed to cluster the word arcs in a lattice according to their positions and then obtain Position Specific Posterior Lattices (PSPL) [23]. Such position knowledge is very useful for the proximity information and $P(\mathcal{W}|d)$ can be easily approximated by each compositional substrings in $\mathcal{W}$ with appropriate positions. More possible approaches were proposed for clustering the word arcs in a word lattice to make the index file as small as possible at very limited performance degradation [24]. In other approaches confusion networks are also proposed for efficient lattice information utilization with much less space requirement compared with a direct lattice inversion [17, 18].

Out-of-vocabulary (OOV) word is always another important issue in SDR. Many keywords in the documents may be OOV words and cannot be recognized. A useful approach to this problem is to represent the query and/or documents as sequences of subword units either in a lattice or in a one-best transcription [22, 25, 26, 27]. The feasibility of such subword-based SDR approaches have been well verified, although it is also true that subword units inevitably bring more false-positives. In our recent work [28], an approach to easily evaluate the position specific posterior probabilities for subword units in a word lattice was proposed, referred to as subword-based PSPL, which was shown to provide significant performance improvements over PSPL for either in-vocabulary or out-of-vocabulary queries. When integrated with the original word-based PSPL, the improvements can be even higher.

All the above formulation is more or less based on the probability $P(d|\mathcal{W})$, which may be somehow assumed to be proportional to $P(\mathcal{W}|d)$, or how often $\mathcal{W}$ occurs in d. Actually, some documents should still be regarded as relevant even $\mathcal{W}$ never appears wholly or partially. Such relationships can be evaluated via semantic analysis by additional approaches like Latent Semantic Analysis (LSA), Probabilistic LSA (PLSA) or Latent Dirichlet Allocation (LDA) or similar. To cover OOV or rare words, subword sequences can be regarded as basic units and included in such semantic analysis [29].

3.2. Multi-Modal User Interface

This is the block at the top middle of Fig. 2. In *type-I dialogue systems* the system output is usually natural language generation followed by text-to-speech synthesis, as presented by a block of "output generator" in Fig. 1. For *type-II dialogue systems*, however, unstructured document archive gives unstructured retrieved information for the user. Such information are usually very difficult to summarize in simple comprehensive utterances. When such information are multimedia, they are even difficult to be shown on the screen and difficult to browse. Therefore special efforts have to be made in the presentation of the system output so as to allow multi-modal interaction between the user and the system, and the system output and user input both can have many different forms in addition to speech. This is why in Fig. 2 there is such a block including Output Presentation, and is another distinct feature of *type-II dialogue systems*. Here a visual display screen is helpful, but we should try to make such screen as small as possible to make the system compatible to hand-held devices.

For text information retrieval, interface design for better user feedback has been an important issue with many interesting approaches. In an example [30], a delicate interface was developed for further interactions including document title, query-biased summary of the document, a list of top-ranking sentences, a sentence in the document summary and each summary sentence in the context it occurs in the document. As other examples, in MIT's lecture browser, the interface includes the video player for the retrieved results and the recognition results with the matched query terms highlighted [31]. In Kyoto University's Information Guidance system, the interface includes the retrieved documents, the explicit answer to the user's question, and also possible following-up questions to draw user's interest [15]. All of these are typical examples of *type-II dialogue systems* with special multi-modal user interface.

In our recent work [19], we propose to construct in real time a topic hierarchy to present the system output as an interface between the user and the system. Every node in this hierarchy represents a set of retrieved documents with similar semantic content, and is labeled by a key term, or the topic. So the user can easily select the set of retrieved documents by the topics, or expand his query by adding extra query terms chosen from the topic hierarchy. In this way, the system can summarize and structure the documents retrieved under user's current input queries, and gives the user some clues for further query term selection. The user can also make his query more specific by approving or disapproving the topics in the hierarchy, or directly expand his query with other key terms. This provides an efficient channel for interactions between the user and the system. This topic hierarchy is constructed in two steps. The first is to extract key terms (topics) from the retrieved spoken documents, and the second is to use these key terms to construct a balanced tree structure. In the first step, PLSA models trained from the back-end archive can be used to evaluate the topic entropy for each term in the transcriptions. Those terms with topic entropy below a threshold carry more topical information and therefore are chosen as the key terms [32]. In the second step, the Hierarchical Agglomerative Clustering and Partitioning algorithm (HAC+P) [33] can be performed on the

key terms extracted above based on some linguistic and statistical features for these key terms [19].

3.3. Dialogue Modeling

This is the block also at the top middle of Fig. 2. Dialogue modeling has been the core module for *type-I dialogue systems*. For type-II dialogues, however, not too much work on this part has been reported, probably because the concept of considering the necessary interactions between the user and the system for retrieval, browsing, question answering and so on as dialogue systems is still new. In our recent work on Interactive Spoken Document Retrieval [34, 35], however, we proposed an approach for dialogue modeling based on Markov Decision Process (MDP) as mentioned in Sec. 2.2. This approach is thus summarized below as an example approach for dialogue modeling for *type-II dialogue systems*.

The approach presented here is based on the output presentation approach we proposed and summarized above in Sec. 3.2 for retrieval of spoken documents, in which a topic hierarchy with nodes labeled by key terms or topics is constructed for system outputs. At the early stage of the dialogue, because the user doesn't know what can be found from the back-end archive and how to enter the query efficiently, very often he only enters very short queries. With such very short queries, the retrieved documents can be many, a large number of key terms can be extracted, and as a result the topic hierarchy constructed can be very large. The purpose of dialogue modeling here is therefore to rank the key terms before constructing the topic hierarchy. The goal of ranking here is to minimize the number of key terms the user needs to enter before his information needs are satisfied. In this way, the key terms ranked the highest will appear on the top of the constructed topic hierarchy, so the user may spend only minimum time in navigating across the hierarchy, and the system may use only limited space in the screen of hand-held clients to show the most important topics first. This is the basic scenario for dialogue modeling discussed here.

First of all, we define an internal state $\mathcal{S}$ for the dialogue as the *And*-combinations of all the query terms the user has entered from the beginning, and the machine action $\mathcal{A}_m$ as the change in the internal state when an extra query term is entered by the user to further expand the query. For example, in the state $S_2 = s[t_i, t_j]$ (t_i, t_j are two query terms), if a new term t_k is entered, this automatically leads to a new state $S_5 = s[t_i, t_j, t_k]$. In this way the state transition function of $P(S^t|S^{t-1}, A_m^{t-1})$ as mentioned in Sec. 2.2 is actually deterministic, which is somehow different from the conventional MDP framework mentioned above. The goal of dialogue modeling here is to minimize the number of query terms a user has to enter before his information needs are satisfied. We thus define the total reward R_0, to be maximized as mentioned in Sec. 2.2 in the MDP framework, as the above number of query terms the user has to enter. But the latter number should be minimized rather than maximized, or the total reward R_0 should be negative of the above number. We therefore define the reward function of $r(S, A_m)$ as negative one if the action A_m leads the state S to a new state S' and the documents retrieved by S' doesn't satisfy the user, and zero otherwise.

With the above definitions we can see that the reward function is determined by each specific user rather than a predefined function. The learning process can then be represented as a state transition tree structure as shown in Fig. 3, in which each node is an internal state, or a series of query terms entered. The tree in Fig. 3 is for a specific user, in which the leaf nodes represented by double circles are those states where the user is satisfied. Each of these leaf nodes are labeled by a score $m(\cdot)$, which is the negative of the number of

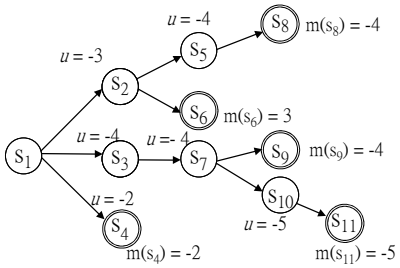

Fig. 3. A typical learning tree constructed for the retrieval states for a specific user.

the query terms successively entered in order to arrive at the state, or the total reward R_0 to be maximized. By Equ. (10) we give the score u to each intermediate state as shown in Fig. 3, which is the maximum score $m(\cdot)$ for all child leaf nodes of the intermediate state , $u = \max_i[m(s_i)]$, where the maximization is performed over all child leaf nodes of the state. Such a learning process can be performed with a huge number of training users to obtain the dynamics of the reward function and a balanced view of how efficient a query term entered at each state can satisfy the user. The scores u for all the states averaged over a huge number of training users is then used to rank the key terms. The query term ranking and the internal states then determines the operations of the dialogue manager, including the construction of the topic hierarchy [34, 35].

4. A PRELIMINARY TYPE-II DIALOGUE SYSTEM

An initial prototype of *type-II dialogue system* was successfully developed at National Taiwan University for retrieval of Mandarin Chinese broadcast news segments, with an archive of 10,000 news stories serving as the back-end unstructured knowledge source. The topic hierarchy presenting the system output for Multi-modal User Interface discussed in Sec. 3.2 and the term ranking approach for Dialogue Modeling discussed in Sec. 3.3 were both implemented [19, 34, 35].In the test, 5,000,000 users were simulated in training the dialogue modeling module, while another 1,000 users were simulated for test. We evaluated the performance of this *type-II dialogue system* in terms of task success rate and the average number of query terms needed for a successful retrieval. The task was defined to be successful if the user is satisfied or the recall is above a given threshold [34]. Recognition errors for queries and documents were simulated by generating feature vectors according to the Hidden Markov Models with increased Gaussian mixture variances, and then recognized normally [35]. The dialogue modeling discussed in Sec. 3.3 is compared against two previously proposed term selection algorithms, the *wpq* method [36] and the *tf-idf* method.

Fig. 4(a) shows the detailed numbers of failure trials and successful trials completed in different number of query terms out of the 1000 simulated testing users. The queries was assumed to be 100% correct, and 1000 out of the 10,000 news stories were assumed to be spoken with character accuracy of 71% (the rest in text form and completely correct). It can be found that with the *tf-idf* method, 746 out of the 1000 trials failed; all successful trials were finished within 7 turns. Much better performance was obtained for the *wpq* method. However, when the proposed dialogue modeling was used, only 120 trials failed, and all trials were completed within 4 turns. Similar plots can be seen in Fig. 4(b), in which query accuracy was reduced to 74% and 1700 out of the 10,000 news stories were spoken with

character recogniztion accuracy of 77%.

In Fig. 5(a)(b) we plot the task success rate and the average number of query terms needed in successful trials for the same three methods as discussed above as functions of the query recognition accuracies, where in case (1) all the 10,000 news stories were 100% correct, and in case (2) 1700 of them has accuracy of 77%. It can be found that the performance of the dialogue modeling was very well, and quite robust with respect to recognition errors.

5. CONCLUSION

In this paper we propose and formulate the framework of type-II dialogue systems, which may become more and more important as the data on the networks are increasing exponentially. The initial prototype mentioned above is still very preliminary though. Much more work is needed in the future.

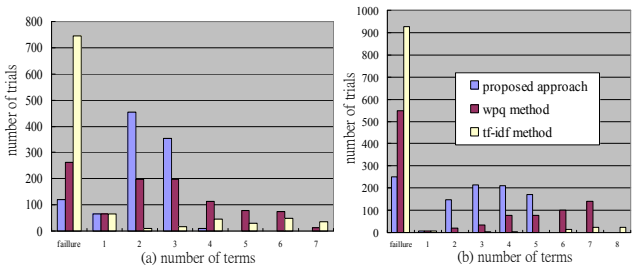

Fig. 4. Number of failure trials and successful trials completed in different number of query terms for the proposed dialogue modeling approach compared to the *wpq* and *tf-idf* methods for two different cases.

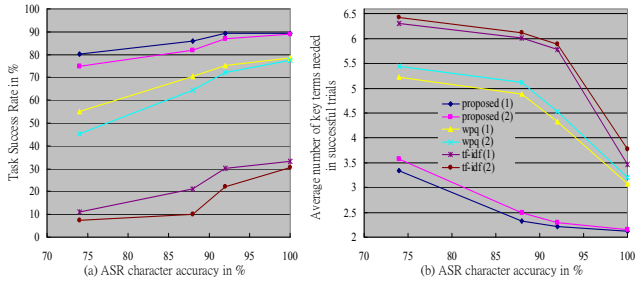

Fig. 5. (a) The task success rates and (b) the average numbers of query terms needed in successful trials for different query recognition accuracies for two different cases.

6. REFERENCES

[1] V.W. Zue and J.R. Glass, "Conversational interfaces: Advances and challenges," *Proc. of IEEE*, vol. 88, no. 8, pp. 1166–1180, 2000.

[2] N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, and M. Gilbert, "The AT&T spoken language understanding system," *IEEE Trans. on Speech and Audio Processing*, vol. 14, no. 1, pp. 213–222, 2006.

[3] S. Young, "Talking to machines (statistically speaking)," in *ICSLP*, 2002.

[4] S. Seneff V.W. Zue, J.R. Glass, J. Polifroni, C. Pao, T.J. Hazen, and L. Hetherington, "Jupiter: A telephone-based conversational interface for weather information," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.

[5] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[6] E. Levin and R. Pieraccini, "Chronus, the next generation," in *ARPA Spoken Language Sysytems Technology Workshop*, 1995.

[7] D. Stallard, "The BBN ATIS4 dialogue system," in *ARPA Spoken Language Sysytems Technology Workshop*, 1995.

[8] Y. He and S. Young, "Hidden vector state model for hierarchical semantic parsing," in *ICASSP*, 2003, pp. 268–271.

[9] S. Miller, R. Bobrow, R. Schwartz, and R. Ingria, "Statistical language processing using hidden understanding models," in *Proc. Human language Technology Workshop*, 1994, pp. 278–282.

[10] S. D. Pietra, M. Epstein, S. Roukos, and T. Ward, "Fertility models for statistical natural language understanding," in *ACL*, 1997, pp. 168–173.

[11] J. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–242, 2007.

[12] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialogue strategies," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[13] M.A. Walker, "An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email," *Journal of Artificial Intelligence Research*, vol. 12, pp. 387–416, 2000.

[14] B.-S. Lin and L.-S. Lee, "Computer-aided analysis and design for spoken dialogue systems based on quantitative simulations," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 534–548, 2001.

[15] T. Misu and T. Kawahara, "Speech-based interactive information guidance system using question-answering technique," in *ICASSP*, 2007, pp. 145–148.

[16] A. Park, T.Hazen, and J.R. Glass, "Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling," in *ICASSP*, 2005, pp. 497–500.

[17] T. Hori, I.L. Hetherington, T.J. Hazen, and J.R. Glass, "Open-vocabulary spoken utterance retrieval using confusion networks," in *ICASSP*, 2007, pp. 73–76.

[18] J. Mamou, D.Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *SIGIR*, 2006, pp. 51–58.

[19] Y.-C. Pan, C.-C Wang, Y.-C Hsieh, T.-H Lee, Y.-S Lee, Y.-S Fu, Y.-T Huang, and L.-S Lee, "A multi-modal dialogue system for information navigation and retrieval across spoken document archives with topic hierarchies," in *ASRU*, 2005, pp. 370–380.

[20] RS. Sutton and AG. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

[21] E. Levin and R. Pieraccini, "A stochastic model of computerhuman interaction for learning dialogue strategies," in *Eurospeech*, 1997, pp. 1883–1886.

[22] M. Saraclar and R.Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT*, 2004.

[23] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *ACL*, Ann Arbor, 2005, pp. 443–450.

[24] Z.-Y. Zhou, P. Yu, C. Chelba, and F. Seide, "Towards spoken-document retrieval for the internet: Lattice indexing for large-scale web-search architectures," in *HLT*, 2006, pp. 415–422.

[25] K. Ng, *Subword-based Approaches for Spoken Document Retrieval*, Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[26] M. A. Siegler, *Integration of Continuous Speech Recognition and Information Retrieval for Manually Optimal Performance*, Ph.D. thesis, Carnegie Mellon University, 1999.

[27] Logan B, P.Moreno, and O. Deshmukh, "Word and sub-word indexing approaches for reducing the effects of oov queries on spoken audio," in *HLT*, 2002.

[28] Y.-C. Pan, H.-L. Chang, and L.-S. Lee, "Subword-based position specific posterior lattices (S-PSPL) for indexing speech information," in *Interspeech*, 2007.

[29] B. Ma and H.-Z. Li, "A phonotactic-semantic paradigm for automatic spoken document classification," in *SIGIR*, 2005, pp. 369–376.

[30] R. W. White, "Evaluating implicit feedback models using searcher simulations," *ACM Transactions on Information Systems*, vol. 23, no. 3, pp. 325–361, 2005.

[31] http://www.galaxy.csail.mit.edu/lectures/.

[32] Y.-C. Hsieh, Y.-T. Huang, C.-C Wang, and L.-S Lee, "Improved spoken document retrieval with dynamic key term lexicon and probabilistic latent semantic analysis (plsa)," in *ICASSP*, 2006.

[33] S.-L. Chuang and L.-F. Chien, "Taxonomy generation for text segments: A practical web-based approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 4, pp. 363–396, 2005.

[34] Y.-C. Pan, J.-Y Chen, Y.-S Lee, Y.-S Fu, and L.-S Lee, "Efficient interactive retrieval of spoken documents with key terms ranked by reinforcement learning," in *Interspeech*, 2006, pp. 1577–1580.

[35] Y.-C. Pan and L.-S. Lee, "Simulation analysis for interactive retrieval of spoken documents with key terms ranked by reinforcement learning," in *1st International Workshop on Spoken Language Technology (SLT)*, 2006.

[36] S. E. Robertson, "On term selection for query expansion," *Jounal of Documentation*, vol. 46, pp. 129–146, 1990.