

WebTalk: Mining Websites for Interactively Answering Questions

Junlan Feng

AT&T Labs – Research
junlan@research.att.com

Srihari Reddy

Johns Hopkins University
srihari@jhu.edu

Murat Saraçlar

Bogazici University
murat.saraclar@boun.edu.tr

Abstract

This paper investigates whether a customer care dialog system can be built automatically by mining and leveraging the wealth of information on a company's website. Our objective is to allow the users to ask FAQ-like questions which may request a specific piece of information, an analytical answer, or a transaction such as checking the status of the payment. We developed an infrastructure, referred to as WebTalk, that constructs different application-specific dialog systems by taking different websites as input. In this paper, we overview the involved technologies in WebTalk, address the challenges that will be shown very different from those in traditional dialog systems, and describe our efforts to approach these challenges. We present an evaluation for one WebTalk component: website based question answering.

1. Introduction

Spoken dialog systems provide a cost effective solution for call center and helpdesk automation. There are a number of successfully deployed dialog applications [8,9]. It is important, however, to be aware of the limitations of the underlying technologies. Most commercially available dialog systems are finite state based, in which the user is taken through a dialog consisting of a sequence of predetermined steps or states. This form of dialog control is suitable for simple and well-structured tasks. The major disadvantage is that it makes the resulting system less flexible. Moreover, designing a dialog flow involves a great deal of human effort. For a complicated task, it is difficult if not impossible to predict all the potential conversational states.

Due to these restrictions and the required high developing cost, only few of the large companies today take advantage in deploying spoken dialog systems for their customer care. By contrast, millions of companies invest in developing and maintaining customer-oriented websites. By the end of 2004, the total number of live .com domains was at a record high of 20 million [13]. A company website is typically professionally designed in order to attract visitors, serve customers, and meet the company's business needs. It contains general information about the company, the products and services the company provides, as well as a large variety of e-commerce or customer care applications. The goal of WebTalk, which was heavily motivated by such wealth of information on company websites, is to automatically provide a conversational interface to the underlying website content. It aims to automate the process of building spoken dialog systems.

Currently, the users of a website obtain information and services by reading the web page content, navigating hyperlinks, searching keywords, and filling out HTML forms. A website may consist of hundreds, thousands, or even millions of web pages, each of which assembles heterogeneous content such as menus, tables, images, forms and different formats of text. It is difficult to locate a piece of wanted information or a wanted transaction form from this large collection. Complex websites ease this frustration by providing a site search engine which accepts keywords and returns a ranked list of links to relevant web pages. However, identifying the relevant piece of information from a list of web pages is still a tedious and time consuming task. This problem has been addressed by the question-answering (QA) literature [10], where the attempt is to respond to users' natural language questions with exact answers rather than a list of documents. Several companies have deployed company specific question answering agents on their websites [11, 12] as a relatively more natural way to interact with customers. The obvious limitation of a customer-oriented QA system is that many questions that customers want answers for can not be satisfied with a simple answer. Some of these questions inherently are initiations for dialogs. Some of them are too complicated, broad, narrow, or vague resulting in lack of a good answer or many good answer candidates. Sub dialogs like disambiguation and relaxation are needed in these cases. Furthermore, in real world, users ask questions naturally as part of contextualized interaction. For instance, a question "Could you tell me more about AT&T CallVantage service?" is likely to be followed by "How does it work?". It's unrealistic to limit users to only submit isolated questions. For all these situations, a dialog interface is more suited than a QA machine.

The rest of this paper is organized as follows. In the next section, we describe the components of WebTalk and address the feasibility and challenges for website-based QA and website-based dialog managing. In Section 3, we describe our experiments and present the experimental results. We conclude this paper in Section 4.

2. Architecture of WebTalk

Figure 1 shows a schematic diagram of the six major technology components in WebTalk including Website Understanding, Automatic Speech Recognition (ASR), Question Answering (QA), Dialog Manager (DM), Language Generation (LG), and Text-to-Speech synthesis (TTS). The website understanding component translates a website into Task Data in a form that can be used to configure the dialog modules. We expect this framework to support automation for building spoken dialog systems from a given website.

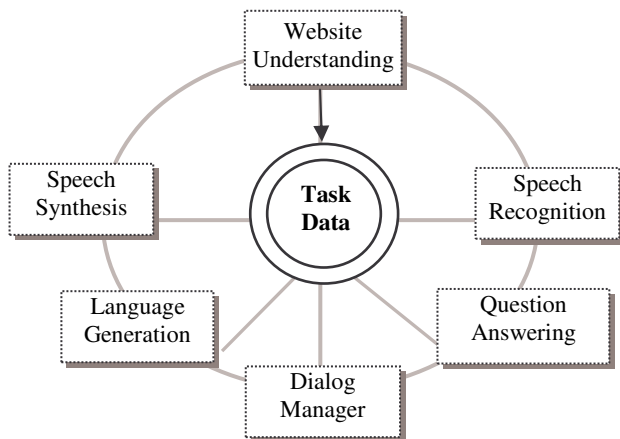


Figure 1: A schematic diagram of major technology components in WebTalk

Among these components, ASR, LG, and TTS are speech related. The task of ASR is to convert the user's input utterance into a sequence of words. The LG component constructs the message that is to be spoken to the user. The role of the Text-to-Speech component is to translate the text response into spoken form. In WebTalk, all these components have to be automatically customized by website data. This requirement poses a number of new challenges including constructing language models that reflect human-machine interaction by using only the web data, designing a language generation component in a manner that can intelligently present the system's knowledge state, and building a good quality web-based task-specific speech synthesis component. Each of these challenges is an interesting research area in its own right. In the following, we focus on the rest components that could constitute a chat-based dialog system.

2.1 Website Understanding

Dialog technologies currently do not operate as an interface to find information from large document collections. Website Understanding is a component to automatically convert the website content into a more structured format that would allow dialog modules to exploit. In our solution to this task we employed learning mechanisms to build a web page semantic structure parser that segments each individual web page into smaller semantic units, implemented a website mining tool that extracts structured task knowledge from the underlying website content, and built a technique to create summaries for website directories.

Webpage Parsing. Information conveyed on web pages is carried not only by their stream of texts, but also by the semantic structure of these pages, which are implicitly encoded in web documents. We formulate discovering web page semantic structures as a task involving web page segmentation - segmenting a web page into smaller information blocks and information block classification - identifying the semantic categories of these smaller information units. An information block is defined as a coherent topic area according to its content or a coherent functional area according to its associated behavior. We defined 12 semantic categories for classifying web page information blocks including *Page-Title*, *Form*, *Table-Data*, *FAQ-Answer*, *Menu*, *Bulleted-List*, *Heading*,

Heading-List, *Normal-Content*, *Heading-Content*, *Picture-Label*, and *Other*.

Web page segmentation is to group text nodes on a web page into a sequence of information blocks. We cast this task as a binary classification problem. For each pair of contiguous text nodes, we build a set of features to represent the distance and difference between them, and then classify this feature set into the information block boundary class or the non-boundary class. These two nodes are separated into two information blocks, if a boundary is identified between them. Webpage information block classification is a 12-class classification task. We employed two machine learning algorithms, Adaboost and SVMs, to build these classifiers from a labeled web page corpus. Experimental results have been reported in [1].

Website Data Mining. The second task for Website Understanding is to extract structured task knowledge, including products and services that the company provides, properties of these products and services, corporate contact information, as well as acronyms the website uses. Structured task knowledge would facilitate the QA and DM components to more precisely respond users' requests. We developed a boosting algorithm to extract products and services and implemented a set of rules for extracting other entities. The evaluation of these efforts will be reported in future work.

Website Structure Understanding: Web pages on a company website are often systematically organized into subdirectories and are linked to each other through meaningful hyperlinks. Most web pages have meaningful page titles. We built a technique to make use of these clues to create summaries for website directories.

In summary, the website understanding component outputs four types of data: semantic text data units, transaction forms, structured task knowledge as well as website directory summaries. Text data units including information blocks falling into the categories like *Table-Data*, *FAQ-Answer*, *Bulleted-List*, *Normal-Content*, and *Heading-Content* have the potential to be answers for some customer requests. Transaction forms, a category of webpage semantic units, are the entries to the online transactions. Structured task knowledge is the output of the website data mining module and is represented in XML grammars. Website directory summaries are explored by the dialog manager as help prompts.

2.2 Website-Based Question Answering

We incorporate a QA component into WebTalk, which takes a natural language question and dialog context as input and finds a number of answers from the Task Data. Dialog manager prepares appropriate dialog context and determines the way to negotiate with the user based on the returned answers from the QA component.

The QA process consists of five stages, namely question parsing, question classification, query formulation, information retrieval, and answer extraction. Question Parsing labels the posed question with part-of-speech tags, general named-entity tags and product and service entities that are specific to the company and are extracted in advance by the Website Understanding component. The second module is a question classifier, which categorizes the question into five categories - *Generic Information Request*, *Problem Reporting*, *Factoid Questions*, *Transaction Request*, and *Information Search*. The third module called query formulation transforms a natural language question

and the dialog context into a set of query terms. The fourth module is an answer retrieval engine which takes a query as input and returns a list of answer candidates deemed to be relevant to the query in a ranked manner. The fifth module does answer extraction by checking the ranked list of answer candidates. It selects those candidates with confidence scores higher than a predefined threshold, containing the names of products or services mentioned in the question, and having entities or structures matching the question type. If none of them meets these conditions, the system returns an empty answer. The QA performance of WebTalk built on the above technologies had been evaluated in [2] and was shown to be comparable with a handcrafted company specific question answering system.

We recently focused our research on improving the query formulation module and the answer extraction module using statistical approaches. This effort is mainly motivated by the following two observations:

(1) Some words are more likely than others to co-occur in a pair of question and answer. For instance, the word “discount” is more likely, in general, to be a QA common word than the word “give”.

(2) There exists a lexical and stylistic gap between questions and answers. For instance, a question containing the phrase “travel” is likely to be paired with an answer containing words like “airline”, “flight” and “reservation”.

The approach we propose attempts to learn the lexical relevance between questions and answers from a large corpus of question-answer pairs. We prepared this corpus through mining answered FAQs (Frequently Asked Questions) from the World Wide Web. More specifically, our learning procedure employs the perceptron algorithm [14] to estimate the weights of a linear model. An absolute 27% improvement in answer accuracy was observed over a baseline of 41% that used an IR model relying on exact word matches. The detailed experiments will be presented in Section 3.

2.3 Dialog Managing

According to the nature of the website data, we characterize the dialog tasks into three categories:

A Dialog interface to Question Answering: The QA component finds answers for a posed natural language question from the text data units. Dialog ability is needed when (1) the question is not clear; (2) the question is posed as part of a contextualized interaction rather than in isolation; (3) the QA component finds multiple good answers which have similar high answer confidence scores; (4) the QA component doesn’t have a good responsive answer but finds multiple moderately relevant answers; (5) the QA doesn’t have any relevant information. The purpose of the dialog management is to keep appropriate dialog context, allow the user to narrow down or expand the answer space by soliciting more information from the user, or to allow the user to navigate the answer space. Some of these challenges have been addressed in the QA literature [5] [6].

Form-Filling Dialog: Applying spoken dialog interfaces to online forms and response pages will enable enormous number of internet-based services to become available over the phone.

A Dialog interface to structured task knowledge: Based on the structured data returned by the website understanding component, more sophisticated sub dialogs such as making

suggestions, clarification, and disambiguation could potentially be initiated by the dialog manager.

To date we have implemented a dialog manager that is capable of supporting task independent discourse dialog, preparing dialog context for the QA component, allowing the users to navigate through the answer space and providing context-based help when the QA component fails to fetch an answer.

3. Experiments

In the following, we evaluated statistical approaches to website-based question answering. The dataset we used in the experiments consists of 52,449 FAQ-answer pairs (q, a) collected from 1800 company websites. The system ranks the answers in a website according to a question-answer relevance measure $f(q, a)$. The training procedure seeks to maximize the number of times the correct answer is retrieved as the most relevant, namely

$$f(q_i^{(n)}, a_i^{(n)}) > f(q_i^{(n)}, a_j^{(n)}), i \neq j, i, j = 1, \dots, N^{(n)} \quad (1)$$

where $a_i^{(n)}$ is the given correct answer for $q_i^{(n)}$ on the n^{th} website and $N^{(n)}$ is the total number of FAQs on the n^{th} website.

In testing we use answer accuracy as the evaluation metric that is defined as:

$$\text{accuracy} = \frac{\text{number of times the correct answer is ranked first}}{\text{total number of questions in the test set}}$$

In the experiments, we used 10-fold cross validation with this dataset.

We begin with the standard document retrieval algorithm *tf.idf* as our baseline, where q and a are represented as vectors of word frequencies \vec{q} and \vec{a} . The similarity between \vec{q} and \vec{a} is the cosine distance between them:

$$f(q, a) = \frac{\vec{q} \cdot \vec{a}}{\|\vec{q}\| \cdot \|\vec{a}\|} \quad (2)$$

This relevance measure between a question and an answer relies on their common words. It performs with 42% answer accuracy as shown in the first row in Table 1.

We first attempt to improve this performance by query expansion. From the training set, we learn a mapping between each query word w and its correlated answer words $\{v\}$. The correlation is measured by mutual information:

$$I(w, v) = H(p(v \in a)) - p(w \in q)H(p(v \in a | w \in q)) - p(w \notin q)H(p(v \in a | w \notin q))$$

where $H(\cdot)$ is the entropy function. :

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$$

We denote the top N correlated answer words for the word w as $w_{\text{exp}}^{(N)}$. In our experimental setting, N is set to 20. In testing, we expand each query word w to the words $v \in w_{\text{exp}}^{(N)}$. The similarity between the expanded \vec{q} and \vec{a} is also measured using the cosine distance. As given in Table 1, with this query expansion the answer accuracy rose 2% from the baseline.

Our second effort is towards more significantly improving the performance using machine learning techniques. In our

approach, we applied a general linear model [14] to this QA task:

$$\text{Answer}(q) = \arg \max_{y \in \text{GEN}(x)} \phi(q, a) \cdot \bar{\lambda} \quad (3)$$

where GEN is a function enumerating a set of candidate answers for a given question q ; ϕ maps each question answer pair (q, a) to a d dimensional feature vector $\phi(q, a) \in \mathbb{R}^d$; $\bar{\lambda}$ is a d dimensional parameter vector. In our task, for a given question $q_i^{(n)}$, GEN returns all the answers from the same website, namely $\{a_j^{(n)}, j = 1, \dots, N^{(n)}\}$. The

training process is to learn the parameter vector $\bar{\lambda}$ from the training examples, which in our case are question answer pairs. In the experiments, we employed the perceptron algorithm [14] to learn the vector $\bar{\lambda}$. In testing, the returned answer is the one maximizing the equation (3).

Based on a variety of intuitions and related previous work in the question answering community, we extracted the following features to constitute the feature vector $\phi(q, a)$:

- (1) The number of common n-grams between the n^{th} sentence of the answer and the question.
- (2) Binary word match features, which indicate if a word occurs both in the question and the answer.
- (3) *tf.idf* valued word match features, which are defined as

$$f(q_i^{(n)}, a_j^{(n)}, w) = \begin{cases} tf(w, a_j^{(n)}) \cdot (idf(w)^{(n)})^2 & w \in q_i^{(n)}, w \in a_j^{(n)} \\ 0 & \text{otherwise} \end{cases}$$

where $tf(w, a_j^{(n)})$ is the term frequency of the word w in the answer $a_j^{(n)}$ and $idf(w)^{(n)}$ is the inverse answer frequency of the word w in the n^{th} website.

- (4) *idf* valued query expansion features, denoted as:

$$f(q_i^{(n)}, a_j^{(n)}, w, v) = \begin{cases} (idf(w)^{(n)})^2 & v \in w_{\text{exp}}^{(N)}, w \in q_i^{(n)}, v \in a_j^{(n)} \\ 0 & \text{otherwise} \end{cases}$$

We conducted two experiments with this framework. In one experiment, we exploited the first three types of features and achieved 54% answer accuracy with the learned weights $\bar{\lambda}$. In the second experiment, we used all the four types of features above. This increased the answer accuracy to 68%, a 27% rise from the baseline. This shows that the query expansion feature makes a big contribution. Table 1 summarizes the experimental results.

4. Summary

This paper describes WebTalk, a general framework for automatically building customer care dialog applications from given websites. The goal is to enable companies, which have already set up their websites, to extend their customer service with a spoken dialog interface whether over the phone or through the Internet.

In this paper, we addressed the challenges confronting WebTalk components, presented our efforts on implementing website understanding, interactive question answering and dialog manager, and provided the evaluation of a learning approach for

Model	Accuracy	Improvement
<i>tf.idf</i> vector space model	41%	-
<i>tf.idf</i> vector space model with query expansion	43%	2%
Linear model without query expansion features	54%	13%
Linear model with query expansion features	68%	27%

Table 1: Experimental results: answer accuracy

website-based question answering. The learning procedure uses the perceptron algorithm to estimate the lexical relevance between questions and answers. The dataset consists of 52,449 FAQ-answers pairs which were crawled from a variety of business websites. In our experiments, we used 10-fold cross validation with this dataset. An absolute 27% improvement in answer-finding accuracy was observed over a baseline of 41% that used an information retrieval model relying on exact word matches.

5. Acknowledgements

Part of the work presented in this paper was done while the second author was visiting AT&T Labs – Research and the third author was with AT&T Labs – Research.

6. References

- [1] J. Feng, S. Bangalore, M. Rahim, “WebTalk: Mining websites for Automatically Building Dialog systems”, Proc. of IEEE ASRU 2003.
- [2] J. Feng, S. Bangalore, M. Rahim, “Question-Answering in WebTalk: An Evaluation Study”, Proc. of ICSLP-2004
- [3] R. Soricut, E. Brill, “Automatic Question Answering: Beyond the Factoid”, Proc. of HLT-NAACL 2004
- [4] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, Vibu Mittal, “Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding”, Research and Development in Information Retrieval, pages 192-199
- [5] Andrew Hickl, John Lehmann, John Williams and Sanda Harabagiu, “Experiments with Interactive Question Answering in Complex Scenarios”, Proc. of HLT-NAACL 2004
- [6] Sharon Small; Ting Liu; Nobuyuki Shimizu; Tomek Strzalkowski, “HITIQA: An Interactive Question Answering System: A Preliminary Report”, Proc. of the ACL 2003 Workshop on Multilingual Summarization and Question Answering
- [7] G. Salton and M. McGill, “Introduction to Modern Information Retrieval”, McGraw-Hill, 1983.
- [8] Michel F. McTEAR, “Spoken Dialogue Technology: Enabling the Conversational User Interface”, ACM Computing Surveys, Vol. 34, No. 1, March 2002, pp. 90-169
- [9] A.L. Gorin, B.A. Parker, R.M. Sachs and J.G. Wilpon, “How May I Help You”, Proc. of IVTTA 1996
- [10] E. M. Voorhees, “Overview of the TREC 2002 Question Answering Track”, Proc. of TREC 2002
- [11] <http://www.sbc.com/>
- [12] <http://www.att.com/>
- [13] <http://www.verisign.com/>
- [14] M. Collins, “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms”, Proc. of EMNLP 2002.