

Putting Your Customers' Questions to Work

Brian Ulicny

Jeeves Solutions
24 Prime Park Way, Natick MA 01760
bulicny@jeevessolutions.com

Abstract

The JeevesOne system is described which enables companies to set up and maintain question-answering functionality for structured and unstructured data on their own web sites.

What Do Our Customers Want?

Jeeves Solutions provides enterprise software for companies to construct and maintain question-answering functionalities on their own web sites.

Our customers want to be able to answer end users' questions online and automatically. Because the customer interaction is automatic, it saves the customer the costs of hiring, training and retaining a customer service representative. But just putting all the answers to known customer questions in a searchable place on the company web site isn't enough. Search too often stinks, as Forrester Research put it. What our customers want is this: our customers want accuracy in the information returned to end users in response to their questions; they want the site to provide a brand-enhancing experience to the end user in guiding the user to the information they need; they want easy ways to discover what information that users are requesting is lacking from their web site; they want the software to integrate with other systems already in place for customer relationship and content management; they want to be able to determine what is on their end users' minds by analyzing their queries; they want to service all of their global end users across languages; and they want to be able to escalate a problem to a live customer service representative automatically when the end user is not finding the information they need.

Very importantly, our customers want to control both the content of the information provided to end users and control the software and data that guides the end user to the requested information. However, they do not want to maintain a staff of computational linguists to achieve this.

What is JeevesOne?

JeevesOne is a family of products that allow customers to guide their end users to the information they need. I will focus here primarily on JeevesOne Enterprise, JeevesOne

Enterprise leverages natural language processing as an effective entry point into a wide range of existing data structures, enabling users to ask questions and receive a comprehensive answer to a series of related tasks. Information from nearly any source, including legacy systems, enterprise applications, Web services and Internet resources can be accessed and interrelated in a unified "View", described below.

Initially released in September, 2001, JeevesOne customers include Ford Motors, F5, High Jump Software, the State of Washington, Novartis, and Nestle Purina PetCare, among others representing a variety of vertical markets.

Natural Language Processing in JeevesOne

Natural language processing is employed both online in JeevesOne, to answer end user queries, and offline, to produce the knowledge layers that underlie the question-answering process. In what follows, we'll briefly describe the cycle from online question-answering to offline knowledge-layer creation and maintenance.

Online NLP in JeevesOne

The online question-answering system of JeevesOne is based on a set of templates, originating in Jeeves Solutions Knowledge Packs or produced by the JeevesOne customer via a graphical user interface (GUI).

Each template consists of a dynamic question text, a set of one or more left-hand side patterns and a set of one or more right-hand side commands. Technically, the pairings of left-hand sides and right-hand sides constitutes a variant form of finite-state transducers. It is well known that Finite-state techniques for processing natural language are robust and efficient, as well as easily maintainable and extensible.

The online processor examines the result of tokenizing the user query. The tokenizer segments the user's query into tokens by examining the user query character by character, while consulting the JeevesOne dictionaries, character-level patterns for items such as numbers, file

types and email addresses, and user-derived misspellings and variant forms (e.g. Jello for Jell-O™). Tokens may include whitespace or have non-whitespace boundaries. For example, “New York” is a single token, and (wo)men might be taken to correspond to two tokens.

The JeevesOne dictionaries consist of general purpose dictionary entries for eight major languages, English, Spanish, French, German, Italian, Dutch, Swedish, and Portuguese (Brazilian). There are also vertical industry dictionaries available for the pharmaceutical, automotive, banking, brokerage, insurance, hardware and software industries. The customer can customize local dictionaries particular to that company. (Additionally, the JeevesOne engine can provide basic keyword searching and indexing for any single-byte or UTF-8 encoded language. The JeevesOne engine also provides language recognition and filtering for 34 languages.)

JeevesOne uses dictionary-based rather than algorithmic stemming techniques like the familiar Porter Stemming Algorithm, widely used in other search engines. Stemming, of course, enables the expansion of searches to match root and inflected forms of keywords (e.g. ‘buy’ and ‘bought’, ‘buying’, ‘buys’). Dictionary based stemming results recognize actual words and concepts in context, while the Porter Stemming Algorithm rigidly shaves off common inflections (such as “ing”, “s” or “ly”, transforming “Notting Hills” to “Not Hill”).

The matcher matches incoming tokenized and stemmed user queries against left-hand side patterns, which denote sequences of tokens, any of which may be optional or required. These sequences are sensitive to token ordering and may require that the entire sequence match the user query or a subsequence. The matcher allows patterns over word groups, thus allowing a compressed notation for matching against synonym groups (words that denote the same thing) and hyponym groups (words that denote the same kind of thing) as well as subsequences at the token or character level. The pattern-matching syntax is expressive enough to match a wide range of queries with a compact expression.

The right-hand side passes information from the user query that has matched a pattern and formulates it into a command in the internal JeevesOne command language. For Search commands, this language provides term weighting, word group expansion, term stemming, and a variety of Boolean operators including and, or, soft and, must include, must not include, term boost, and term discount. Question templates with search commands thus allow a matched user query to be mapped to a dynamically generated, optimized search command in which terms have been expanded or contracted, weighted in various ways, or substituted as appropriate. For example, a Search

command for a pharmaceutical site might match queries about whether certain drugs are appropriate for children, and add the stem ‘pediatric’ in order to match documents that speak of ‘pediatric uses’ rather than ‘children’.

For commands other than search commands, word group expansion, stemming, and various conditional substitution operations can also be performed. For example, it may be necessary to transform a user’s query in terms of a city into the corresponding three-letter airport code in order to query a back-end database.

Once all of the transformations of the user’s query into the commands specified in the right hand side are complete, these commands are then executed by the query engine; the results are collated and scored and displayed to the user as detailed above. Scores for results are based both on the extent to which a query matches a template left-hand side, and how well a document or passage matches the right-hand side command.

The question text uses elements of the user’s query to dynamically echo the system’s understanding of the user’s intention. Functionality to support agreement, elision and capitalization in the question text has been enabled. The dynamic question text allows multiple possible interpretations of the user’s query to be disambiguated; allowing the user to select the results that correspond to the desired intention. Results are grouped under the instantiated question text for a template. The question text functionality has the ability to make agreement corrections in the text to insure grammaticality. For example, a user query “open a account” can be echoed correctly as “How do I open *an* account?”

JeevesOne’s dictionary-based spelling checking allows for contextual spell correction. A user’s query for “soup receip” may be echoed as “Do you have any soup recipes?” whereas “sales receip” may be echoed as “Where is my sales receipt?” This also bridges the gap between end user usage (e.g. “Jello”) and customer content (e.g. Jell-O™ Brand Gelatin).

The right-hand side passes information from the user query that has matched a pattern and formulates it into a command. These commands can be Search commands, or URL commands, SQL commands, Exact Answer commands, or EIS (Enterprise Information System) commands. (URL commands represent the starting point of Jeeves Solutions technologies, in Ask Jeeves. The early Ask Jeeves paradigm constrained human editors to map patterns matching user queries to URLs that were judged to contain the answer. This approach was obviously not very automated. The various Ask Jeeves web properties do not rely on this paradigm today.) On the right-hand side, synonym groups are expanded to match documents that contain any synonym. Other

functionality allows the right-hand side to look only for the element of a word group that the user wrote or a preferred term (the stem term). Furthermore, terms can be introduced on the right-hand side that don't occur in a user's query but are useful for identifying content. This may be done conditionally, to allow the right-hand side to add a certain term only if a certain word-group is found. For example, the right-hand side can say "look for pages that contain the term TDD (telecommunication devices for the deaf) if the user queries on deaf or its synonyms".

In the case of URL commands, parameters can be dynamically added to URLs based on the user's query. Thus, a CGI-argument may be appended to a URL based on the user's query. For example, in a movie search application, the title of a movie in a user's query (an element of a hyponym group) might be appended to a URL command, e.g. <http://www.mymoviesearch.com?Title=userquerytitle>. The result set returned would then include the page corresponding to that CGI command

Exact Answer from Unstructured Data

Certain commands in Jeeves One retrieve just a specific piece of information, rather than a document, in response to a question. We will discuss retrieving exact answers from structured data below.

Text Exact Answer commands are the command service in JeevesOne that most closely corresponds to the TREC question-answering paradigm. Exact Answer question templates specify patterns that specify patterns for matching end user questions on the left hand side and one or more exact answer commands on the right hand side. Exact Answer commands specify that a specific passage from a document that should be presented as the result (with informative highlighting and hyperlinking to the target passage of the source document) if it meets constraints specified in the command.

For example, a mileage Exact Answer template might be designed to match mileage queries such as "What kind of mileage does the Mini Cooper get?" The corresponding command would then look for passages in the document repository that meet certain constraints. Some of these constraints are constraints on terms that must or should not occur in the document as a whole. Other constraints are constraints on the passage

More specifically, JeevesOne Exact Answer commands require the specification of a certain target term(s) that constitutes the answer (e.g. a number) and which will be highlighted in a distinctive way. One must also specify a term that must appear near the target item ("miles per gallon" or its synonyms). One must specify terms that

must appear in the document, to be considered (e.g. the model of the car). Finally, one can optionally specify terms that are desirable on such pages (e.g. "specifications", "EPA", "highway", "city"), and terms that are undesirable on such pages (e.g. "concept car").

An obvious limitation of this approach is that the system presupposes that the type of each answer target can be specified in advance. In some cases, this can be done via a wordlist that would be necessary anyway (e.g. the list of vehicle models on an automotive site). In other cases, the target can be specified as matching a certain character-level pattern (e.g. a numeric regular expression). In other cases, it may be difficult to specify all of the target terms; for example the list of all possible chemical compounds that would answer "What is the active ingredient in Drug D?"

The system also allows one to specify that the answer will be a list, for example a list of colors in which a car model is available. These lists are not collated from multiple passages, but must occur contiguously. However, multiple Exact Answer results for a template are possible. Similarly, yes/no questions will be answered with an affirmative instance.

This approach to online question-answering is somewhat conservative; it says that an exact answer should not be specified unless the components of such an answer can be predicted. In practice, many limitations of this type can be overcome by specifying a series of exact answer commands that back off from the original request.

Consider for example, the question "Do Snickers bars contain peanuts?" To approach this question in the most flatfooted way, within the JeevesOne Exact Answer paradigm, the Exact Answer command would constrain the engine to look for a document that contained 'Snickers' and had 'peanut' or its variants close to 'ingredient' or its variants. This is fine if it does contain peanuts, but what if it doesn't? In anticipating such cases, the system administrator should specify an additional exact answer command that looks for ingredients on documents that contain 'Snickers'.

From the customer's perspective, answering such questions should be done in the most 'brand-enhancing' way. In order to do this, one must look beyond the specific question asked, especially if it is asked frequently enough, to its purpose, or presuppositions.

In noticing many questions about peanut ingredients, the savvy JeevesOne administrator will take a step back, at this point, and realize that a question about peanut ingredients can have a very serious purpose. Many children now suffer from potentially very serious allergies

to peanuts and their oils. So, the most appropriate way to answer the user's question is not only to simply identify passages that answer the user's question, but to acknowledge the user's possible concern about allergies and link to information that specifically addresses that concern via a URL command attached to that question template. That is, the question template that matches variants of "Does x contain (potential allergen) y" should contain a command to bring up relevant allergy information, an Exact Answer command designed to fire in positive instances (when the allergen is among the ingredients) and a separate Exact Answer command designed to fire in negative instances (to verify to the end user that the product contains none of the allergen.)

This example shows that it is necessary to think about the purposes behind the queries that users ask and how best to serve those purposes, as well as the company's own purposes, in designing an online question-answering system for online self-service.

Exact Answers from Structured Data

JeevesOne facilitates exact answers from structured data via two command services, SQL commands and EIS commands.

In JeevesOne Standard as well as Enterprise, SQL Command services are provided to transform the matched tokens of a user query into terms in an SQL query. The various WHERE clauses of the SQL query may be made contingent upon the existence of corresponding terms in the user query or provided with default values. Text transformations and normalizations at this level can be accomplished via customized stem groups.

In JeevesOne Enterprise, an entirely new level of access to structured data as answers to end user questions is provided via Enterprise Information Services (EIS), using technology acquired from Octopus, Inc.

JeevesOne Enterprise provides an Enterprise Information Services (EIS) component dedicated to connecting to and combining information retrieved or "called" from multiple data sources. The EIS component serves as a gateway to tie enterprise applications, Internet resources, and web services to the search solution.

The product currently integrates with Siebel, RDBMS databases, public web sites such as Yahoo Finance and UPS Tracking. WebMethods and SAP coming soon. JeevesOne product includes an entire API for JeevesOne customers to create their own data integration. Jeeves

Solutions also offers Jeeves Solutions Professional Services or partner staff to build custom data integrations.

The EIS component is fully J2EE compliant and uses a 3-tier architecture to promote separation of logic and reusability of code.

EIS Commands link end user questions to EIS 'Views'. Views are pre-defined, dynamic templates that consolidate information elements from relevant data sources. That is, Views are an intelligent dashboard of information pulled together to answer a specific question accessible via a GUI.

Views are easily created by application administrators using a visual drag and drop method from an element catalog. Views are accessed when a user asks a question, or even from primary navigation if that is preferred.

A View is more than disparate data on a single screen; it shows interrelated data. What does this mean? Let's use an example to illustrate the functionality.

A user asks a question such as "Has my order shipped?" The questions would launch a View that contained a status of, say, "order shipped" with a shipping number pulled from the customer's back-end SAP system, perhaps by passing the end user's id from a stored cookie to the command service via JeevesOne's pass-to-runtime functionality. The shipping number would then fire a FedEx inquiry that reported on where the shipment was sent. Finally, the "ship to" address would fire a map from mapquest.com. The result is that the user quickly and easily is told the precise status of his or her order without having to interact with several separate systems to get the same information.

Offline NLP in JeevesOne

Offline, user queries are mined to formulate the knowledge layer of the online system.

In the first instance, JeevesOne uses an iterative covering algorithm to process huge numbers of user queries to produce candidates for Knowledge Pack word groups and left-hand sides automatically. A set of word groups and left-hand sides for a given domain of questions is produced to discover variants of the same questions and to cover the most frequent question patterns. JeevesOne Knowledge Packs are designed to cover the most frequent queries in a given domain, based on query logs from our enterprise customers and general purpose Web properties (Ask Jeeves, AJ Kids, AJ UK, and Teoma). User queries are known to follow what is known as a Zipf or Bradford

distribution: the frequency of a question and its variants is inversely proportional to its rank. Informally, this is known as the 80/20 rule: a small number of possible cases cover a large proportion of the actual uses.

In addition, JeevesOne uses a proprietary method to discover significant n-grams in the customer text that is indexed that are not in the dictionary. These phrases are not just the most frequent sequences (“of the”, “to the”, “and the” being frequent but uninteresting example). JeevesOne uses statistical and linguistic techniques to determine linguistically significant phrases automatically, such as “XML development tool” or “online content management system”. These candidates can then be reviewed and efficiently added to the dictionary, to improve pattern matching and document retrieval accuracy.

JeevesOne system administrators can maintain their online dictionaries and word groups (synonyms and hyponyms) via a Web-based GUI. They may also specify new question templates in the Knowledge Editor. This component allows users to specify new question texts for a new template; then, new left-hand sides are generated by providing an example of the question pattern. A left-hand side pattern (or set of patterns) for this example is then generated via a component called the Fuzzifier. This allows the user to select among various levels of generality for pattern elements corresponding to a term: do you want ‘who is the CEO?’ to match only variants of that question, or all instances of ‘who is the <corporate officer>?’

An optimized right-hand side command for every template is generated automatically. These automatically generated commands may be overwritten with a command or commands specified directly in the JeevesOne command language.

Finally, JeevesOne uses sophisticated document structure recognition techniques to segment and weight documents that are indexed. These techniques allow JeevesOne to emphasize the significant parts of a web page (e.g. titles, lead paragraphs, and so on) in search over insignificant but frequent portions, such as copyright notices, navigation bars, and so on.

JeevesOne uses the lexical information encoded in the Knowledge Layer to stem and index the customer’s documents, which may be grouped into separate collections. JeevesOne’s indexing architecture is capable of handling hundreds of simultaneous requests per second, and performing searches through millions of indexed documents. This easily handles the largest Jeeves Solutions customers to date. The server architecture is linearly scalable, allowing the customer to add additional processing power to the JeevesOne installation as the

corpus of content and number of users grows. A “Controller /Worker” architecture allows a mix of servers, big or small, to be leveraged, allowing you to re-use and maximize existing hardware investments.

For dynamic content management systems an HTML jump page is constructed to programmatically access the unique qualifiers for each piece of content. This approach bypasses the dynamic presentation layer introduced by systems like Vignette and Broadvision.

Index updates can be scheduled to occur as frequently as necessary to accommodate quickly changing content. In addition the document collection API can be leveraged to insert and delete content dynamically based upon changes in the underlying content management system.

The product supports enhanced Incremental Indexing, allows partial snapshots or “snaplets” to be generated. This alternative process takes less time than a full snapshot by only indexing content that has changed, allowing the index to stay in tighter synchronization with the actual content.

How do Customers Monitor JeevesOne’s Performance?

Using an extensible data mart consisting of 4 fact tables, 9 dimension tables and 2 aggregation tables that offers 107 report views and 15 report templates, JeevesOne Analytics allows companies to measure the performance of the JeevesOne application and gather actionable feedback from customers to continuously improve their Web site and online business initiatives.

Administration Reports include: series of reports to track

- broken links
- spider duration
- average size of documents
- number of documents found.

Analytics Reports include:

- User session data (including session and user IDs), times, level of satisfaction, number of queries, number of counts, number of picks, browser type
- Query data, including query text, session IDs, times, picks, word counts, category, scores on matches and picks
- Match data, including times, scores, associated users and queries, documents
- Pick data, including times, ranks, associated users and matches, documents

- Concepts and keywords, including run dates, tokens, query counts, and frequency counts
- Knowledge, including tokens, stems, entities
- Content holes, including run dates, tokens, query counts, and frequency counts

JeevesOne also offers Knowledge Packs for vertical markets as a framework for creating a highly relevant question answering system. Each Knowledge Pack offers an expansive list of recognized vocabulary, pre-built questions and variations on common queries that help vertical markets like pharmaceuticals quickly customize JeevesOne for their sites. For example, hundreds of questions such as, "Are there any health risks associated with your medication?", "Are your medications covered by my insurance provider?", and "Where can I read about possible side effects?" come ready to deploy within the Knowledge Pack

JeevesOne Usability Evaluation

Jeeves Solutions commissioned eTesting Labs -- formerly ZD Labs -- to perform a study comparing the JeevesOne 2.0 -- a prior version-- self-service interface to traditional keyword search interfaces. The tests were done using a consumer financial site -- www.Prudential.com -- and a technical site -- www.BEA.com. For each site, 100 test participants were asked to perform four typical tasks using (a) the keyword search interface currently in use on these sites, and (b) the JeevesOne self-service interface installed on a mirror of these sites.

Here are two of the tasks:

Consumer site You have just opened a new online 401(k) account and are not sure of the maximum amount of money that you can contribute to your 401(k) for 2001. Using only the question box on the home page, find out the maximum amount of money you may contribute.

Technical Site: You need to install a copy of BEA WebLogic Workshop on a new developer's system. The developer is currently running a copy of Red Hat Linux 6.2, so you're not sure if Workshop is compatible with this older version of Linux. Using the question box, find out which operating systems Workshop supports.

The results of this study, conducted in April 2002, are summarized below:

Task completion in five minutes:

JeevesOne/Consumer Site: 98%

Keyword Search/Consumer Site: 73%

98% JeevesOne/Technical Site

95% Keyword Search/Technical Site

% positive responses to "How easy was it to find the answer to your question?"

84% JeevesOne/Consumer site

29% Keyword Search/Consumer Site

66% JeevesOne/Technical site

46% Keyword Search/Technical Site

% positive response to "How well do you feel this interface understands your questions?"

85% JeevesOne/Consumer Site

26% Keyword Search/Consumer Site

67% JeevesOne/Technical Site

50% Keyword Search/Technical Site

% positive responses to "Where you able to find the information you were looking for?"

94% JeevesOne/Consumer Site

65% Keyword Search/Consumer Site

88% JeevesOne/ Technical Site

77% Keyword Search/Technical Site

Median time to complete tasks in seconds (smaller is better)

150 JeevesOne/Consumer Site

229 Keyword Search/Consumer Site

121 JeevesOne/Technical Site

149 Keyword Search/Technical Site

Median page hits to complete tasks (smaller is better)

3.0 JeevesOne/Consumer Site

7.0 Keyword Search/Consumer Site

4.5 JeevesOne/Technical Site

6.5 Keyword Search/Technical Site

The study found that 63% of the test participants said that JeevesOne self-service interface would cause them to return and use this service to answer their technical questions in the future while only 47% of the test participants said the existing BEA search interface would cause them to return and use this service to answer their technical questions in the future.

The study found that 73% of the test participants said that JeevesOne self-service interface would cause them to return and use this service to answer their financial questions in the future while only 29% of the test participants said the existing Prudential Financial search interface would cause them to return and use this service to answer their financial questions in the future

The study found that the JeevesOne self-service interface allowed test participants to find the answers to their scenario searches in less time and with fewer mouse clicks, keystrokes, and page hits than the search interface sites that did not use the JeevesOne self-service interface.

Conclusion:

JeevesOne puts a wide range of functionality in its customers' hands to set up and maintain question-answering functionality on their own sites to meet their customers' and their own needs. The system provides highly accurate guided search and question-answering to user questions, drawing upon materials stored in a variety of structured or unstructured format. Through analytic feedback, a JeevesOne system administrator is always able to keep the system in tune with user queries without requiring any sophisticated knowledge of natural language processing. Templatic searching provides the ability to guide end users in a brand-enhancing way that customers require. The result is a system that provides great usability for customer self-service.

References

P. Hagen, H. Manning, and Y. Paul. *Must search stink?*
The Forrester report, Forrester, June 2000

eTesting Labs, 2002, JeevesOne Usability Study,
<http://www.jeevessolutions.com/technology/etesting/>

E. Voorhees, *The TREC-8 Question Answering Track Report*, **NIST Special Publication 500-246:**
The Eighth Text REtrieval Conference (TREC 8)

D. Warthen, 2000, *Grammar Template Query System*,
WIPO Application , WO 00/57302