Automated FAQ Answering: Continued Experience with Shallow Language Understanding

Eriks Sneiders

Department of Computer and System Sciences
Stockholm University / Royal Institute of Technology
Electrum 230, S - 164 40 Kista, Sweden
eriks@dsv.su.se

Abstract

The subject of this research is development of an evolving automated FAQ (Frequently Asked Question) answering system that provides pre-stored answers to user questions asked in ordinary English. The natural language processing technique developed for FAQ retrieval does not analyze user queries; instead, analysis is applied to FAQs in the database long before any user queries are submitted. Thus, the work of FAQ retrieval is reduced to keyword matching without inferring; the system still creates an illusion of intelligence. Additional research is done in order to process phrases.

The system is designed for ordinary websites such as those belonging to university laboratories, software developers, etc.

Introduction

People sitting in front of their computers expect quick solutions. People browsing a website want to get quick answers to their questions. In order to enable the latter, an evolving WWW-based automated FAQ answering system, which provides pre-stored answers to users' questions asked in ordinary English, has been developed.

In an FAQ collection, the information supplier tries to answer in advance typical questions that the information users may have; FAQs are intended to solve certain problems. Traditionally such a collection is organized as an ordinary list, which has several deficiencies:

- *An FAQ user is not given a chance to ask any questions.* Instead, the user is forced to scan through a long list (or several lists) of various questions in order to find a question, similar to the user's own question, which may not exist in that list.
- *The information supplier does not know the actual questions that arise.* Rather, the information supplier answers possible questions in advance. Nonetheless, these possible questions do not always satisfy the users' needs.
- *FAQs in a list may be poorly organized.* If the number of FAQs is large and their order chaotic, then there are two options of navigation through the list: (1) to read all the questions or (2) to search for keywords by free-text substring search facility. It is, however, not possible to use substring search if the list is spread over several documents. It is an advantage if FAQs in a list are semantically grouped. But even in this case the grouping may be ambiguous, and a user may not know where exactly to look for a particular FAQ.
- *There may be several FAQs in a list that answer the user's question.* Some more FAQs may be related to the question. If the list is not well structured, a user has to scan through the whole list in order not to loose possibly valuable information.
- *An FAQ list may be too long, sometimes scattered over several documents.* It is difficult to locate a small piece of information in a large text mass. Often people either easily find what they want or give up.

These deficiencies pertain to FAQ lists whatever medium carries them – WWW, Usenet newsgroups, CD, paper, etc. An FAQ answering system overcomes them by retrieving FAQs upon a request expressed in natural language, and by storing the request for further analysis.

In order to give an idea of the system's functionality, an example of asking a question follows (Figure 1).

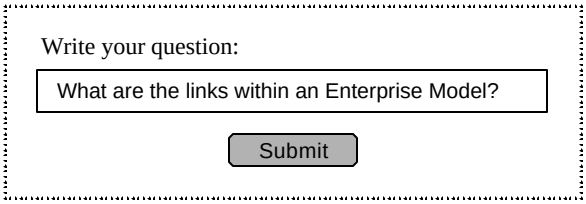

Figure 1 Example of a user question to the FAQ answering system.

The subject of the sample question is one of the techniques of Enterprise Modelling (Bubenko 1994), which is a general business planning methodology. By using a Web-browser, the user submits his or her question to the system. The

system receives the question and searches through its database in order to find pre-stored FAQs that correspond to the question. The system recognizes different formulations of the question. After one or several, if any, relevant FAQs are found, the system sends them and their answers back to the user, as showed in Figure 2.

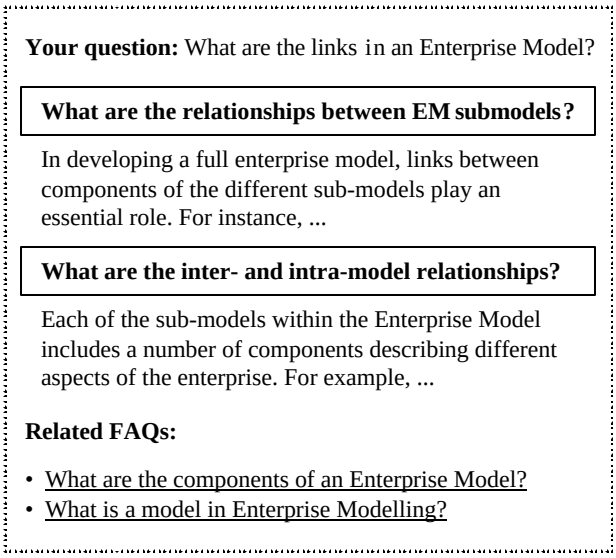

Figure 2 Reply to the question (the text is cut).

A new natural language processing technique for FAQ answering has been developed within the scope of this research. The technique is called Prioritized Keyword Matching. It uses *shallow language understanding*, which means that the FAQ answering system does not comprehend a user question. The system formally matches the question to FAQ entries in the database; the matching is based on keyword comparison. The system performs no syntactic parsing of the question and does not extract semantic concepts. Lexical and morphological analysis of keywords, however, is done in order to enhance the language processing. The first version of the technique implemented in an FAQ answering system was introduced in (Sneiders 1998); a thorough discussion can be found in (Sneiders 1999). This paper continues the discussion with an improved version and preliminary evaluation of the technique.

Approaches and Roles of Automated FAQ Answering

Unlike Artificial Intelligence question answering systems that focus on generation of new answers, FAQ answering systems retrieve existing question-answer pairs from their databases. Two representatives – FAQ Finder and Auto-FAQ – illustrate two types of FAQ answering systems.

FAQ Finder (Hammond et al. 1995) (Burke et al. 1997) is a system designed in order to improve navigation through already existing external FAQ collections. The system has an index – FAQ text files organized into questions, section headings, keywords, etc. In order to match a user question to the FAQs, the system (1) does syntactic parsing of the question, identifies verb and noun phrases in the question, and (2) performs semantic concept matching in order to select possible matches between the question and target FAQs in the index.

Auto-FAQ (Whitehead 1995) maintains its own FAQ set; the system does not perform indexing of an external FAQ collection. The system uses other approach to automated FAQ answering – that of shallow language understanding – where the matching of a user question to FAQs is based on keyword comparison enhanced by limited language processing skills. Question answering is more like text retrieval than traditional natural language processing.

The FAQ answering system developed within the scope of this research follows the approach of Auto-FAQ. Nonetheless, the language processing and FAQ retrieval technique is not published in (Whitehead 1995). After a version of Auto-FAQ had been built, the development of the system stopped (according to personal communication with S. D. Whitehead).

The target system of this research is designed following several principles:

- By using the system, a user can ask natural language questions, perform keyword-based search through the FAQ collection, and browse all the FAQs in the collection.
- The system maintains its own FAQ set, as opposed to indexing an external FAQ source, and uses shallow language understanding when matching a user question to FAQ entries in the database.
- The system is evolving because its question answering ability improves as more questions are asked, recorded, analyzed, and new FAQ entries in the database created.
- The system is Web-based because WWW is a rapidly growing medium convenient for asynchronous communication and popular in the business and academic environments. Besides, a Web-browser is a ready-to-use graphical user interface tool; there is no need to build another one.

The system's server side operating system is Microsoft Windows NT 4.0. The user query processor is a CGI script linked to an HTTP server. The administration tool uses its own graphical user interface and is not connected to the Internet. Both the query processor and administration tool are written in Borland Delphi Pascal. The reasoning, architecture, functionality and implementation of the system, as well as creation and maintenance of the system's FAQ set are presented in (Sneiders 1999). A version of the system is available at (see references: EKD url).

We can distinguish the following roles of an automated FAQ answering system in the community of its users:

- *Means of information acquisition.* The system's natural language based user interface lets people formulate their

problems and submit them as questions to the system that records these questions before answering.

- *Form of organizational memory.* We can perceive organizational memory as storage bins containing information about past decision stimuli and responses (Walsh and Ungson 1991: 61). An FAQ answering system contains:
  - Identified problems (stimuli mentioned above). Each FAQ identifies a problem that has appeared within the community of the users of the system.
  - Solutions to these problems (responses mentioned above). Each FAQ has an answer that explains the solution to the problem expressed in the FAQ.
- Means of information retrieval. The system retrieves FAQs and their answers upon request expressed in natural human language.

This research is devoted primarily to the last role of an FAQ answering system – means of automated information (i.e., FAQ) retrieval.

Prioritized Keyword Matching

The statistical methods of Information Retrieval (Salton and McGill 1983) (Salton 1989) count frequency of common terms in the documents being compared in order to determine similarity between these documents. These methods process large documents and are not appropriate for FAQ answering: single sentences are too short for calculations of term frequency.

On the other hand, semantic language processing like that used by FAQ Finder either requires a very rich lexicon and a knowledge base dealing with the meanings of FAQs (an average website cannot afford such a lexicon and knowledge base) or yields low quality of FAQ retrieval otherwise.

The Prioritized Keyword Matching technique was developed in order to make automated FAQ answering affordable for virtually any website.

Basic Idea

The idea of Prioritized Keyword Matching is based on the assumption that there are three main types of words in a sentence within a certain context in a certain subject:

- *Required keywords* are the words that convey the essence of the sentence. They cannot be ignored.
- *Optional keywords* help to convey the meaning of the sentence but can be omitted without changing the essence of the sentence. The nuances may change though.
- *"Irrelevant" words*, like "a", "the", "is", etc., are words that are too common in ordinary language or in the subject. The meaning of "irrelevant" words is close to that of stop-words in Information Retrieval. The only difference is that stop-words are assumed always unimportant in a given collection of documents, whereas any of the "irrelevant" words in Prioritized Keyword Matching may suddenly become relevant if used in order to emphasis nuances in a particular sentence in a given collection of sentences. The latter happens rarely.

Let us consider an example with "What is the relationship between Business Goal Models and Business Process Models?" In this sentence we distinguish:

- required keywords "relationship", "goal", "process";
- optional keywords "business", "models";
- irrelevant words "what", "is", "the", "between", "and".

If we modify this selection of words with their synonyms and various grammatical forms, we obtain a new, broader selection of words, which characterizes a set of different sentences that are semantically related to the one given above. We assume that these sentences are related although we do not comprehend them.

Let us define that each keyword is always represented by a number of synonyms and their grammatical forms, and that irrelevant words are the same for all the sentences. Hereby, if the same required and optional keywords can characterize two sentences, we declare that both sentences have about the same meaning, i.e., they match each other. This is the basic idea of Prioritized Keyword Matching.

There is also the forth type of words – *forbidden keywords* – whose possible presence in a sentence is not compatible with the existing meaning of the sentence. For instance, for the sentences "Why do we use it?" and "How do we use it?", "how" and "why" are respectively forbidden keywords: the formulation of the first sentence is not expected to contain "how", the formulation of the second one is not expected to contain "why". In practice, we do not consider all the possible words not expected in the formulation; we consider forbidden keywords only when we need to distinguish two similar sentences having the same required keywords. Forbidden keywords emphasize the difference between both sentences.

One may wonder why "business" and "models" in the example above are optional keywords, i.e., less relevant. The reason is that, in the context of Enterprise Modelling, Business Goal Models and Business Process Models are often referred to as simply goals and processes. A user may formulate the question as follows: "What is the relationship between goals and processes?" "Business" and "models" do not appear in this formulation.

Conceptual Data Structure

Let us assume that we have a database consisting of FAQ entries where each FAQ has its required, optional, and forbidden keywords specified.

According to the basic idea of Prioritized Keyword Matching, each FAQ becomes a pattern that identifies a class of questions with similar meanings, where the keywords of the FAQ identify the concepts relevant to this pattern. After an arbitrary user question is asked the system uses the Prioritized Keyword Matching algorithm to match the question to each FAQ entry separately in order to determine whether or not the question belongs to the

class of questions identified by the FAQ. Hereby, the algorithm has the following input (Figure 3 illustrates it):

- an arbitrary sentence – a user question; and
- an FAQ entry with required, optional, and forbidden keywords.

The output of the algorithm is a statement denoting whether or not the user question matches the FAQ in the entry. The algorithm uses a list of "irrelevant" words introduced earlier; there is one such list for all the FAQ entries in the database.

Figure 4 shows the concepts involved in the algorithm and the relationships between these concepts if the FAQ answers the user question. It is important to note that the *only user's concern is his or her own question*. When typing the question, the user knows nothing about the structure of the database, the keywords, and the matching algorithm. All the data, except the question itself, either come from the database or is created during the matching process. The data concepts are explained in the next subsection along with the algorithm.

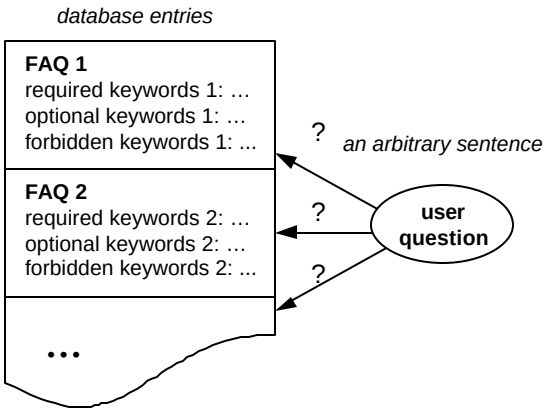

Figure 3 Input to the algorithm: an FAQ entry with identified keywords and an arbitrary user question.

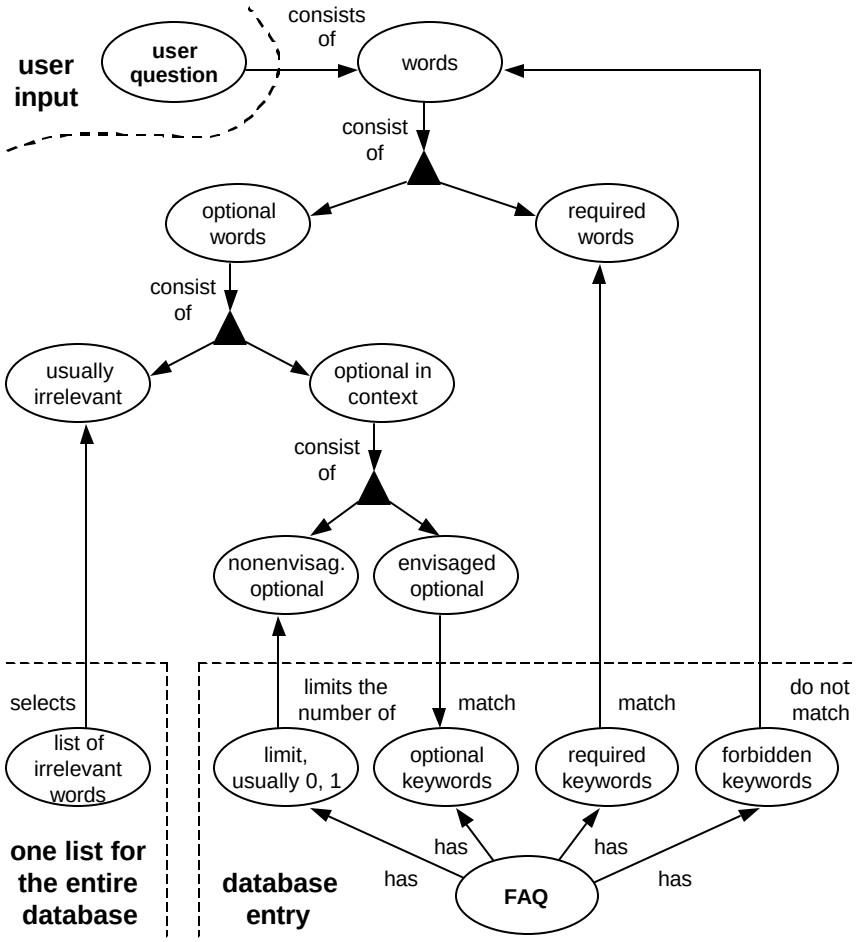

Figure 4 Concepts involved in Prioritized Keyword Matching and the relationships between them if the FAQ answers the user question.

Description of the Algorithm

In order to better understand the Prioritized Keyword Matching algorithm, let us observe it together with an example in the context of Enterprise Modelling. After a user has asked a question, the system matches this question to an FAQ entry in the database.

The ***user question***: "How are substantial business goals related to business processes?"

The ***FAQ*** "What is the relationship between Business Goal and Business Process Models?" and its keywords:

- *Required*:
  a) "goal", "goals";
  b) "process", "processes";
  c) "relation", "relations", "relationship", "relationships", "dependence", "dependencies", "connection", "connections", "association", "associations", "link", "links", "linked", "linking", "relate", "relates", "related", "relating", "connect", "connects", "connected", "connecting", "associate", "associates", "associated", "associating".
- *Optional*: "business", "businesses", "model", "models".
- *Forbidden*: none.
- *Limit of non-envisaged words*: 1 (described in Step 6 of the algorithm).

The human common sense says that the user question and FAQ convey roughly the same meaning. The system has to formally determine this by performing the following steps:

1. The system splits the user question into separate words.
   In the example, the question is split into "how", "are", "substantial", "business", "goals", "related", "to", "business", "processes".
2. The system matches the ***required keywords*** in the entry, usually two or three, to the words of the user question. If there is at least one required keyword that is not represented among the words of the user question by at least one synonym or grammatical form, the system *rejects* the match between the user question and the FAQ.
   In the example, all three required keywords of the FAQ are represented among the words of the user question: "goals" (a), "processes" (b), and "related" (c).
3. The system matches the ***forbidden keywords*** in the entry, if any, to the words of the user question. If there is at least one forbidden keyword that is represented among the words of the user question by at least one synonym or grammatical form, the system *rejects* the match between the user question and the FAQ.
   In the example, there are no forbidden keywords. These keywords are rarely used only to emphasize the difference between similar in appearance but still different in meaning FAQs.
   After matching the required and forbidden keywords, the system removed their counterparts among the words of the user question and proceeds with the optional words: "how", "are", "substantial", "business", "to", "business".
4. From the optional words, the system filters out those listed as usually irrelevant ("a", "the", "is", etc.). The filtering is based on the ***list of irrelevant words***, one list for all the FAQs in the database.
   In the sample question, irrelevant are the words "how", "are", "to". After they are filtered out, there are only context dependent optional words left: "substantial", "business", "business".
5. The system matches the context dependent optional words of the user question to the ***optional keywords*** in the entry. The system identifies and filters out the context dependent optional words that match these keywords.
   In the sample question, the only context dependent optional word that matches the optional keywords is "business"; in Figure 4 it is referred to as envisaged optional. The other one – "substantial" – does not match the optional keywords; in Figure 4 it is referred to as non-envisaged optional.
6. The system considers the words left – non-envisaged optional words – which match neither required nor optional keywords, and are not in the list of irrelevant words. If there are too many such words, the system *rejects* the match between the user question and the FAQ in the entry. How does the system determine this "too many"? For this purpose, there exists a ***limit*** of non-envisaged words, usually 0 or 1, stated in the entry and dependent on the complexity of the FAQ. The number of non-envisaged optional words may not exceed this limit.
   In the sample question, the only non-envisaged optional word is "substantial", which does not exceed the limit in this FAQ entry equal to 1. Therefore there is no reason to reject the match between the user question and the FAQ.
7. Already three times the system had an opportunity to reject the match – in Steps 2, 3 and 6. It did not use this opportunity. It *accepts* the match between the user question and the FAQ in the entry.

Required, optional, and forbidden keywords in an FAQ entry may be represented by both single words and phrases (phrases are discussed further). In order not to corrupt phrases in the user question during the matching process, the words in the question are not removed physically; they are just marked as matching.

A user would loose much information if the system retrieved only FAQs that are very close to the user question. Therefore the system retrieves so called related FAQs as well, as showed in Figure 2. An FAQ is considered related to the user question if all of its required and no forbidden keywords are represented among the words of the question; optional words are ignored. This is checked in Steps 1 through 3 of the above algorithm.

What is a Good FAQ Entry?

There is a simple answer: a good FAQ entry is one which *does* match a large variety of differently formulated user

questions with the meaning close to that of the FAQ, and *does not* match not related user questions. Three features characterize a good entry:

- *Thorough selection of required and optional keywords* in an entry highlights representative concepts of the FAQ.
- *Good context dependent controlled vocabulary* (i.e., lexicon) ensures the ability of the system to resolve context dependent synonyms and grammatical forms of each keyword.
- *Sufficient number of auxiliary entries* helps to meet a large number of formulations of a user question. Although the approach of matching required, optional, and forbidden keywords is flexible, sometimes one FAQ entry in the database cannot represent all conceivable formulations of the corresponding user questions. Therefore several entries for one FAQ may be introduced. For instance, "What is Actor and Resource Model?" and "How do we describe actors in Enterprise Modelling?" are two formulations of the same FAQ, each in its own database entry with its own keyword set. In the real system there are 1-2, less often 3 auxiliary entries for each FAQ.

Each FAQ entry in the database has a small lexicon. Synonyms and various grammatical forms of each keyword are considered so that the entry covers as many different ways of asking the same question as possible. Typical grammatical variations are:

- singular and plural forms of nouns;
- tenses of verbs;
- different spellings, American vs. British English (e.g., "model<u>i</u>ng" vs. "model<u>l</u>ing", "formula<u>s</u>" vs. "formula<u>e</u>", "analy<u>z</u>e" vs. "analy<u>s</u>e");
- split and merged words (e.g., "sub-model" vs. "submodel", "non-existent" vs. "nonexistent" vs. "not existent").

Typical cases of synonymy are:

- ordinary language synonyms: "related", "connected", etc.;
- switching between related verbs, nouns and adjectives: "In what cases do we <u>apply</u> Enterprise Modelling?" vs. "What are the cases of <u>application</u> of Enterprise Modelling?" vs. "When is Enterprise Modelling <u>applicable</u>?";
- words that are not ordinary language synonyms, but act like synonyms in a particular context: "Why is Enterprise Modelling <u>beneficial</u>?" vs. "Why do we <u>use</u> Enterprise Modelling?";
- generalization and specialization of a concept (not common).

Prioritized Keyword Matching vs. Techniques of Information Retrieval

The surroundings of the use of Prioritized Keyword Matching resemble those of Information Retrieval: we have a free-text user query and a collection of indexed documents where we perform exhaustive search. In case of Information Retrieval the index means a term vector for every document and a common stop-list; in case of Prioritized Keyword Matching the index means required, optional, forbidden keywords for every document and a common list of "irrelevant" words. Sounds similar.

The principal difference between both techniques is following: importance of a term in a term-vector is denoted by its scalar weight whereas importance of a term in case of Prioritized Keyword Matching is denoted by its *non-scalar* role (i.e., required, optional, etc. keyword). Knowing the role of a term in the document we make much better conclusions about different properties and the importance of the term than just knowing the weight as the only property. This core difference has the following consequences:

- Term-vectors are effective only if they are long enough unless there is additional information, other than scalar proportion of the importance of the term, encoded in the numerical weight. On the contrary, the roles assigned to the terms in a document do not require many terms in the document in order to compare it to another document.
- In Prioritized Keyword Matching, the roles are assigned to the terms in the collection of documents only; the user query is not indexed. On the contrary, the term-vectors in Information Retrieval require indexing of both the query and the documents in the collection.

The major drawback of roles is that we need intelligent reasoning in order to assign a role to a term. On the contrary, in Information Retrieval we assign weights to the terms according to the corresponding term frequency with no reasoning whatsoever.

Idea of Multiple Lexicon

FAQ entries in the system's database, once created or updated, are static. The keywords of each FAQ are known long before any user questions are asked. Therefore the synonyms and grammatical forms of each keyword are put into the entry along with the keyword. No external source of lexical information is used during the matching of a user question to this entry. Lexical and morphological analysis of the keywords in the entry is done before these keywords are used.

Multiple vs. Single Lexicons

One may suggest that the system uses no lexicon. This is not true. Each FAQ entry has a small lexicon that implements one function – identification of mutually exchangeable words (synonyms and their grammatical forms) for every keyword within the context of a given FAQ. The system uses a multiple lexicon assembled from numerous independent small lexicons, where each of them is attached to its own FAQ entry. Figure 5 illustrates the difference between multiple and the ordinary single lexicons.

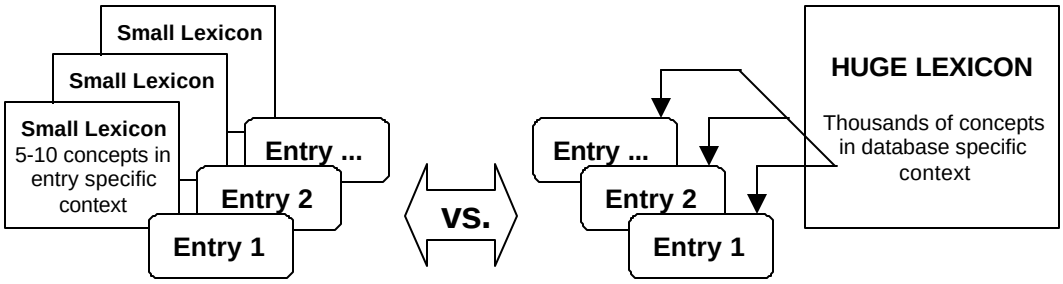

Figure 5 Multiple (many small units) vs. single (one large unit) lexicons.

The principal difference between both types of lexicons lies in the size and context of autonomous units. A single lexicon has one large unit containing thousands of concepts within the context of the entire database, whereas a multiple lexicon has many small autonomous units containing 5-10, or even less, concepts within the context of one database entry (i.e., one FAQ). A multiple lexicon has certain advantages:

- Semantic relationships between words may be entry rather than database specific, like in "How do we develop a Business Process Model?" vs. "How do we acquire a Business Process Model?". It is not possible to include so specific relationships in a lexicon one for the entire database.
- It is difficult to maintain a large lexicon. As new concepts are entered into such a lexicon they get "frozen". If a concept or its relationships with other concepts are modified, we must keep track of how the changes interact with the context of every single entry where this concept is used. The units of a multiple lexicon are more localized, therefore their maintenance is less error prone.
- Traditionally in natural language processing, grammatical analysis is applied to a user query. A lexicon, traditionally the single one, is a tool of this analysis. A multiple lexicon offers another model of the analysis: in case of one-sentence database entries like in FAQ answering, an autonomous unit of the lexicon stores the result of pre-made analysis of the sentence (i.e., FAQ) rather than a tool for this analysis. Hence:
  - we can use whatever advanced language processing tool we want (because we store the result of the analysis); and
  - the query processing is reduced to keyword matching, which requires simpler data structure and less processing power than analysis of the query using a single lexicon.

A possible drawback of a multiple lexicon is redundancy. Yet this is a minor drawback. If the database has 300 entries with 10 concepts each, there are only 3000 concepts in total, which is less than a small dictionary anyway. The target system of this research reduces redundancy by using substitutes (substitutes are discussed further).

The idea of multiple lexicon is analogous to that of object-oriented programming (OOP):

- *Definition of a lexicon* (in OOP – definition of a class). The necessary functions of lexical analysis – resolving of synonyms, grammatical forms, generalization, specialization, etc. – are defined (in OOP – definition of methods) using empty concept slots (in OOP – properties or attributes) since the actual concepts are not known yet at the definition stage.
- *Instances of the lexicon* (in OOP – instances of the class, i.e., objects). Within a series of different contexts, the concept slots created during the definition phase are filled by the actual concepts. Now there is a series of analogous, autonomous, narrow context dependent lexicons where each lexicon may contain five concepts as easy as five thousand.

Role of Human Reasoning in FAQ Answering Using a Multiple Lexicon

The Prioritized Keyword Matching technique, which uses a multiple lexicon, was developed for an FAQ answering system considering the following peculiarities:

- The task of the system is automated FAQ answering; the system itself does not introduce new FAQs.
- Human intelligence is easier to "implement" then artificial intelligence.
- Since FAQs about the topic (particularly, Enterprise Modelling) were not collected previously, the FAQs must be written and the database must be populated manually by the administrator (not a user!) of the system. Therefore we can use full advantage of the present human reasoning and ask the administrator to select the keywords and create autonomous units of the multiple lexicon as well. The administrator does enjoy computer assistance during this work, but this is a subject of other research, not that of FAQ retrieval.

We can imagine FAQ entries as pieces of conserved human intelligence; pre-maid decisions are applied upon a user's request as a query is submitted, as opposed to artificial intelligence where decisions are made upon a user's request. Hence, using a multiple lexicon we can reduce the task of FAQ retrieval to keyword matching and still have an illusion of an intelligent system.

Well, FAQ entries for the target system of this research are created manually. Yet this work is neither too tedious nor difficult. The entries are reusable and copy-and-paste manipulations applicable. Synonym groups in the context of different FAQs are not always the same but usually similar. Computerized tools – dictionaries, grammar prompters, spelling checkers, you name it, incorporated in the administration tool or as stand-alone applications – are possible and welcome as long as the human supervision is preserved. The result of the efforts – an FAQ entry – is important whatever methods are used in order to create it.

The approach to FAQ answering using a multiple lexicon was designed for a particular system operated under particular circumstances – the system maintains its own FAQ set. Therefore the approach should *not* be misleadingly generalized; it should *not* be misapplied to systems built for a different purpose – navigation through an external frequently changing FAQ source.

Regarding involved human resources, the Prioritized Keyword Matching technique was developed, the FAQ answering system designed and implemented, a particular FAQ set created and maintained – this work was done by one postgraduate student.

Substitutes

Let us come back from a theoretical discourse to more practical issues. Since the Prioritized Keyword Matching technique performs formal keyword matching without understanding the meanings of the words, we can introduce a shortcut for a group of context dependent synonyms and their grammatical forms with similar appearance. For instance, "relat*" can be a shortcut for "relation relations relationship relationships relate relates related relating". The only meaning of the shortcut is a graphical substitute for a group of words. While shortcuts are not visible to the users of an FAQ answering system, they make administration of the system easier. With shortcuts, the sample FAQ entry discussed along with the Prioritized Keyword Matching algorithm looks more attractive. The FAQ: "What is the relationship between Business Goal and Business Process Models?" The keywords:

- *Required*:
  a) "goal*";
  b) "process*";
  c) "relat* depend* connect* associat* link*".
- *Optional*: "business* model models".

The optional keyword "model" has no shortcut in order to distinguish it from "modelling".

One may object that "goal*" matches both "goal" and "goalkeeper". It is not likely, however, that the system maintaining the above FAQ could get a question where soccer players and processes along with their relationships would be combined into one sentence within the context of Enterprise Modelling. While shortcuts make the work of the administrator easier, they are not enforced where they are not appropriate.

We may observe that, although synonym groups differ from context to context, they may have common, repeating words. In order to save writing efforts, we can create a repository of substitutes for repeating groups of words. For instance, we can define "$models" as a substitute for "model models", put it into the repository of substitutes, and use like this:

- *Optional keywords*: "business* $models".

Here "$models" has no other meaning as a graphical substitute for the two words. There can be shortcuts used in the definition of a substitute.

Existence of a repository of substitutes does contradict with the idea of multiple lexicon because the units of the lexicon stop being autonomous – they have common substitutes. Nonetheless, the advantages of a multiple lexicon are preserved if substitutes are used carefully. Substitutes save writing efforts, and it is up to the administrator of the system to decide where and how to use them.

Phrases

The first version of the FAQ answering system developed within the scope of this research did not recognize phrases; it did not distinguish "process modelling" from "modelling process", which was an obvious disadvantage to be eliminated.

What is a Phrase for Prioritized Keyword Matching?

A phrase in a user question is a sequence of words where their order is important. A phrase represented in an FAQ entry is a sequence of concepts where each concept is represented by a group of synonyms and their grammatical forms. Each synonym may be a single word or another, embedded phrase. The administrator of the system enters a phrase into an FAQ entry along with the keywords as one of the synonyms of a keyword according to the following syntax: "<" denotes the beginning if a phrase, ">" denotes the end of a phrase; ";", ":", and "#" are delimiters between the concepts in the phrase. Examples:

- <process*; modelling modeling>
- <<modelling modeling; process*> # <in; spite; of> despite # <process*; modelling modeling>>

There are three types of concepts in a phrase:

- "<" and ";" are delimiters in front of a mandatory concept: "<one; of; two three>" matches either "one of two" or "one of three" and nothing else.
- ":" is a delimiter in front of an optional concept: "<on: the; other; hand>" matches "on the other hand" and "on other hand" with dropped "the".
- "#" is a delimiter in front of a concept that allows having any number of any words between this and the previous

concept: "<modelling modeling # process*>" matches both "modeling process" and "modelling of many different kinds of various processes" (note that both have different meanings).

A user of the system does not see how phrases are represented in an FAQ entry.

Main Ideas behind the Phrase Processing

The reasoning in this subsection is not even of the concern of the administrator of an FAQ answering system; the subsection discusses the principles of matching a phrase to a user question implemented in the target system of this research.

During the matching process, the system constructs a graph for each phrase in an FAQ entry. Matching of a phrase starts when the first concept in the graph matches some expression – a single word or another, smaller phrase – in the question. Supposedly, the rest of the concepts in the graph should match the rest of the expressions (mostly single words) in the question. Nonetheless, the matching is not straightforward because concepts may be optional, there may be variable distance between adjacent concepts, or several synonyms (which may be embedded phrases of different length, and so on recursively) in a concept can match an expression (not necessarily the same) in the user question. If we can match the same phrase graph in many different ways and get different results, we have alternative paths in the control flow of the matching. It is possible to construct a representation of a phrase so that no alternative paths ever appear; a reliable system, however, must be able to process them in case if they do appear. In order to make it possible, the control flow in the phrase graph must be organized properly. Figure 6 shows incorrectly and correctly organized control flows.

An incorrect control flow goes from concept to concept: the system discovers that there is a synonym in a concept that matches an appropriate expression in the user question and proceeds with matching the next concept. Concept B, however, turned out a trap: there were two matching synonyms. The system took the first one – Syn.B.1 – and failed at Concept C. It was too late to return to Concept B and try the alternative Syn.B.3 because the information about previous alternatives was already lost.

A correct control flow goes from synonym to concept. After the system had selected Syn.B.1 and failed at Concept C, it came back to the "fork" in Concept B and took Syn.B.3, proceeded with Concept C one more time (the dashed line), and reached the happy end.

A correct phrase graph should be constructed so that there is a link from each synonym of the current concept to the next concept. If the synonym is a single word, the link goes from this synonym to the next concept. If the synonym is an embedded phrase, the link goes from each synonym of the last concept of the embedded phrase to the next concept in "this" phrase, and so on recursively.

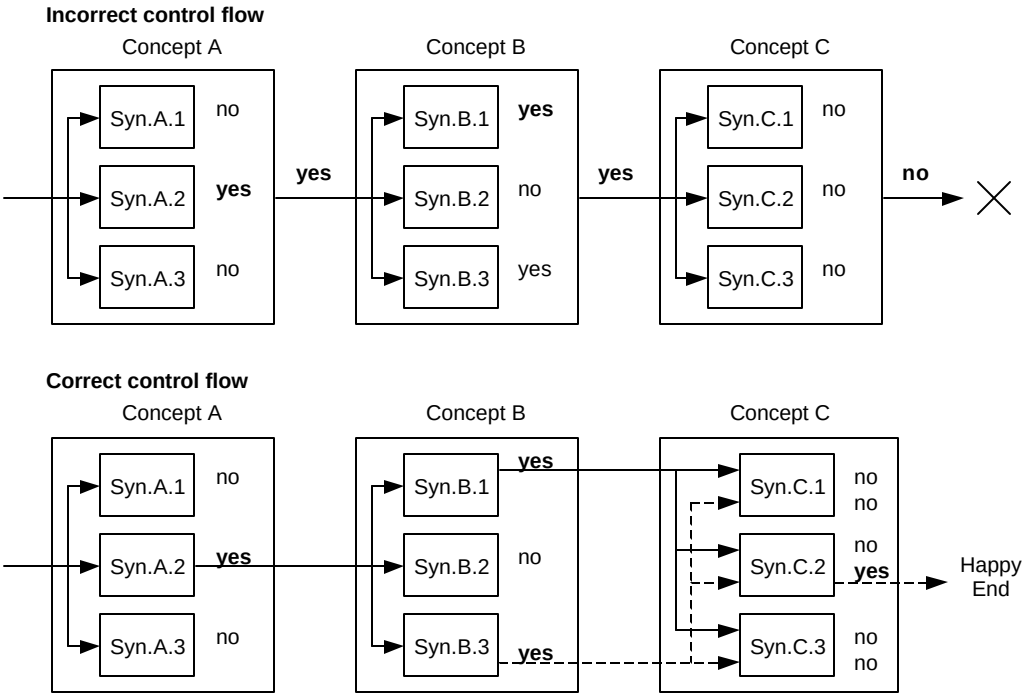

Figure 6 Incorrect and correct control flows of matching a phrase.

Preliminary Evaluation of an Implementation of Prioritized Keyword Matching

Two core features influence the performance of an FAQ answering system: quality of the language processing (as a question is asked, the system must find the corresponding FAQ) and completeness of the FAQ set (the corresponding FAQ must exist in the database).

A discussion on completeness of the FAQ set is presented in (Sneiders 1999). The main issue we should consider here is following. In the beginning, the system is able to answer only a few questions. We may ask a number of questions and do not get them answered, which indicates that the system has a bad question answering ability. After these questions are incorporated into the FAQ set, which is normally done, we may ask them once more and get them answered, which indicates that the system has a perfect ability to answer questions. The universe of conceivable questions of any reasonably broad topic is infinite, completeness of the FAQ set is ever improving, the system is evolving.

The quality of Prioritized Keyword Matching technique is determined by its ability to retrieve relevant existing FAQs from the database upon a user's request. In Information Retrieval there are two parameters to measure the quality of such retrieval – recall and precision (Salton and McGill 1983: 164-172) (Salton 1989: 248-249, 277-278). *Recall* R characterizes the system's ability to retrieve all the relevant items existing in the collection of documents (i.e., FAQs):

$$R = \frac{\text{number of relevant documents retrieved}}{\text{total number of relevant documents in collection}}$$

Precision P characterizes the system's ability to retrieve only relevant items:

$$P = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}$$

Although all the features of language processing discussed in this paper are implemented and tested, there is not enough empirical data for a formal evaluation. Instead, recall and precision of the first implementation of Prioritized Keyword Matching is presented. This implementation has no phrase processing, no forbidden keywords, no shortcuts, no substitutes. At the moment of taking the measurements, there were more than 80 questions asked to the system (asked by people other than the administrator of the system) and logged. From those more than 80 recorded question / reply pairs, selected were those where the questions were not duplicate (which happened if questions were posted for demonstration purpose) and without spelling mistakes. In order not to be influenced by completeness of the FAQ set, only those question / reply pairs were considered where the question had the corresponding FAQs in the database at the moment of asking it. Eventually, not so many – 17 – question / reply pairs were selected that satisfied the criteria above.

At first, only close answers were observed ignoring related FAQs. The average recall was 0.65; the average precision was 0.95. Nonetheless, it proved that often the system incorrectly classified a close answer as a related FAQ. Since the user obtained the FAQ anyway, recall and precision ignoring the difference between close answers and related FAQs was worth measuring: the average recall was 0.88, the average precision was 0.85. These figures are high. Although the measurements with only 17 queries may be criticized for not being representative, they shed positive light on the potential of the technique.

Prioritized Keyword Matching showed good query processing time – 1 to 2 milliseconds to match a user question to a separate FAQ entry (roughly 0.5 milliseconds more after phrase processing was introduced). The processing time is subjective: it depends on the hardware, efficiency of the compiler, skills of the programmer. The particular FAQ answering system was operated in the Microsoft Windows NT 4.0 environment using a Pentium 166 MHz processor and 48 MB RAM.

An experienced administrator of the system needs 5-15 minutes in order to select and test the keywords for an FAQ entry.

Further Research and Conclusions

There are a number of possible directions of developing the ideas implemented in the target system of this research. Several of them are mentioned below.

Although the administrator of the system enjoys computer support when he or she creates an FAQ entry, an integrated tool-set would ease the tasks of selecting and analyzing the keywords. Additional support is needed in order to create the initial set of FAQs before the system is put into operation since no one is going to ask any questions to a system with no FAQs. Automated analysis of manuals and similar literature could suggest raw material for the empty database of a newly created FAQ answering system.

The Prioritized Keyword Matching technique uses a multiple lexicon. It would be a challenge to move forward into deeper language analysis using this kind of lexicon. We could change the categorization of required, optional, and forbidden keywords so that it is stated who does what. Then the system could better suggest related FAQs and give better hints on what kind of related information exists in the database. Combining this with parsing of user queries, the system may try to resolve pronoun references "it", "this", "that", etc., and make kind of alive dialogue with the user.

According to the basic idea of Prioritized Keyword Matching, an FAQ in the database is a pattern that identifies a class of questions with similar meanings. Required and optional keywords identify the concepts relevant to this pattern. Each keyword is represented by a group of synonyms. In FAQ answering, the pattern is a human language sentence, the keywords identify concepts

expressed in human language, and the synonyms are natural language words. We may generalize the pattern mechanism of Prioritized Keyword Matching and apply it to social, physical, chemical, etc. phenomena. We can organize generic patterns of these phenomena like FAQs in a database and search through the database by using the Prioritized Keyword Matching technique.

Conclusions

This paper presents continued research in automated FAQ answering by using shallow language understanding. The Prioritized Keyword Matching technique discussed here was developed in order to match an arbitrary user question to an FAQ entry in the database. The use of shallow language understanding means that the matching is based on keyword comparison; the system performs no syntactic parsing of the question, it does not extract semantic concepts. In Prioritized Keyword Matching, lexical and morphological analysis is applied to the keywords in the FAQ entries rather than user questions long before any questions are submitted. This implies use of a multiple lexicon assembled from numerous autonomous FAQ-context dependent small lexicons: each of those small lexicons is attached to its own FAQ entry. We can imagine FAQ entries as pieces of conserved human intelligence; pre-maid decisions are applied during the matching process as a user query is submitted. Hence, by using a multiple lexicon we can reduce the task of FAQ retrieval to keyword matching. A system using Prioritized Keyword Matching may attain good recall and precision of FAQ answering. Two earlier obtained (recall, precision) value pairs are (0.65, 0.95) and (0.88, 0.85). The lion's share of this success is attributed to the context dependence of the autonomous units of a multiple lexicon.

The approach to FAQ answering using a multiple lexicon was designed for a particular system operated under particular circumstances – the system maintains its own FAQ set. Therefore the approach should *not* be misleadingly generalized; it should *not* be misapplied to systems built for a different purpose – navigation through an external frequently changing FAQ source.

The paper introduces an original approach to processing phrases within the framework of shallow language understanding. A phrase is recursively represented as a series of concepts, where each concept contains synonyms, where each synonym may be a single word or another, embedded phrase. While matching a phrase, the system is able to process alternative paths.

Relative simplicity of the Prioritized Keyword Matching is aimed at making automated FAQ answering affordable for an average website. By having a natural language based user interface, the system adds one more dimension – limited human language understanding – to the traditional notion of multi-media technology (images, sounds, animation) on WWW. One person with at least normal intelligence is able to install the software and populate the FAQ set of the system. The system is ready to work with the first FAQ entry in the database; neither a large lexicon nor a knowledge base for inference and deduction are needed.

There exists a version of the working system which answers questions on Enterprise Modelling. Another version, which answers questions on Internet protocols, is being introduced.

References

Bubenko, J. jr. 1994. Enterprise Modelling. *Ingènierie de Systemes d'Information*, vol. 2, issue 6: 657-678.

Burke, R.; Hammond, K.; Kulyukin, V.; Lytinen, S.; Tomuro, N.; and Schoenberg, S. 1997. Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System. *AI Magazine*, vol. 18, no. 2: 57-66.

EKD url. *EKD - Enterprise Knowledge Development*. http://ekd.dsv.su.se/, valid in August 1999

Hammond, K.; Burke, R.; Martin, C; and Lytinen S. 1995. FAQ Finder: a Case-Based Approach to Knowledge Navigation. *Proceedings. The 11th Conference on Artificial Intelligence for Applications*, 80-86. Los Alamitos, CA, USA: IEEE Comput. Soc. Press

Salton, G., and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.

Salton, G. 1989. *Automatic Text Processing. The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Sneiders, E. 1998. FAQ Answering on WWW Using Shallow Language Understanding. Information Systems in the WWW Environment. *IFIP TC8/WG8.1 Working Conference, 15-17 July 1998, Beijing, China*: 298-319. Chapman & Hall.

Sneiders, E. 1999. Automated FAQ Answering on WWW Using Shallow Language Understanding. Thesis in partial fulfillment of the requirements for the degree of Licentiate of Technology, Dept. of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, Sweden.

Walsh, J. P.; and Ungson, G. R. 1991. Organizational Memory. *Academy of Management Review*, vol. 16, no. 1: 57-91.

Whitehead, S. D. 1995. Auto-FAQ: an Experiment in Cyberspace Leveraging. *Computer Networks and ISDN Systems*, vol. 28, no. 1-2: 137-146.