

KOM2002

eContent 22071Y2C2DMAL2

REPORT D2.2-part B Revision 1.0

Natural Language Question Answering System Classification Manual

Written by:

Jacob Palme and Eriks Sneiders

Contractual Date of Delivery: 31 March 2003

Latest Revision: 13 Aug 2003

Nature of the Deliverable: SP

Deliverable Type: PU

Work package WP2

Responsible Partner: KTH

Table of contents

Table of contents	2
1 Executive summary	3
2 Basics of FAQ Retrieval.....	3
2.1 How is an FAQ Found?	3
2.2 FAQ Retrieval Techniques.....	5
3 Classification of FAQs.....	5
3.1 FAQ Attributes	5
3.2 How to Classify FAQs	8
3.3 Phrases.....	9
3.3.1 Genitive	10
3.4 Homonyms.....	10
3.5 Examples	11
3.5.1 Example 1: “can't”	11
3.5.2 Example 2: Why do people with eating disorders often accept an invitation, but then not show up?	11
3.5.3 Example 3: Is it Easier to Eat Sensibly Together with Other People	11
3.5.4 Example 4: Which food should I eat and which food should I avoid?	11
3.6 Aliases	12
4 Dictionary files.....	14
4.1 Substitutes List	14
4.2 Stop List.....	15
5 Automatic Generation of Groups.....	16
6 References	16
Appendix A: Technical Description of the Question-Answering Algorithms	18

1 Executive summary

The natural-language question-answering system used in Web4health is based on manually specifying templates for each FAQ (Frequently Asked Question). The templates are matched against the questions asked by users. The success of the question-answering thus depends a lot on the quality of these templates. This manual describes the format and method of producing templates.

2 Basics of FAQ Retrieval

People explore by asking questions. Therefore, since the early days of the Internet, informative websites tend to have compilations of frequently asked questions, i.e., FAQs, and their answers that cover the knowledge domain of the website (Figure 2.1). Each FAQ embodies a problem statement, and the FAQ answer presents a solution to the problem. As small pieces of often requested information, they are like patches on the knowledge domain.

Usually FAQs are grouped in lists; the lists are placed in a subject tree. If the number of FAQs is large, it becomes increasingly time consuming for a user to find FAQs that satisfy his or her information need (imagine that one needs to read more than a hundred questions). That's where question-answering systems help.

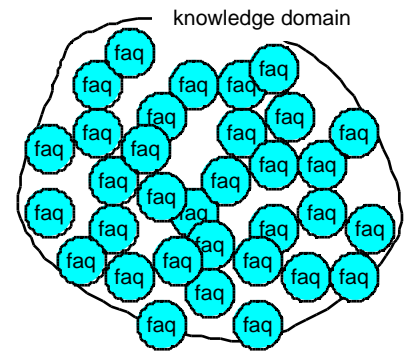


Figure 2.1 FAQs covering a knowledge domain.

2.1 How is an FAQ Found?

One of the tasks within the KOM 2002 project is to create an FAQ database on health disorders. We do not know how “frequently asked” all these questions will be. We do, however, know that they cover important problems because they are selected by the project’s medical partners.

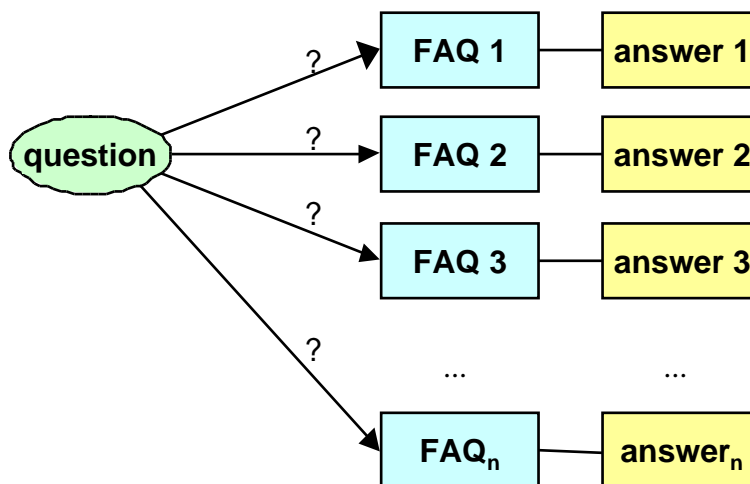



Figure 2.2 Matching a user question to FAQs.

The question-answering system used in KOM 2002 operates the FAQ database. As the user asks a question, the system scans through the database and retrieves related FAQs, if any (Figure 2.2). The user selects those retrieved FAQs and their answers that best suit his or her information need.

How does it work in reality? Let us consider a sample question “What is Gestalt Therapy?” submitted to the prototype¹. In Figure 2.3, we see that the system has retrieved three FAQs, labelled “Answers”: (1) “Gestalt Therapy”, (2) “A session with an Eating Disorders Therapist” and (2) “Psychotherapy; Links”.

Home News Forums Login/out Private Ask the expert Help

Go to: New Question Facts

 **Answers**

Your question:

Answers: [1. Gestalt Therapy](#)
[2. A session with an Eating Disorders Therapist](#)
[3. Psychotherapy; Links](#)

[Top](#) [New question](#) [Next answer](#)

Gestalt Therapy [Link](#) [Discuss](#)
[Ask an expert](#) [Print](#)

Written by: [Gunborg Palme](#), certified psychologist and certified psychotherapist, teacher and tutor in psychotherapy.

First version: 15 Jul 2002 Latest revision: 23 Jan 2003.

Question(s):

What is meant by Gestalt therapy?

Answer:

Figure 2.3 Retrieval of sample FAQs.

FAQ answers follow further on the page. These answers is that information the user is actually looking for.

¹ <http://web4health.info/en/answers/>, valid in January 2003

2.2 FAQ Retrieval Techniques

As the FAQs and answers are entered into the database, there are three generic approaches how a question-answering system can retrieve them:

- *AI (Artificial Intelligence) solutions* use complex knowledge bases in order to comprehend both the query (i.e., user question) and FAQs. AI solutions render good quality of question answering. Unfortunately, they require high qualifications and rare skills of the involved personnel. Development and maintenance of the knowledge bases is expensive.

In a multi-lingual environment, the development and maintenance is required not only for the application domain data but also for language specific features in each language.

- *Statistical techniques of Information Retrieval* evaluate common terms in the query and each FAQ – how many and how representative they are – and decide whether or not the query and the FAQ are equivalent. Unfortunately, statistical techniques work rather poorly if pieces of text are 1-10 words short. They do not work at all if the query and FAQ use different wording to carry the same meaning.
- *Manual indexing of documents* is done for specialised collections. For example, all pieces of legislation issued by the Swedish parliament have manually assigned keywords used by search systems. Precision of the retrieval of manually indexed documents is high because the keywords are representative, selected using human intelligence.

Some question-answering systems, the most well-known of which is Ask Jeeves, use manual indexing of their question templates alias FAQs. A shortcoming of this approach is that each document needs to be processed manually. A good news is that question templates are re-usable; created once they are used over and over again.

Apparently the multi-lingual environment and the human resources available in the KOM 2002 project call for *manual indexing of FAQs*. As far as automatisations of such work is considered, today's technologies cannot communicate the meaning of a natural language sentence from the human mind to the computer without any manual work done at some point of the development of the system itself or its imported components.

Because the process of specifying FAQ templates requires slightly more than simply assigning keywords, we will call it "classification of FAQs".

3 Classification of FAQs

The term *FAQ* is here used to describe the record in the data base describing an answer. The term *query* refers to the actual query written by a user.

3.1 FAQ Attributes

The discussion here is based on the following attributes for each FAQ:

Attribute	Description
ID:	Unique identifier to separate this from other FAQs.
Title:	A user-oriented textual description of the FAQ. Important for finding the document by search engines such as Google.
Question	One or more examples of questions in natural language. This text is shown before the answer, and can also be used as input to the classification process. It is not mandatory.
Required keywords:	Keywords which must exist in the user query for a match to be done.
Priority keywords:	Keywords which will cause this FAQ to be shown before FAQs without this keyword.
Optional keywords:	Keywords which may or may not exist in the user query. Words in “Priority” need not be repeated here.
Forbidden keywords:	Keywords which may not occur in the user query.
Number of non-envisaged words:	If the user query contains no more than this number of non-specified keywords, the response is shown to the user as a <i>probable match</i> . If the user query contains more than this number of non-expected keywords, the response is shown to the user as a <i>possible match</i> .
Body:	The HTML or XHTML text of the answer to the FAQ.

Example of an FAQ:

ID:	ed-dia-bulimic-obese	A globally unique identifier for this FAQ. Should never be changed. For more information, see
Group:	e-dia - Eating Disorders Diagnosis	Used to sort the list of all FAQs produced by the command http://salut.dsv.su.se/w4h-faq-??/faq.exe where “??” is replaced by the language, see chapter 5 on page 16.
Title:	Are bulimic people obese?	Shown in response to user.
Required keywords:	bulimic* [binge*; eat*] ; obes* overweight* fat* corpulent* stout* fleshy [weigh* # \$much]	Indicates, that for this template, a matching query must contain two terms, one term matching bulimic [binge*; eat*] and one term matching obes* overweight* fat* corpulent* stout* fleshy [weigh* # \$much].

The semicolon “;” separates terms. The terms can occur in arbitrary order outside square parentheses. Thus, this template will match both the question “Are bulimics overweight?” and “Are fat people bulimic?”.

The asterisk “*” at the end of a word means any additional characters, for example obes* will match both “obese” and “obesity”.

Square parentheses “[]” denote phrases. Inside square parentheses, “;” requires that the terms directly follow each other, whereas “#” requires that the terms appear in the designated order without necessarily being adjacent.

It is very important to list many synonyms for each term, since users will often ask the query using different terminology than the person who produces the FAQ.

If the same synonym occurs in many FAQs, you can put it in a separate synonym list, and refer to this list with a name that starts with “\$”. For example, “\$much” refers to a list of synonyms for “much”. This list is named the *substitutes* list. See chapter 4.1 on page 14 for more information.

Example: This FAQ will match the following queries:

Are bulimic people obese?

Will binge eaters weigh unusually much?

Optional keywords: \$people unusual*

Keywords that may or may not occur in the query. There is also a built-in stop list of common words like “the”, “on”, etc. which are always allowed as optional.

Forbidden keywords: anore*

Keywords which may not occur for a match to occur. Use this very sparingly. An example where forbidden keywords may be needed if you want to distinguish between two different answers to two similar FAQs.

Number of non-envisaged words:	1	If the user query contains no more than this number of non-specified keywords, the response is shown to the user as a “likely match”. If the user query contains more than this number of non-expected keywords, the response is shown to the user as a “possible match”. If a query contains many unexpected terms, there is a larger risk that an incorrect match is found. Usual values for this number is 0 or 1.
Priority keywords:	bulimi*	Optional field. If several FAQs are retrieved, the system counts how many priority words match the user query and boosts the rank of those FAQs that have more matching priority keywords.
Body:	Usually, the term <i>bulimic</i> is used for people with bulimic behaviour but who compensate for the overeating so as not to become overweight.	The HTML text of the main body of the answer. XHTML is a dialect of HTML with somewhat more rigorous requirements on the syntax. WEB editors like Dreamweaver can automatically produce XHTML, but if you edit the HTML manually, you will have to know the rules for XHTML. (Note: Our implementation will work even if the HTML you specify is not perfect XHTML.)

3.2 How to Classify FAQs

The FAQs are composed and their answers are written by subject experts. After this is done, a person classifies the FAQ: manually selects *required*, *optional*, *forbidden*, *priority* keywords, and the *number of non-envisaged words*. *Required and optional keywords are most important* and most difficult to select. Priority keywords are not obligatory, forbidden keywords are rarely used, and the number of non-envisaged words is 0 for FAQs with simple wording and usually 1 for FAQs with more complex wording.

The person who classifies FAQs, need not to be the same person who composes FAQs and writes the answers. Still, some expertise in the subject is advantageous.

The main method of classifying an FAQ can be described as follows:

1. Try to think of every possible question, which anyone might ask, for which this FAQ is a good answer. Imagine the language that the users will use. Try to think of different people, whom you know, and how they would phrase this question.

2. Group the conceivable questions so that each group can be described by one set of required and optional keywords. Different groups of questions requires different sets of keywords, which means that there may be different wordings for questions that refer to the same answer. Imagine that each group requires a separate FAQ, even if it has the same answer.
3. For each FAQ (i.e., group of questions) distinguished in step 2, write down the keywords as it will be described further.

Consider context-dependent synonyms. It is usually best to treat a synonym in a wide sense. For example, the same FAQ may be an answer to both the question “Is drug addiction hereditary” and the question “Is drunkenness genetic”, so for this FAQ, “drug addiction” and “drunkenness” are synonyms, and “hereditary” and “genetic” are synonyms. If the same long list of synonyms is used often, you can put it into the *substitutes* list (see chapter 4.1 on page 14, and then reuse it in the classification of many FAQs.

Consider phrases and expressions. There is special syntax how to denote them (see chapter 3.3). A synonym of a term can be not only a word but also a phrase of several words.

4. Test the FAQ by submitting various questions. See how it responds to the query and fine-tune if necessary.

When I work on classifications in English, I use the very good English synonym list at <http://www.wordsmyth.net/>. When I work on classifications in Swedish, I use a printed synonym lexicon. In addition to these, I also sometimes use the thesaurus built into Microsoft Word, and some times Roget's thesaurus for English, which is available both online, as a machine-readable text and as a printed book.

3.3 Phrases

A phrase is a sequence of terms separated by punctuation characters, such as white space, a semicolon, dash, apostroph, or colon. Each term is represented by a number of synonyms. A synonym is a word, possibly truncated and appended with an asterisk, or another phrase. It is important to realise that by default the order of keyword terms does not matter. It does matter only when a phrase is explicitly indicated.

Examples of phrases:

[binge; eating]	Matches “binge eating” and “binge-eating”
[binge; eat*]	Matches “binge eating”, “binge eaters”, “binge-eaters”, etc.
[binge; eating munching]	Matches “binge eating”, “binge-eating”, “binge munching”, “binge-munching”

There are three types of terms in a phrase, where the type is denoted by the delimiter:

- "[" and ";" are delimiters in front of a mandatory term: "[one; of; two three]" matches either "one of two" or "one of three" and nothing else.

- ":" is a delimiter in front of an optional term: "[caused ; by : a an ; \$eatingdisorder]" matches "caused by an eating disorder" and "caused by eating disorders" with dropped "a".
- "#" is a delimiter in front of a term that follows the previous terms but is not necessarily adjacent: "[modelling modeling # process*]" matches both "modeling process" and "modelling of many different kinds of various processes" (note that both have different meanings).

Phrases use the following format in the Required, Priority, Optional and Forbidden fields:

[alta; vista] above is a phrase, these two words in sequence. More than two words can be listed, for example [on; the; other; hand].

[begin start ; operations] means either “begin” or “start” followed by “operations”.

[begin start # operations] means either “begin” or “start” followed by arbitrary text, followed by “operations”.

[begin start : the ; operations] means either “begin” or “start” followed by “operations” with the optional word “the” in-between.

Note: All punctuation is ignored, so to match both the words “co-ordinator” and “coordinator” you need to specify [co; ordinator*] coordinator*. Digits are also ignored, only letters are counted as part of a word. And “can't” has to be encoded as [can ; t].

Nesting is allowed, for example [project research ; [co; ordinator*] coordinator* manager*] will match phrases like “project coordinator” or “research co-ordinator”.

Note: In phrases inside brackets, the semicolon (;) means that the terms must come in this sequence. Note that this is different from the meaning of semicolons outside brackets in the Required field, where the terms can come in any order in the actual question.

A warning: Words with less than 4 characters followed by an asterisk will often match many other words than those you intended. For example, you might use hit* to find hit, hitting, etc., but hit* will also match “Hitler” and “hitherto”, which perhaps was not what you intended. Often this is not important, since other words in the template will stop unintended matches.

3.3.1 Genitive

Note that the English genitive apostroph (') is handled like any punctuation character. Thus, to match the English word “girl's” the phrase [girl ; s] must be used.

3.4 Homonyms

A problem is homonyms, the same word with two different meanings. The software does not distinguish upper and lower case characters, so for example the term bed will match both

bed = where you sleep, and

BED = Binge Eating Disorder

If this is a problem, you can combine bed with other words in a phrase, or with other keywords, for example, [the a my his her our your; bed] could be entered as priority keywords.

3.5 Examples

Here are some examples of English terms and suggested classifications of them:

3.5.1 Example 1: “can't”

Required	[can; t not] cannot cant [not no; able abilit* capable capabilit*]
----------	---

3.5.2 Example 2: Why do people with eating disorders often accept an invitation, but then not show up?

Required	decline* refus* abstain* rebuff* [pass* ; up] demur* [show* ; up] arriv* appear* present [drop* ; in] come came
Optional	thank* invit* accept* agree* consent* decid* \$eatingdisorder promis* \$people meet* welcom* greet* face* appoint* contact* [get `# together]
Limit	2

Examples of sentences this template matches:

- “Why do people with eating disorders often accept an invitation, but then not show up?”
- “Why does she refuse to show up when we agreed to meet?”
- “Why did he not come when we had decided to get together at the Opera house?”

3.5.3 Example 3: Is it Easier to Eat Sensibly Together with Other People

Required	\$eat ; sensibl* rational* levelhead* reasonab* unreasonb* prudent* intelli* sane* insane* unrealist* realist* thoughtful* credib* understand* know* clearhead* bright* perspica* precept* astut* smart* apt suitab* witt* shrewd* \$good together party*partie* band* company* bunch* group* gathering* alone lone* solitar* gregarious* secluded* single* desolate* separat* friend* accompan* unaccompan* [on ; \$people ; own] [with by at in # \$people]
Priority	\$eatingdisorder

Examples of questions which are matched by the template above:

- “Is it easier to eat sensibly together with other people?”
- “Why do I binge eat more when I am alone?”
- “Why am I more sensible at parties?”

3.5.4 Example 4: Which food should I eat and which food should I avoid?

Required	diet* weightplan [\$eat \$food nutrit* calory* calorie* ; plan control regulate
----------	---

	control* regulat* keep* regulat* manag* restrain* guid* supervis* subdue* curb* hold* check* restrict* restrain* govern* harness* contain* confine* restrict* tame tune* modulat* adjust* keep* stead* stationar* balanc* stable* level stabilize* uniform* flat even constant*] / Seat ; \$goodbad sensible right right* good* proper* [not ; too] correct* normal* sound* responsi* appropria* well suitabl* / \$goodbad ; \$food \$slim /\$obesity ; \$cause
Priority	eat avoid choose select buy [\$cause ; \$obesity] right* well* correct* good*
Optional	
Forbidden	[eating ; disorder] personalit* charact* temperament* psych* persona
Limit	1

Examples of questions which match this template:

- Help me develop an eating plan.
- How should I eat sensibly?
- What food is good and healthy?
- What can cause obesity?

The Forbidden clause above is included in order to avoid matches for questions like:

- Which personality factors can cause obesity?
- What is wrong with my psyche causing me to eat badly?

There are other, more appropriate answers to these questions.

3.6 Aliases

Aliases describe different equivalent FAQs that refer to the same answer.

When you classify FAQs (chapter 3.2), you try to think of every possible question, which anyone might ask, for which this FAQ is a good answer. Group the conceivable questions so that each group can be described by one set of keywords. One set of keywords in a Required, Optional, Forbidden, or Priority field is an alias. In the database, aliases are separated by “/”. Example:

Question:	What are the dangers with nuclear power?/Can a nuclear power plant be used to make an atomic bomb?
Required:	risk* danger* hazard* peril* bad* wretch* deplorable* terribl* dreadful* awful* guilt* unfavor* unfavour* disapprov* harmful* detriment* unhealth* damag* evil atrocious nasty nastiness rotten malignant* reprehens* mean lamentab* deplorab* ; nucle* atom* fission* ; power* / nucle* atom* fission* ; power* ; bomb* war*

	weapon*
Optional:	plant* nucle* atom* fission*

In the example above, the first alias will match for example “What are the perils of atomic power?” and the second alias will match for example “Can nuclear power plants be used to make atomic bombs?”.

The example above is identical to two separate FAQs with the same body text:

Question:	What are the dangers with nuclear power?
Required:	risk* danger* hazard* peril* bad* wretch* deplorable* terribl* dreadful* awful* guilt* unfavor* unfavour* disapprov* harmful* detriment* unhealth* damag* evil atrocious nasty nastiness rotten malignant* reprehens* mean lamentab* deplorab* ; nucle* atom* fission* ; power*
Optional:	plant* nucle* atom* fission*

and

Question:	Can a nuclear power plant be used to make an atomic bomb?
Required:	nucle* atom* fission* ; power* ; bomb* war* weapon*
Optional:	plant* nucle* atom* fission*

Note: Alias markers“ /” can occur in Required, Proirity, Optional, Limit and Forbidden. If such markers are used in several of these fields, they match as follows:

Question:	A1 / B1 / C1
Required:	A2 / B2 / C2
Optional:	A3 / B3 /C3
Prority:	A4 / B4 / C4
Limit:	A5 / B5 / C5
Forbidden:	A6 / B6 / C6

The above is identical to

Question:	A1
Required:	A2
Optional:	A3
Prority:	A4

Question:	B1
Required:	B2
Optional:	B3
Prority:	B4

Question:	C1
Required:	C2
Optional:	C3
Prority:	C4

Limit:	A5
Forbidden:	A6

Limit:	B5
Forbidden:	B6

Limit:	C5
Forbidden:	C6

If only some fields have alias markers, the value in the field without alias markers will be copied to all aliases. Example:

Question:	A1 / B1 / C1
Required:	A2 / B2 / C2
Optional:	A3
Priority:	A4
Limit:	A5
Forbidden:	A6 / B6 / C6

is identical to

Question:	A1
Required:	A2
Optional:	A3
Priority:	A4
Limit:	A5
Forbidden:	A6

Question:	B1
Required:	B2
Optional:	A3
Priority:	A4
Limit:	A5
Forbidden:	B6

Question:	C1
Required:	C2
Optional:	A3
Priority:	A4
Limit:	A5
Forbidden:	C6

4 Dictionary files

4.1 Substitutes List

The substitutes list contains synonyms of common words, so that you can include a reference to the synonym instead of listing all the synonyms in keywords. Example:

If the substitutes list contains:

\$describe = describ* depict* illustr* specif* character* clarif*

Then the query “**\$describe ; pottassium**” is identical to the query “**describ* depict* illustr* specif* character* clarif* ; pottassium**”.

The substitute entries can contain phrases and asterisks. Example:

\$bad= bad* wretch* substandard* deplorable* lous* terribl* [second ; rat*] poor* useless* base* dreadful* mean* inadequate* inferio* fault* shodd* awful* deficient*

worthless* imperfect* sorry sick* guilt* unfavor* unfavour* disapprov* harmful* detriment* unhealth* damag* evil atrocious nasty nastiness corrupt* stale* rotten malignant* sordid reprehens* [too ; much little high low] useless* [not ; worth] [low ; qualit* standard*] tawdry* [second third ; class rate] mean piti* lamentab* deplorab* risk* danger* hazard* peril* threat* endanger*

The currently valid substitutes list for Web4health can be found at <http://cmc.dsv.su.se/eu/kom/substlist1;login> in all languages, all partners can modify this list using the “Translate or modify command”.

At the bottom of the above web page, the following text is shown:

Language: [Swedish](#) ([0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#)) , **English** ([0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#))

That English is in bold face above indicates that you are just now seeing the English version of the list. By clicking on Swedish, you will get the Swedish version. The series of numbers are previous versions of the list, the last [30](#) before) is the latest version in this language. The brown **O** indicates that this is the original of this FAQ.

When this is written (November 2002) the English substitutes list contains the following terms:

\$addiction, \$advantage, \$anorexia, \$avoid, \$bad, \$benefits, \$bingeeating, \$bmi, \$book, \$bulimia, \$buy, \$category, \$cause, \$change, \$child\$, \$children, \$company, \$control, \$depressed, \$describe, \$do, \$eat, \$eatingdisorder, \$exercise, \$fastfood, \$find, \$food, \$get, \$goal, \$good, \$goodbad, \$guide, \$health, \$healthy, \$help, \$hungersatisfaction, \$hungersatisfied, \$hungrysatisfied, \$identify, \$influence, \$information, \$intoxicated, \$make, \$meaning, \$methods, \$obesity, \$people, \$put, \$raise, \$satisfied, \$sensation, \$sexabuse, \$should, \$similar, \$slim, \$sorts, \$suggest, \$therapy, \$violence, \$weight.

Why are there three very similar items “\$hungersatisfaction”, “\$hungersatisfied” and “\$hungrysatisfied” or two very similar items “\$child” and “\$children”? Answer: They are in fact identical, all three are included to cater for misspellings in templates. Only “\$children” actually contains a list of synonyms. “\$child” just is a reference to “\$children”.

Note: You might save work by using the same list of substitutes in more than one language. The content, of course, must be different for each language, but the names of the substitutes can be the same. This will make classification work easier, since some of the work done in producing the English classification need not be repeated for each language. For example, I have done this for the Swedish language. I can then use the *Required* clause “\$obesity ; \$cause” in both the English and the Swedish version of an FAQ about the causes of obesity.

4.2 Stop List

The stop list contains a list of words that are regarded as *optional* in all user questions. Stop-list words can, however, occur in the keywords of a question, and are then matched as usual. Typical words to include in a stop list are very common words like “and”, “or”, “to”, “the”, etc. In the stop list, the words

are exact and single, they may not end with “*” nor they may be phrases. Thus, when we list the word “recommend”, we include “recommended recommends recommending” as well.

The currently valid stop list for Web4health can be found at <http://cmc.dsv.su.se/eu/kom/stoplist;login> in all languages, the different language versions can be selected in the same way as described above in chapter 4.1 for the Substitutes list.

5 Automatic Generation of Groups

Every answer has an ID, which is a text string of not more than 28 characters. Examples: ed-treat-surgery or life-work-nasty-customer. These IDs are matched to a group list. This group list will be extended throughout the project. The current groups list can be found in:

<http://web4health.info/??/sfaq/groups-list.txt>

where ?? is the language. When this is written (12 Nov 2002) only group titles in Swedish (sv) and English (en) are ready. The part before the “-”, however, is always in English.

The *Group* for a particular FAQ is found automatically by taking the part before “ - ” in the groups list, and matching it against the beginning of the ID of the FAQ. The first found match is used.

The value of *Group* is only used in generating the list of all FAQs whose current content can be found at

<http://salut.dsv.su.se/w4h-faq-??/faq.exe>

where “??” is replaced by the language (in November 2002 only Swedish, se, and English, en). It is not used automatically to produce the subject trees.

6 References

Sneiders, E. and Larsson, K. (2001)	Application and Maintenance Aspects of an FAQ Answering System. Report series No. 01-007, Department of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, Sweden
Sneiders, E. (1999)	Automated FAQ Answering: Continued Experience with Shallow Language Understanding. <i>Question Answering Systems. Papers from the 1999 AAAI Fall Symposium</i> . Technical Report FS-99-02, November 5-7, North Falmouth, Massachusetts, USA, AAAI Press, pp.97-107, http://www.dsv.su.se/~eriks/Sneiders1999.pdf .
Sneiders, E. (1999)	Question Answering by Automated FAQ Retrieval. <i>Proceedings of the Workshop on Futures in Information Systems and Software Engineering Research</i> , April 1999, Stockholm, Sweden
Sneiders, E. (1999)	<i>Automated FAQ Answering on WWW Using Shallow</i>

	<i>Language Understanding</i> . Thesis in partial fulfillment of the requirements for the degree of Licentiate of Technology. Stockholm University / Royal Institute of Technology, Sweden
Sneiders, E. (1998)	FAQ answering on WWW using shallow language understanding. <i>Information Systems in the WWW Environment. IFIP TC8/WG8.1 Working Conference</i> , 15-17 July 1998, Beijing, China, Chapman & Hall on behalf of IFIP, pp.298-319
Sneiders, E. (1998)	Prioritized Keyword Matching of Natural Language Sentences in Database Querying. <i>Proceedings of the Second Conference on Information Management Methodologies</i> , April 1998, Växjö, Sweden
Palme, J (2002A)	KOM2002 Groupware Use in Web4Health, KOM 2002 report

D 2.1B Dreamweaver templates for Web4Health web site design, by Jacob Palme. KOM2002 deliverable D 2.1B.

D 2.1C Naming of Web4Health Informational Pages, by Jacob Palme, KOM 2002 deliverable D 2.1C.

Appendix A: Technical Description of the Question-Answering Algorithms

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 emphasize 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 fourth 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):

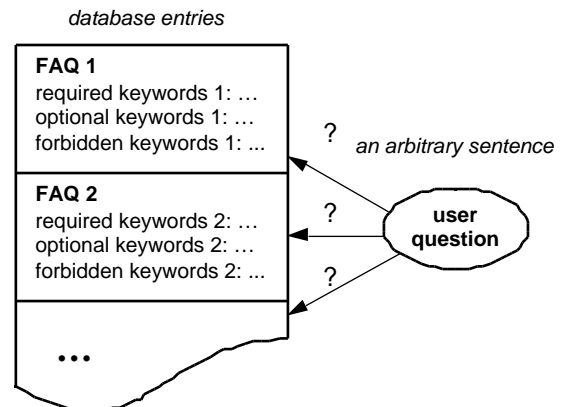


Figure 3 Input to the algorithm: an FAQ entry with identified keywords and an arbitrary user question.

- 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.

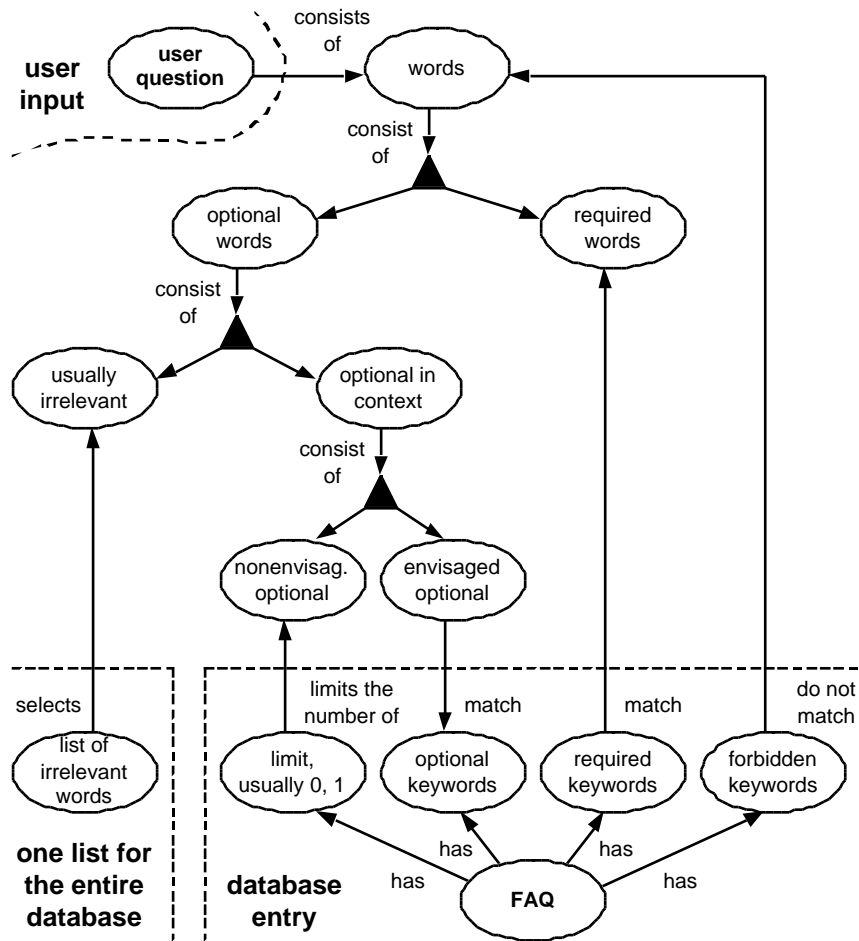


Figure 4 Concepts involved in Prioritized Keyword Matching and the relationships between them if the FAQ answers the user question.

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 lose 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.

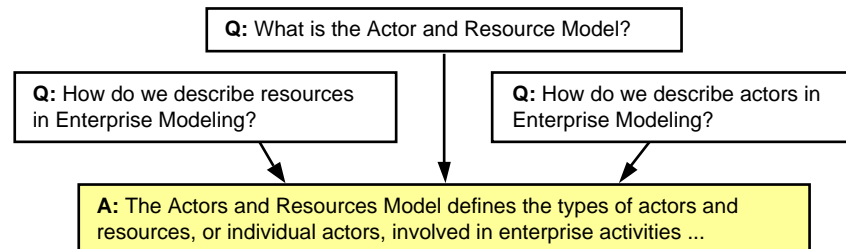


Figure 5 Three questions having the same answer.

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., "modeling" vs. "modelling", "formula_s" vs. "formulae", "analyze" vs. "analyse");
- 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 apply Enterprise Modelling?" vs. "What are the cases of application of Enterprise Modelling?" vs. "When is Enterprise Modelling applicable?";
- words that are not ordinary language synonyms, but act like synonyms in a particular context: "Why is Enterprise Modelling beneficial?" vs. "Why do we use Enterprise Modelling?";
- generalization and specialization of a concept (not common).

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 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 of 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.

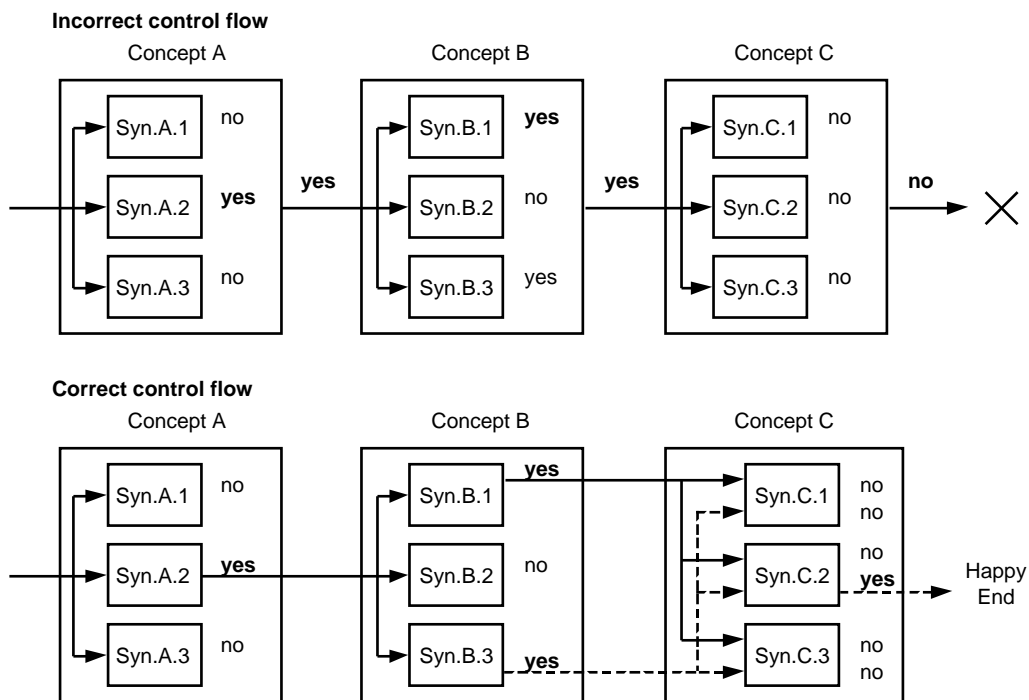


Figure 6 Incorrect and correct control flows of matching a phrase.