



Knowledge Management Best Practices and Adaptive Learning

Efficiently using Knowledge Management and adaptive learning in Service Manager



Introduction	2
Explanation of Terms	2
Adaptive Learning	2
Exact Match	2
Hot News	2
Knowledge Centered Support (KCS)	2
Phrase	2
Stemming	3
Stopwords	3
Thesaurus	3
Wildcards	3
Word density	3
How does the search engine interpret the search text entered?	3
What are considered "stop-words"?	4
How does the thesaurus work?	4
How are file names handled?	4
Is Knowledge Management case sensitive or case insensitive	4
What determines the position of a document in the hitlist based on its terms?	4
How does adaptive learning information influence the position in the hitlist?	5
How can Service Manager influence the weight of a document?	5
What to do about aging / retiring documents?	6
What are best practices for performance when searching using Knowledge Management?	6
For more information	8

Introduction

The Adaptive Learning feature was introduced to Knowledge Management to give the knowledge administrator the opportunity to promote a document in the hitlist based on similar words that were used to find the document or by adjusting the weight of a certain term in the document. Adaptive learning in Knowledge Management allows the administrator to manipulate which documents will be found based on the search criteria. This document will discuss only adaptive learning as used in Service Manager Knowledge Management.

Explanation of Terms

Adaptive Learning

In adaptive learning, the document content is adjusted manually or through software to manipulate the search results to include information gathered in prior successful searches. Service Manager's Knowledge Management enables adaptive learning automatically by entering successful search terms into an adaptive learning record whenever a found document was used as a solution, or by manually entering adaptive learning information. The adaptive learning record introduces new keywords to the document and is indexed along with the related document.

Exact Match

An exact match search will return any document that has the exact word(s) entered in the search. For example when searching for cars, cars will be an exact match, while car will be a stemming match.

Hot News

Knowledge documents that inform the customer service representative or the customer on current outages or workarounds to current issues are called hot news. Hot news is typically displayed continuously for the time it is relevant and is not usually searched by a knowledge search. Adaptive learning should not be used to promote a hot news article to a higher priority than all searches, but rather hot news should be displayed as a marquee in Service Manager or as a link on the IT website.

Knowledge Centered Support (KCS)

Knowledge Centered Support (KCS) is a methodology and a set of practices and processes that focuses on knowledge as a key asset of the customer/technical support organization. Its premise is to capture, structure, and re-use technical support knowledge.

KCS seeks to:

- Create content as a by-product of solving problems
- Evolve content based on demand and usage
- Develop a knowledge base of an organization's collective experience to date
- Reward learning, collaboration, sharing and improving

KCS is usage and demand driven and thus provides on-time knowledge.

Phrase

A phrase is a related group of words. In the Knowledge Management search, a phrase is defined by a set of words enclosed in double quotes, such as "this is a test search". When searching for a phrase, the exact same words in the exact same sequence will be searched in documents and returned. Phrases are only taken into consideration when entered in a search, text entered in a document or via adaptive learning that is within double quotes is not considered a phrase.

Stemming

Each word has a single stem, namely the part of the word that is common to all its inflected variants. For example car is the stem for the words cars, phonetic is the stem for phonetics and phonetically. The Search Engine performs stemming for all search words in the query and returns both exact matches – at a higher weight – and stemmed matches in the hitlist.

Stopwords

Stopwords are any of a number of very commonly used words, such as a, and, in, and to, that add no real meaning to the search and thus are normally excluded by computer search engines. When searching with Service Manager Knowledge Management, all stop-words are automatically stripped from the query prior to sending the query to the Search Engine.

Thesaurus

A thesaurus is a dictionary of synonyms (words with similar meanings) and antonyms (words with opposite meanings).

The Search Engine thesaurus is used to return documents with similar words in addition to exact matches. For example, searching for car will return automobile as well. The thesaurus can be found in the Search Engine install directory under /K2/common/uni. To add new words or change existing synonyms, the Search Engine thesaurus has to be decompiled as described in the online help, then edited and recompiled.

Wildcards

A wildcard is a character that may be used in a search term to represent one or more other characters. In Service Manager's Knowledge Management, the * symbol can represent a variable number of characters. Searching for *ing will return all documents that have the words ending in "ing".

Word density

Word density is highest when all search words occur in the same sentence. It is still considered high, if all search words are in the same paragraph. The word density score is lower, if all words in the search occur in the document, but further apart. Word density is used by the search engine to determine how high a priority the result will get in the document hitlist.

How does the search engine interpret the search text entered?

The Search Engine's business is to figure out what the user was looking for when entering a query by using the natural language query translation, looking at word density, word proximity, thesaurus lookups, etc.

First the query statement is analyzed and transformed as follows:

- White space is removed
- Defined stop-words are stripped
- Wildcards are identified and parsed separately
- Phrases are identified and parsed separately
- Weight is assigned in the KMQuery JavaScript record as follows:
 - Exact matches to the search words are assigned an accrual of 45
 - Exact matches to search phrases are assigned an accrual of 45
 - Keywords are assigned an accrual of 35 (this feature is not enabled out of box)

- Words that were found via the Thesaurus are assigned an accrual of 30
- Words based on the same stem are assigned an accrual of 14

What are considered “stop-words”?

For detailed information on how to manage stop-words, please refer to the online help section “Manage Stopwords” in the Knowledge Management section of Service Manager.

Special characters such as [] () < > - + ! , . / \ ' " & ? = are also removed from natural language queries as stop-words since they have special meaning to the search engine. To search for special characters, enclose them in quotes, for example “[]”. It is not possible to search for a double quote as part of the simple search. Double quotes always assume a phrase is enclosed.

In the standard search box “and”, “or” and “not” are considered stop words. They can be used as Boolean operators in the advanced search.

How does the thesaurus work?

Instructions on the thesaurus can be found in the online help under Knowledge Management - Knowledgebase administration and maintenance - Search engine thesaurus.

In the script library “KMQuery”, Verity tags are added to request thesaurus expansions on terms as well as stems for our terms. Synonyms are weighted less than the actual term the user entered. Stemmed variations of the original term are weighted higher, ie: search term “car”. The stem “cars” is weighted nearly as high as “car” and higher than “automobile”. In our documentation, we provide instructions for modifying the default thesaurus as well.

The feature to search for alternative terms for misspelled words is not available in Knowledge Management.

How are file names handled?

Attachments in knowledge articles do not have regular filenames since they are renamed to attachment1, attachment2, etc. Due to this, only the attachment’s content can be searched as its name is not indexed. If a term in a document refers to a filename and you want to search for the filename in the document body, you will have to include it as a quoted phrase such as “readme.txt”.

Is Knowledge Management case sensitive or case insensitive

The Search Engine attempts to match the case sensitivity provided in the query expression when mixed case is used. For search terms entered completely in lowercase or completely in uppercase, the search engine looks for all mixed case variations. Search terms with mixed case automatically become case-sensitive. For example, a query on Apple would find only the precise string Apple, while a query on apple finds all of the following: APPLE, Apple, apple. A query all in uppercase does not turn on case-sensitive searching. A query on APPLE finds all: APPLE, Apple, apple.

What determines the position of a document in the hitlist based on its terms?

The Search Engine takes many parameters into account. The weight is influenced amongst others by word count in this document versus the word count in the collection, as well as word proximity. Additionally Service Manager manipulates the weight based on whether it was a full match or a thesaurus or stemming match.

All words that were part of the search will be highlighted in the hitlist, whether they were direct matches, found via the thesaurus or stemming. Terms will not be highlighted in the found documents when opened, just in the summary of the hitlist.

How does adaptive learning information influence the position in the hitlist?

The weight of kmadaptivelearning is less than terms contained inside actual knowledge documents. When taking into consideration word density, exact matches, stemming, number of times a particular term is present within the document and/or attachments and the actual term being searched on the document may only move up two places within the hitlist. Adaptive learning will not take a document that would be on page 8 of the hitlist and move it to the first position on page 1. Additionally, adding more counts to a term in kmadaptivelearning can hamper the indexing itself because that term is written out that number of times before it is sent to the search engine. While kmadaptivelearning can assist in positioning documents within the hitlist, it cannot replace well authored Knowledge Documents.

Adaptive learning is not intended for adding weight to documents but is a way to add searchable terms that are not contained in the document. For example, a document may contain information about "Windows Server 2003". Most users may search for "Win2k3" but will not get a hit on this document unless the admin has added "Win2k3" to the thesaurus. If "Win2k3" is added to the adaptive learning table, the term "Win2k3" will now be indexed with this document. Adding "Win2k3" multiple times will only slightly boost the relevance.

Adaptive learning works on a per document basis unlike the thesaurus that is set globally. It helps people find the document they were searching for by adding specific keywords to the adaptive learning record.

The adaptive learning document is scored separately and then normalized with the document body's score and any attachment score. By altering the number of times a word is written in the adaptive learning file, you are changing its score which is a small part of the overall score when combined with the document body and attachments. If you give a word a 500 in adaptive learning, it will add that word 500 times to the document, all in the same area. If the search engine sees 500 words right next to each other, it assumes that the word was entered too often and cannot be as important as *500. A lower weight will be added relative to the number of times the word was included.

The exact relevance ranking formulas are proprietary to the search engine and are not available to HP.

How can Service Manager influence the weight of a document?

The KMQuery Script library is used to process Natural Language Queries to Verity Query Language. During this process, stop words are removed, phrases and wild cards are parsed out, etc. When the Verity Query is built, modifiers are added for scoring. For example, a word that matches the search criteria exactly gets a score of 45. To add a weight of 50 to the word "test" and a weight of 80 to the word "help", the following syntax is added to the query that is sent to the search engine:

```
<ACCRUE> ([50]<WORD>(test), [80]<WORD>(help))
```

ACCRUE is a Verity Query Language (VQL) concept operator that selects documents which include one or more of the search elements specified. The score will increase with more search elements that are present.

In the KMQuery ScriptLibrary record, the weight value of the exact same word from the query word list is set to 45. The weight value of a word found from the thesaurus for a word from the query is set to 30. The weight of a word that has the same stem as a word in the query list is set to 14. The weight of a phrase exactly the same as a phrase in the query is set to 45, and the weight of a keyword that matches a word in the query (keywords are defined on the Verity side, not inside SM) is set to 35.

For example on a search for "Test windows printing print "windows print" print*" the VQL will be:

```
<ACCRUE>([45]<Lang/en><ACCRUE>(<MANY>test, <MANY>windows, <MANY>printing,
<MANY>print ),[35]<ACCRUE>(test<IN>SecondaryDoc, windows<IN>SecondaryDoc,
printing<IN>SecondaryDoc, print<IN>SecondaryDoc ),[45]<ACCRUE>("windows
print",[30]print*],[30]<ACCRUE>(<Lang/en><Thesaurus>test, <Lang/en><Thesaurus>windows,
<Lang/en><Thesaurus>printing, <Lang/en><Thesaurus>print
),[14]<ACCRUE>(<Lang/en><MANY><STEM>test, <Lang/en><MANY><STEM>windows,
<Lang/en><MANY><STEM>printing, <Lang/en><MANY><STEM>print ))
```

To add weighting based on whether a field occurs in a specific field, such as the title, the following addition to the KMQuery JavaScript can accomplish this:

```
// to add weight for a field or exact phrase in title:
// initialization section of the nlqToVQL function
manywordInTitle="[60]"

// add the following section at the end of the
// for (var words =0; words < wordarray.length; words++)
// loop in the nlqToVQL function
    manywordInTitle += wordarray[words];

// if (words == wordarray.length-1)
// manywordInTitle += "<IN>Title ";
//     where Title is the field name as defined in Manage KB
//     - use Alias to make sure that all Title fields are the same.
// else
// manywordInTitle += "<IN>Title, ";
*/
    vql+=manyword+", "+thesaurus+", "+manystem+", "+manykeyword;
// vql+=manyword+", "+thesaurus+", "+manystem;

// add "manywordInTitle here to add it to the vql:
vql+=manyword+", "+thesaurus+", "+manystem+", "+manykeyword + ", " +
manywordInTitle;
```

What to do about aging / retiring documents?

Documents have an expiration date that is typically set when the knowledge is contributed. Expired documents can be searched and then reviewed or archived. Feedback should be monitored by the knowledge administrator and every document that receives negative feedback should be reviewed and updated or archived.

Knowledge Centered Support (KCS) is demand driven. People will quit searching for data in outdated documents. If you want to clean up your Knowledgebase occasionally to remove these outdated documents, you can check the usage statistics and, if a document does not get used for half a year or a year, you should retire the document. The kmusagestats and kmusagehistory tables can help with finding outdated documents, but finding them and retiring them is a manual effort.

What are best practices for performance when searching using Knowledge Management?

The more detailed the query is, the more time it needs to execute. Any additional selection criteria from the advanced search will affect query performance, since it will be added to the original query statement.

Knowledge Management uses a full text search engine that is most efficient doing text searches not fielded queries. When you do a fielded query, it will do a field by field comparison for every document in the collection or collections, which is very inefficient and is equal to an un-indexed query

in SQL. To visualize the effect imagine that an indexed query is like a set of bowls filled with M&Ms each color having its own bowl. To find a yellow M&M, you just have to take it out of the yellow bowl. This technique is what the search engine does for full text searches. In comparison to that, searching for yellow in a mixed bowl of M&Ms, with all colors together, you will need to pick up every M&M and see if the color attribute of the M&M is equal to the required color value of yellow. When you do an advanced search on a specific field, the Search Engine will have to work that way as well, since the field is not specified in the index and thus we need to compare the value for every single record in the set.

In addition, when doing a fielded query, the Search Engine has a 32K limit on data it can search within a specific field. To get around the 32K limit, you can add any field you want to include in your text searches in the document body using the Manage Knowledgebases menu option. The document body does not have the 32K size limit.

For more information

Please visit the HP OpenView support web site at:

<http://www.hp.com/managementsoftware/support>

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note: Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following URL:

<http://www.managementsoftware.hp.com/passport-registration.html>

© 2010 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained in examples in this document regarding OpenSSL technology is provided by Hewlett-Packard Development Company, L.P. as a courtesy to our customers and partners. This documentation does not replace an OpenSSL reference, and HP encourages you to conduct additional research regarding OpenSSL technology by consulting with sources outside of this document. HP hereby disclaims all liability associated with the use and accuracy of this information. As OpenSSL technology evolves, HP may or may not update this reference.

Peregrine Systems, Service Manager, and Evolve Wisely are registered trademarks of Hewlett-Packard Development Company, L.P. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. SiteMinder is a registered trademark of Computer Associates International, Inc. Sun Microsystems, Java, JRE, and JDK are trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of the Open Group. RSA is a registered trademark of RSA Data Security Inc. WebSphere is a trademark of International Business Machines Corporation in the United States, other countries, or both. WebLogic Server is a registered trademark of BEA Systems, Inc. All other trademarks are the property of their respective owners.