Retrieval and Classification (TREX)
Features and Functions

Retrieval and Classification (TREX) 5.0

Copyright

1 Knowledge Management: Organizing and Finding Information

Knowledge – a fundamental factor in production

Knowledge is one of the most important production factors. However, the lack of knowledge and information is constantly becoming more obvious. Information exists in numerous file formats that are stored using different kinds of media and in widely distributed storage locations. Because such information is often hidden in the darkest, most inaccessible corner of a company, it can be impossible for companies to retain an overview of these piles of information and to make the best use of their resources. This causes a situation in which many companies do not even know the full extent of their knowledge and information. The average company has enough information to fill a medium-sized library, but this information often lies unexploited simply because it can not be tapped and used efficiently. A company that is able to access this wealth of knowledge quickly and systematically has a considerable market advantage.

Searching for information in natural-language text documents

Most information available in a company exists in the form of descriptive, natural language text documents: Letters, internal correspondences and notes, emails, presentations, tables, and so on. These document, stored in different data sources and formats, all contain text in some form or another. However, it is difficult to use the content of the documents efficiently since it is not structured. In contrast to information and data stored in a structured manner in a database, where the content can be easily found, these documents have mostly unstructured content in different file formats. It is possible to find such text documents in a file system using their metadata or document attributes – file names, file name extension, creation data, author, and so on. However, the user can only make a judgement on the content of documents when he has opened them using an applicable application and read them. This procedure quickly becomes impractical in the case of large document collections.

This gives rise to the task in hand: To tap into the contents of text documents by structuring and classifying it so that the relevant information is available to the company for the everyday work and decision-making of employees. This can make it possible to reach company-relevant information quickly, to make it available for use in business communication, knowledge and content management, and to convert it into organized knowledge.

2 What is TREX; and what can it do?

TREX provides intelligent search and classification functions

The information retrieval system RETRIEVAL AND CLASSIFICATION (TREX) provides various software applications with a wide spectrum of intelligent search, retrieval, and classification functions for documentation development. You can use TREX to search extensive electronic collections of text documents flexibly, and to structure document classification in a way that gives a clear overview of what is available. The TREX text-mining functions allow interesting and relevant information to be extracted from text documents for the user.

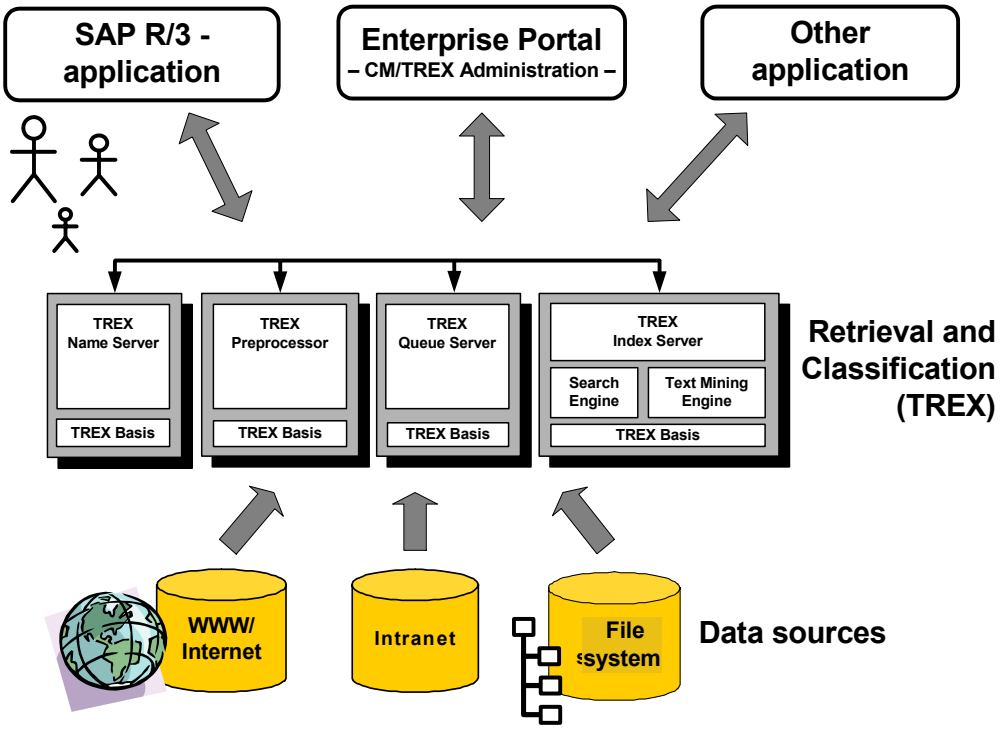
TREX supports many file formats and language.

In principle, TREX can process, search, and classify any file format that can be rendered as text. Filter software integrated into TREX converts all current standard document formats (HTML; XML; DOC; TXT; RTF; PPT (Microsoft PowerPoint), XLS (Microsoft Excel) and PDF (portable document format) files and so on) into text. Text documents in numerous European and non-European languages can also be processed by TREX. All central and western European languages are supported, as are Korean, Japanese, and Chinese.

3 TREX as part other applications

Where is TREX implemented?

Because TREX does not have its own graphical user interface (GUI), it is integrated into other applications and called up and used from them. In this way, TREX provides an effective and powerful retrieval and text-mining tool for a myriad SAP applications. As a fundamental part of knowledge management within SAP Enterprise Portal, TREX can search in unstructured information within text documents in any form, and structure documents found in the World Wide Web or in local datasets, and classify them. However, many other SAP applications also use the TREX retrieval functions to find all types of unordered and unstructured content in the form of text documents. These include SAP CRM (CUSTOMER RELATIONSHIP MANAGEMENT) INTERNET SALES, SAP B2B (BUSINESS TO BUSINESS PROCUREMENT), SAP DB (SAP Catalog), SAP MARKETS, SAP KW (KNOWLEDGE WAREHOUSE), SAP DOCUMENTATION, BW (BUSINESS INFORMATION WAREHOUSE), SAP PLM (PRODUCT LIFECYCLE MANGEMENT) and SAP SYSTEM.



Using TREX in other applications.

4 TREX Core Components: TREX Search Engine and TREX Text Mining Engines (TREX Engines)

The information retrieval functions are realized in RETRIEVAL AND CLASSIFICATION (TREX) by the two TREX core components. These are the TREX SEARCH ENGINE and the TREX TEXT-MINING ENGINE. These two components carry out the actual retrieval tasks.

4.1 The TREX Search Engine – Fundamental Search and Retrieval Functions

TREX search engine: Fundamental search functions

The TREX search engine provides fundamental search functions similar to the standard search engines in the Internet. For example, if a user does an *exact search*, only documents that contain the exact search term are found. However, you can search for phrases as well as for individual words. The *Boolean search* allows you to use Boolean operators (AND, OR, NOR) to build more complex search queries. If only parts of a term are known, you can use the placeholders '?' and '*' (*masked* or *wildcard* searches). You can also search for words or phrases that are similar to the search phrase (*fuzzy* or *error-tolerant* search). Words that represent a language variant of the search query can also be found (*linguistic search*). A document can also have metadata or document attributes (for example, author of document, creation date, change date) where important documentation is stored. You can search by these document attributes as well as by terms in the document text. This is called an *attribute search*.

The individual retrieval functions of the TREX search engine are in detail:

1) An **exact search** for individual words or phrases (**phrase search**)

2) A **Boolean search** for individual words or phrases

3) A **masked** or **wildcard** search

4) A **fuzzy** or **error-tolerant** search

5) A **linguistic search**

6) An **attribute search** or **search for document properties**

4.1.1 Exact Search for Individual Words or Phrases (Phrase Search)

The exact search for an individual word is the standard function of a search engine. This quick and efficient search finds documents that contain the term (word or phrase) specified. The information found has to match the specified search term exactly, although lowercase, uppercase, and accents are not taken into account. You can often obtain better search results if you enter groups of words or a phrase instead of a single word. When you carry out a phrase search, any amount of word groups can be entered.

Example of an exact phrase search: If you do an exact search for <"SAP AG Walldorf">, only documents that contain the exact same phrase are found. The inverted commas signify that the words are to be treated as a phrase.

4.1.2 Boolean Search for Individual Words or Phrases

A Boolean search allows you to link individual search terms by using Boolean operators such as AND and OR. These operators must be written in uppercase; otherwise they are treated as search terms.

Example of a Boolean search for individual words: A search for <President AND USA> only finds documents that contain both words. A search for <President OR Minister> finds documents that contain either word.

Search terms for Boolean searches can be single words or word groups. Word groups are handled like single words and can also be linked to other terms using Boolean operators.

Example of a Boolean search for word groups: A search for <"SAP AG" AND “USA“> finds documents that contain SAP AG **and** USA. To structure complex expressions, use parentheses ( ). Remember to leave a space before and after every parenthesis. The query <President AND (Bush OR Clinton)> finds documents that contain President in addition to either Bush or Clinton, or that contain all three words.

4.1.3 Masked or Wildcard Search

The masked or wildcard search enhances search options by allowing the use of placeholders (also called truncation marks or wildcard characters) for part of the search term. The * asterisk is used as a placeholder. It represents any amount of letters in the search term, and can occur at the beginning, end, or in the middle of a word.

Example of a masked search: A search for <**Comp***> finds documents that contain Company. A search for <*Comp*code*> finds documents that contain Company Code.

4.1.4 Fuzzy or Error-Tolerant Search

The fuzzy or error-tolerant search is another important TREX search engine function. The fuzzy search includes non-exact, similar search terms and provides more flexibility than the exact search. This search method finds information even if your search query does not exactly match the document contents. This finds words or phrases that are only similar to the search term or phrase. This is especially useful if you are not sure of the exact spelling of a word, if you make a spelling mistake when entering a search term, or if a spelling mistake has been made in the document that you are looking for.

Example of an error-tolerant search: A search for <presidant> (typing error) also finds documents that contain the correct spelling <president>. A search for Meier also finds Meyer and Maier.

4.1.5 Linguistic Search

A linguistic search uses linguistic aids for the search. This also finds terms that are linguistically related to the search term and that have the same root as the search term (reduction to root).

Example of a linguistic search: A search for <mice> also finds <mouse>. If a document contains the sentence <The mice were caught in the trap>, the search query <mouse> or <catch> will find it.

4.1.6 Attribute Search or Search for Document Properties

A document can also have metadata and document properties such as the author, title, creation date, change date, and size, that contain important information. You can also search by this metadata as well as by words in the document text. These document properties are also called attributes, which is why this search is sometimes known as an attribute search. A prerequisite for this search is that the metadata has been maintained appropriately for the document set that is being searched. Each document property has a name (for example,<`author`>) and a corresponding value (for example, <`Schroeder`>). It is also possible to combine an attribute search with the search for document content.

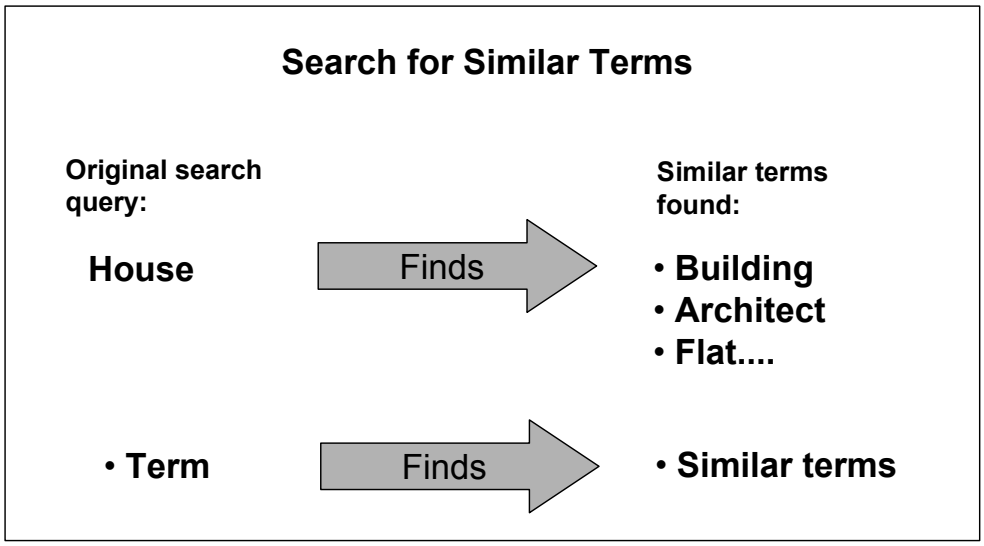
4.2 The TREX Text-Mining Engine – Enhanced Retrieval and Classification Functions.

The TREX Text-Mining Engine – Enhanced Retrieval and Classification Functions

The TREX text-mining engine enhances the functions of TREX by reducing document content to meaningful words with the help of text operations such as reduction to root form, normalization of spelling, the removal of stop words, and so on. This improves retrieval results. The search results are refined because the relevance of the documents can be evaluated differentially. This is possible because the similarity between the documents is calculated on the basis of the tried and tested **vector space model**: Each document and term in the document set is represented by a multi-dimensional space. This representation allows TREX to carry out enhanced retrieval and classification methods. In this way, documents can be ordered sensibly according to certain criteria by being assigned automatically to a generic category or class (***document classification***). When a document is found, the user also receives a list of key words that give an overview of document content and help to characterize the document and distinguish it from others (***determination of key words***). The user can also look for documents that are similar to the one found. This helps to restrict the number of documents that may be relevant (***Search for similar documents***). TREX offers terms that are similar to the original search term in content and meaning. The user can then reformulate the search request (***Search for similar terms***).

The following are the TREX text-mining engine functions in details:

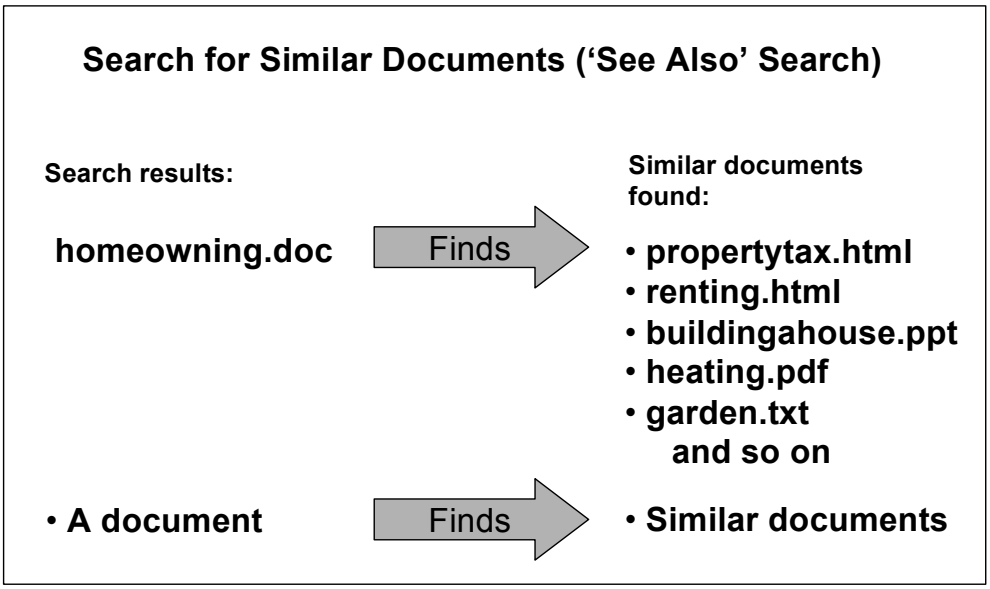
1. **Search for similar terms**
2. **Search for similar documents and classes (‘See Also’ search)**
3. **Determination of key words (feature extraction)**
4. **Classification of documents**



4.2.1 Search for similar terms

When searching for similar terms, the system lists all words that are similar to the search term. This can include synonyms. The similarity is calculated according to how many times the terms in question appear in the same document as the search term originally entered. For example, if you analyze a large number of press reports, you will find that the term ‘Telecom’ is often associated with ‘shares’ and the ‘stock market’, or that the term ‘oil prices’ is often connected to ‘taxation’ and ‘costs’. These words will often appear together within the same document.

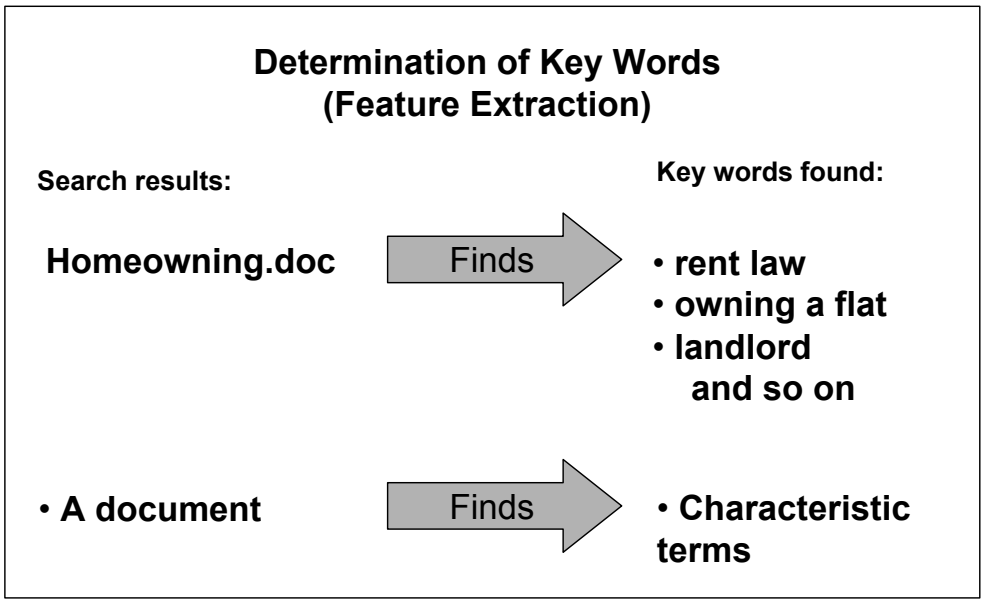
The result is a list of related words or phrases that serve as the basis of a new search query. If the original query did not obtain the results you wanted, you can easily formulate a new, more precise search query using similar or related terms.



4.2.2 Search for similar documents and classes ('See Also' search)

When an original search has found a number of documents, a search for *similar documents* offers the option of selecting certain documents within the search results, and using them to search for other documents that are similar to those selected for the search. This search uses complete documents as the basis of the search query, instead of simply one or more search terms, as was the case for the original search. This allows the user to find similar documents even if the original search term is not contained in them. If, for example, the search for 'President' finds documents with information on the 'American presidential elections' and 'Peace talks in the Middle East, you can select the documents found on the peace talks, and start a search for similar documents. The new results may no longer contain the original search term 'President', but they will reflect the new context of the search and intelligently enhance search options.

The 'See Also' search allows you to search not only for documents that are similar to documents found, but also for documents that are similar to a predefined group of class of documents and for similar document classes for a predefined class (*Similar Classes*).



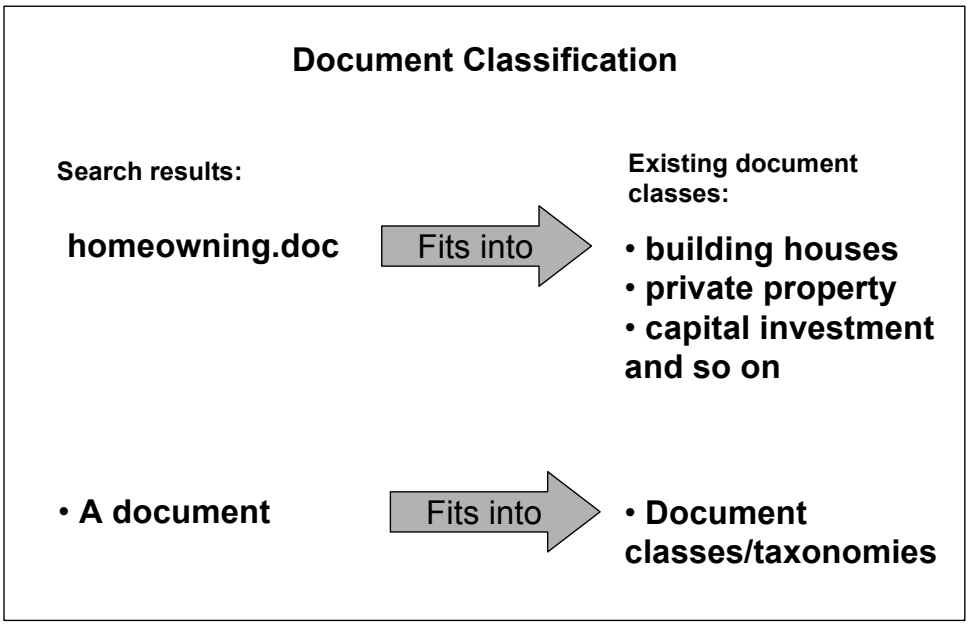
4.2.3 Determination of key words (feature extraction)

When determining key words for a document (feature extraction or *document features*), characteristic words or phrases that describe and identify the document content are also determined for each document found. These key words give a concise overview of the document content and help the user to distinguish the document from the mass and evaluate its significance for the search query. The extracted key words can also serve as the basis for a new search.

Characterizing the contents of a document by using key words is often better than only displaying the first lines of the document, as happens with many search engines. When you obtain a range of documents when searching for the term ‘President’, one document found may have the key words ‘Palm Beach’, ‘Vote’ and ‘Republican’, whereas another may have ‘Palestinians’, ‘Israel’, and ‘Peace talks’. Obviously the content of the two documents is completely different, and yet both documents might being with the words ‘Washington – The president of the United States...’. The key words determined can therefore help to identify the thematic content of a document.

Characteristic key words can be determined and extracted not only for individual documents, but also for groups or classes of documents (*Class Features*). Feature extraction is an efficient way of quickly obtaining a picture of the contents of a document or of a class of documents.

The TREX text-mining functions that have been mentioned up until now, feature extraction, search for similar terms, and ‘See Also’ search (search for similar documents and classes), can all help to sensibly and intelligently enhance the thematic area of the original search, and to create a new search query in the information retrieval system if the original search query did not return the required results.



4.2.4 Document Classification

Documents that are being added to a document collection can be assigned to one or more existing document classes (categories) according to prescribed criteria by the TREX text-mining engine for classification purposes. TREX calls this function *Classify Documents*. TREX analyzes the new document and then returns a list of proposals for classification/categorization. The user can then choose the categories in the list to which the document is to be assigned. Initial document classes can be manually defined by the user using the TREX *Document Clustering* text-mining function (see *Creating a Taxonomy*). A taxonomy is an ordered hierarchy of document classes. You can also classify new documents automatically. A prerequisite for this is that there are existing document classes with sufficient correctly classified documents that meet the classification criteria. In this case, new documents are automatically assigned to one or more document classes according to the prescribed criteria. When indexing new documents, the classes are then continuously 'learned'.

This type of classification, using sample documents, is called example-based classification or EbC. An alternative to example-based classification is query-based classification (QbC). In the case of query-based classification, documents are assigned to prescribed categories or taxonomies in a category based on Boolean search queries. A sequence of terms, linked with Boolean operators, defines the taxonomy nodes (the classes in a class hierarchy).

Conceivable usage scenarios for automatic document classification.

There are various usage scenarios for automatic document classification in which these retrieval functions make particular sense.

- Expert finder: Classification of people according to specialist area based on the documents they use.
- Automatic classification and categorization of questions on a bulletin board.
- Automatic classification of e-mails into categories such as ‘invoices’, ‘orders’, and ‘complaints’.
- Automatic forwarding of incoming documents.
  - Forwarding customer mails to the department that is responsible for them.
  - Forwarding announcements to people who are likely to be interested according to interest profiles.
  - Automatic classification of incoming e-mails for a customer service center, and forwarding them to the appropriate experts.
  - Automatic classification of incoming e-mails and the triggering of a workflow for them.
  - Merging two catalogs (two hierarchies) by finding similarities and redundancies.