Document Classification in TRIP

White Paper

Abstract:
This document describes classification techniques in general, and how those techniques are applied and exposed within the TRIP product framework.

End User License Agreement

All rights to this software, its documentation and logotypes of the TRIP product family and software (altogether “Software”) supplied by Smaser AG (Smaser) are exclusively owned by Smaser.

The transfer of this Software, solutions or parts thereof requires the prior written agreement of Smaser. Furthermore, the customer has the right to use licensed Software and / or process solutions supplied by Smaser to the extent specified in his contract with Smaser.

The free-to-use non-commercial version doesn’t require a prior written agreement with Smaser but such customers, organizations and/or third parties agree by using the software and / or solution of Smaser to be strongly obliged to keep all rights to this software, documentation and logotypes of the TRIP product family absolutely uninfringed and protected.

Table of Contents

Introduction

The act of classifying a document, i.e. attaching one or more "tags" to the document that define its place in some organizational metaphor, relies on two discrete processes: training and assignment. We use the term "scheme" to encompass whatever organizational metaphor the user decides to implement as part of this process.

Training, as the name suggests, consists of instructing the classifier in how to recognize those topics that best distinguish or describe the scheme's component structure. Typically, a user of such a system defines the scheme as consisting of some number of "categories" and then introduces the system to a variety of documents that the user has manually determined *a priori* to belong to one or more of these categories. The system uses this data to build a predictive model describing how to recognize and assign tags to uncategorized documents in the future.

Assignment, therefore, is the act of using the predictive model created during training to attach one or more category tags to an uncategorized document.

For example, a simple categorization scheme for news articles could be:

- Local
- International
- Financial
- Entertainment
- Miscellaneous

The user would create these categories, and then introduce the system to documents relating to each category, for example documents relating to domestic politics might be provided as training material for the "Local" category, whilst documents relating the latest scandals in the movie industry might be provided as training material for the "Entertainment" category.

General Framework

The design for document classification within TRIP introduces a number of new concepts and reuses a number of existing concepts to accomplish the tasks of maintaining classification schemes, and assigning tags.

Specifically, classification schemes are defined within special purpose TRIP databases, known as classification containers. The exact structure of each container is defined by the classification algorithm in use by the system (which can vary), but follows a general structure whereby the container provides a map between category names and category IDs.

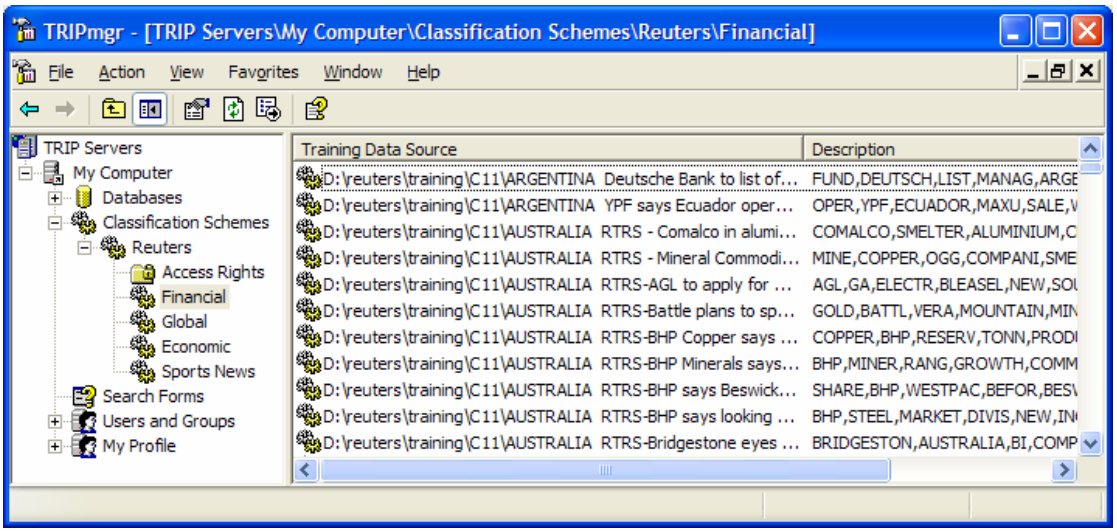
A container is associated with a database by the database's FM and whilst each database may only reference a single classification container, a single container may be used to classify documents in many different databases.

Tag assignment is performed at index time as part of the normal process of database maintenance. Whenever a database has an associated classification container, the classification process is invoked for all new and modified records found when scanning the database for updates.

Searching within databases that have been classified (i.e. databases that have an associated and trained classification container, and that have been re-indexed to take advantage of such a container) is accomplished with a new search function as described in the section on CCL modifications, below.

Classification in TRIPmgr

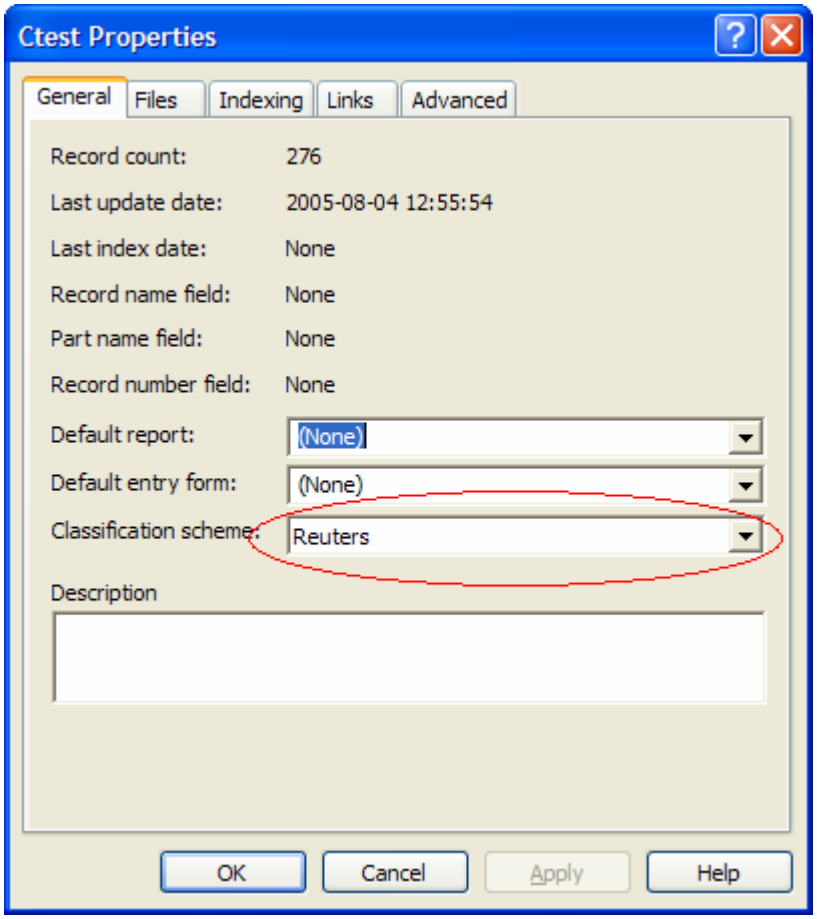
The TRIPmgr interface exposes classification in terms of schemes, categories within schemes and training data associated with categories. For example, the following screen shows a sample scheme that has four categories, each of which has been trained with a series of documents:



Access rights are defined as normal (and everybody who needs to interact with the scheme must have read access, of course), whilst categories can be trained one document at a time, or by reading an entire folder of documents.

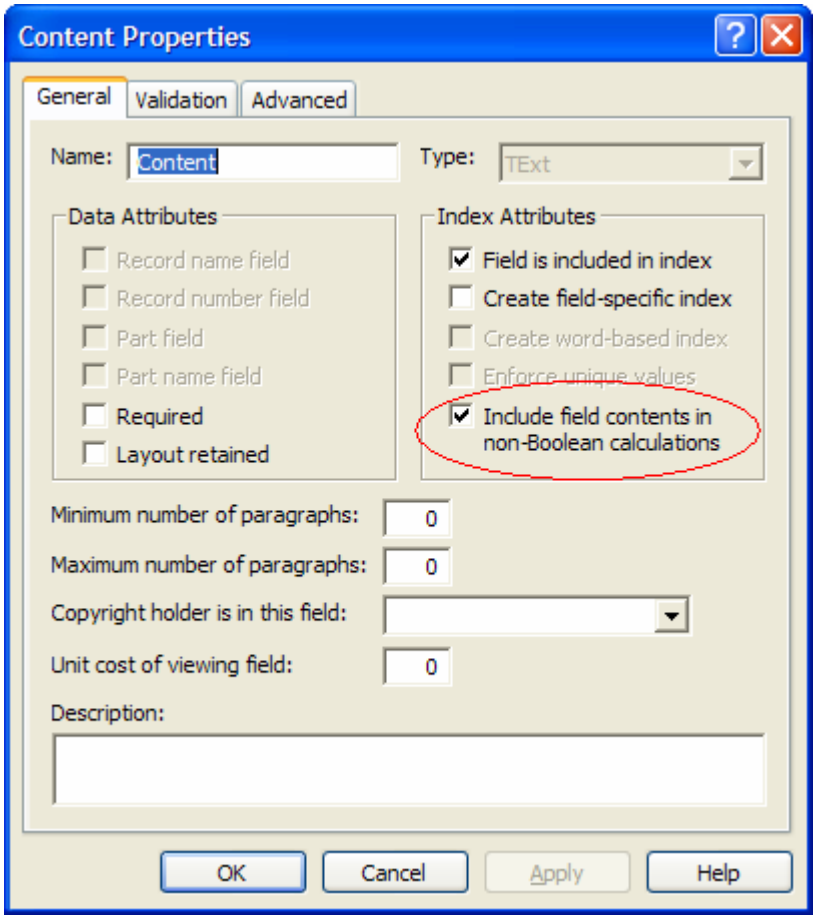
Likewise, categories can be created one at a time, or an entire folder tree of documents can be used in one operation to both create categories and then to train those categories with the documents in the folders corresponding to the categories created.

Assigning a scheme to a database is accomplished via the database properties sheet, which is modified as shown below:



Each record in such a database that gets indexed is assigned one or more category tags. When determining these category tags, TRIP uses a new field flag to decide whether to include each field in the database within the text that gets classified, as shown below.

Note that category tags are not written to the database's BAF, and so cannot be viewed using a report. All category tags are written to the database's index (BIF) during the index update process.



Scheme Definition

As described above, a classification scheme in TRIP terms is a special-purpose database. The exact structure of this database is defined by the classification algorithm configured for use, but all such databases share a common sub-schema, as follows:

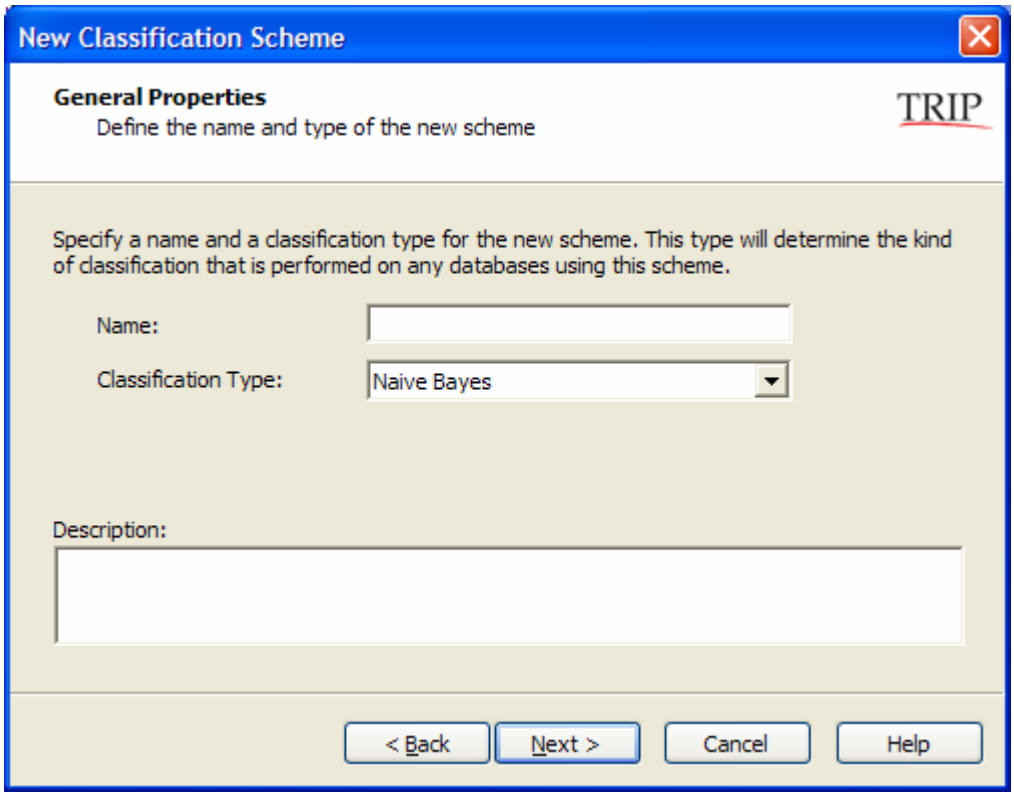
Name	Type	Required?	Indexed?
NAME	PHrase	Y	Y
COMMENT	PHrase	N	N
LABELS	PHrase	N	N
INFO	PHrase	N	N

The NAME and COMMENT fields are self-explanatory. The LABELS and INFO fields are tupled together and serve to provide a labeling mechanism for training data. For each item of training data that is submitted to the engine, the calling application provides a label (e.g. the file name of the original document) and the engine generates an entry in the INFO column containing the set of most-relevant terms from that document, typically high-occurring nouns from the document.

The category tags that are written to a database being classified are actually the record numbers of the corresponding category within the container. Using record numbers for tagging allows the name of any category to be changed at any time without requiring the database to be reclassified.

Creating a new scheme is simply a matter of deciding upon a name, and an algorithm that will be used to train categories and assign tags, as shown below. Currently, the only algorithm available is

Naïve Bayes, but this is intended to be expanded in the future to include kernel learning methods such as the well known Support Vector Machine.

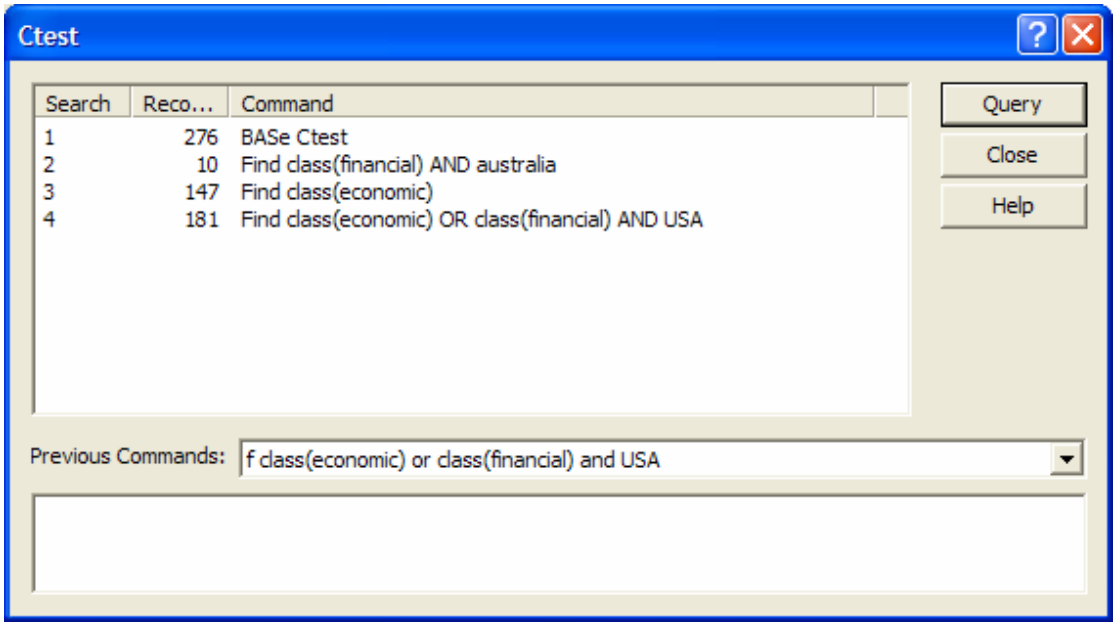


Searching using categories

To support category-based searching, CCL has been enhanced to include a new search function: CLASS(). This function finds records that have been assigned category tags, and can be combined within any Boolean expression as normal.

The search function accepts a string as argument, and this string is interpreted as the name of one or more categories within the current scheme (note that each open database may be assigned a different scheme and that this indirect mapping is performed for each open database separately). Once the names are matched, any category tags (record IDs) found are then located within the open database.

An example of this function being exercised is shown below.



Note that this search function does not produce specific hit locations within documents, thus a simple search for a category on its own (e.g. S=3, above) will not result in highlight points being shown in a report.

Naïve Bayes

The first algorithm implemented as part of the TRIP document classification framework is the popular Naïve Bayes algorithm. This technique aims to predict the probability of a new document being associated with a known category, and then simply picks the maximum of all such predicted associations.

The TRIP database specification for a Naïve Bayes container adds the following fields to the common sub-schema:

Name	Type	Required?	Indexed?
MAX	INteger	N	N
COUNT	INteger	N	N
DATA	TExt	N	N

The MAX and COUNT fields are used by the algorithm to record term frequency information for the category, whilst the DATA field is used to store a concordance of term occurrence information (i.e. a dictionary of terms and counts).

Also note that the first record in a Naïve Bayes container is reserved for global information and will not contain a DATA field. This results in category IDs being assigned starting from 2.