IBM Corporation

IBM Content Classification

Decision Plan Design, Implementation, and Use Considerations

Richard Joltes, Software Developer

Contents

Introduction

Development of IBM Content Classification decision plan scenarios is often far simpler than creation of a properly trained knowledge base environment. The decision plan solution is often easier for users and administrators to visualize, more deterministic, and quicker to implement. Unlike undertaking the task of gathering hundreds of representative documents to create a knowledge base instance, decision plan implementations can be created relatively quickly using a set of known document characteristics and desired behaviors.

This is not to say, however, that correct implementation of a decision plan is easy. A great deal of document and workflow analysis must still be performed in order to ensure that decision plan behaviors will be consistent across all cases.

Decision plan implementations can be used to extend and build upon the power of the knowledge base model. A decision plan can reference a knowledge base to return a “best fit” category as part of decision plan processing, thus providing a statistics-based suggestion that the decision plan can then use or discard based upon the level of confidence returned.

Additionally, decision plan-based classification can also automate the process of metadata modification or normalization. The rules in a decision plan can affect the metadata for an in-process document, such as adding a date or a document class prior to the forwarding of the document and its associated metadata to the next step in the document management process.

Correct, appropriate use of IBM Content Classification decision plan implementations takes the product far beyond basic classification tasks by transforming it into a document-cleanup and reformatting engine. The decision plan process can help organizations avoid storage of useless metadata (such as corporate document headers and footers) that might otherwise hinder the process of searching for specific archived materials at a later date. A well-developed decision plan can also help improve overall storage and document usefulness by eliminating incorrect data such as common misspellings, badly formatted phone numbers, or accounting data. Properly implemented, an IBM Content Classification decision plan can also be used to remove sensitive information such as credit card data from files prior to storage.

Converting business requirements to decision plan rules

As with all major document management implementations, planning and forethought are paramount when a decision plan model is to be implemented in a given environment. The IBM Content Classification engine cannot start processing documents without the user providing adequate rules. These rules must be defined using a solid sampling of all document types that will traverse the IBM Content Classification system for classification purposes.

The process of formalizing business rules starts with identifying the business problems to be solved and establishing a number of possible solutions to those problems. Examples might include removal of outdated documents, moving personal mail to a designated folder, or categorizing items according to a taxonomy.

It is useful to view demos of IBM Content Classification in action, possibly using the built-in decision plan rules and knowledge base implementations, before commencing the process of rule creation. The audience for these demos should be individuals who will be charged with rule creation, and any other interested stakeholders who might provide input during the process. Seeing IBM Content Classification in action will help these users understand what the engine is actually capable of doing. This knowledge should dramatically improve the overall result of the basic rules-creation process.

Such a demo should occur very early in the process in order to prevent early establishment of unrealistic expectations regarding the system's capabilities. Users should not come to expect, for instance, that IBM Content Classification can magically clean up poorly formatted documents or create missing metadata out of thin air. Every transformation performed by IBM Content Classification must be based on a rule or rule set.

It is important to clearly identify and document fixed criteria for success of a given implementation.

Decision Plan design includes the following stages:

- Isolate and formalize the actions needed to solve the business problem
- Look for possible rule triggers and actions by using Classification Workbench.
- Formulate decision plan rules

Mapping real world actions to the decision plan

Real-life scenarios should be envisioned in order for correct decision plan rule creation to occur. It can also be extremely useful to write down the exact behavior that is desired for a particular case by using a natural language description of the problem to be solved. Such a description, when agreed upon by all affected groups, can then be translated into an appropriate set of IBM Content Classification decision plan rules. Rules are effectively a series of logical constructs that are designed to isolate specific characteristics of a given document in order to allow proper processing and classification to occur.

An example of such a natural-language rule might be: *all files which are not emails, and which contain the phrases 'budget control' and 'auditing' within the same paragraph, should be (a) classified as 'accounting and finance' documents AND (b) moved to Folder XXXX on the FileNet server.*

Another example: *If a document is classified by the IBM Content Classification knowledge base instance as a 'taxation' document, then its expiration date should be set to 20 years from its creation date in order to satisfy corporate document retention policies.*

A third example involves metadata extraction, and might be phrased as follows: *if a file contains the phrase 'sales order' and also includes a code that looks like '3 digits, then a dash, then 5 characters, then a dash, then 5 more digits', then we need to extract that code and store it in the 'billing code' column.*

A more complex example is: *if the document body contains the term 'financial' and the phrase 'fiscal year' within the same paragraph, and the document title includes the word 'audit', then extract the document metadata that contains the creation date. If this date is blank, use the current date. Assign the document an expiration date ten years in the future from that date, and place it in the IRS folder on the FileNet server.*

These examples demonstrate the use of real-world examples that require no technical understanding of the IBM Content Classification decision plan creation process. The same processes could be applied to a system of physical document classification involving file cabinets and folders. The only users who need to understand the technical aspects of the actual IBM Content Classification deployment are personnel such as system administrators or developers who are charged with technical implementation of the agreed-upon rule set.

It is important to note that all such rules, once created by various departments and subject matter experts, should be compared closely to discover possible conflicts or overlapping intents. How should the decision plan handle a case in which, for example, the accounting department wants a specific document type tagged in one way while corporate controls needs it handled in a different manner? What if one department wants documents containing emails from Corporation XYZ tagged as engineering materials, while another department deals with a different Corporation XYZ that supplies cleaning materials to the company? All such questions should be evaluated, and a coherent set of rules produced, before proceeding to the next step.

Writing the initial rules for document classification in plain language largely eliminates the need for all involved parties to learn the decision plan rules creation process. Users involved in the creation of such plain-language rules should have a strong sense of the business need and be at least basically familiar with the decision plan engine in general terms so that they do not attempt to create rules that are beyond the capabilities of the product as a whole.

The IBM Content Classification decision plan is capable of modifying or setting document metadata, tagging documents with specific identifiers (all of these are implementation specific), and performing match or modification tasks based on regular expression (regex) syntax.

Here are a number of aspects that might be considered:

- Are documents to be classified according to a taxonomy? If so, has that taxonomy been clearly defined and tested against physical documents to ensure its accuracy and completeness?
- Does the document type have any impact on its relevancy or subject classification? For example, should HTML documents be treated differently from Microsoft Word files?

- For emails, can the metadata fields (cc, sender, recipient, etc.) help in classifying the document, or in choosing a destination folder?
- Does the file creation date help in determining what actions should be chosen?
- Should more weight be given to a match within the document title than to a match for the same phrase in the document body?
- Should headers and footers be processed at all?
- Are there formal text entities (such as names, addresses, telephone numbers, dates, and social security numbers) that can be extracted and used either in another rule, or sent as metadata to IBM FileNet P8 or IBM Content Manager?
- Is the beginning of the document more relevant than its continuation?
- Should very small documents or very large documents be processed differently?

In general, pre-creation of real life rules prior to implementing such rules in IBM Content Classification should help streamline the overall process by identifying possible points of conflict, difficult rules, overlap, and other problems.

Common rule triggers and actions for content management systems

Each rule can contain one or more *triggers* and one or more *actions*. For specific cases, a rule can contain only a trigger (which triggers a group attribute to stop processing) or only an action (by setting the trigger to 'true').

This section describes some common examples of triggers and actions.

Move to Folder

Determine document taxonomy.

Triggers: Always

Action: The rule runs a match operation against a knowledge base to identify the top-scoring category for the document

Move documents with scores above 70%

Triggers: When the category score is equal to or greater than 70%.

Action: Move the document to a folder with the name of the top category

Move documents with scores below 70%

Triggers: When the category score is less than 70%.

Action: Move the document to the `ManualReview` folder (by setting the `FileNetP8: File` field.)

Assign document class

Set document class for scores above 60%.

Triggers: When the top category score is equal to or greater than 60%.

Action: Set the `DocumentClass` field to the value of the category name.

Set document class for scores below 60%

Triggers: When the top category score is below 60%.

Action: Set the `DocumentClass` field to the value `ManualReview`.

Find best category

Triggers: Always

Action: The `BestCategory` field is assigned the name of the top-scoring category

Locate the stores in California

Triggers: When the `BestCategory` field is assigned the value 'Locations', and the word 'California' occurs in the Body field.

Action: Set the IBM Content Manager item type to 'California'

Move to a new address

Triggers: When the `BestCategory` field contains the phrase *Address Change* (case sensitive), and the `message_body` field contains the words *move* (case insensitive with a wildcard) and *address* (case sensitive with a wildcard). The word *move* must precede the word *address*, and the distance between them must be at least five words.

Action: The `MovedToANewAddress` field is set to 'Send Flowers'.

Declare record based on string match

Triggers: When any field contains the words *item* and *stock* (case insensitive).

Action: Set the `FileNetP8_RecordsManager:Declare` field to declare documents as records in IBM Enterprise Records.

File in folder based on categories

Triggers: When the top category score is greater than 50%.

Action: Set the `FileNetP8:File` field to the name of the top category. Classification Center uses this value to move files to a folder with this name.

After an initial set of rules has been defined, an attempt can then be made to implement some or all rules as actual decision plan rules in order to test the viability of the desired effect. This phase of the project will likely result in the need for numerous clarification statements regarding the desired effect of a given rule.

Search methods for use in trigger expressions

There are several methods that you can use in a rule trigger to search within content fields. For more information, see Expression syntax for advanced triggers and actions in the IBM Content Classification online documentation.

Words and phrases

You can search for text that contains specific words or phrases, regardless of the number of spaces between the words of a phrase. You can also look for words and phrases that are at specific distances from each other. To search for many words, use a word list file.

Character sequences

You can search for a specific string. Use this method to search for exact sequences of special characters. For example, you can search for the string *123abc<space><space>**456xyz* with exactly two spaces between the tokens. To search for many strings, use a string list file.

Although word and phrase searches can also handle special characters, there are some exceptions. For example, you cannot use a word search to look for an exact number of spaces. If you search for an asterisk (*) character, it might be interpreted in a word search as a wildcard character.

Regular expressions

You can use regular expressions to specify complex search conditions. However, some expressions can take a very long time to process when applied to certain texts.

Patterns

You can search for patterns such as phone numbers or social security numbers by defining a pattern search. Because pattern searches have limited syntax compared to regular expressions, they do not take

a long time to process. Pattern searches can be included in word searches. They can be used to search for specific word distances. For example, you can define a trigger that returns true based on the distance between two patterns.

Decision plan design guidelines

The following best practices should be considered when creating a decision plan in Classification Workbench.

Use rule groups

Group related rules together in order to achieve more efficient processing and ease of management.

There is an option to stop rule or group processing after a rule fires or returns an error. You can use this option to determine the optimal rule or group order, or ensure that all fields are populated (fields can be given default values in the first rule or group and overwritten with more specific values if an appropriate rule fires). When an appropriate rule is reached (perhaps as the last rule of a group), all further processing can be stopped, as shown in the following example:

```
Rule Name:        WordDocs
Rule Status:      Enabled
When Triggered:   Stop all processing
On Action Error:  Stop rule processing
Trigger: true
Actions:
```

You can also group rules according to some document attribute (extension, for example) and then include a trigger-only rule as the first rule of each group. If a document does not match the criteria, the processing skips to the next group. The following example shows a rule that can be used to stop processing of the current group and skip to the next group:

```
Rule Name:        WordDocs
Rule Status:      Enabled
When Triggered:   Stop group processing
On Action Error:  Stop group processing
Trigger: not($FileExtension : ~doc~)
Actions:
```

Search specific fields when possible

When using keywords to identify and manage documents, identify specific document elements (such as subject line or sender) that should be searched for the requested keyword, when possible. Searching entire documents for a keyword match is inefficient, and might result in undesirable behavior such as irrelevant matches or use of inappropriate data elements.

Search the beginning of large fields for the most relevant words

Often the first few lines or paragraphs in a large text field contain the most relevant keywords. In order to search for the most relevant terms, assign a substring to a new field:

```
set_content_field $short_body {substring($Body,1,200)}
```

Alternatively, you can assign the substring to a temporary variable:

```
set_temporary_variable temp {substring($Body,1,200)}
```

Then search the newly created variable or field.

Check that fields exist and have enough content

A search for a keyword in a field will fail when the field does not exist, or is empty (or almost empty). This possibility should be considered for special handling.

If you search the field that contains the main document content (such as the Body or Message field), this constitutes a special case that can require a specific action (such as moving to a special folder for deletion).

To check whether a field exists, use the following trigger:

```
$Body exists
```

To check the minimum length of a field, use the following syntax:

```
strlen($Body) <6)
```

These two checks can be combined into a single trigger:

```
(strlen($Body) <6) or (not (exists $Body))
```

Clean up data

If you use a knowledge base in conjunction with a decision plan, implement rules to allow the decision plan to remove extraneous metadata such as email address blocks, corporate signature blocks, and other common elements prior to passing the document to the knowledge base for classification. Removing this extraneous metadata will generally improve overall knowledge base statistical matching operations, and also eliminates extraneous processing of irrelevant data.

Use the 'cleanup' action to remove substrings according to regex matches. Other string manipulation actions can be used according to need.

Detect documents that cannot be processed

Create rules to handle specific errors that might be encountered during processing. For instance, if a rule determines that a document cannot be opened due to password protection on the file, you can address

the situation by placing such documents in an “inaccessible files” folder for later manual processing. The same type of handling could be used for documents that cannot be processed by IBM Content Classification due to format incompatibility issues or corrupted data.

Detect documents that require manual review

Integration with a knowledge base can be used to filter out documents. For example, a decision plan could call a knowledge base instance and base its subsequent actions on a returned classification threshold. If the result is greater than the threshold, the document is placed in the appropriate folder. But if the result falls below the threshold, the document is placed in a “review” folder for later manual processing. Obviously, this necessitates a balancing act between excessive use of the review folder and possible errors in automated classification.

Use decision plan rules to override classification

Often 'rules of thumb' are the best way to choose a category for a document. For instance, an organization might decide that any email sent by the human resources department should always be categorized as internal information. The organization then needs to determine how to clearly identify all such emails, such as by using a string match on a fixed email address (e.g. “hr@mycorp.com”), and then use this knowledge to categorize the email message.

Statistical (knowledge base) classification can be used in cases in which the results of the rule-based document evaluation do not succeed in assigning a category. A rule set could be created to determine a document’s properties based on certain metadata fields. As a final step, if none of the rules were evaluated as true, a knowledge base could be called to provide a statistical result. The rule for utilizing the knowledge base could use a threshold: if the result of a knowledge base call is above 0.75 (or some other level) the document is classified into a given folder. Another rule can be used to override the statistical classification based on a threshold and place the document in a “review” folder for further manual processing.

Use multiple knowledge bases in a decision plan

It should also be noted that a decision plan can reference more than one knowledge base, so it is possible to handle cases in which multiple statistical classification tasks are required. For example, an organization needs to determine both the document category and whether a customer’s sentiment is positive or negative based on statistical processing. A decision plan could call one knowledge base that is trained to recognize a fixed list of categories (for example, by department, such as “engineering” and “accounts payable”) and also call a second knowledge base that is trained to return “positive” or “negative” sentiment results. The subsequent use of these results is implementation dependent. For example, the results can be used to populate metadata fields, assign the document to a specific folder, or notify personnel via email regarding the need to review a specific document.

Handling multiple languages

IBM Content Classification does not perform language translation. A best practice is to train each knowledge base in a single language. A single decision plan can support multiple languages. The decision plan is capable of identifying a language and sending the document to relevant rule groups and

knowledge base for appropriate processing. For a list of supported languages, see What languages are available in the IBM Content Classification online documentation.

Use hooks to extend the decision plan functionality

Sites which require custom content transformation or other specialized functionality might also wish to make use of the IBM Content Classification external hooks feature. This capability allows document managers to add references to external hooks that are written in standard languages such as Perl, shell, Windows batch, C, and Java. This functionality can also be useful when sites want to make use of previously written utilities when implementing a decision plan. See the IBM Content Classification documentation for additional details about the use and limitations of this functionality.

There are two types of hooks: simple hooks and in-memory hooks.

- Simple hooks are called into memory each time the rule triggers.
- The in-memory hooks (C++ and Java) require the compiling of a single class that has these three methods: Init(), Run(), and Finish(). These methods improve performance time by remaining in memory until the server is reset.

Export the decision plan to a text file for analysis and editing

Classification Workbench has an option for exporting the decision plan as a text file to a system folder. The results can then be viewed and edited using any text editor. The advantage of viewing the decision plan as a text file is that it can be easily browsed. All groups, rules, triggers and actions can be viewed simultaneously.

You can also use this method to copy groups and rules and alter the content. The original syntax must be adhered to in order to allow re-importing into Classification Workbench.

Use case examples showing the mapping between real world problems and decision plan rules

If – then – else clause usage

Use case

A site designing its content repository is faced with the problem of inconsistent formatting in certain non-classified documents. They have standardized the use of a metadata field called *Found* in a site's repository, yet some documents also use a similar field called *MyCat*. The site wishes to copy all cases of the *MyCat* field to the *Found* field in order to standardize storage within the content repository.

Solution

For all documents, create a field called `$Found`. The content of the `$Found` field will be based on the content of another field that might exist in a given document. Set the value of the `$Found` field to 'none' if the other field does not exist in the document.

```
set_content_field $Found {if (exists($MyCat) ) then ($MyCat)  else ('none')
}
```

You can use this concise syntax rather than writing the following, more elaborate syntax:

```
trigger: true
set_content_field $Found 'none'
trigger: exists($MyCat)
action : set_content_field $Found $MyCat
```

The requirement for if-then-else is that both clauses are of the same type.

Format dates

Use case

A site is ingesting data into IBM Content Collector. The data includes metadata fields such as the email sent (origination) date and retention period. The retention period is specified as a fixed number of days, hours, minutes, and seconds. The site does not need to save the origination date, but needs to set the IBM Content Collector expiration date appropriately.

Solution

In this case, a two-step process is required. First, IBM Content Classification extracts both the origination date and the retention period. It then converts the retention period into the corresponding number of seconds and updates the document's expiration date.

First action: The sent date is extracted by IBM Content Classification into the field ICM_SentDate.

Second action: The rule computes the expiration date and reformats into the required format.

This example produces an expiration date in the format that is required by IBM Content Collector.

The first action includes:

- Converting the `$ICM_SentDate` field from a 'yyyy-MM-dd' string to a date structure.

  `import_date($ICM_SentDate,'yyyy-MM-dd')`

- Converting the `$retention_period` field from a 'days hours:minutes:seconds' string to a count of seconds.

  `period($retention_period)`

- Adding the newly formatted retention period to the reformatted date structure.

  `import_date($ICM_SentDate,'yyyy-MM-dd') + period($retention_period)`

- Exporting the result into a string.

```
{export_date(import_date($ICM_SentDate,'yyyy-MM-dd') +
period($retention_period) , '+yyyy-MM-dd\\'T\\'hh:mm:ss:SSSZ-')}
```

- Assigning the resulting values to the original_date field (final format):

```
[on]  set_content_field $original_date
{export_date(import_date($ICM_SentDate,'yyyy-MM-dd') +
period($retention_period) , '+yyyy-MM-dd\\'T\\'hh:mm:ss:SSSZ-')}
```

The second action is reformatting the date to the format expected by IBM Content Collector:

```
[on]  set_content_field $ContentManager:ICC:ExpirationDate
{substring($original_date,1,27) concat ':' concat
substring($original_date,28,3)}
```

This reformatting of the string requires string concatenation to ensure that it conforms to the required IBM Content Collector format.

The date format that IBM Content Collector expects is:

+YYYY-MM-DDThh:mm:ss:SSS+hh:mm- (hh:mm for timezone)

The following message is an example of an IBM Content Collector date error log in the case where the date format is not correct:

```
'ibm.ctms.utilityconnector.ICMClassificationTask' failed for entity with id
'234@175@171@32@33D19F998988C62C852578B2006B926E-1@-1@-
1@123@IMPL==DOMINO;TYPE==MAILBOX;SERVERID==CRAWATLM09/CRAWFORDCO/US;STOREID==
mailjrnl\MJ001115.nsf;EMAILADDR==null;FILEPATH==null36@bb364177-5fc2-42ca-
81f0-c9bb9a5167722@{}7@INITIAL-1@': Status=error; Message='"+2021-06-
14T12:00:00:0000-0:400" does not match expected date format "+YYYY-MM-
DDThh:mm:ss:SSS+hh:mm-".'Reason: Task Method
'ibm.ctms.utilityconnector.ICMClassificationTask' failed for entity with id
```

Use wordlists

Use Case

A site needs a way to extract names (such as the first name followed by the last name) from texts in order to store them in a separate metadata field in their content repository. However, administrators do not want to resort to creation of an all-encompassing list of first name/last name combinations.

Solution

To avoid having to create a list of first name/last name combinations, the site creates two wordlists, one for first names only, and one for last names. IBM Content Classification then searches the field for a string that appears in the 'first_names' wordlist, followed immediately by a string that appears in the 'last_names' wordlist. A wordlist is usually a large, domain specific list of words.

The following example shows a wordlist condition in the trigger:

```
$Body : wordlist[ (name of wordlist) ]
```

This rule succeeds if any of the words included in the wordlist are found in the text.

Wordlists can be used anywhere a word can be used. For example:

```
$Body : 'John Smith'
```

This is equivalent to the following distance operation:

```
$Body : 'John' s/1 'Smith'
```

Complete the following steps to create a condition that looks for any first name followed by any second name.

1. Create two wordlists with 500,000 words each:

   ```
   wordlist_FirstName.txt
   wordlist_LastName.txt
   ```

2. Use the wordlists in a trigger:

   ```
   $Body : wordlist[FirstName] s/1 wordlist[LastName]
   ```

Use rules to adjust or override statistical classification

Use case

A site is using a knowledge base in conjunction with a decision plan to determine the location of a customer, but has decided that in certain cases the returned results need to be refined based on certain words that might be found in the document. As part of this processing, the site wants to isolate specific city names and relate them more closely to areas within a state.

Solution

In the following case, if the State_KB knowledge base returns a score greater than 0.7 for the state of Massachusetts when identifying a customer location, *and* the city name Newburyport is found within the document, then the category will be refined to Northern Massachusetts with a score of 0.9 before other processing takes place.

```
Rule Name:        United States/MASSACHUSETTS
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop rule processing
Trigger:  $__all__ :  ~Newburyport~ and (score('State_KB','Massachusetts) >
0.7)
Actions:
     [on]    set_category 'Region_KB' 'Northern Massachusetts' 0.9
```

More examples of using rules to override statistical classification

You might want to override the statistical classification based on information from non-statistical sources. For example, you might want to increase the score of a category when a specific token or phrase is found in one of the non-statistical fields. You can try to influence the statistical classification by adding the field to the statistic processing (by setting a content type for that field in Classification

Workbench) and providing lots of feedback for it. Alternatively, you can use decision plan rules to override the category scores that are returned by the knowledge base.

Tip: Be careful when you add rules to override the statistical classification. Adding such rules might solve one problem, but it can harm the classification of other documents. For example, if you create a rule to increase the score of the Health category when the word *doctor* is found in a document, you might unintentionally cause documents about doctors of physics to be misclassified in that category. In general, it is more effective to influence the statistical classification by adding fields to the statistic processing rather than using decision plan rules, unless you need to create a rule about something that the statistic engine has no way of learning.

Before you run rules to adjust or override statistical classification, you must run a **Match** or **Match once** action to generate category scores based on the knowledge base. For example, the following rule submits content items for matching against the TestKB knowledge base if category scores do not already exist.

```
Rule Name:        New Rule
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop all processing
Trigger: true
Actions:
     [on]  match_once TestKB 0:0
```

To override the top-scoring category that was returned by the knowledge base, use the **Set top category** action. The following rule sets Radiology as the top-scoring category.

```
Rule Name:        Rule 2
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop all processing
Trigger: true
Actions:
     [on]  set_top_category TestKB Radiology
```

To change the score of a category, use the **Set category** action. The following rule sets the score of the Radiology category to 95.

```
Rule Name:        Rule 3
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop all processing
Trigger: true
Actions:
     [on]  set_category TestKB Radiology 0.95
```

Tip: You can use multiple **Set category** actions to adjust the scores of different categories and change the overall category ordering. But do not use the **Set top category** action more than once. The **Set top category** action sets the score of the specified category to be higher than 100% so that it overrides whatever category is currently in the top slot, even if the category has a score of 100%.

You can use rules to remove categories from the results that are returned from the knowledge base. For some documents or scenarios, you might not want to return results unless you are highly confident that they are accurate. For example, you might want to be more cautious about the results that you return for documents that come from a sensitive source. In this case, you might remove all but the most relevant results by using a high threshold, such as 90%. For other documents, you could prune the results with a low threshold to remove just the irrelevant results.

To remove low-scoring categories from the match results, use the **Prune categories** action. The following rule keeps the top 5 results that have a score over 50.

```
Rule Name:        Rule 4
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop all processing
Trigger: true
Actions:
     [on]  prune_categories TestKB 5:0.5
```

You can also remove categories from the match results based on a threshold file. The following rule removes results that did not achieve the thresholds that are specified for each category in the `_threshold_high_90` threshold file.

```
Rule Name:        Rule 5
Rule Status:      Enabled
When Triggered:   Continue
On Action Error:  Stop all processing
Trigger: true
Actions:
     [on]  prune_categories TestKB 0:0:TestKB/_threshold_high_90
```

Perform looping on multi-valued attachment fields

Some triggers and actions, such as search, can be performed over multiple field instances. However, more complex triggers need to consider each multiple field instance separately.

For example, a customer wants to extract social security numbers from the `$Body` field and then validate each number. The following action will extract all numbers that conform to the established syntax for social security numbers and place each social security number into a separate value of a multi-value field:

```
Actions:
 [on]  extract_regular_expression $ssn $body 'i:\\s((\\d\\d\\d)-\\d\\d-
\\d\\d\\d\\d)\\s'
```

This action extracts all social security numbers to the multi-value field `$ssn`.

However, a further constraint must be checked against each number. Create another group (the value above needs to be set before entering the loop group) and set the number of iterations to equal the length of the multi-value field `$ssn`.

The first rule of the loop group will extract the first (or next) value of the multi-value $ssn field:

```
Trigger: true

Actions:
     [on]  extract_by_index $this_ssn $ssn var[loop]
```

The second rule of the group will extract first part of the number delimited by a dash:

```
Trigger: true

Actions:
     [on]  set_content_field $ssn_parts {split($this_ssn, '-')}
     [on]  set_content_field $part1 $ssn_parts[1]
```

The third rule will check the value of the first part (the additional constraint):

```
Trigger: ($part1 > 0) and ($part1 < 723)

Actions:
     [on]  add_to_content_field $good_ssn $this_ssn
```

Note: If the value of a field is a legal numerical value, no conversion is needed to perform calculations.

Validating business goals during the development stage

Analyzing the decision plan results

Analysis of decision plan results is an iterative process. During the design stage, the analysis is necessary to verify the initial assumptions and hypotheses. The analysis process results in a re-evaluation and reformulation of the rules.

Later, during the production stage, additional analysis is required to assure that consistent and expected results are maintained. Content can vary over time, and the decision plan might need to be adjusted to compensate for such changes.

Rules do not exist in a vacuum. Whether a given rule is triggered or returns a given result might depend on the result obtained by rules that were triggered earlier in the decision plan rule set. This is true whether a rule lies in the same or a different group. If the decision plan is set to stop all processing after a specific rule returns a true result, no subsequent rule will ever be triggered for that document.

It is therefore necessary to evaluate the relationship among the various rules in a given decision plan in order to determine which should and should not be triggered in a given case. Such analyses can be performed within a Classification Workbench decision plan project. For example, you can observe how

many times a given rule is triggered and under what circumstances. Actions can be added to populate fields that can be used in statistics, correlating to chosen fields. Data can also be saved on the server and analyzed in Classification Workbench.

When you run the decision plan over a large content set, you can begin to evaluate and assess how the decision plan will behave in the eventual production system. Such testing might often be conducted within Classification Workbench. A typical workflow for decision plan analysis consists of the following steps:

1. Establish a content set and import it to Classification Workbench.
2. Run Analyze Decision Plan.
3. Run Reports.

Alternately, you can test decision plan implementations by using the Java GUI Decide sample application that is provided with IBM Content Classification. In specific organizations, use of this sample might simplify testing when testing is to be performed by employees who lack access to the Classification Workbench application. Before testing begins, the decision plan developers must first export the decision plan from Classification Workbench to a running server instance so that testers have access to the most recent changes.

Content set for testing a decision plan

While decision plan processing does not require the extensive corpus of training documents that are generally required for knowledge base training, it is still necessary to test rules in order to ensure that they exhibit correct behavior across a wide variety of documents. As such, decision plan development efforts require a set of documents to be tested in order to determine whether decision plan rules function correctly.

Such a test set can contain various types of data.

Documents that have been previously hand-reviewed and assigned specific categories or behaviors

These documents will be used for initial rules testing and vetting. When the newly developed decision plan (and knowledge base, if required) is allowed to process these documents, the results must align with the desired behaviors previously assigned to this document set.

For example: Document A is hand designated as an “accounting and finance” document. The decision plan is supposed to assign documents of this type an expiration date of 10 years from the date on which the documents are processed. The decision plan is also supposed to examine a metadata field called “author” and place its contents in a matching metadata field in the FileNet repository. If the decision plan is run against this document, all these results should occur. If they do not, then the decision plan

should be examined to determine why the results did not occur. Is the author metadata missing from the file? If so, how should the decision plan handle such a case?

Documents known to contain ambiguous data

This set can be used to test how the decision plan responds to more difficult circumstances, such as missing metadata or, in the case of integrated knowledge base processing, unclear document intent or category. These documents can be used to test handling of, for instance, rules intended to place ambiguous documents into "for review" folders or otherwise flag them for later manual processing.

Previously unexamined documents

This set, which can be relatively large and created on an ad-hoc basis from documents that have not previously been otherwise reviewed or classified, represents typical material that will be processed by the decision plan. Analyzing a content set of this type might provide the most realistic assessment of how the decision plan will behave in production.

For example, a decision plan developer might capture a group of 100 random emails, Microsoft Word documents, and unusable files such as JPEG images, import them into Classification Workbench, and then run the documents through the decision plan and examine the results. Are JPEG images correctly marked as unable to be processed? Are the rules for email (presuming any exist) followed correctly? Do the Microsoft Word documents have their metadata correctly extracted and placed in designated fields?

Note that a great deal of per-rule or other initial testing can be conducted by using the Classification Workbench application. Users might generate new rules and test them against individual documents or sets of documents to determine whether the rules function as designed. For localized testing, a user could also export the decision plan (and knowledge base if applicable) to a running IBM Content Classification server instance on their own laptop, and then use the *Java GUI Decide* sample application to feed in documents or text for testing purposes.

Establishing Thresholds

Within the decision plan there are numerous techniques for working with thresholds that control which classification results (returned by the knowledge base) are accepted.

Since thresholds can only be adjusted after analyzing results, this process cannot be carried out *a priori.* Classification results need to be constantly monitored as part of the maintenance process.

Thresholds can be:

- Set per category, in the knowledge base itself.
- Set in threshold files (which map thresholds to categories) that can be accessed by the decision plan rule.

For example, the following trigger checks that the score of the category `'Account Info'` is higher than the relevant score in the `online retail/online retail_ByCurrent_thresholds` threshold file.

```
score('online retail' , 'Account Info') > score('online
retail/online retail_ByCurrent_thresholds' , 'Account Info')
```

- Checked in rule triggers and combined with other triggers and actions. For example:

```
score('online retail' , 'Account Info') > 0.85
```

Thresholds within the decision plan are usually implemented in order to deal with less than optimal scores that are returned by a referenced knowledge base. Ideally, high scores (for example over 85%) indicate that a document belongs to a specific category. However, less well-defined categories might return a much lower score (for example, <70%). The best way to deal with such cases is by establishing per-category thresholds, or thresholds based on external factors that can be detected by the decision plan.

Establishing metrics

Organizational metrics should be established, such as *achieve a 75% reduction in documents classified in the wrong folder,* or *successfully automate the population of the 'credit card number' field in the customer database to within a 95% accuracy rate.* Without such metrics, it might be difficult to assess the overall effectiveness of the project. Also note that metrics established early in the project might require revision as the rules-creation process progresses. Depending on the use case and its relationship with actual documents processed by the IBM Content Classification system, the desired results might be difficult to achieve due to inadequate or missing data, or as the result of unforeseen issues uncovered during the process of decision plan creation and testing.

For example, while a given use case might seem very clear (such as "all documents containing email addresses should be moved to FileNet Folder X"), such an assumption might result in grossly incorrect results. Is "Folder X" designated simply as the final destination for all email messages? If so, what about Microsoft Word or text documents that contain an embedded email address as part of a header, footer, or body? Under the above rule, all these documents could potentially be moved to the Folder X destination, which is not the desired behavior. Obviously, the proposed rule inadequately describes the task to be performed and is in need of revision.

Establishing priorities

After the system is running, a benchmark needs to be set to measure success.

Although any business rule aims for 100% accuracy, in practice this rarely occurs. Therefore it is important to choose between false negative and false positive. This prioritization is a key factor in establishing thresholds.

False positive is a rule that is triggered when it should not be triggered.

False negative is a rule that does not fire when it should have triggered.

Prioritizing false positives:

To insure that all relevant documents are gathered for a category or set of actions, it is necessary to include a few false positive hits. Documents used to run benchmark tests should include a few documents that contain irrelevant data that should trigger false positive results.

Prioritizing false negatives:

In some situations the inclusion of false negatives can have overwhelmingly negative consequences. For instance, in choosing documents that can be exposed to the public, the inclusion of a few confidential documents can be disastrous.

Deploying the system

Deployment of a configured IBM Content Classification project consists of exporting the knowledge base and decision plan from Classification Workbench to a running IBM Content Classification server instance on one or more computers. The decision plan is then available to accept documents and provide classification services.

Again, it must be emphasized that no connection exists between a project in Classification Workbench and the exported server instance that is actually classifying documents. Once the decision plan and knowledge base has been exported, Classification Workbench is no longer involved in the processing. After a knowledge base running on the server has accepted feedback, it needs to be imported back into Classification Workbench in order to be analyzed. If a knowledge base or decision plan is changed in Classification Workbench, it needs to be again exported to the server.

Although knowledge base and decision plan analysis is best carried out in Classification Workbench, final testing should be conducted on a running IBM Content Classification server. This stage of testing includes any integrations (such as IBM Content Collector and the Classification Center component of IBM Content Classification server), as well as the both the source and destination for the processed documents (such as an IBM FileNet or Content Manager instance, file system, or other repository). The

testing should involve actual documents, not specific sets pre-created or sanitized prior to ingestion into the system.

Prior to executing tests, ensure that the following conditions are met:

- All decision plan and knowledge base instances to be tested have been exported from Workbench to a running server
- The deployed knowledge base and decision plan are the most recently developed (or approved) versions. This requirement is especially critical in cases where multiple developers have been creating decision plan or knowledge base instances, since it is possible to create multiple conflicting instances of a given decision plan or knowledge base when they are developed on multiple computers.

This phase of the development process is also an ideal time to test performance of the overall, non-decision plan aspects of the system. If the process can be initiated using a script that will time the results, then a metric in terms of documents per second can be obtained. This metric, especially against larger random test sets, will help determine whether the system has been sized appropriately for the expected number of documents per day. If a test run of 10,000 documents takes 60 minutes, this results in a throughput of 166.67 documents/minute (10,000 / 60) or 2.7 per second (166.67 / 60).

Testing cycles might be conducted in batches of 100, 1000, or 10,000 documents, with reviews and tweaking of decision plan rules between each set. Simply feed the documents to the IBM Content Classification instance (by whatever method works in a particular case), allow processing to occur, and then review the results for consistency and the presence of the desired behavior. If problems occur or if re-testing with the same documents is desired, clean the repository, correct the errors if necessary, and re-run the test set.

Maintaining the decision plan and knowledge base post-production

Monitoring decision plan and knowledge base performance during production is similar to the analysis performed at the end of the development stage. In order to allow for optimal monitoring, some planning is required.

Retention of Classification Workbench material

Projects created in Classification Workbench are not fully exported to the running production server. Only the decision plan or knowledge base itself is transferred to the server instance, while the training documents and other data are retained on the Windows computer on which Classification Workbench was run.

Therefore, after an knowledge base has been trained during the development stage, it is important to keep the corpus that can replicate the original knowledge base. Similarly, it is helpful to keep the

content set that was used for decision plan analysis so that any degradation in performance can be accounted for. Because the decision plan running on a server is read-only, any changes in behavior is due to changes in the nature of the processed content.

Saving production data

In order to fully analyze how well the decision plan is performing in production, it is necessary to save the history of all decisions made by that decision plan. For example, on the Classification Center server, back up the *Classification_Home*\ECMTools\logs directory to store a history of all classification decisions.

In the Management Console component of IBM Content Classification, you can select an option for saving analysis data on the server. This server data can be exported and imported directly into Classification Workbench. The resulting content set can be analyzed similarly to the original content set.

Feedback

Feedback can be used in two ways:

1. To analyze the accuracy of the knowledge base by comparing the suggested category with the feedback category. Ideally, they should be the same.

2. To improve the knowledge base by adjusting the internal statistics ("learning").

Analyzing feedback

One of the most important aspects to monitor and analyze is user feedback. Experienced users will often feel the degradation in classification accuracy without any formal analysis. In order to analyze feedback formally, it is necessary to save the server data (as previously mentioned) that includes user feedback and can be similarly analyzed in Classification Workbench.

Initially, feedback should be reviewed and processed frequently (perhaps daily, depending on the organizational requirements) for at least the first week of operation. In this situation, “feedback” includes both documents that were successfully classified by IBM Content Classification and those that were retained in an “unclassified” folder or otherwise left for manual review. Regular attention to feedback, especially during the first few weeks or months of processing, will help the knowledge base accuracy level quickly improve and should have the effect of diminishing the need for manual reclassification of incorrectly classified documents. The IBM Content Classification instance should be treated like a new employee. It should be provided with frequent review and correction during the early stages of employment in order to ensure that proper processing occurs.

Using feedback to improve the knowledge base

In spite of the obvious advantages of giving feedback, sometimes inappropriate feedback can actually damage a knowledge base.

Also, when feedback is not distributed evenly over all categories, the knowledge base might become skewed. Categories which do not get feedback will begin to perform poorly. Specifically, they will begin to return low scores in relation to other categories, resulting in misclassifications.

The ideal way to improve a knowledge base consists of the following steps:

1. Export the server data to a content set. Each item consists of the submitted text, the suggested category and the feedback category.
2. Manually review the feedback (by correcting and deleting).
3. Possibly add additional feedback.
4. Combine this data with the original data used to create a new content set.
5. Create a new knowledge base with this content set.
6. Review the performance of the knowledge base in Classification Workbench before exporting it to the server

After the knowledge base has reached the levels of accuracy established during the project definition phase, feedback reviews might become less frequent. However, it is strongly suggested that feedback be reviewed and processed regularly since workloads tend to "drift" over time. Characteristics of a given category might change, resulting in a slow degradation in accuracy. This situation can be corrected through the judicious application of feedback, both positive and negative, in order to minimize the effect of such "category drift." Failure to process feedback and review the performance of the IBM Content Classification instance in production over long periods of time might result in a large number of incorrectly categorized documents, and the need for an increased level of manual intervention.

Conclusion

This document presented tips for each stage of the IBM Content Classification decision plan development and implementation process. However, these stages should be treated as modular. Very rarely do things work as planned, and at any point in the process new problems can be uncovered that require a reassessment and refining of the decision plan. The analysis, retraining of the knowledge base, and decision rule tuning are all activities that should be carried out regularly, with special care to documenting changes, saving data, knowledge base files, decision plan files and Classification Workbench project folders in an organized manner.

More information

For more information about implementing a decision plan, see the following topics in the IBM Content Classification information center:

Building a decision plan

Decision plan analysis

Analyzing data from a production server

Setting thresholds

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks