This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Content Analyzer Plug-in for GAX

eServices Digital Administration 9.0.0

1/28/2022

Table of Contents

Content Analyzer Plug-in for GAX

Genesys Content Analyzer is an option to Genesys Knowledge Management, requiring an additional license. It adds natural language processing technology to Genesys Knowledge Management.

Content Analyzer Plug-in for GAX is an administration tool for managing Genesys Content Analyzer. The rest of Knowledge Management is handled in eServices Manager Plug-in for GAX.

> Tip
>
> - Content Analyzer is designed to be viewed at a minimum screen resolution of 1280x1024, or a full HD resolution of 1920x1080.
> - For Content Analyzer to work with Microsoft Internet Explorer 11 you might have to adjust your browser settings. Ensure that the following Internet Explorer Security options are enabled:
>   - Downloads/Font download
>   - Scripting/Active scripting

What Does Content Analyzer Do, and How Does It Do It?

Content Analyzer looks at a text-based interaction and tells you where on your category tree it is most likely to belong.

- It does this using a classification model, which is basically a statistical representation of a category tree.
- It applies the model to an incoming text interaction (email, SMS, chat, and so on), and produces a list of categories. For each category, it tells you how likely it is that the interaction belongs to the category.
- The actual analysis is done by Classification Server when it is triggered by a Classify or Analyze object in a routing strategy.

> Tip
>
> Classification, like screening, can operate on any interaction that has text somehow associated with it, whether as the body of the interaction (email, chat), or otherwise (as user data, for example). This document calls all of these things *training interactions* or *messages*.

So Where Do Models Come From?

First you need a *Training Data Object.* That is a category tree plus a set of text objects (such as emails, SMS messages, Twitter interactions, and so on), with each text object assigned to one category in the tree. Those text objects are called *Training Messages* or *Training Interactions*.

Then you run the training, and that produces the model.

In a little more detail,

1. Create a Training Data Object.
2. Modify the Training Data Object as needed:
   - Create some new Training Messages.
   - If you have uncategorized Training Messages, move them into categories.
   - Use cut/copy/paste to move Training Messages between categories.
3. Get ready to run the training.
   - Choose when to run it and how to run it.
   - If you want, you can have it filter out confusing or irrelevant text strings before it runs.
4. (coming soon!) Test the model.

Importing and Exporting

You can import and export training objects and models as part of the process of importing and exporting objects in eServices Manager. This is also a means of copying training objects.

Warning

Once you import an archive that includes any Training Data Object, you cannot import the archive again to a different language. If you attempt to do so, you will receive a message asking you to change the root category name, but when you do that you receive an error message saying that a Training Data Object with that name already exists.

As a workaround, you can rename the Training Data Object before importing the second time.

Installing Content Analyzer Plug-in for GAX

Prerequisites

- Genesys Administrator Extension 8.5.290.09 or later
- UCS 8.5.200.17 or later
- Classification Server 8.5.200.05 or later
- If the a previous version of Content Analyzer Plug-in for GAX was installed on the current host, uninstall that version.

Creating a Zip File

1. Create a temporary folder on your desktop.
2. Copy the IP folder from the original location (typically, the product CD) into the temporary folder.
3. Copy the templates folder from the original location into the temporary folder.
4. Zip the temporary folder.

Alternatively, contact Genesys Customer Care to obtain the required Zip file containing the installation package and associated templates.

Installing the plugin

Local Control Agent 8.5.100.31 or higher

1. Upload the Content Analyzer installation package (IP) to GAX. Refer to the GAX documentation for more information.
2. Extract the IP to any folder.
3. Navigate to the **ip** folder in the extracted folder.
4. Do one of the following:
    - On Windows, run **setup.exe**.
    - On Linux, run **install.sh**.
5. Restart GAX.

Local Control Agent 8.5.100.29 or lower

1. Add the following option to the **Application Options** tab of the Genesys Administrator Extension (GAX) Application object:
   - Section name: **[asd]**
   - Option name: **plugin_ip_list**
   - Option value: `CntAnlzPlgnAdm64` (for the Windows host) or `CntAnlzPlgnAdm` (for the Linux host)

   > #### Tip
   > This option is also used by eServices Manager Plug-in for GAX and Privacy Manager Plug-in for GAX. If you are running both Privacy Manager and eServices Manager with Content Analyzer, the value can be a comma-separated list; for example, `eSMngrPlgnAdm,PrivacyMng,CntAnlzPlgnAdm`.

2. Restart GAX.
3. Upload the Content Analyzer installation package (IP). If you previously uploaded the IP, you must do so again now.
4. Carry out the plug-in installation process.
5. Restart GAX.

Configuring the plugin

Local Control Agent 8.5.100.31 or higher

1. Create an Application for Content Analyzer with the type Application Cluster.
2. Add tenants to the Content Analyzer Application.
3. Specify the Host and Port. The Application Cluster application object and the GAX application must be configured with the same host in order to enable mutual TLS connections.
4. Connect the Content Analyzer Application to UCS, UCS Proxy, and Classification Server. In addition, Training Server must be connected and run with the same UCS.
5. Configure these connections: TLS, ADDP, and so on.
6. Navigate to the GAX Application Options for the Content Analyzer Application.
7. Create a section with the name **[gax-content-analyzer]**.
8. Create the following options in the **[gax-content-analyzer]** section:
   - **app-name** = Content Analyzer Application name
   - **interactions-per-request** = `1000`
   - **connection-timeout** = 3

9. If you need multi-language support, set the UTF-8 JVM parameter for all Java components (UCS, GAX Server, Browser) in the corresponding .ini file (or the .bat file if you start the component from the command line):
   ini file: `[JavaArgs] -Dfile.encoding=UTF-8 ...`
   bat file: `set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 ...`
10. Restart GAX.

> ### Tip
> On multi-language support: In the GAX **User Preference** window (under the Preferences menu), **Use system settings** refers to using the settings in the **System Preferences** window. It does not refer to the Region and Language setting of the host machine.
>
> If **Use system settings** is selected in the **Language** field of GAX's Locale User Preferences, it refers to the language that is selected in the **Language** field of **System Preferences**. Note that if you want to have any language other than English available in these fields, you must install the relevant language pack plug-in.

Local Control Agent 8.5.100.29 or lower

1. Create an Application for Content Analyzer with the type Application Cluster.
2. Add tenants to the Content Analyzer Application.
3. Specify the Host and Port. The Application Cluster application object and the GAX application must be configured with the same host in order to enable mutual TLS connections.
4. Connect the Content Analyzer Application to UCS, UCS Proxy, and Classification Server. In addition, Training Server must be connected and run with the same UCS.
5. Configure these connections: TLS, ADDP, and so on.
6. Navigate to the GAX Application and specify the Content Analyzer Application name as the value of the **app-name** option in the **[gax-content-analyzer]** section.
7. If you need multi-language support, set the UTF-8 JVM parameter for all Java components (UCS, GAX Server, Browser) in the corresponding .ini file (or the .bat file if you start the component from the command line):
   ini file: `[JavaArgs] -Dfile.encoding=UTF-8 ...`
   bat file: `set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 ...`
8. Restart GAX.

> ### Tip
> On multi-language support: In the GAX **User Preference** window (under the Preferences menu), **Use system settings** refers to using the settings in the **System Preferences** window. It does not refer to the Region and Language setting of the host

> machine.
>
> If **Use system settings** is selected in the **Language** field of GAX's Locale User Preferences, it refers to the language that is selected in the **Language** field of **System Preferences**. Note that if you want to have any language other than English available in these fields, you must install the relevant language pack plug-in.

Role-Based Access

To grant a user access to the GAX menu and Content Analyzer Plug-in for GAX, assign the following privileges to the users in GAX:

- **COM/Access Configmanager**
- **content-analyzer/Content Analyzer Plug-in for GAX Access**

Uninstalling the plugin

On Linux

1. Stop GAX.
2. Go to **<GAX_HOME>/webapp/WEB-INF/lib** on the file system (where **<GAX_HOME>** is your home folder for the GAX application).
3. Delete the **gax-content-analyzer-<$version$>.jar** file (where **<$version$>** is the version of the plugin).
4. Go to **<GAX_HOME>/plug-ins** on the file system.
5. Delete the **gax-content-analyzer-<$version$>.jar** file.
6. Go to **<GAX_HOME>/webapp/plugins** on the file system.
7. Delete the **gax-content-analyzer** folder.
8. Start GAX.

On Windows

1. Stop GAX.
2. Go to **Programs and Features**.
3. Find and run **Genesys Content Analyzer Plug-in for GAX <$version$>** (where **<$version$>** is the

version of the plugin).

4. Select the **Remote** check box.
5. Click **Next**.
6. Click **Yes** in **Confirm Windows**.
7. Click **Finish**.
8. Go to **<GAX_HOME>/webapp/plugins** on the file system (where **<GAX_HOME>** is your home folder for the GAX application).
9. Delete the **gax-content-analyzer** folder.
10. Go to **<GAX_HOME>/plug-ins** on the file system.
11. Delete the **gax-content-analyzer-<$version$>.jar** file.
12. Start GAX.

Training Data Objects

A training data object is a category tree plus a set of text objects (such as emails), called *Training Messages* or *Training Interactions*, with each text object assigned to one category in the tree.

There are several ways that a text object can end up assigned to a category.

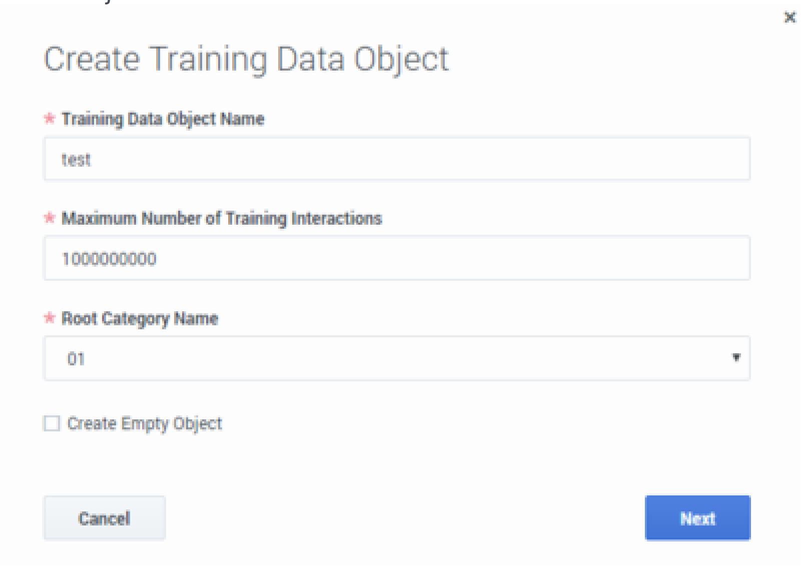
- Whenever an agent uses a standard response in handling an interaction, that interaction gets assigned to the category that the standard response belongs to.

  You may find that some agents are more likely to select the right standard response than others. You can choose to use only the category assignments made by those agents when creating a new Training Data Object—see "Who Categorizes?" below.

- Standard responses are associated with categories, by definition.
- You can create a new Training Interaction and add it to an existing Training Data Object.

New Training Data Object

1. In Content Analyzer Plug-in, on the **Training** tab, click the plus-sign icon to bring up the New Training Data Object window.



Create Training Data Object

* Training Data Object Name: test

* Maximum Number of Training Interactions: 1000000000

* Root Category Name: 01

□ Create Empty Object

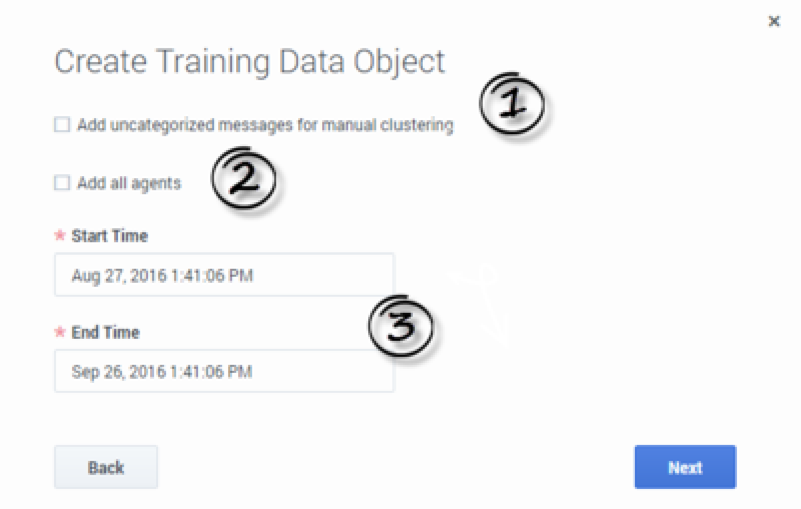
New Training Data Object

- Be sure you use a name that observes the limitations on characters: the permitted characters are the same as for eServices Manager objects, and the maximum number of characters is 64.
- If you don't have a set of categorized text objects on hand, you can select **Create Empty Object**

and add training interactions to it later.

- The figure shows the maximum allowed number of training interactions. The default number is 50,000, and minimum is 1.

2. Then make some decisions about what text objects to include.



Create Training Data Object

□ Add uncategorized messages for manual clustering

□ Add all agents

* Start Time

Aug 27, 2016 1:41:06 PM

* End Time

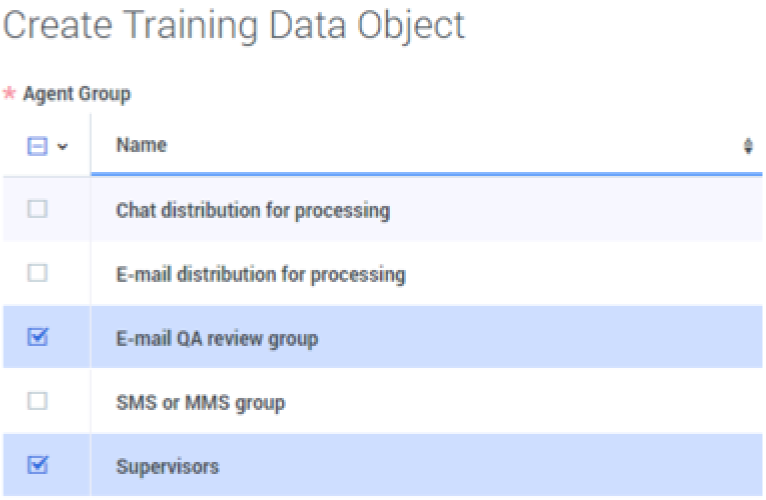
Sep 26, 2016 1:41:06 PM

New Training Data Object 2

1. If you selected **Add uncategorized messages**, you'll be able to move them to categories later.
2. Use the categories assigned by all agents? (select **Add all agents**.) or select the more reliable agents? (next step.)
3. This timespan sets the dates of the messages used for training.

> ### Tip
>
> In defining the time interval, be careful to avoid these common errors:
>
> - Identical start and end dates, resulting in zero messages in the Training Data Object
> - Too large a time span, which can result in too many messages in the training set and the latest emails having less of an impact on training of the model.

3. Use the categorizing done by which agent groups? (If you selected **Add all agents** this dialog doesn't appear.) See "Who Categorizes?" below for more on this.



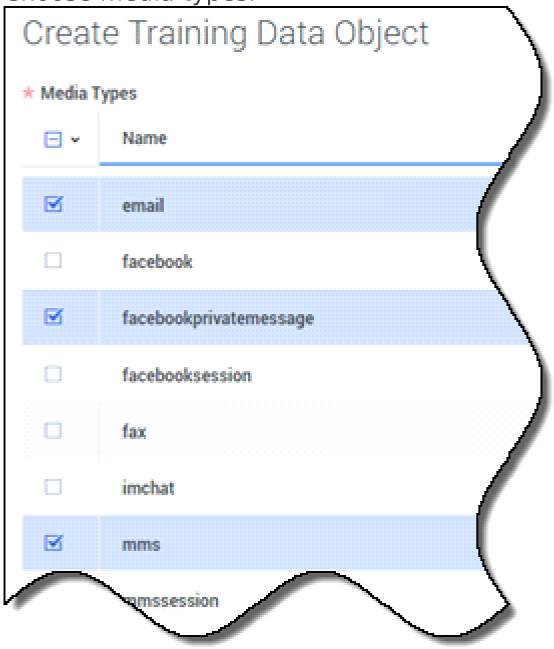
Create Training Data Object

* Agent Group

⊟	Name
☐	Chat distribution for processing
☐	E-mail distribution for processing
☑	E-mail QA review group
☐	SMS or MMS group
☑	Supervisors

New Training Data Object 3

4. Choose media types.



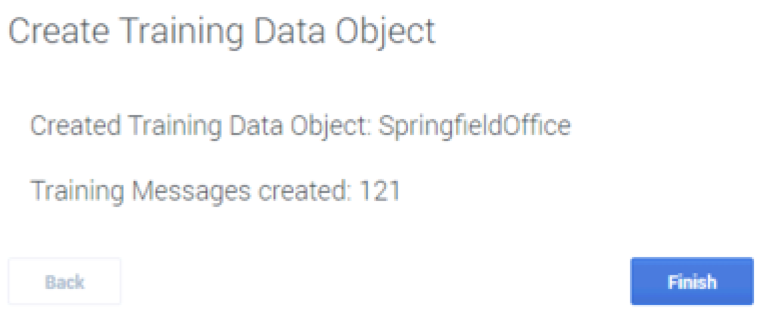
Create Training Data Object

* Media Types

⊟	Name
☑	email
☐	facebook
☑	facebookprivatemessage
☐	facebooksession
☐	fax
☐	imchat
☑	mms
	mmssession

New Training Data Object 4

5. The final screen shows how many training messages the new Training Data Object includes.



Create Training Data Object

Created Training Data Object: SpringfieldOffice

Training Messages created: 121

New Training Data Object Final

Notes

Who Categorizes?

Creating a new Training Data Object process can take from a few seconds to many hours, depending on the number of interactions and agents included. Because of this, Genesys recommends that no more than 200–300 agents participate in the categorization process. You can create an agent group specifically for creating Training Data Objects, consisting of agents whose categorization is trustworthy, and who understand that their categorization decisions will have a direct effect on the classification model. You can then select that group in Step 3 above.

You cannot select individual agents.

Deleting

You cannot delete any training data object (or any root category) that has at least one model assigned to it.

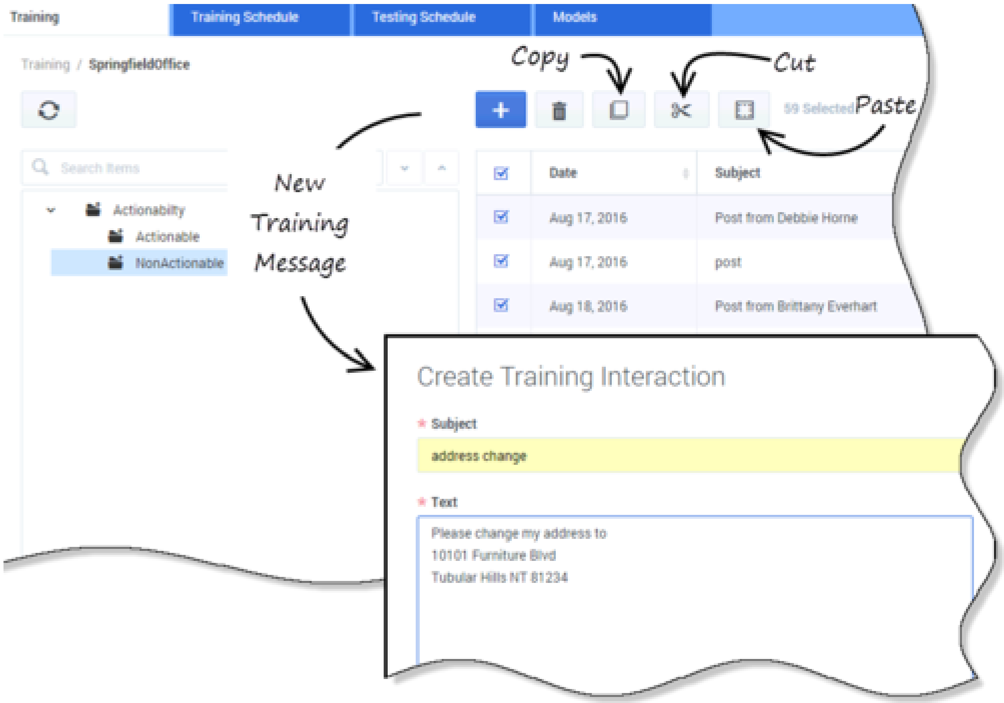
Next Steps

You can

- Assign uncategorized messages to a category
- Create new training interactions

Modifying Training Data Objects

To modify an existing Training Data Object, select it on the **Training** tab, then click the Edit (gear) icon. The following figure shows some of what you can do:



☑	Date	Subject
☑	Aug 17, 2016	Post from Debbie Horne
☑	Aug 17, 2016	post
☑	Aug 18, 2016	Post from Brittany Everhart

Create Training Interaction

* Subject

address change

* Text

Please change my address to
10101 Furniture Blvd
Tubular Hills NT 81234

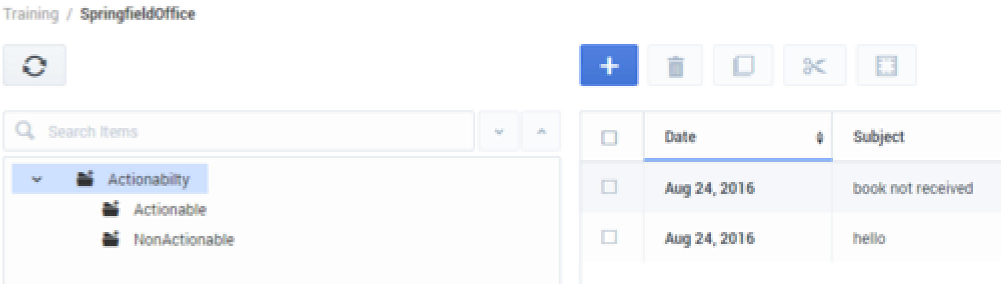
Modify a Training Data Object

Add New Training Interactions

Select a category, then click the plus-sign icon, as shown in the figure **Training Data Object Operations**. Add text to the **Subject** and **Text** fields.

Put Uncategorized Messages in a Category

Uncategorized Training Interactions show up in the root category. In the following figure, the uncategorized Training Interactions **book not received** and **hello** appear in the root category **Actionability**.



□	Date	Subject
□	Aug 24, 2016	book not received
□	Aug 24, 2016	hello

Uncategorized Messages

To categorize them, use **cut** (or **copy**) and **paste** to move them into non-root categories.

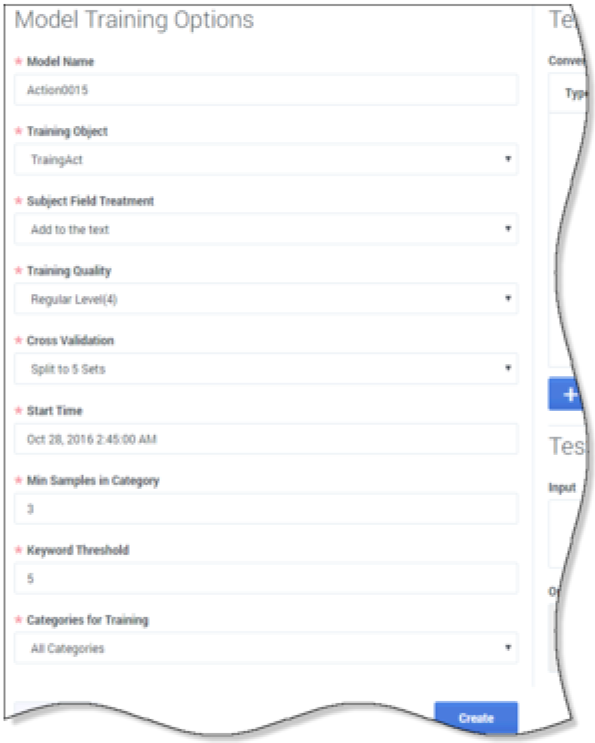
You can also use cut/copy/paste to move Training Interactions from one category to another.

Schedule Training

On the **Training Schedule** tab, set up a training session.

- To populate fields with the values of an existing training session, click the clone icon. This is a handy way to change the time that an existing session is scheduled to run.
- To populate fields with default values, click the plus-sign icon.

In either case, the **Model Training Options** dialog appears.



Model Training Options

* Model Name
Action0015

* Training Object
TraingAct

* Subject Field Treatment
Add to the text

* Training Quality
Regular Level(4)

* Cross Validation
Split to 5 Sets

* Start Time
Oct 28, 2016 2:45:00 AM

* Min Samples in Category
3

* Keyword Threshold
5

* Categories for Training
All Categories

Model Training Options

1. **Model names** must be no more than 21 characters long, and must use only the allowed characters.
2. **Subject Field Treatment**
    - **Ignore**—Training does not consider the content of the **Subject** field.
    - **Add to the text**—Training considers the content of the **Subject** field.
    - **Add with double weight**—Training gives the content of the **Subject** field twice as much importance as the content of the e-mail body.
3. **Training Quality**—If you know that the Training Data Object contains many wrongly categorized text objects, use Unreliable Levels 10–12. Otherwise, use Draft, or Regular Levels 1–6. Note the following:

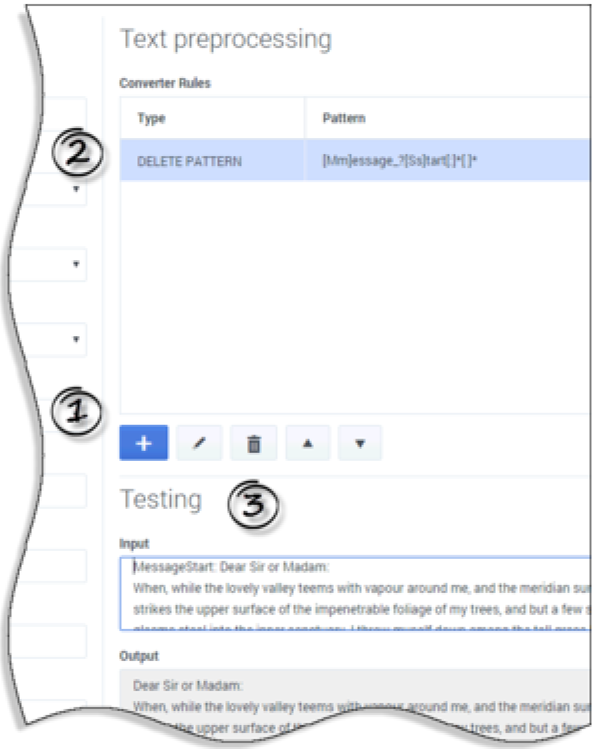
- The Regular Levels and the Unreliable Levels form two independent scales that are not easily comparable. Within each, a higher number means better quality. The only way to know for sure whether, for example, Unreliable Level (11) will produce better or worse results than Regular Level (4), is to create one model with each setting and test them.
- These levels actually determine the number of words that the system considers and the number of iterations the training process runs. Increasing both of those should increase the quality of the resulting model, but at higher levels it may not. Again, the only way to know is to test the resulting models, preferably with cross-validation.
- Training time increases as you move from Draft quality to Regular Level 3 quality. But once the quality goes above 3, there is not much difference in training time. Genesys recommends that you use the lowest quality only when you want to obtain a preliminary reading of the model's quality estimation. For production, use quality 2–6.

4. **Cross-Validation** is explained on a separate page. Select either no cross-validation, or cross-validation that splits the data into 3, 6, or 10 sets. If you select cross-validation, training produces an accuracy rating for the model along with the model itself. This has the advantage of not requiring an extra testing step, but it increases the training time.
5. For **Start Time**, because training can use a large proportion of system resources, you will probably want to schedule it for nonpeak hours. Be sure to set a time later than the present moment.
6. **Min Samples in Category** is the minimum number of text objects that a category must have in order to be included in training. Categories with no or few text objects make poor subjects for training.
7. **Keyword Threshold** is the minimum number of text objects that a keyword must occur in for that keyword to be considered in training. A relatively high value for this setting can reduce training time, but it can also reduce quality. What counts as a high or low value for this setting depends on the total size of the Training Data Object. For example, if a Training Data Object has 5 to 10 text objects per category, a high keyword threshold might be **2** or **3.** If a Training Data Object has 30 to 50 text objects per category, a high keyword threshold might be **20**.
8. **Categories for Training** is **All Categories** or **Terminal Categories Only**. A "terminal category" is one that contains no subcategories. It may be that a category tree uses nonterminal categories mostly for organizing the terminal categories. If so, few or no text objects are associated with the nonterminal categories, and there is little to be gained by including the nonterminal categories in training.
9. You can clean up your Training Data Object by using the Text Preprocessing pane (to the right of **Model Training Options**) to remove extraneous text.

Text Preprocessing

As part of scheduling training, you can use the **Text preprocessing** pane (to the right of the **Model Training Options** pane) to remove extraneous text from the Training Messages of a Training Data Object. You create filters (patterns) that search for text and perform various deletion operations. This can be helpful when the emails that you want to use for training contain significant amounts of text that is:

- Irrelevant or misleading for classification purposes, and
- Identifiable by a regular expression.



Text preprocessing

Converter Rules

Type	Pattern
DELETE PATTERN	[Mm]essage_?[Ss]tart[:]*[ ]*

Testing

Input

MessageStart: Dear Sir or Madam:
When, while the lovely valley teems with vapour around me, and the meridian su
strikes the upper surface of the impenetrable foliage of my trees, and but a few s

Output

Dear Sir or Madam:
When, while the lovely valley teems with vapour around me, and the meridian su
the upper surface of t trees, and but a fe

Text Preprocessing

1. Click the plus-sign icon to create a new rule.
2. Types of rule:
   - **DELETE AFTER**—Search for a match to the pattern, then delete all text after and including the matching text.
   - **DELETE BEFORE**—Search for a match to the pattern, then delete all text before and including the matching text.
   - **DELETE ALL IF FIND**—Search for a match to the pattern, then delete the entire Training Message

that includes the matching text.

- **DELETE ALL IF NOT FIND**—Search for a match to the pattern, then delete the entire Training Message if it does not include the matching text.
- **DELETE PATTERN**—Search for a match to the pattern, then delete only the text that matches the pattern.

3. Test the pattern. Enter text, the result appears. If you modify the rule, you'll have to enter the text again to see the result from the modified rule.
   For the two DELETE ALL types, if you test the pattern and it finds a match, the **Output** window is empty. In actual use if there is a match, the entire Training Message is deleted from the Training Data Object.

Testing Models

Once you’ve created a model, you can test it. Testing takes a model and has it analyze a training object. A moment’s thought will tell you that the training object must be

- One with the same root category as the model
- Not the one that was used to create the model

Schedule the test on the Testing Schedule tab: simply select the model, the TDO to test it on, and the start time for the test.

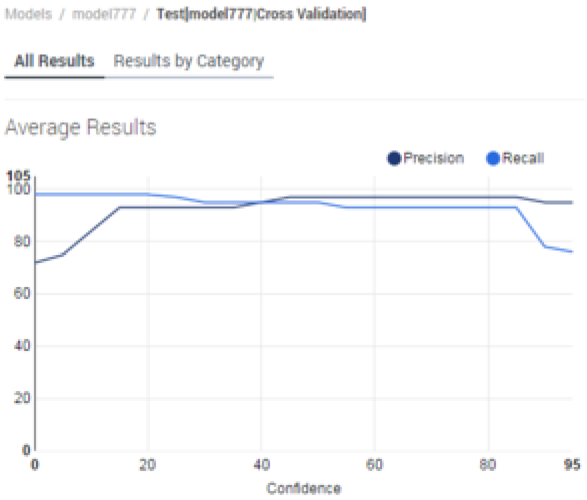
To see the test results for a model, select the model, on either the Testing Schedule or Models tab, and click the eye icon. The eye icon will be active only if you’ve select a model that has been tested.

Understanding the test results is where it gets interesting.

All Results tab

This tab shows the Average Results and Correct in Top N graphs, and the Category Confusion table.

Average Results



Precision, Recall, and Confidence

This graph shows the Precision (black) and Recall (blue) ratings (vertical axis) at a given Confidence level (horizontal axis). But what do those terms mean? Read on:

Confidence

This is a numerical score, from 1 to 100, that indicates the percent likelihood, according to the selected model, that a text object belongs in a certain category.

(In contrast, *accuracy* is an assessment, produced by testing, of the correctness of a model's assignment of text objects to categories. In other words, confidence expresses a model's guess about a categorization; accuracy rates the correctness of that guess)

Precision and Recall

To understand Precision and Recall, consider several possible ways of looking at the performance of a model. If your model attempts to assign a certain number of items to a category X, you can make the following counts:

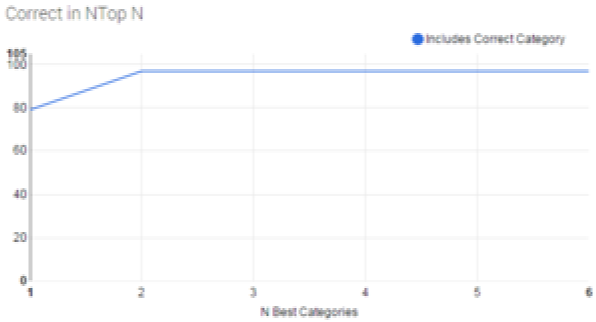
- *a* = the number of items the model correctly assigns to X
- *b* = the number of items the model incorrectly assigns to X
- *c* = the number of items the model incorrectly rejects from X (that is, items that the model should assign to X but does not)

From these quantities, you can calculate the following performance measures:

- Precision = *a* /(*a* + *b*)
- Recall = *a* /(*a* + *c*)

Generally, for increasing precision you pay the price of decreasing recall. That is, the model assigns an item to a category only when it is very sure that the item belongs. But by insisting on being very sure, it runs the risk of rejecting items that really do belong in the category. In the figure **Precision, Recall, and Confidence**, you can see this effect above the 85 percent Confidence level.

Correct in Top N



Correct in Top N

When a model classifies a text object, it returns a list of categories and the probability (the Confidence rating) that the object belongs to them. Ranking the returned categories with the highest probability first, how likely is it that the correct category appears within the top two, the top three, and so on?

- The horizontal axis, **N Best Categories**, shows the best (top-ranked) category, best two, best three, and so on.
- The vertical axis, **Includes Correct Category**, is the percent likelihood that this many top-ranked categories include the correct one.

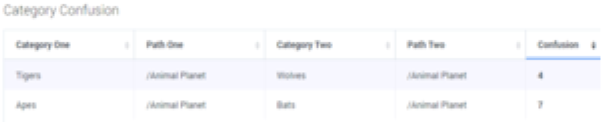
Imagine that the highest-probability returned category is the model's first guess. If we go down to the fourth-ranked category, that's like giving the model four guesses. The more guesses, the easier it is to get the right answer. The fewer guesses, the better the model is at classifying.

The model in the figure is quite good: its first guess (top-ranked returned category) is right just under 80 percent of the time, and we only have to give it one more guess to achieve close to 100 percent correct.

One way to use this rating is to advise agents how many categories to look at when choosing a standard response. If there is a 95 percent probability that the right category is in the top three, you can advise agents to consider only the top three categories.

Category Confusion

The Category Confusion table lists up to 10 pairs of categories that the model is likely to confuse.



Category Confusion

Category One	Path One	Category Two	Path Two	Confusion
Tigers	/Animal Planet	Wolves	/Animal Planet	4
Apes	/Animal Planet	Bats	/Animal Planet	7

Category Confusion

The Confusion column shows the probability, as a percentage, that the model will mistakenly classify a Category 1 item as Category 2. For example, the figure **Category Confusion** shows that this model classifies tigers as wolves 4 percent of the time.

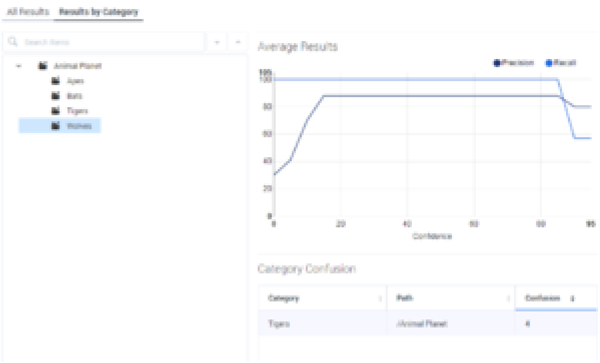
A rating of 50 would mean total confusion: the model cannot distinguish wolves from tigers. A rating of 100 would mean that the model always calls wolves tigers and always calls tigers wolves—a complete reversal.

If a pair of categories has a rating of over 20 and both categories have more than three or four members, you should consider modifying them. You can modify them in either of two directions:

- Merge them; that is, decide that they are so similar that they amount to a single category.
- Further differentiate them by adding more highly contrasting training interactions to them in the Training Data Object.

Results by Category tab

This tab displays the same ratings as the **All Results** tab, but for a single category.



Category Confusion

Category	Path	Confusion
Tigers	/Animal Planet	4

Results by Category

Category Confusion shows the category/ies that are likely to be confused with the category that's selected on the left.

Cross-Validation

In cross-validation, Training Server follows these steps:

1. It builds one model using all of the data.
2. It divides the data into *x* partitions, where *x* = 3, 5, or 10.
3. It builds a number of partial models: as many as there are partitions, each one using a different combination of *x* -1 partitions.
   For example, if the data is divided into the three partitions A, B, and C, Training Server builds model X using partitions A and B, model Y using partitions A and C, and model Z using partitions B and C.
4. It tests each of these partial models against the partition that it omitted when it was built.
   In the example, it tests model X against partition C, model Y against partition B, and model Z against partition A.
5. It aggregates the results of all these tests and presents them as the rating of the entire model.

These ideas underlie the concept of cross-validation:

- The best way to test a model is to apply it to data that was not used in building the model.
- A model built using most of the data is usefully similar to the model built using all of the data, so the results of testing (for example) all possible 90-percent models are a good indication of the quality of the 100-percent model.

Because cross-validation adds to the time required to build a model, you may not want to select cross-validation for very large training objects or for objects for which you selected training quality Regular Level 6.

Samples for Detecting Language, Sentiment, and Actionability

The installation of eServices Manager includes samples that detect

- Language
- Sentiment
- Actionability

Screening Rules

Genesys supplies sample screening rules that analyze interactions for sentiment and actionability

Models

Language Detection

Classification Server 8.5.3

[+] Click to show section

As part of the installation of eServices Manager 8.5.3, Genesys provides a model that classifies e-mails as one of the following languages:

- English
- Bulgarian
- Checz
- Danish
- Dutch (Netherlands)
- Finnish
- French
- German
- Greek
- Italian
- Korean

- Norwegian
- Polish
- Portuguese
- Romanian
- Russian
- Slovakian
- Spanish
- Swedish
- Turkish

To import this model and its training object,

1. Open eServices Manager and select **unknown** as the language. If there is no such language you must create one in Genesys Administrator or Configuration Manager (under **Business Attributes > Languages**).
2. Then import the file `lang-plus.kme`, located in the `<eServicesManagerHome>\LanguageModel` directory.

> ## Important
>
> - If you accept the default path offered by the GAX installation, <eServicesManagerHome> is normally
>   - Windows: `C:\genesys\GCTI\`
>   - Linux: `/home/genesys/GCTI`
> - Alternatively, you can import the `lang.kme` file located in the <eServicesManagerHome>\LanguageModel directory if you want to use a model that only classifies emails in the following languages: English, French, German, Italian, Portuguese, Russian, and Spanish.

The training object consists of 20 categories, one for each language. Each category contains a number of text objects in its language. You can add more text objects to these categories as well. This could be especially valuable if you have a collection of text objects (such as e-mails) whose subject matter relates to your business. After you add text objects, you must train a new model to take advantage of the added data.

You can also add other languages to the model, as follows:

1. eServices Manager: On the **Standard Responses** tab (still with **unknown** selected as the language), add a category for the new language to the **LanguageDetection** category tree.
2. Content Analyzer:
   1. On the **Training** tab, select the **LanguageDetection** training object, select the new language category in the training object, then add text objects in the language.

2. Train a new model that includes the new language.

You can do this for any language supported by E-mail Server (E-mail Server supports all languages that are supported by the version of JRE that is supplied with Genesys eServices). However, Genesys has not tested any language other than those listed above.

Classification Server 8.5.2 and earlier

[+] Click to show section

As part of the installation of eServices Manager 8.5.2 and earlier, Genesys provides a model that classifies emails as English, French, German, Italian, Portuguese, Russian, or Spanish.

To import this model and its training object,

1. Open eServices Manager and select **unknown** as the language. If there is no such language you must create one in Genesys Administrator or Configuration Manager (under **Business Attributes > Languages**).
2. Then import the file `lang.kme`, located in the `<eServicesManagerHome>\LanguageModel` directory.

> ### Tip
>
> If you accept the default path offered by the GAX installation, <eServicesManagerHome> is normally
>
> - Windows: `C:\genesys\GCTI\`
> - Linux: `/home/genesys/GCTI`

The training object consists of seven categories, one for each language. Each category contains a number of text objects in its language. You can add more text objects to these categories as well. This could be especially valuable if you have a collection of text objects (such as e-mails) whose subject matter relates to your business. After you add text objects, you must train a new model to take advantage of the added data.

You can also add other languages to the model, as follows:

1. eServices Manager: On the **Standard Responses** tab (still with **unknown** selected as the language), add a category for the new language to the **LanguageDetection** category tree.
2. Content Analyzer:
   1. On the **Training** tab, select the **LanguageDetection** training object, select the new language category in the training object, then add text objects in the language.
   2. Train a new model that includes the new language.

You can do this for any language supported by E-mail Server (E-mail Server supports all languages that are supported by the version of JRE that is supplied with Genesys eServices). However, Genesys has not tested any language other than those listed above.

Actionability

To use the actionability sample, import the file `Actionability.kme`, which is located in the `<eServicesManagerHome>\ActionabilityModel` directory.

This provides:

- A model **Actionability** for analyzing actionability.
- The training object **Actionability** that created that model.
- A category tree **Actionability** that contains the categories to assign to interactions as a result of the analysis.

Sentiment

To deploy the sentiment sample, use the following procedure.

1. In Configuration Manager or Genesys Administrator, create a language called **English_Sentiment**.
2. With eServices Manager set to that language, import the file `EnglishSentiment.kme`, which is located in the `<eServicesManagerHome>\SentimentModel` directory.

This provides:

- A model **SentimentSampleModel** for analyzing sentiment.
- The training object **Sentiment** that created that model.
- A category tree **SentimentDetection** that contains the categories to assign to interactions as a result of the analysis.