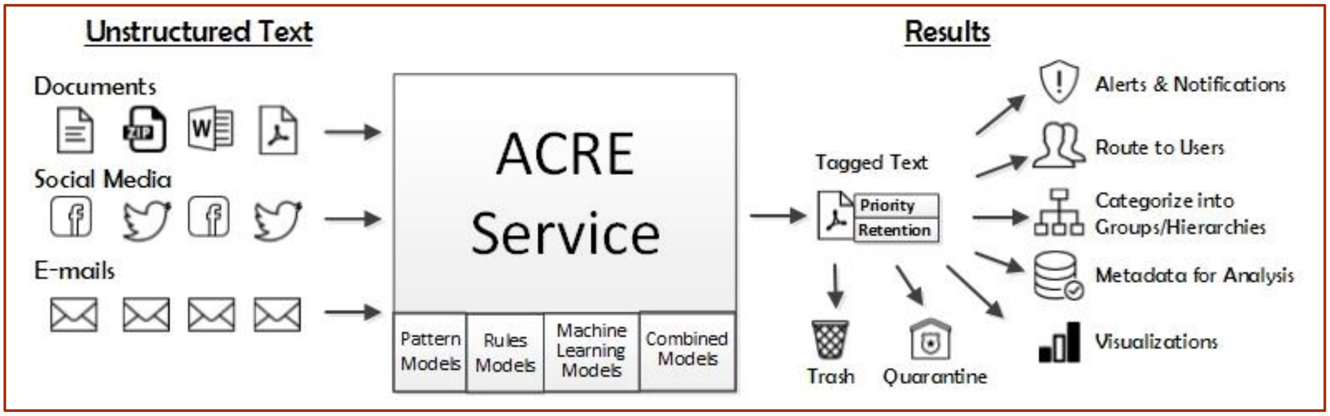
Extracting More Value from Text using ACRE



Vertical Data, LLC
P.O. Box 38, Bath OH, 44210
http://vertical-data.com
contact@vertical-data.com

Executive Summary

Businesses today are increasingly inundated with **unstructured text** in documents, e-mails, social media, survey results and more. More and more, text is becoming a part of the Big Data challenge.

This text often contains embedded information that is of considerable value to corporate stakeholders, including marketing, records management, customer service, security, and many others. Information from text can be used to reveal customer insights, determine data value and retention, identify potential threats, improve business processes, ensure regulatory compliance, and provide rich data for further business analytics. But much of this value is often lost due to the effort required for human review and labeling of text.

The **Analytics, Categorization and Retrieval Engine (ACRE)** is a new general-purpose **text categorization** service that analyzes text to quantify and extract its business value and then tags, monitors, filters, summarizes and routes this text data. Analysis is driven by a set of ACRE **Label Models**, each of which controls the assignment of one **label** or **tag** (such as Positive / Negative or Discard / Retain) to each piece of text using natural language processing (NLP), patterns, rules, machine learning and/or a combination of these techniques. Any number of label models can be executed on any text repository or data stream, resulting in a virtually unlimited number of new variables that can be used to gain insight and drive further analysis. ACRE is not tied to any particular document or content management system, but ACRE results can be incorporated into these systems using tag import mechanisms.

ACRE offers a library of pre-defined label models, but also provides an intuitive and open model development environment where text knowledge workers can develop customized models for labeling text. Powerful artificial intelligence tools, such as part-of-speech tagging and nearest-word-cloud machine learning, can be utilized without prior technical knowledge. As with some other intelligent tagging systems, labels can be generated based on Named Entities, but ACRE offers many other labeling options as well. Models can be quickly prototyped and then incrementally tested and improved until ready to deploy.

Any set of label models can be automatically executed on arbitrarily large text datasets or streaming sources using **ACRE jobs** that execute on the **ACRE Web Service** - a public or private cloud service that can be scaled across hundreds of processors. Job setup, execution, and results can be controlled by authorized users or applications using any web browser or RESTful SaaS client.

The ACRE Web Service can also provide **enhanced search** services on categorized text. Search results can be filtered and sorted by labels to quickly narrow results, providing a type of conceptual search. ACRE can also execute **find similar** ("more like this") and **find anomalies** searches. These capabilities can be used alone or in combination with other search techniques.

Contents

What is ACRE?

The Analytics, Categorization and Retrieval Engine (ACRE) is a platform for making decisions about text. The specific decision methods and parameters are stored in the cloud as ACRE Label Models, described below. Executing a label model on a text item generates a label or tag that stores the result of that decision.

A variety of predefined label models are available in the *Label Model Library*. New customized models can be created, tested and modified by authorized users or applications. In addition to tagging, ACRE can also take actions based on label values (such as filtering or routing the text), provide summaries and visualizations of results, and pass results on to other business analytics applications for further processing.

The **ACRE Service** is a cloud service that allows authorized users (through any web browser) or applications (through any RESTful SaaS client) to execute any of the following command:

1. Define text data sources, which can be spreadsheets of text, folders of data files, Twitter streams (extracted by hashtag or search terms), or database connectors.
2. Create, modify, view, or train an ACRE Label Model.
3. Define an ACRE Job, which specifies a data source, label models, parameters, notifications and outcomes.
4. Execute an ACRE Job and monitor its progress
5. View job results, including label assignments, summary results, word frequency tables and word clouds for all label sets.
6. Search and download job results, optionally filtering results by label values and/or sorting by similarity.

Users and applications are authenticated upon login. Access to all commands, data, and models is controlled and audited. Detailed commands, parameters and options for the ACRE v1.3 command-line release can be found in the *ACRE v1.3 User's Guide*, *Command Reference* and *Options Reference* available at http://vertical-data.com.

Accessing the ACRE Service

Figure 1 shows Users, Model Managers and Apps accessing an ACRE server. Model Managers are authorized to create, modify, train and test models, while Users can only view, search and download results or execute previously defined models. Authorized Apps can also perform any of these functions, depending on their access rights.

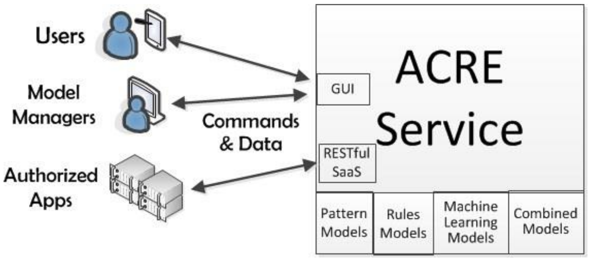


Figure 1: Transactional ACRE access

User Example: Joe has a set of new customer survey results in an Excel file. He logs onto the ACRE service, uploads the Excel file, executes Sentiment and Topic models, examines the results on the server and then downloads the results as label tables, summary tables and word clouds for further use.

App Example: An Oracle database receives a set of updated text forms, which triggers a script that uploads the revised forms to the ACRE server, executes 3 models, and then stores the resulting labels into several tables, automatically storing metadata for the new text.

Since model management capabilities are the same for human Model Managers and Authorized Apps, we will use the term "modeler" for both in the discussions below.

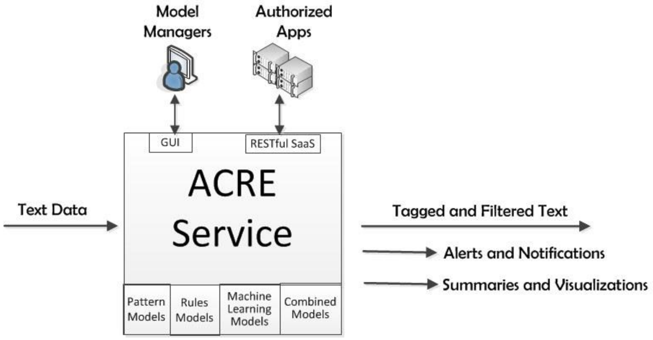


Figure 2: Streaming Text through ACRE

Figure 2 illustrates how ACRE can work with streaming text, such as e-mails or social media, that flow through from a defined source to a defined destination. Each text item will be tagged, and may be filtered or quarantined, allowing only approved content to pass through. Certain label outcomes can be set to generate e-mail or SMS text alerts. In addition, this text data may, at the modeler's discretion, be indexed and stored in the cloud, providing advanced search capabilities that incorporate the tag results to any authorized user from any web browser.

Throughout this white paper, we will use the word "document" to refer to an unstructured text input, which may be a file, e-mail, social media post, or other form of text.

Label Model Types

How does a label model select a label value? There are currently 3 basic model types and 2 combined model types, each of which can be configured in countless variations:

1. ***Pattern Models*** extract label values from the text data by matching regular expression[1] patterns within the text. Regular expressions provide extended search capabilities that go well beyond what is offered by standard search queries.

[1] https://en.wikipedia.org/wiki/Regular_expression

Example use case: *Data Loss Prevention:* Company A executes ACRE label models that detect all social security numbers, credit card numbers, project numbers, and other sensitive information patterns in outgoing e-mails and documents. Tagged documents can then be quarantined or forwarded for further inspection.

2. ***Rule Models*** assign label values from a **Category Tree (CT)** (defined below) based on a set of **rules** (keywords and regular expression patterns) associated with each label value. When a rule pattern is matched, the corresponding label is assigned (with some exceptions).

   Example use case: *Drug Name Tagging:* A health services IT organization uses ACRE to tag each incoming document with a standardized name for the drug discussed in the document. By defining a label for each standard drug name and then creating Rules with lists of alternate drug names, abbreviations, and foreign language equivalents, the model will add a label with the correct standard drug name to each document.

3. ***Machine Learning Models*** assign label values from a Category Tree based on the similarity of the document word cloud to a **trained word cloud** associated with each label value. Modelers create and modify the trained word clouds by training the model with example documents for each label value.

   Example use case: *Sentiment Emulation:* A marketing organization has a spreadsheet with a thousand Twitter tweets that have previously been tagged for Sentiment by a social media aggregator. They also have thousands of other tweets that are untagged. They upload the tagged document to the ACRE Service, select the Sentiment column and click “Train”. This creates a machine learning model (with trained word clouds corresponding to “Positive” and “Negative”) that can now be used to tag all other tweets for Sentiment.

4. ***Combined Models*** assign label values from a Category Tree using both rules and machine learning, providing significant modeling and performance advantages over using either method alone. Vertical Data, LLC, believes that these combined model designs are unique and has filed USPTO patent application #14676500 covering these categorization methods. Two ACRE Combined Models are:

   a. **Rules with ML Fill-in** models extend rule-based label assignments to documents that do not match any rules, by matching new document word clouds to trained word clouds corresponding to the rule matches. This can solve the problem of diminishing returns often seen in rules-only decision systems as additional rules are added.

      Example use case: *Language Labeling:* A political organization has collected 4000 survey responses, with some in English, French and Spanish. In order to split the responses by language, they defined a model with 3 labels (English, French, Spanish) and then chose 5 common words from each language and created Rules with these. Executing this as a Rules model labeled only about 30% of the responses (that is, the responses that actually contained at least one of the 15 words). Then executing this model in “Rules with ML Fill-In” mode correctly labeled the Language for 99.5% of the responses.

b. **Rule-Seeded Machine Learning** models are trained directly from rule matches. These trained models are then executed in ML mode on new documents. This can be used to quickly create machine learning models that evaluate document vocabulary similarity based on the occurrence of rule keywords and other words that typically co-occur with them. This eliminates the need to manually select training documents.

Example use case: Medical Record Clustering: A medical research team wishes to identify patients potentially at risk for diabetes through analysis of narrative clinical notes in electronic medical records (EMRs). They train a Diabetes model using the clinical notes from EMRs that are coded for diabetes or that contain diabetes-related keywords in their clinical notes. Executing this model on new patient EMRs will rank them based on the similarity of their clinical note vocabulary to the training set and then tag EMRs whose confidence values are above a given threshold.

Note that every parameter and option of every model type is fully visible to any authorized user and fully configurable by any authorized modeler. ACRE is a fully open model development environment. Further, every labeling decision can be traced back to the specific rule keyword or pattern (for rules models) or the set of specific trained word clouds (for ML models) that triggered the label choice. So ACRE labeling decisions are fully transparent and can be audited.

ACRE’s dynamic data visualization tools allow users to sort or filter each dataset using multiple label results, providing opportunities for new insights into data patterns and trends.

Modelers are given a rich set of advanced parameters that can be modified to fine-tune model performance. All options have default values that work well in most cases.

Category Trees

The Category Tree (CT) specifies the possible label values for all model types except Pattern Models (which require only a pattern string). The Category Tree is a hierarchical set of label nodes, where the top node (the root) contains the model name and label values are assigned from the leaf nodes. For example, Figure 3 shows the category tree for a basic Sentiment label model.

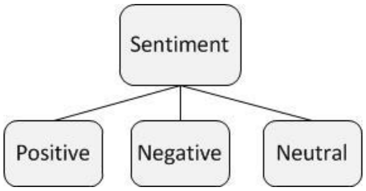


Figure 3: Category Tree for Sentiment Model

When this model is executed on a set of documents, each document will be assigned to one or more label values, as shown in figure 4. Depending on model design, some documents may be assigned a "No label" value.

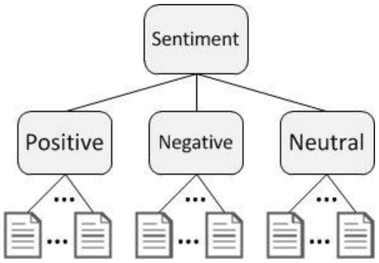


Figure 4: After model execution – Each document assigned to single label value

Sentiment is an example of a single-value model, which assigns only one label to each document. For example, it would make no sense to assign both "Positive" and "Negative" labels to a document just because it contains both positive and negative words (though some other sentiment labeling systems do exactly that!). Instead the model must choose the "best" choice among multiple matches. ACRE Rule models use rule weights to make this decision, choosing the label with the highest summed weight over all rule matches. By default, this will choose the label with the most matches, but this behavior can be changed if the modeler modifies the default rule weights. For ML models, the label with the highest Confidence level (defined below) is chosen.

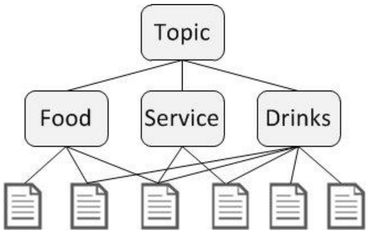


Figure 5: Example: Each survey response assigned to one or more label values (Multi_Values = YES)

On the other hand, for some models, assigning multiple labels is the right thing to do. Figure 5 shows a tree for categorizing restaurant survey responses based on whether words related to Food, Service or Drinks are mentioned in the response. In this case, a survey response that discusses both food and service should be assigned both labels. The modeler sets option Multi_Values=YES to enable this multi-labeling. In this case, for rules models, all matched rule labels will be assigned and, for ML models, all labels with Confidence > Threshold (where Threshold is another configurable model parameter) will be assigned.

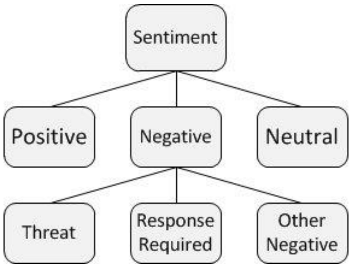


Figure 6: Adding a level to the tree

Category trees can have any number of levels. Figure 6 shows an extended Sentiment tree that splits the “Negative” node into three sub-nodes. A document that would have been labeled “Negative” in the basic model will now be labeled “Threat” if it contains threatening or violent words, “Response Required” if it contains demands or high emotional content, and “Other Negative” otherwise.

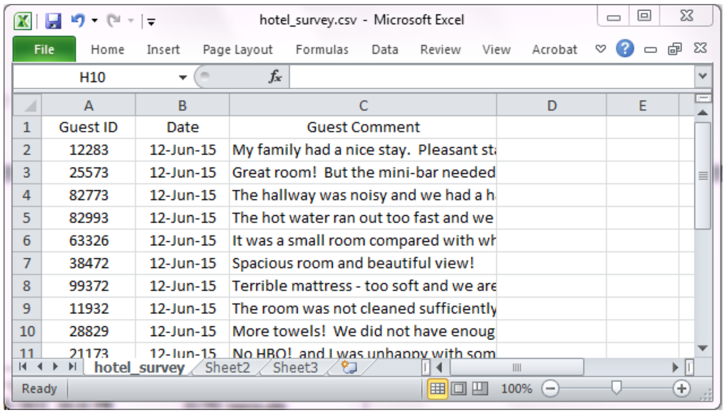
This extended Sentiment tree would provide added value to an organization that uses it to monitor customer feedback and wants to respond quickly when appropriate. A separate Request Priority model can also be designed that sorts the “Response Required” results by keywords and/or emotional content so that higher priority requests are seen first.

Organizations can also use existing hierarchical schema, such as organizational charts or Active Directory trees, as Category Trees and create ACRE models that map incoming documents into this schema. For example, mapping documents onto an organizational chart containing e-mail addresses allows the associated model to automatically send categorized documents by e-mail to the selected individual.

Increasing the number of nodes in the category tree also increases the set of analytic results available after model execution. Each tree node represents a different set of labeled documents (i.e., the documents assigned at or below that node), and all model performance results, including label tables, summary tables, word frequency tables and word clouds, can be generated separately for each node in the tree.

An Example: Motel5

Motel5 is a hotel/motel chain that uses a web application where guests can submit text comments about their stay. Motel5 has collected 452 guest comments in file hotel_survey.csv – shown below.



	A	B	C	D	E
1	Guest ID	Date	Guest Comment		
2	12283	12-Jun-15	My family had a nice stay. Pleasant st		
3	25573	12-Jun-15	Great room! But the mini-bar needed		
4	82773	12-Jun-15	The hallway was noisy and we had a h		
5	82993	12-Jun-15	The hot water ran out too fast and we		
6	63326	12-Jun-15	It was a small room compared with wh		
7	38472	12-Jun-15	Spacious room and beautiful view!		
8	99372	12-Jun-15	Terrible mattress - too soft and we are		
9	11932	12-Jun-15	The room was not cleaned sufficiently		
10	28829	12-Jun-15	More towels! We did not have enoug		
11	21173	12-Jun-15	No HBO! and I was unhappy with som		

Figure 7: Original Motel5 customer survey data

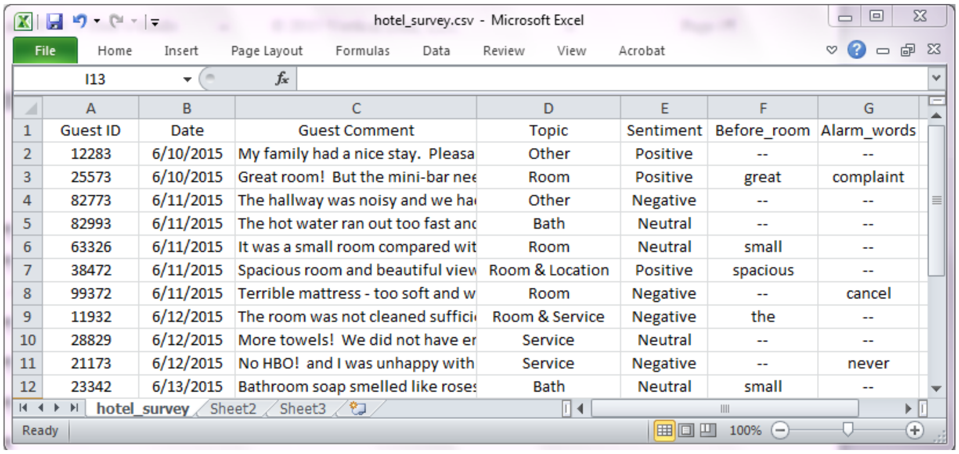
Motel5 decides to execute 4 label models to add value to this data:

1. A **Topic** model that identifies topics mentioned in each response. Label values = {Room, Bath, Service, Location, Other}.
2. A **Sentiment** model. Label values = {Positive, Negative, Neutral}.

3. **Before_Room** - a Pattern model that will **capture the word before "room"** for any comments containing that word. This will be used to help identify what clients are saying about their rooms. The regular expression pattern for this model is "(\w+) room".

4. A Pattern model that will **capture "alarm words"** from a list of threatening or violent words that merit further investigation.

Results

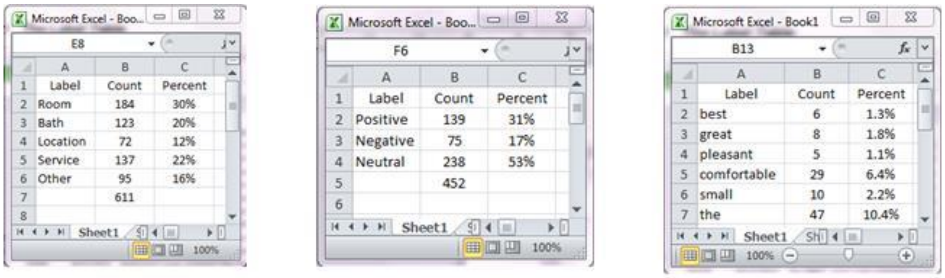


	A	B	C	D	E	F	G
1	Guest ID	Date	Guest Comment	Topic	Sentiment	Before_room	Alarm_words
2	12283	6/10/2015	My family had a nice stay. Pleasa	Other	Positive	--	--
3	25573	6/10/2015	Great room! But the mini-bar nee	Room	Positive	great	complaint
4	82773	6/11/2015	The hallway was noisy and we ha	Other	Negative	--	--
5	82993	6/11/2015	The hot water ran out too fast anc	Bath	Neutral	--	--
6	63326	6/11/2015	It was a small room compared wit	Room	Neutral	small	--
7	38472	6/11/2015	Spacious room and beautiful view	Room & Location	Positive	spacious	--
8	99372	6/11/2015	Terrible mattress - too soft and w	Room	Negative	--	cancel
9	11932	6/12/2015	The room was not cleaned suffici	Room & Service	Negative	the	--
10	28829	6/12/2015	More towels! We did not have er	Service	Neutral	--	--
11	21173	6/12/2015	No HBO! and I was unhappy with	Service	Negative	--	never
12	23342	6/13/2015	Bathroom soap smelled like roses	Bath	Neutral	small	--

Figure 8: Label Table Results

Figure 8 shows the Label Table resulting from the execution of the 4 models on the Motel5 survey data. Four new columns have been appended to the survey data, each containing the results from executing one ACRE Label Model. Each new column contains the model name at the top and the assigned label values for each of the 452 survey comments. Since the modeler configured "Multi_Values = YES" for the Topic model, some cells in that column have multiple label values.

Summary Tables



	A	B	C
1	Label	Count	Percent
2	Room	184	30%
3	Bath	123	20%
4	Location	72	12%
5	Service	137	22%
6	Other	95	16%
7		611	
8			

Summary for Topic

	A	B	C
1	Label	Count	Percent
2	Positive	139	31%
3	Negative	75	17%
4	Neutral	238	53%
5		452	
6			

Summary for Sentiment

	A	B	C
1	Label	Count	Percent
2	best	6	1.3%
3	great	8	1.8%
4	pleasant	5	1.1%
5	comfortable	29	6.4%
6	small	10	2.2%
7	the	47	10.4%

Summary for Before_room

Figure 9: Summary Tables for Motel5 Model execution results

ACRE automatically generates summary tables, as shown in Figure 9. Totals for the Topic model (611) are higher than the number of comments (452) due to the multi-value assignments.

Note that, since ACRE generates multiple new variables, Motel5 can use these label results to generate new multivariate analyses, or crosstabs. For example, Figure 10 shows Sentiment results per Topic. From this data, Motel5 can learn, for example, that most comments about

Service were negative, while comments about Location were mostly positive. These results provide further insights, drive further actions and extract more value from the text!

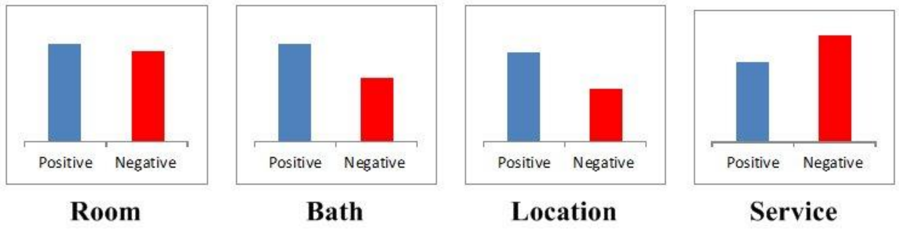
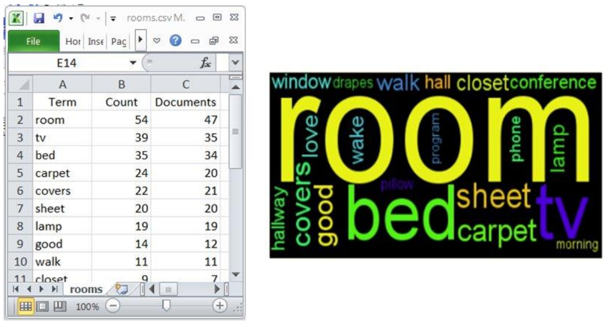


Figure 10: Examples: Multivariate results

Of course, as the number of ACRE models executed on a data set is increased, the opportunities for multivariate analysis and insights grow exponentially!

Word Counts and Word Clouds

ACRE also provides word count tables and corresponding word clouds for each node in the Category Tree. For ML models, each node has 2 associated word clouds: the Trained Word Cloud generated by the Modeler training, and the Documents Word Cloud reflecting the contents of the documents processed during model execution. Figure 11 shows the Documents Word Cloud for the "Room" node of the Topic model.



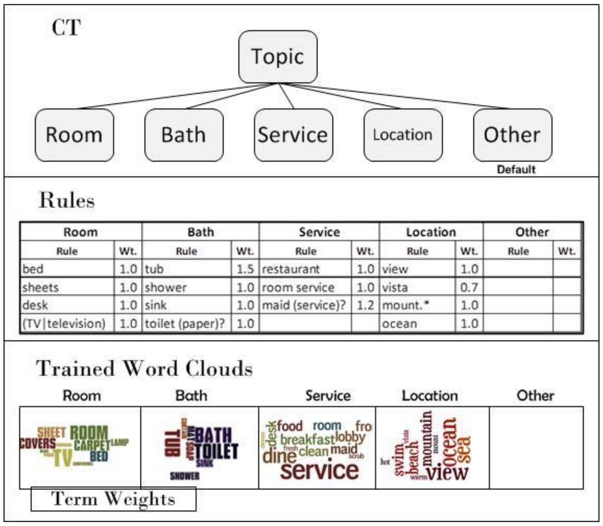
	A	B	C
1	Term	Count	Documents
2	room	54	47
3	tv	39	35
4	bed	35	34
5	carpet	24	20
6	covers	22	21
7	sheet	20	20
8	lamp	19	19
9	good	14	12
10	walk	11	11
11	closet	9	7

Figure 11: Documents Word Frequency Table and Word Cloud for "Room" node

Depending on how the modeler configures the model, these word clouds may not include all words in the documents, but only the words within the **vocabulary of analysis** that is used in ML analysis (defined below). For example, insignificant words, such as "the", "a", "and", etc., are typically eliminated from this vocabulary.

Model Features

The major model components stored in the cloud by the ACRE Service are shown in Figure 12 below. The defined Category Tree structure is stored, and one leaf node can be designated as the Default value, which is assigned in the case when no other label value would be assigned.



Room		Bath		Service		Location		Other	
Rule	Wt.	Rule	Wt.	Rule	Wt.	Rule	Wt.	Rule	Wt.
bed	1.0	tub	1.5	restaurant	1.0	view	1.0		
sheets	1.0	shower	1.0	room service	1.0	vista	0.7		
desk	1.0	sink	1.0	maid (service)?	1.2	mount.*	1.0		
(TV|television)	1.0	toilet (paper)?	1.0			ocean	1.0		

Figure 12: Stored contents of an ACRE model

Rule Model Parameters

For each CT leaf node, modelers can view and set (1) Rule patterns (keywords or regular expressions) and (2) Rule Weights, which default to a value of 1. When a Rules Model is executed, all Rule patterns are compared against each input document. If Multi_Values = YES, then all rule match labels are assigned. If Multi_Values = NO, then the single label with the highest sum of matched rule weights is assigned. Modelers can modify rule weights to influence this outcome.

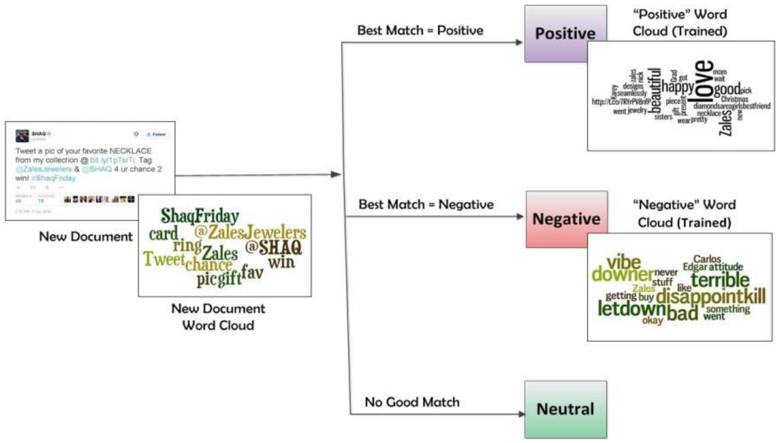
Label table results can also be sorted by the total matched rule weight for each result, which provides a method to rank results by number of matches or significance of matches, based on how weights are set.

Machine Learning Model Parameters

By default, ACRE ML models execute the **Nearest Word Cloud** machine learning algorithm, which assigns each document to the label value whose Trained Word Cloud (TWC) is most similar to the word cloud of the document.

Trained Word Cloud values are set as Modelers identify training documents for each label value. When the model is executed on a new document, a ***Confidence*** value is calculated for each TWC, with higher Confidence values corresponding to greater similarity between the TWC and the word cloud of the document. The label with the highest Confidence value is selected.

Figure 13 shows an example of Nearest Word Cloud model execution to select a Sentiment label value for a social media document. The document's word cloud is compared against Trained Word Clouds for Positive and Negative label values. The document will be assigned to the label value with the greatest Confidence value that exceeds the Threshold value. If no Confidence value exceeds Threshold, then the document will be assigned the Default value of Neutral.


SHAQ

Tweet a pic of your favorite NECKLACE from my collection @ bit.ly/1pTslTi. Tag @ZalesJewelers & @SHAQ 4 ur chance 2 win! #ShaqFriday

Figure 13: Label Selection by Nearest Word Cloud classification

The Nearest Word Cloud (NWC) algorithm was chosen as the ACRE default based on the following properties:

a) Intuitive and Transparent

ACRE Trained Word Clouds and Document Word Clouds are easily viewed at any point in the analysis, which provides a visual method for modelers to verify why each particular label was assigned. In contrast, most other ML algorithm use methods that are opaque and difficult to understand. This visual verification, in combination with word frequency table comparisons, can guide the modeler in determining whether additional rules or training are needed to improve results.

b) Scalable

Both model training and execution tasks can be distributed across multiple processors using architectures such as Hadoop, which provides high levels of scalability. Since Nearest Word Cloud is a linear-time algorithm (notation: **O**(n)), execution time will not grow unduly large as the number and size of input documents increases.

c) Accurate

Academic studies[2] have shown that Nearest Word Cloud provides higher accuracy than other linear-time text classification algorithms, such as Bayesian, k-nearest-neighbors and C4.5.

To further test ACRE algorithm accuracy, we have executed it on the RCV1 corpus[3], in which Reuters editors manually categorized hundreds of thousands of news articles using a category tree with 55 second-level labels. After training on 1000 articles, an ACRE ML model was executed on 500 new articles, selecting the

[2] E.-H. Han, G. Karypis, "Centroid-Based Document Classification: Analysis and Experimental Results", 2001

[3] http://trec.nist.gov/data/reuters/reuters.html

single best label for each new article. The ACRE label selection was correct (that is, it matched a label selected by a Reuters editor) for more than 70% of the new articles.

A critical factor in ML model accuracy is the choice of the vocabulary of analysis, that is, the set of words and phrases that are used in calculating Confidence values. Modelers can significantly reduce noise in the ML calculations and increase accuracy by narrowing this set down to only those words and phrases that should be significant in document comparisons.

ACRE offers many useful tools for managing and evaluating this vocabulary of analysis, including drop lists, go lists, part-of-speech and named-entity tags, and bi-grams - which automatically include all 2-word phrases in this vocabulary. Modelers can also fine-tune ML model performance by changing Term Weights, which are numerical values that can be configured to either boost or diminish the influence of particular individual terms (words or bi-grams) in the calculation of Confidence values.

Finally, label table results can be sorted by Confidence value, which provides a simple method to implement **Find Similar** (highest Confidence) and **Find Anomalies** (lowest Confidence) functions.

Natural Language Toolkit

ACRE utilizes the well-tested Natural Language Toolkit (NLTK) library[4] to provide Natural Language Processing (NLP) functions that enhance modeling capabilities and improve accuracy.

1) Stemming

Modelers can make use of several NLTK stemming algorithms, each of which substitutes the stem, or root, of each word in place of the word itself in the analysis. This allows the model to treat words that have the same root, such as “child” and “children”, as identical, which increases accuracy. In our experiments, enabling stemming typically increases ML model accuracy by 3% - 5%.

2) Part-of-Speech (POS) Analysis

Part-of-speech analysis tags each word with its part of speech (i.e., noun, verb, adjective, etc). This allows more specific rule matches (for example, a rule could match “care” as a noun, but not as a verb) and also allows a more precise specification of the ML vocabulary of analysis (for example, a modeler can specify that ML analysis should be done using nouns or adjectives only).

3) Named Entity Recognition (NER)

Named Entity Recognition detects and tags named persons, organizations, locations and times within the text. These NER tags can be used for more precise rule matches (for

[4] http://www.nltk.org/

example, match “brown” as a name, but not as a color), more precise specification of ML vocabulary of analysis, and enhanced search capabilities.

Conclusions

In summary, the Analytics, Categorization and Retrieval Engine provides a service platform to build, test and execute text categorization models that allow organizations to extract more value from their text. In particular, ACRE allows an organization to:

- Automatically Categorize (tag or label) text using both rules and machine learning
- Monitor text for keywords and patterns that require immediate action
- Cluster or Group text items that have similar content
- Filter text to eliminate unrelated items or spam
- Gain insights into customer concerns, preferences, sentiment, and other issues within surveys and social media.
- Turn text into numbers for analytics by providing quantitative results which can be integrated into numerical data analytics processes
- Retrieve and Sort text using enhanced search, Find Similar and result ranking
- Summarize text and topics using label counts and word frequency tables
- Visualize text and topics using word clouds

All of these allow an organization to extract more value from text data.

Let us demonstrate what ACRE can do for your organization! Please contact us:

Vertical Data, LLC
P.O. Box 38, Bath OH, 44210
http://vertical-data.com
contact@vertical-data.com

Some cover icons from http://icons8.com