SAS587-2017

Exploring the Art and Science of SAS® Text Analytics: Best practices in developing rule-based models

Murali Pagolu, Christina Engelhardt, and Cheyanne Baird, SAS Institute Inc.

ABSTRACT

Traditional analytical modeling, with roots in statistical techniques, works best on structured data. Structured data enables you to impose certain standards and formats in which to store the data values. For example, a variable indicating gas mileage in miles per gallon should always be a number (for example, 25). However, with unstructured data analysis, the free-form text no longer limits you to expressing this information in only one way (25 mpg, twenty-five mpg, and 25M/G). The nuances of language, context, and subjectivity of text make it more complex to fit generalized models. Although statistical methods using supervised learning prove efficient and effective in some cases, sometimes you need a different approach. These situations are when rule-based models with Natural Language Processing capabilities can add significant value. In what context would you choose a rule-based modeling versus a statistical approach? How do you assess the tradeoffs of choosing a rule-based modeling approach with higher interpretability versus a statistical model that is black-box in nature? How can we develop rule-based models that optimize model performance without compromising accuracy? How can we design, construct, and maintain a complex rule-based model? What is a data-driven approach to rule writing? What are the common pitfalls to avoid? In this paper, we discuss all these questions based on our experiences working with SAS® Contextual Analysis and SAS® Sentiment Analysis.

INTRODUCTION

In the last two decades, the world of analytics saw a lot of interest and research in analyzing data available in text. Extracting meaningful insights from text data is a Herculean task due to the fact that human language is complex, unstructured, nuanced and generally comes with a very low signal to noise ratio. It is a great advancement in science when humans can impart capabilities to machines for analyzing and interpreting text. SAS® Text Analytics depends on computationally intensive algorithms, statistical techniques and natural language processing methods. Broadly, there are two different methodologies that you can use for analyzing text data in the SAS world – the statistical approach and the linguistic approach (Chakraborty, Pagolu, and Garla, 2013).

In the statistical approach (also known as bag-of-words approach), the frequency of occurrence and co-occurrence of terms in the document collection (also known as the *corpus*) play a key role. Those numbers are generated in a table named the term-by-document matrix, and then condensed further by means of dimension reduction techniques such as singular value decomposition. In the linguistic approach, you deal with the semantics of the terms and the context in which they appear in each document, but not how frequently they appear across the entire corpus. You can develop linguistic rules that use keywords and phrases, Boolean operators, linguistic qualifiers, lemmatization (the capability to roll up term variations and inflections to the root word), part-of-speech recognition, regular expressions, and entity/fact definitions.

These two approaches fundamentally differ in the sense that the statistical approach characterizes an entire corpus by considering all the documents in the collection at once (*inter*-document analysis), whereas the linguistic approach only inspects a single document at a time, evaluating it in isolation against a set of predefined rules (*intra*-document analysis).

In the statistical approach, the large number of terms which make up the entire corpus are analyzed in order to discover topics or themes which depict the document collection. SAS® Text Miner calculates frequency weights (local weights) and term weights (global weights) based on factors such as document-specific term frequency, frequency of most frequent term, number of documents and the number of documents in which a term appear (Chakraborty, Pagolu, and Garla, 2013). While frequency weights help

determine the importance of a term in the overall composition of a document, term weights help you understand which terms can better discriminate between the documents. This fundamental assumption that those terms that are moderately frequent across the corpus but are highly frequent within those documents in which they appear can very well discriminate between groups of documents is the basis for the unsupervised techniques such as clustering and text topic extraction in SAS® Text Miner. A text analytics model built using the statistical approach can also be termed probabilistic since it quantifies the probability of a document belonging to a particular cluster and then finds terms that can explain those clusters using the calculated weights.

By way of contrast, in the linguistic approach there is no such weighting mechanism prescribed. A rule-based model doesn't characterize the entire corpus in any way, rather every linguistic rule/definition in the model is evaluated for each document individually. It either extracts a piece of information from the full text, or classifies the document into zero, one, or multiple categories. A model developed using the linguistic approach is deterministic in nature, not probabilistic. Either a document satisfies a given rule completely or it doesn't; there is no ambiguity about the outcome.

SAS® Contextual Analysis offers its users the ability to develop linguistic rules to define a framework for classifying a corpus into pre-defined labels that are also known as categories. Developing linguistic rules requires using subject-matter expertise as well as an understanding of the grammatical structures and the nuances of the language in the corpus. SAS Contextual Analysis also offers the flexibility for the analyst to write rule-based models for extracting important pieces of information from each document, regardless of the statistical importance of those terms or phrases in the corpus. For certain types of data, the linguistic approach can yield higher accuracy than the statistical approach, although the tradeoff is that a linguistic model typically takes a longer period of time and more planning to develop.

As the field of text analytics matures and incorporates deep learning and dependency parsing, these two worlds tend to converge; the mathematical approach derives semantic and syntactic relationships between terms that can further inform linguistic rules. In this paper, our focus is on discussing best practices for textual data analysis using the linguistic approach and occasionally comparing the pros and cons with the statistical approach. Readers are expected to have a basic understanding of Boolean operators, qualifiers, regular expressions, concepts and fact extraction rules. We strongly encourage you to refer to *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS*.

ASPECTS OF TEXTUAL DATA ANALYSIS USING LINGUISTIC APPROACH

Before we delve a little further into the intricacies of linguistic approach, let us understand what SAS offers in this space today. **Contextual Extraction** is the ability to extract important bits and pieces of information from documents using various types of rule definitions. For example, you can extract names of individuals holding office in important positions at the White House mentioned in news articles for the past year. This task might require a combination of knowledge on the grammatical structure or patterns that are predominant in that collection of articles, as well as a list of names of those holding key positions in the White House. In other words, both an understanding of how to model language *and* subject matter expertise are important. SAS® Contextual Analysis offers regular expressions, part-of-speech tags, and predicate logic that can be used to build simple to advanced contextual extraction definitions.

Document Classification is the term used for the process of assigning documents in the corpus to one or more categories as the use case demands. Depending on the use case, a document can be classified to one and only one category or can be assigned to more than one category. For a vast majority of use cases, a pre-defined set of categories is determined beforehand. The categorization model development framework (also known as *categorization taxonomy*) in SAS Contextual Analysis is built to largely take advantage of linguistic rule-based model development while also capitalizing on some aspects of the statistical approach as a precursor to the model development exercise. This hybrid approach works well for those who want the tool to produce a "starter" rule-based model which they can then manually modify to develop their final taxonomy. The best thing about this approach is that it doesn't require adequate amounts of training data with a defined target variable indicating the actual category to which a document belongs. In addition, there might be cases where certain documents don't belong to any of the pre-defined categories and you might miss them in your reporting if you never perform unsupervised learning to detect and capture these unexpected themes. It is a best practice to define an "Unknown" or

"Unclassified" category, either by defining appropriate linguistic rules to ensure the uncategorized documents fall in this bucket, or by labeling documents which do not match to any of your categories as "Unclassified" during model execution and post-processing.

Sentiment Analysis is the science of picking up the affinity of the author of the text towards a particular object or its features and classifying it into any of the following four categories considered widely as the industry standard: positive, negative, neutral, or unclassified. SAS Contextual Analysis offers automatic document-level sentiment, while SAS® Sentiment Analysis caters to the needs of advanced use cases that require a custom hierarchy for extracting sentiment towards specific objects and/or their corresponding features. Both contextual extraction and feature-level sentiment analysis lend themselves slightly better to the linguistic, rules-based approach.

APPROACHES TO RULE BUILDING: MACHINE VERSUS MANUAL VERSUS HYBRID

Assuming you have sufficient volumes of high quality training data, supervised machine learning models are quick to generate and are often more robust than a human-built taxonomy. That is, they often include term variants, term combinations, or topics that a person might not have intuitively come up with on his or her own. Different techniques are available for comparing machine learning models, and users can modify parameters and weighting strategies. The accuracy/speed-to-value ratio is very favorable, as is the scalability aspect.

However – although manual rule-writing is more time consuming than statistical text modeling, the linguistic approach more readily allows for advantages such as these:

- explicit incorporation of input from Subject Matter Experts (SMEs)
- good alignment with business objectives
- transparent and human-readable/editable logic (critical for highly regulated industries where models must be explainable)
- clean, business-interpretable outputs immediately ready for executive level reporting

As mentioned previously, the hybrid approach may be a good option. Let the machine do the bulk of the heavy lifting but then the analyst comes in afterward to add in subject matter expertise. Remove extraneous logic which may not be aligned with business goals.

MODEL VALIDATION

Having a good amount of hand-scored "gold-standard" data is ideal for effective model validation and testing the robustness of a model – you could train a statistical model on some of it, and use the remainder for validation. With data to use as ground truth, you can calculate precision and recall statistics, confusion matrices, and have benchmarks for percent accuracy. Unlike binary target variables in a typical predictive modeling situation with structured data elements, rule-based modeling for large scale taxonomies has typically several hundreds of categories. It means the confusion matrix is very complicated and difficult to digest. For the purpose of validating categories, we suggest that you generate as many gold-standard data sets as the number of individual categories you are building, with sufficient representative data to validate each category in the taxonomy. An additional benefit of having analysts and SMEs hand-scored data prior to modeling is that they internalize the keywords, patterns, and constructs that apply to that particular type of data. It gives a sense for what can and cannot be feasibly accomplished through text modeling. This knowledge will be invaluable as they build the rules and taxonomy.

MODEL TUNING

Realistic goals should be set for model accuracy (assuming gold-standard data from the previous section) taking these points into account:

- Data from sources such as social media posts or discussion forums will be noisier and harder to model accurately when compared with product reviews, news articles, claims notes, survey responses, and official documentation, etc. Informal chatter, conversation threads, types of slang and

abbreviations used in the online vernacular are more dependent upon heuristics and context for interpretation than other sources. Set your expectations for accuracy according to your needs.

- It is important to assess how accurately the model needs to perform in order to meet your goals. For example: near-perfect accuracy might be required for high-impact use cases such as medical documents or legal contracts, but is less important in gauging directional market sentiment on products.
- Consider which is more important in your situation: sensitivity versus specificity? There can be a cost associated with a false positive case versus a false negative case. Developing a confusion matrix evaluating overall profit or risk based on the prediction accuracy can be useful in those cases.
- There will likely be dependencies within the category rules. For example, Category A is referenced in the rule definition for Category B. In such cases, tuning rule definition for Category A can affect the performance of Category B and both should be evaluated during testing.

PERFORMANCE AND SCALABILITY

When performing contextual extraction of entities, facts, or sentiment, SAS offers many options for rule types, some of which can do the jobs of others. It is a best practice to use the least powerful rule type that will accomplish the task at hand. In other words, don't use a sledge hammer for a nail!

Why? Because more advanced rule types come at a cost of computational intensity, which may affect model performance at execution. For example, don't use a REGEX rule to match a literal string (keyword/phrase), when a simple CLASSIFIER rule will achieve the same goal. Similarly, do not use a PREDICATE_RULE when a C_CONCEPT or a CONCEPT_RULE will do. This guideline is particularly relevant in situations where nested rule definitions are created, or concepts are referenced in other definitions. For example, consider the two PREDICATE_RULE concepts "RuleA" and "RuleB" below, which contain references to "ConceptA" through "ConceptE" which can be CLASSIFIER, CONCEPT, or REGEX based concept definitions.

```
RuleA:
PREDICATE_RULE:(x,y):(AND,(DIST_4,"_x{ConceptA}","_y{ConceptB}"),(NOT,"RuleB"
))

RuleB:

PREDICATE_RULE:(x):(AND ,"_x{ConceptC}","ConceptD",(NOT,"ConceptE"))
```

Concept "RuleA" is a PREDICATE_RULE which extracts matches when ConceptA and ConceptB are found within a distance of four words from each other only when there are no matches found anywhere in the document for the concept "RuleB". In "RuleB", the definition verifies the pass/fail conditions of its referenced concepts, yet nothing is done anywhere with the matched "ConceptC" string returned by "RuleB". Reference to the concept "RuleB" within the PREDICATE_RULE in "RuleA" works as a binary test of yes/no as to whether "RuleB" matches. In "RuleB" only one argument is returned, predicate logic is not necessary and we can convert the PREDICATE_RULE in "RuleB" to the following:

```
RuleB - CONCEPT_RULE:(AND,"_c{ConceptC}","ConceptD",(NOT,"ConceptE"))
```

Some additional best practices:

- Avoid building contextual extraction models when a categorization model is sufficient, as there are performance implications.
- Exercise caution using REGEX (regular expression) rules as they are powerful, yet computationally intensive. Unless written carefully to match patterns precisely, they can cause a lot of performance issues.
- When deciding whether to create one model for all sources versus a model per source, consider performance and whether it's worth running a lot of data through a model that you know won't fit it.
- Consider pre-processing data for performance improvements. For example, stripping out things like

html tags might reduce the processing time, even though having them in there doesn't affect your model. Similarly, consider removing useless tables or plain text representations of images or chart objects, and so on.

MAINTAINABILITY

If multiple users or groups will be building rule-based text models on shared data or for similar purposes, we encourage you to build and maintain centralized, reusable helper/intermediate entities, so that everyone has some base definitions to start with are consistent. Create base definitions/assets for concepts such as brand names, product names, synonyms, and misspelling lists.

A common question is: How can I evaluate the efficacy of specific rule parts, or sub-rules, within a single category/concept definition? This is possible only through regression testing where you need to start with one or two rules within a category which yield maximum recall and reasonable precision. Then as you keep adding more and more rules, you will see that while precision keeps climbing up, recall starts declining. These additional rules can be negation rules to exclude false positives. A well-developed classification model always strikes the best possible balance between precision and recall values. Figure 1 is an illustration of how adding linguistic rules to a category increases recall and precision initially, and then curves down with falling recall and rising precision values. Points marked 1 through 6 in the plot represent the sub-rules within a category. In this example, it is ideal to stop after adding the first four rules.

If you have gold-standard data that's been human verified, you will have a baseline to see if the new model is performing better or worse. Also, you might run diagnostics on the percentage of overall corpus that the model is encompassing. For example, if the original model has hits on 90% of the taxonomy and it's been verified that the remaining 10% are of no value, you should watch this percentage. If it dips to 70% or 80%, it might be time to perform more unsupervised topic exploration on the unclassified data to see what emerging topics you may find that your model is not capturing.

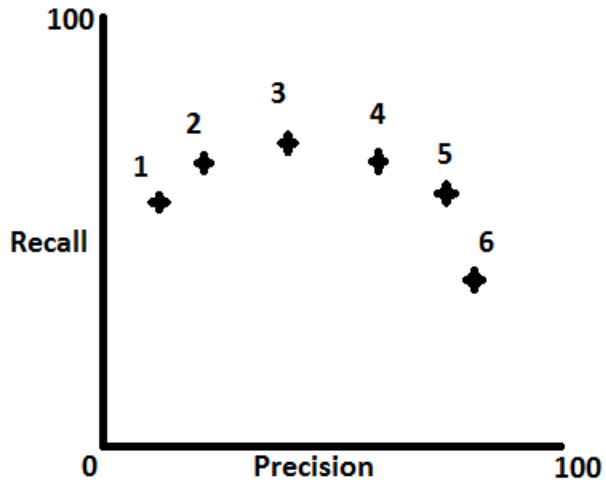


Figure 1. Recall versus Precision Plot for a Linguistic Rule-Based Category

ASSESSING THE IMPACT OF CATEGORY RULE MODIFICATION

It is quite common and feasible to periodically modify your category rules. In such instances, you might want to quantify how changing the rules affects your accuracy metrics (recall and precision) for that category. In the example below, Category1 is an existing rule comprised of certain keyword arguments, and a concept "Rule2" referenced as shown below:

Category1: `(OR, "term1", "term2", "term3", "term4")`

Category2 is similar to Category1 but has omitted terms term3 and term4.

Category2: `(OR, "term1", "term2")`

Now, a simple rule in Category3 like the one below can show documents that match only Category1 exclusively but not when they match Category2. Thus, this rule serves to provide an answer for you to understand how your rule modification has affected document matches and what documents you might have lost from dropping those two terms in Category2.

Category3:

```
(AND,
   _tmac:"@Top/Category1",
   (NOT,
      _tmac:"@Top/Category2"
   ))
```

In cases where you have added some terms, dropped some terms, or modified Boolean logic of any kind, it is a good idea to add another category as shown below while testing to know the impact in both directions, since you might have lost some matches and gained others.

Category4:

```
(AND,
   _tmac:"@Top/Category2",
   (NOT,
      _tmac:"@Top/Category1"
   ))
```

CAVEATS WHEN USING BOOLEAN RULE-BASED OPERATORS

With regard to the usage of Boolean linguistic rules in SAS Contextual Analysis, it is important to note a few things.

- Usage of the AND/NOT operator requires some diligence as they are essentially global in nature. This means that they are applied on the whole document and not limited by the boundaries of an encapsulating DIST or SENT operator. In other words, using AND/NOT operator within SENT or DIST operators will not yield the desired results; in these cases, you should use NOTINSENT or NOTINDIST operators.

  The NOTINSENT operator is used if you need to ensure a condition A is satisfied (that is, a certain set of terms need to appear) yet at the same time, you do not want another condition B to be found within the same sentence where condition A is satisfied. Please see the following examples to understand the usage of AND/NOT and NOTINSENT operators.

  Example 1 (AND/NOT):

  ```
  (AND,
     (OR, "term1", "term2", "term3"),
  (NOT,
     (OR, "term4", "term5", "term6")
  ))
  ```

  This rule will assign a document to this category if any of the desired terms 1, 2, or 3 are mentioned, but not if any of terms 4, 5, or 6 are present *anywhere in the entire document*. Wrapping the entire rule in an encompassing SENT or DIST operator will not change the global nature of this exclusion.

  Example 2 (NOTINSENT):

  ```
  (NOTINSENT,
     (OR, "term1", "term2", "term3"),
     (OR, "term4", "term5", "term6")
  )
  ```

This rule will assign a document to this category if any of the desired terms 1, 2, or 3 are mentioned, but not if any of terms 4, 5, or 6 are present *in the same sentence as the desired term*. Note that if terms 4, 5, or 6 occur elsewhere in the document, it will not prevent the match from occurring to this category. This logic is useful for removing unwanted contexts for a term at a local level, with the understanding that the excluded terms might be used in a valid way elsewhere in the document.

- In addition, use the following approach when you are writing sentence-level rules:

  If you need to verify the existence of conditions within a sentence, it is better to use the individual arguments for the SENT operator directly. The arguments for the SENT operator have an implicit AND relationship; nesting an explicit AND operator as well might not return the results you want. So, the rule below is not the correct way of using the SENT operator.

  Example 3 (Incorrect usage of SENT operator):

```
(SENT,
   (AND,
         (OR, "term1", "term2", "term3"),
         (OR, "term4", "term5", "term6")
   )
)
```

  Instead, use the following syntax.

  Example 4 (Correct usage of SENT operator):

```
(SENT,
   (OR, "term1", "term2", "term3"),
   (OR, "term4", "term5", "term6")

)
```

A DATA-DRIVEN APPROACH TO LINGUISTIC RULE-BASED MODELING

Rule-based model development can sometimes be a painstakingly long and exhaustive process. Depending on your ability to quickly discern patterns and how frequently they occur in the data, rules you develop manually might not effectively grab the majority of true positives and/or efficiently handle false positives. Any additional help in developing these rules can accelerate your rule development process. In this section, we will describe an innovative approach that assists in deriving the rules for categorization taxonomy.

Let us consider a use case where we are required to evaluate medical claims notes and assess which claims belong to the high risk category and which are at low risk. To demonstrate this approach, we created our own examples of sample claims notes where “smoking” is the morbid condition we are looking for in the claim adjustor notes. Our objective is to categorize a claim as high risk if we find at least one instance where it is mentioned in the notes by developing category rules that can detect true positive instances and exclude false positives. Now, in a realistic situation we should have some historical claims that we have manually evaluated as high risk versus low risk for such morbid conditions. Let us make an assumption that we have a “gold-standard”, hand-classified historical data set with the claims notes as well as an indicator telling us if those claims are high risk or low risk claims. Table 1 shows three examples highlighting portions of the text where smoking-related information is found in the claims notes.

Sample Text	Summary
Example 1: …… claimant has informed that he is **not a smoker** but drinks alcohol occasionally ……he **smoked** for 10 years and then quit after he was diagnosed with ……	1 instance of True Positive, 1 instance of False Positive
Example 2: ……. ……. …… alcohol: no, **smokes: yes** ……	1 instance of True Positive

Sample Text	Summary
Example 3: ……he used to consume **2 packs per day** on an average which was below the usual for a regular ……	1 instance of True Positive

Table 1: Sample Claims Notes with Instances of the Smoking Concept

Portions of text highlighted in blue indicate an instance of false positive context, while text highlighted in red indicates a true positive context. Our objective for this exercise is to capture all claims with at least one true positive context occurring anywhere in the claims notes.

A DATA-DRIVEN APPROACH METHODOLOGY

First, we will capture potential candidates for false positive cases by casting a wide net and catching the contexts for smoking-related terms wherever they occur in close proximity to negation terms. Using SAS Contextual Analysis, we can write a predicate rule with two extraction parameters, "neg" and "key", to achieve this first step.

Example: `PREDICATE_RULE:(neg,key):(DIST_10,"_neg{NEGATIVE}","_key{SMOKING}")`

NEGATIVE – represents a Classifier concept for set of commonly occurring negation terms

SMOKING – represents a Classifier concept for SMOKING concept terms

neg – represents the parameter which captures the match returned by NEGATIVE concept from the text

key – represents the parameter which captures the match returned by SMOKING concept from the text

Here are some examples of terms that represent these concepts:

- Smoking – smoke, cigar, Chantix, tobacco, packs per day, nicotine etc.
- Negative – doesn't, didn't, denied, don't, isn't, ruled out, negative, no, none, non, false etc.

Using this predicate rule, we can extract the contexts from sample notes where a smoking concept keyword is within 10 tokens' distance from a negative term. Once we extract the information from the sample notes, we can manually review the contexts to identify actual false positives and true positives. We might perform proximity analysis separately for hand-classified high risk sample claims notes simply based on any true positive contexts we might have found while analyzing the extracted contexts. By applying SAS Contextual Analysis scoring code, we can generate results with the "neg "and "key" parameters extracted with the help of the above PREDICATE_RULE. Using a SAS scan function, search the contexts for identifying the relative positions of the parameters (neg and key in this case). Table 2 shows some examples of how contexts are extracted from the sample claims notes. Proximity/distance between the keywords and relative direction are identified, and then separated as High risk and Low risk items. Occasionally, we may find several overlapping matches identified or extracted by a single PREDICATE_RULE if there are multiple keyword and negation term matches found in the document. In that case, we can consider the longest returned match for our analysis to get a better understanding of the overall context. When the relative positions of the extracted "neg" and "key" parameters are identified, we can record the proximity and direction as per these guidelines:

- **Proximity** – Distance between matches for "neg" and "key" parameters
- **Direction** – Position of negative term (neg) with respect to the concept keyword (key)

Note: + if neg occurs after key, - if neg occurs before key.

Classification (Smoking Concept)	Examples
Low Risk	……not a smoker ……[Distance: 2, Direction: -] ……denies smoking ……[Distance: 1, Direction: -]

Classification (Smoking Concept)	Examples
	……smoker : no ……. [Distance: 2, Direction: +] ……is not into smoking……. [Distance: 2, Direction: -] ……is a non smoker …….. [Distance: 1, Direction: -] ……smokes - false ……. [Distance: 2, Direction: +] ……no tobacco …….. [Distance: 1, Direction: -]
High Risk	…… no alcohol but is a smoker….. [Distance: 5, Direction: -] …… alcohol: no, smokes: yes … [Distance: 2, Direction: -] ……didn't reveal he was a smoker…[Distance 5, Direction: -] …married no children clmt smokes…[Distance 3, Direction: -]

Table 2: Examples of False Positives and True Positives for the Smoking Concept

DISTRIBUTION ANALYSIS

As we derive the direction and distance/proximity values from the extracted information, we can study the distribution of the negation terms extracted as well as the average metrics for the proximity with the direction indicator applied. Display 1 shows a distribution analysis of negation terms with the summary statistics of proximity values on the sample set of claims notes. We can see that certain negation terms have high frequency over others, and they constitute a major portion of the cases with matches in the notes.

Analysis Variable : Proximity						
neg	N Obs	Mean	Std Dev	Minimum	Maximum	N
- no	1	-5.0000000	.	-5.0000000	-5.0000000	1
-denies	2	-2.0000000	1.4142136	-3.0000000	-1.0000000	2
-no	1	-1.0000000	.	-1.0000000	-1.0000000	1
denied	5	-4.0000000	4.1231056	-9.0000000	-1.0000000	5
denies	18	-2.7222222	2.9466942	-10.0000000	-1.0000000	18
did not	3	-4.3333333	3.2145503	-8.0000000	-2.0000000	3
do not	4	-4.2500000	4.2720019	-10.0000000	-1.0000000	4
doesn't	1	-1.0000000	.	-1.0000000	-1.0000000	1
doesnt	1	-1.0000000	.	-1.0000000	-1.0000000	1
don't	2	-1.0000000	0	-1.0000000	-1.0000000	2
family history	1	-3.0000000	.	-3.0000000	-3.0000000	1
neg	1	-2.0000000	.	-2.0000000	-2.0000000	1
never	2	-6.0000000	5.6568542	-10.0000000	-2.0000000	2
no	45	-2.5555556	2.5810107	-10.0000000	1.0000000	45
non	27	-1.0000000	0	-1.0000000	-1.0000000	27
none	1	-5.0000000	.	-5.0000000	-5.0000000	1
normal	3	-2.0000000	3.4641016	-4.0000000	2.0000000	3
not	30	-3.0000000	3.2589400	-10.0000000	3.0000000	30
risk for	1	-3.0000000	.	-3.0000000	-3.0000000	1
slightly	2	-2.5000000	2.1213203	-4.0000000	-1.0000000	2
unrelated	3	-6.0000000	0	-6.0000000	-6.0000000	3

Display 1: Distribution of Frequency of Negation Terms

Along with the individual negation terms' frequency, we can also identify the most frequent combinations for negation and keywords for the smoking concept from the extracted information. Display 2 shows us the list of keyword and negation term combinations found in the sample notes along with the descriptive statistics of the proximity analysis variable. Again, looking at this distribution and how the mean proximity

values show up, we can group certain negation terms and keywords in separate sets. We can write our own category rule as shown below, which qualifies as a data-driven rule based on our analysis and what story our data tells us.

Example: `(OR,(ORDDIST_5,"[SMOK_NEG_PRE]","[SMOK_1]"))`

SMOK_NEG_PRE – Negative terms that predominantly occur before the keyword concept terms.

SMOK_1 – Keyword concept terms that occur most frequently with the terms grouped under the SMOK_NEG_PRE classifier concept definition.

Analysis Variable : Proximity							
key	neg	N Obs	Mean	Std Dev	Minimum	Maximum	N
smoke	-no	1	-1.0000000	.	-1.0000000	-1.0000000	1
	denies	1	-1.0000000	.	-1.0000000	-1.0000000	1
	do not	2	-1.0000000	0	-1.0000000	-1.0000000	2
	doesn't	1	-1.0000000	.	-1.0000000	-1.0000000	1
	doesnt	1	-1.0000000	.	-1.0000000	-1.0000000	1
	don't	2	-1.0000000	0	-1.0000000	-1.0000000	2
	no	6	-1.0000000	0	-1.0000000	-1.0000000	6
smoker	denies	5	-1.0000000	0	-1.0000000	-1.0000000	5
	no	7	-1.0000000	0	-1.0000000	-1.0000000	7
	non	22	-1.0000000	0	-1.0000000	-1.0000000	22
smokes	-denies	1	-3.0000000	.	-3.0000000	-3.0000000	1
	denies	3	-6.3333333	2.8867513	-8.0000000	-3.0000000	3
	neg	1	-2.0000000	.	-2.0000000	-2.0000000	1
	never	1	-10.0000000	.	-10.0000000	-10.0000000	1
	no	6	-5.0000000	1.5491933	-7.0000000	-3.0000000	6
smoking	-denies	1	-1.0000000	.	-1.0000000	-1.0000000	1
	denied	3	-1.0000000	0	-1.0000000	-1.0000000	3
	denies	7	-2.2857143	3.4016803	-10.0000000	-1.0000000	7
	did not	1	-2.0000000	.	-2.0000000	-2.0000000	1
	do not	1	-10.0000000	.	-10.0000000	-10.0000000	1
	no	13	-2.6923077	3.3262746	-10.0000000	-1.0000000	13
	non	5	-1.0000000	0	-1.0000000	-1.0000000	5
	none	1	-5.0000000	.	-5.0000000	-5.0000000	1
	not	2	-6.0000000	5.6568542	-10.0000000	-2.0000000	2
	unrelated	3	-6.0000000	0	-6.0000000	-6.0000000	3

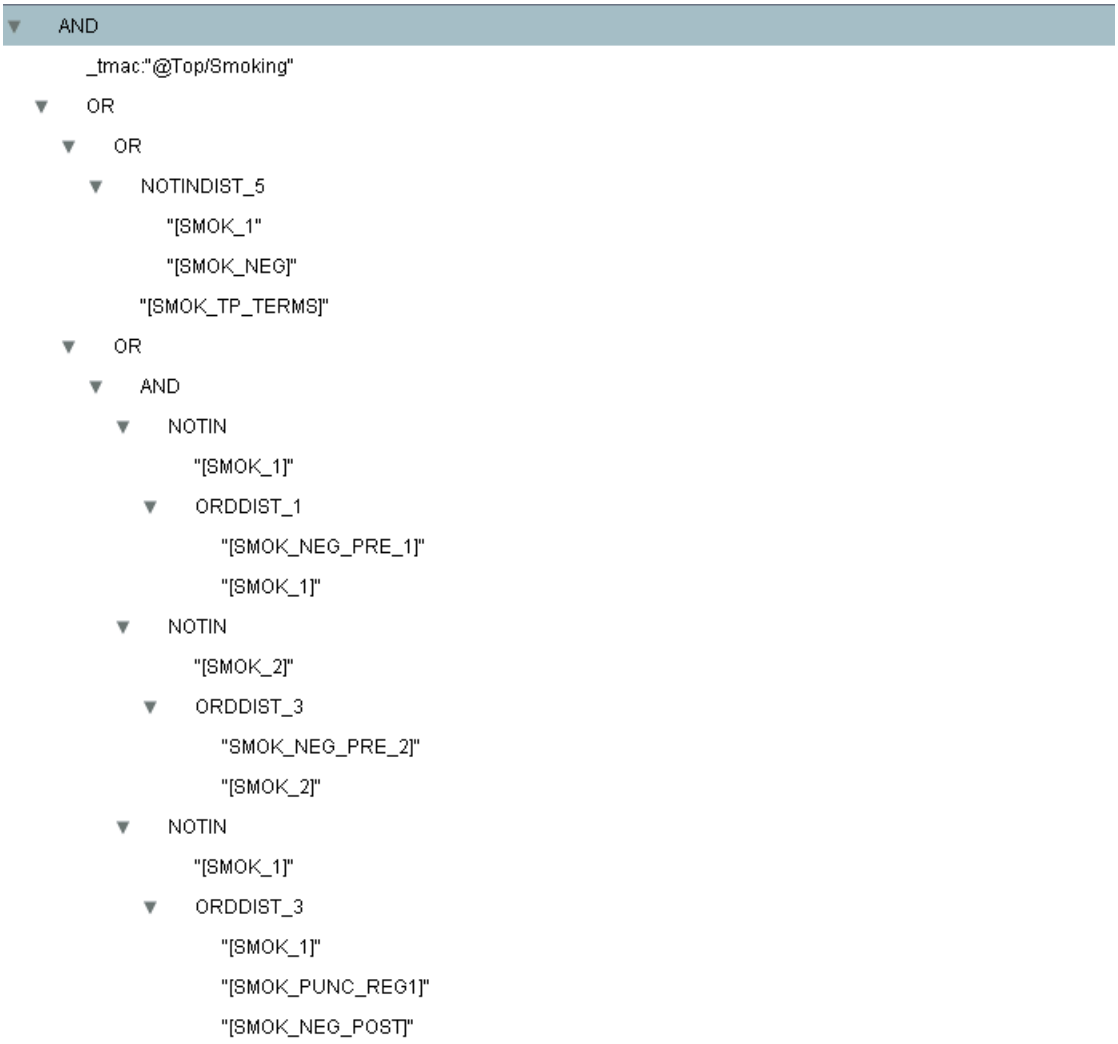
Display 2: Distribution of Frequency of Negation Terms and Smoking Concept Terms

RESULTS

Based on the above approach, we can develop and build several rules and then combine them under a single category. We create this category for identifying claims notes that have at least one instance of a true positive context for the smoking concept anywhere in the entire document. Display 3 is an example of how we can build a category rule based on the analysis we perform with negation terms and concept keywords on our data for a sample condition (smoking, in this case). In this example rule, we can see several rules stitched together using appropriate Boolean operators. This rule helps in catching claims notes with at least one true positive instance of the smoking concept mentioned anywhere in the document. This entire rule looks confusing and complicated, but as we develop these rules, make a habit of briefly noting down how we formulated these rules based on our analysis. In this way, maintainability and interpretability of these rules over a long period of time with or without our presence will be easier within our organization.

SMOK_1, SMOK_2, SMOK_NEG, SMOKE_NEG_PRE_1, SMOKE_NEG_PRE_2, SMOK_NEG_POST, SMOKE_TP_TERMS are all intermediate, or helper, CLASSIFIER concept definitions developed based on the distribution analysis of negation and keyword terms along with the proximity metric as described in

the previous distribution analysis section. These helper concept definitions come in handy in building the individual rules that, when joined together, make one bulky category rule.



Display 3: An example showing a Category Rule for the Smoking Concept in SAS Contextual Analysis

NOTE: The default indention within the tree view (shown above) of the rule editor in SAS Contextual Analysis is really helpful for maintenance purposes when someone other than the developer of this rule needs to modify some portions of the rule. Imagine trying to initially comprehend the rule's logic if displayed in text view!

```
(AND,_tmac:"@Top/Smoking",(OR,(OR,(NOTINDIST_5,"[SMOK_1","[SMOK_NEG]"),"[SMOK
_TP_TERMS]"),(OR,(AND,(NOTIN,"[SMOK_1]",(ORDDIST_1,"[SMOK_NEG_PRE_1]","[SMOK_
1]")),(NOTIN,"[SMOK_2]",(ORDDIST_3,"SMOK_NEG_PRE_2]","[SMOK_2]")),(NOTIN,"[SM
OK_1]",(ORDDIST_3,"[SMOK_1]","[SMOK_PUNC_REG1]","[SMOK_NEG_POST]"))))))
```

In the following paragraph, we show you an example of how you can interpret and explain a complicated category rule such as this.

Categorize a claims-notes document as a "True Positive" for the 'Smoking' concept if,

a) A 'smoking' concept term is found anywhere in the notes

AND

b) At least one of the following two conditions listed in 1 and 2 is satisfied:

1. A ‘smoking’ related concept term from SMOK_1 classifier concept definition is found such that it is not within a distance of 5 words from a ‘negative’ term (SMOK_NEG)

   (Or)

   Any term strongly suggesting that the claimant is a smoker (SMOKING_TP_TERMS)

   Examples: long term smoker, significant smoker, Chantix, emphysema, varenicline, packs a day, packs per week, pks/day, and so on.

2. A ‘smoking’ related concept term from SMOK_1 is found such that **it is not** within an ordered distance of 1 word from a negative term found in SMOK_NEG_PRE_1 concept definition.

   Examples: non smoker, doesn’t smoke, no smoking, no tobacco, and so on.

   (And)

   A ‘smoking’ related concept term from SMOK_2 is found such that **it is not** within an ordered distance of 3 words from any of the negative terms found in SMOK_NEG_PRE_2 concept definition.

   Examples: not a smoker, not into smoking, no habit of smoking, denies use of tobacco, negative for smoking, and so on.

   (And)

   A ‘smoking’ related concept term from SMOK_2 is found such that **it is not** followed by a punctuation mark (: - \ /) and any of negative terms from SMOK_NEG_POST in that particular order.

   Examples: smoker – no, smokes ? No, smoker : false, smoking – neg, smoke – denied, etc.

Note: Regardless of any number of false positive contexts identified in a claims-notes document, this rule will override them and tag the entire notes as “true positive”.

Summary of Data-driven Approach

- Using data-driven analysis helps you to develop linguistic rule-based categorization or contextual extraction models based on the patterns found in the data.
- Contextual extraction rules help you to understand the patterns in the data.
- Using powerful factual extraction rules such as PREDICATE RULE, you can not only extract the matching parameters (concept keyword and negative term) but also extract the concordance (a certain number of words or characters before and after the matching context).
- Using both categorization and contextual extraction features simultaneously has its own benefits. However, exercise caution when using REGEX (regular expression) rules since you might be matching several thousand terms in the documents with a small mistake in the rule.
- Performing the distribution analysis of negation and keyword terms over proximity measure separately for hand-classified “gold-standard” High risk Versus Low risk documents will help develop precise rules.
- The categorization rules framework in SAS Contextual Analysis provides powerful operators to incorporate negation scenarios to exclude false positive contexts very easily.

GENERAL BEST PRACTICES FOR RULE-BASED SENTIMENT ANALYSIS MODELING

As when creating categorization or contextual extraction taxonomies, for sentiment analysis we also encourage you to create intermediate entities as “helpers”, or reusable building blocks, that you can reference in other rules. This allows you to create definitions to capture the essence of a generic feature such as “Price”, and then just combine that definition with other contextual indicators, such as a specific product or brand, in order to compare sentiment toward Price in a fair fashion across products and brands. In addition, by creating this symbolic reference to the single Price definition, it is simple to later add more keywords and rules in that single place to extend the definition and have those updates automatically propagate to the other rules that reference it. This makes for very efficient, interpretable, and maintainable models.

Keep in mind that some words are not universally positive or negative – the tone depends on the context. For example, the word “cheap” might be positive in the context of price, but negative in the context of product quality. In such cases, you can add these ambiguous keywords to the feature level positive/negative tonal keyword lists, rather than the global lists.

SAS® Sentiment Analysis Studio supports testing folders for gold-standard data that is pre-classified as positive, negative, and neutral. Whenever possible, we encourage you to upload gold-standard data in this format to simplify the process of testing and enable you to readily see what type I and type II errors your rules are generating.

It can be tempting to try to capture every single nuance of language correctly in your sentiment model right away. For example, you might want to properly detect the following:

- Sarcasm (for example, “Great. The train is late AGAIN!”)
- Conditional statements (for example, “If the service had been better we would have come back.”)
- Model operators (for example, “I might consider upgrading to the new model”, “The food could’ve been better.”)
- Comparatives and Amplifiers (for example, “it was way worse than before.”)
- Negation (for example, “I was not happy with the experience.”, “The crew could not have been nicer!”)

While these more subtle language patterns can sometimes be modeled with a bit more effort, we recommend that you handle the more straightforward expressions of sentiment first, and then tackle these more complex cases in subsequent phases of tuning. (Negation is the exception; this pattern can be captured and assigned properly in most cases with a few additional pieces of logic, and is typically part of an initial sentiment model.)

In situations where the above types of tricky cases only represent a small percentage of your overall corpus and the rules added to catch them carry the risk of losing good matches or causing false positives in the rest of your data, it may be worth ignoring these edge cases in your model altogether and chalking them up to misclassification error. Remember – even humans only agree on how to interpret text 80% of the time; do not expect your models to be perfect!

CONCLUSION

The statistical approach is fast and easy to maintain. It easily scales up to increasing data volumes or changing data patterns for exploratory and predictive modeling needs. The linguistic approach is a time-consuming and sophisticated process, but can yield incrementally more accurate results if the assets are built using domain knowledge and subject matter expertise and the models are well-maintained. In our experience, the statistical approach and the linguistic rule-based approach each have their own benefits and drawbacks. Depending on the use case or application purpose, one might take precedence over the other. Generally, one approach outperforms the other depending on the nature of data and objective/goal of the analysis. In our experience, the statistical approach works best for internal data such as surveys, call center logs, manufacturer warranty claims, technician notes, and so on, where exploration of the data for generating themes or predictive modeling is the priority. Linguistic rule-based modeling is best suited

for applications requiring classification of documents into pre-determined categories/sub-categories and contextually extracting information from dense documents such as medical claims notes, legal documents, academic publications, and so on. In those cases, it is important to contextually verify the occurrence or absence of desired concepts to disambiguate between false positives versus true positives. Text Analytics is as much an art as it is a science, and each individual use case offers its own unique opportunity for you to apply creativity, data mining techniques, and domain knowledge to best solve the problem at hand.

REFERENCES

Aizawa, A. 2003. "An Information-Theoretic Perspective of tf-idf Measures." *Information Processing & Management*. 39 (1): 45-65.

Booth, A. D. 1967. "A 'Law' of Occurrences for Words of Low Frequency." *Information and Control.* 10 (4): 386-393.

Chakraborty, G., M. Pagolu, and S. Garla. 2013. *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS*. Cary, NC: SAS Institute Inc.

Manning, C. D., and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.

Text Analytics Using SAS Text Miner. SAS Institute course notes. Course information: https://support.sas.com/edu/schedules.html?ctry=us&id=1224

Zipf, G.K. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA: Addison-Wesley.

ACKNOWLEDGMENTS

We would like to thank Saratendu Sethi, Praveen Lakkaraju, Teresa Jade, and Michael Wallis from SAS for their support and guidance over the years.

RECOMMENDED READING

- *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Murali Pagolu
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Murali.Pagolu@sas.com
http://www.sas.com

Christina Engelhardt
124 Preserve Way
Mooresville, NC 28117
SAS Institute Inc.
Christina.Engelhardt@sas.com
http://www.sas.com

Cheyanne Baird
SAS Institute Inc.
Cheyanne.Baird@sas.com
http://www.sas.com