Paper 100-2013

Relate, Retain, and Remodel: Creating and Using Context-Sensitive Linguistic Features in Text Mining Models

Russell Albright, Janardhana Punuru, and Lane Surratt, SAS Institute Inc.

ABSTRACT

Text mining models routinely represent each document with a vector of weighted term frequencies. This bag-of-words approach has many strengths, one of which is representing the document in a compact form that can be used by standard data mining tools. However, this approach loses most of the contextual information that is conveyed in the relationship of terms from the original document. This paper first teaches you how to write pattern-matching rules in SAS® Enterprise Content Categorization and then shows you how to apply these patterns as a parsing step in SAS® Text Miner. It also provides examples that improve on both supervised and unsupervised text mining models.

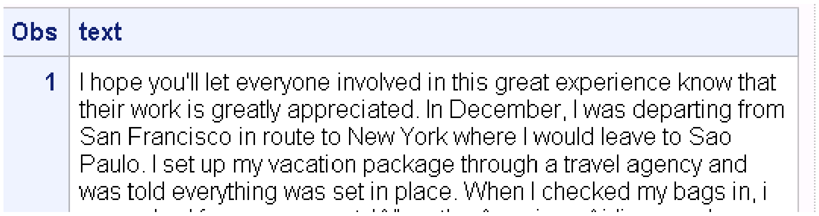
INTRODUCTION

Building good exploratory and predictive text models is a challenge. As a SAS Text Miner user, you want to discover relevant and meaningful clusters and topics, and you hope that your predictive models are effective at their classification and predictive tasks. So when your models fail to meet expectations, your first step is to attempt to improve the performance by using the many controls and options available to tune your model. However, sometimes even tuning is not successful, and you must dig deeper by examining the representation of your documents themselves to address concerns about underperforming models.

This paper uses the term *features* to refer to the terms, multiword terms, term-tag pairs, and other indicators in the vector representation of your documents. The primary way that you control which features are used to represent your collection is by using the options on the Text Parsing node in SAS Text Miner. You can use options such as part-of-speech tags, stop lists, entities, synonyms, and stemming to either expand or restrict the set of terms used. However, none of these settings can compare with the amount of power and precision over feature creation that SAS Enterprise Content Categorization provides in conjunction with SAS Text Miner. By using SAS Enterprise Content Categorization, you can create rules that define additional features for your collection. These rules then enable the parsing component of SAS Text Miner to extract these features and include them in the representation of your collection. The goal of this paper is to help you realize the potential of this node property and to enable you to use it effectively.

Because creating and using custom features is so important, an add-on product to SAS Text Miner is available, SAS® Concept Creation for Text Miner. This product enables you to create the custom entities without requiring a license for SAS Enterprise Content Categorization. Either product produces the results shown here. For simplicity, this paper refers to SAS Enterprise Content Categorization, but all methods apply to either product.

This paper demonstrates and motivates the analysis by examining parts of a collection of approximately 6,000 complaints, recommendations, and commendations that were culled from the web about airlines and their customer service. The length of most complaints varies from a paragraph or two to a page. Figure 1 shows a sample complaint.



Obs	text
1	I hope you'll let everyone involved in this great experience know that their work is greatly appreciated. In December, I was departing from San Francisco in route to New York where I would leave to Sao Paulo. I set up my vacation package through a travel agency and was told everything was set in place. When I checked my bags in, i

Figure 1. Sample Observation in the Airline Complaint, Recommendations, and Commendations Data Set

The following section shows the importance of your document representation with a short example. Then the paper describes the entity property of SAS Text Miner so that you understand its potential in defining features. Next, it briefly describes how to define these features Enterprise Content Categorization. Finally, it turns to Text Miner to show how you can apply these new features to improve your models.

DOCUMENT REPRESENTATIONS FOR TEXT MINING

SAS Text Miner uses the bag-of-words vector representation of documents. The parsed terms become variables that are implicitly represented. (Text Miner does not usually form the explicit document-by-term representation because of the sparsity in the table.) After the document collection is represented as a collection of vectors, the text mining algorithms can learn patterns across the entire collection. These patterns are the essential component to useful modeling; without them, text mining cannot succeed. By choosing and creating distinctive linguistic features that enhance the aspects of the collection you are examining, you can improve the meaningful co-occurrence patterns between documents and you can find patterns based on those co-occurrences.

For example, consider the sentence shown in Table 1. This sentence is encoded with three different sets of features. Representation 1 shows only the term strings themselves. In Representation 2, a stop list, stemming, and tagging have been applied in the feature creation. Finally, Representation 3 demonstrates the theme of this paper. Some relational information is encoded by creating a “baggage_fee_ent” term, a baggage fee entity that is created because the sentence contains indicators that its author was discussing how much money he or she spent specifically to check bags.

Sentence	I was charged $98 for two bags.						
Representation 1	I	was	Charged	$98	for	two	bags
Representation 2	charge:Verb	$98:CURRENCY	two:Num	Bag:Noun			
Representation 3	charge:Verb	baggage_fee_ent	two:Num				

Table 1. Three Representations of the Same Sentence

Representation 3 is distinguished from the others because of how the term “baggage_fee_ent” is automatically created. First, you write a couple of rules in SAS Enterprise Content Categorization to generalize the $98 to a general monetary concept and then to relate the monetary concept to the “bags” term. Then when you apply the rules in SAS Text Miner, the “baggage_fee_ent” term is created. Because the string “$98” is unlikely to exactly match other monetary values (and even if it does, it might relate to some expense other than baggage fees), the generalization to a monetary value and its relationship to bags makes for a powerful feature that is likely to introduce effective co-occurrence patterns.

Exactly how and to what extent you customize your feature selection is up to you and the goals of your analysis. In the preceding example, do you want two documents to be more similar because they share exactly the same monetary value of “98,” or do you want documents to be similar merely because they both mention *any* monetary value? In the latter case, a generalization of your feature is in order to map it to a canonical form. When one or more of the terms “bag,” “bags,” or “baggage” occur in your collection, should these always increase the strength of relationship between any two documents that contain them? Or are you hoping for more refinement—an ability to distinguish between checking baggage and losing baggage, for example? If so, maybe you should create a feature that encodes a relationship when two or more terms are near each other in the same document.

WHAT ARE SAS TEXT MINER ENTITIES?

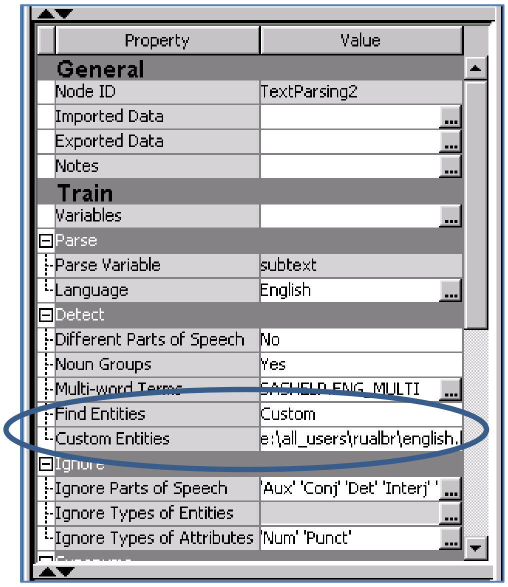
In SAS Text Miner, entities are terms (often multiword) that exist in the document and represent some predefined concept or class. The concept usually represents real-world elements such as a company, a person, or a date. Entities become terms in the terms table and have a role that corresponds to the class that they belong to.

To extract an entity, a classifier must make a prediction about every term in the collection. For each term, the classifier asks the question, “Is this the beginning of an entity of type X or not?” After the classifier finds the beginning it makes another prediction about the end of the entity. Sometimes, the classification is made in a deterministic way, based on the properties in the text (such as whether the first letter is capitalized or whether the term is in the predefined list of company names). But other times, the entity rule might be a pattern match on a specific pattern such as ***-***-****, which implies a phone number. Still other rules can be quite complex and might depend on part-of-speech tags and surrounding terms. In these cases, there might be much uncertainty about which class, if any, a term might belong to.

Figure 2 shows the entity properties for the Text Parsing node. You can select the following entity settings:

- None: no entities will be detected.
- Standard: uses the default set of entity rules, which are described in the next subsection.
- Custom: enables you to specify the location of your own set of rules that you created in SAS Enterprise Content Categorization.
- All: text parsing will extract both your custom entity types and standard entity types.

These entities are applied very early in the parsing process. As a result, entities are discovered as features before synonyms are applied and stop or start lists are enforced.



Property	Value
General	
Node ID	TextParsing2
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
⊟Parse	
Parse Variable	subtext
Language	English ...
⊟Detect	
Different Parts of Speech	No
Noun Groups	Yes
Multi-word Terms	SASHELP.ENG_MULTI ...
Find Entities	Custom
Custom Entities	e:\all_users\rualbr\english.l
⊟Ignore	
Ignore Parts of Speech	'Aux' 'Conj' 'Det' 'Interj' ' ...
Ignore Types of Entities	...
Ignore Types of Attributes	'Num' 'Punct' ...

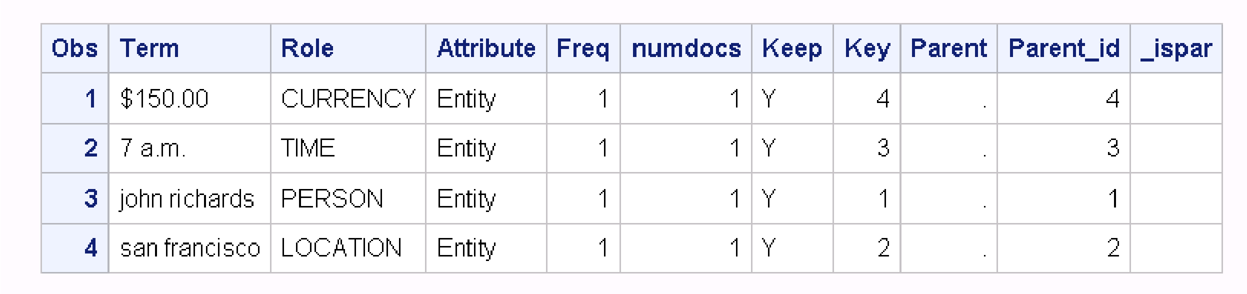
Figure 2. Entity Properties in the Text Parsing Node

STANDARD ENTITIES IN SASTEXT MINER

Standard entities are the entities that can be detected by default in SAS Text Miner. All entity types except PROP_MISC represent a common concept that might be apply in almost any domain, and the PROP_MISC entity type represents proper nouns in general that do not match any of the other types. The complete list of the default standard entities for English follows:

- ADDRESS
- DATE
- MEASURE
- PERSON
- SSN
- TITLE
- COMPANY
- INTERNET
- ORGANIZATION
- PHONE
- TIME
- VEHICLE
- CURRENCY
- LOCATION
- PERCENT
- PROP_MISC
- TIME_PERIOD

SAS Text Miner finds these standard entities by using a preconfigured binary file that is created by Enterprise Content Categorization (which is shipped with Text Miner). The file is accessed by the parsing procedure, TGPARSE. Table 2 shows the output terms table, which contains some common, standard entities.



Obs	Term	Role	Attribute	Freq	numdocs	Keep	Key	Parent	Parent_id	_ispar
1	$150.00	CURRENCY	Entity	1	1	Y	4	.	4	
2	7 a.m.	TIME	Entity	1	1	Y	3	.	3	
3	john richards	PERSON	Entity	1	1	Y	1	.	1	
4	san francisco	LOCATION	Entity	1	1	Y	2	.	2	

Table 2. Sample Entity Output from PROC TGPARSE

The primary purpose of the standard entities is simply to identify these different classes of items. They provide a mechanism for you to further pursue the entities by looking at documents that contain them. Then, based on your own goals, you can write your own SAS code to relate the entities. For example, you might want to explore how different company names and people’s names relate to each other.

Standard entities can also be useful in the other modeling nodes. They can serve to disambiguate two cases of the same term string used in different ways. (This is particularly important because Text Miner lowercases all terms. So distinguishing terms by case is not maintained unless the roles are different.) You can also focus your analysis by choosing to include only specific entity types, or you can manually treat all entity types to be a synonym of the same term. The next sections describe the latter technique in more detail.

CUSTOM ENTITIES IN SAS TEXT MINER

The custom entity feature of SAS Text Miner enables you to input a file that you create in SAS Enterprise Content Categorization to control the features that you identify in the Text Parsing node. You can define custom entities to discover items that belong to some new class for your domain. In the case of the airline data, you might be interested in creating an airline entity or an airport entity. After you decide which linguistic properties define these things, you can then identify them, use them in reports, and disambiguate cases of one type versus another.

But you don’t want to stop there in your creative use of custom entities. Custom entities do not need to be interpreted in the same way as standard entities; they do not necessarily have to be limited to representing only real-world elements in your text. Instead, incorporating custom entities enables you to become creative in using specific, helpful information that you extract from the text to use as features. Ultimately, you will be able to identify elements that represent relationships between multiple terms, detect co-references, create general pattern matches of specific linguistic elements in the text, and build increasingly complex rules for extraction that are based on earlier rules that you have already written. The hope is that you will be able to create useful features for model building.

The following subsections emphasize a pair of complementary perspectives on custom entities. The first perspective is an interpretation as a type of programmatic synonym list that generalizes a variety of distinct terms into a canonical form. That form will match across documents and, when encoded into a feature, indicates to the model that all documents that contain this entity have a similarity, no matter what diverse strings represent them. The second perspective capitalizes on relational elements between terms to create a new feature that is a refinement. Together, the two perspectives (entities as synonyms and entities as relational features) are even more powerful than either one alone.

Custom Entities as Synonyms

Entities can be interpreted as providing a complex but powerful approach to synonym discovery and assignment when all entities of a given type are mapped to some canonical term. When you use custom entities, you control what types of text strings you would like to treat as synonyms. And, unlike the synonym functionality in SAS Text Miner that is based on predefined lists of terms, SAS Enterprise Content Categorization rules, which are applied in Text Miner as custom entities, enable you to generalize your synonyms to match much broader patterns that would be impossible for you to specify explicitly.

For example, return to the task of creating an entity out of the airport codes, such as SFO for San Francisco International Airport. Suppose you do not have a list of all airport entities, but you merely want to use the pattern of three uppercase letters. The following syntax in Enterprise Content Categorization enables you to extract all terms of this type:

```
    Top: AIRPORT_CODE:   REGEX:[A-Z][A-Z][A-Z]
```

The preceding syntax might generalize too much and pick up random instances of uppercase terms (such as “AND”) in your text, but there are ways to refine your rule that are described in the next subsection.

Suppose you also want to make your documents similar as long as they mention *any* airport code. Because SAS Text Miner uses the term-role pair to distinguish terms, finding two airport codes in two different documents, such as D1 and D2, shown in Table 3, does not create any relationships between the documents.

Document	Term	Role	Parent	ParentRole
D1	SFO	AIRPORT_CODE		
D2	LGA	AIRPORT_CODE		

Table 3. Airport Codes without Synonyms

However, you can create a relationship if you represent the term with a parent term that is formed from the Role. In Table 4 the two terms found are represented by the same parent term, AIRPORT_CODE_ENT, because they share the same entity type. In the Text Filter node, you can manually assign all terms that have the AIRPORT_CODE role to a single representative term. This can be tedious, however. Appendix A provides SAS code that enables you to do this mapping in a SAS Code node that follows the Text Parsing node. At the beginning of the code, you specify a macro variable for which roles you want to be mapped to a canonical form as a parent. The macro adjusts the terms table and the term-document frequency table to account for the parents that are introduced. A third alternative is to use the Text Topic node to create a user-defined topic.

Document	Term	Role	Parent	ParentRole
D1	SFO	AIRPORT_CODE	AIRPORT_CODE_ENT	AIRPORT_CODE
D2	LGA	AIRPORT_CODE	AIRPORT_CODE_ENT	AIRPORT_CODE

Table 4. Airport Codes Treated as Synonyms

Custom Entities as Relational Features

As mentioned before, entities typically represent some real-world concept such as a company name or a person’s name. But by using the custom entity property, you can assign a very flexible meaning to entities so that you can capture powerful features. For example, the airline industry, like any industry, uses many terms that have special meanings when they occur in a given context. For example, being “bumped” often means that the airline overbooked the flight and you are being assigned to the next flight. It is clearly a different meaning from being “bumped” by a passenger while you are stowing your bag. You can use context and relationship between terms to capture this information so that the two meanings can be distinguished. For example, the following rule could help you capture the cases in which the text means being “bumped” from a flight:

```
BUMPED_FLIGHT: CONCEPT_RULE:(ORDDIST_10, "_c{bumped}", (OR, "flight",
"airplane", "plane")).
```

The longer your documents, the more beneficial it is to create an entity that encodes terms appearing near one another. Imagine a long complaint written to an airline that covers everything from the service, to the food, to the airfare. Suppose that early in the document the complaint is about “narrow-minded employees” and that later the complaint is that the “seat was dirty.” The bag-of-words approach to representing your document correlates the term “narrow” with the term “seat” as much as it does with “minded.” The same holds for the term “dirty.” If there is a pattern of dirty seats in flights coming out of New York, you might miss it. A collection of rules that encode when the notion of dirty and seat are near each other might be necessary to uncover this pattern.

Although the previous examples are for very specific cases, there are many linguistic aspects that you can use in your analysis. Many of the SAS Enterprise Content Categorization rules that might be used for sentiment (SAS Institute Inc. 2010; Lange and Sethi 2011) can automatically be included in your model and can potentially drive your analysis. These include incorporating rules to capture when the meaning of a term has been reversed by the use of “not” or “didn’t,” for example.

When you use SAS Enterprise Content Categorization rules, you don't need to stop with associating actual terms in your document to create a new term. You can also refer to earlier rules to make increasingly powerful rules. For example, after you define an entity for lost baggage, you can relate that to an entity for flight delay.

SAS ENTERPRISE CONTENT CATEGORIZATION STUDIO FOR CONTEXTUAL EXTRACTION

This section describes details about building contextual extraction models in SAS® Enterprise Content Categorization Studio. You can then use these models in SAS Text Miner to control the features that are used in each document vector. SAS® Contextual Extraction enables you to define custom concepts (which are referred to as entities in SAS Text Miner) by using lexical and syntactic patterns, Boolean operators, contextual disambiguation operators, regular expressions, and so on. You can also use combinations of these.

SAS Enterprise Content Categorization, which enables you to define and extract valuable information from customer-specific unstructured data repositories, consists mainly of three components: SAS Content Categorization Studio, SAS Contextual Extraction, and SAS® Content Categorization Server. The package shown in the following section is SAS Contextual Extraction.

SAS Content Categorization Studio enables you to define models for document classification. Its goal is to categorize, based on the rules defined in the model, each document into one or more predefined classes. SAS Contextual Extraction Studio, on the other hand, enables you to define rules and patterns to extract useful information from inside the individual documents. Finally, SAS Content Categorization Server applies the categorization and contextual extraction models to incoming texts and returns the results for further analysis.

SAS ENTERPRISE CONTENT CATEGORIZATION STUDIO

SAS Enterprise Content Categorization Studio provides a free-form editor for defining a taxonomy of useful concepts and for defining patterns and rules for individual concepts. It also provides a testing environment that enables you to test against individual documents or against the entire collection of documents at one time. The entities that are mentioned previously in relation to SAS Text Miner are referred to as concepts in this product.

When you create a new project, Enterprise Content Categorization Studio opens with an empty project. You can then create or load previously saved high-level concepts. The tool supplies an editor that you use to define the rules and patterns. For example, the comments in the Airline corpus were reviewed, and the taxonomy shown in Figure 3 was developed. It defines many of the classes of items that you might think belong to documents that talk about the airline industry.

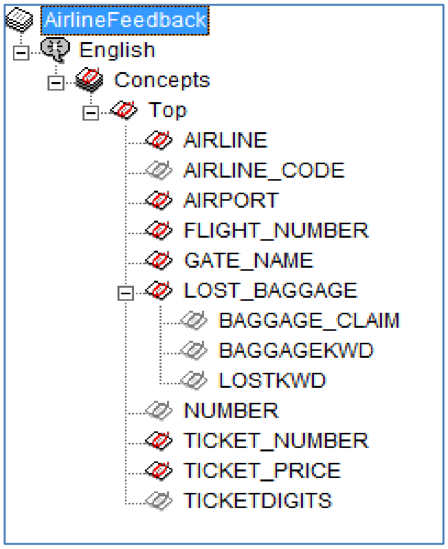
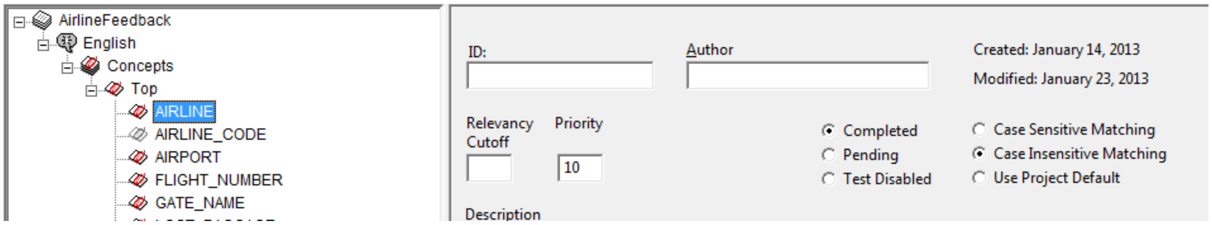


Figure 3. An Airline Taxonomy in SAS Contextual Extraction Studio

As shown in Figure 3, Top is a placeholder for the root node in the taxonomy. All the nodes under Top are custom concepts for Airline Taxonomy. In the taxonomy in Figure 3, the concepts with red icons are the concepts for which matching results can be seen from a testing window or from SAS Text Miner. All the concepts with icons that are shown in gray are unavailable. The unavailable concepts act as intermediate concepts that can be used as contextual requirements to build on for matching other concepts. Of course, available concepts can also be used as intermediate concepts, but matching results for unavailable concepts are not shown to the end user. Each concept can be made available or unavailable, assigned a priority, and made sensitive to case as shown in Figure 4. You can assign these settings separately to individual concepts or globally from the project menu.



```
AirlineFeedback
  English
    Concepts
      Top
        AIRLINE
        AIRLINE_CODE
        AIRPORT
        FLIGHT_NUMBER
        GATE_NAME
```

ID:

Author

Created: January 14, 2013

Modified: January 23, 2013

Relevancy Cutoff

Priority: 10

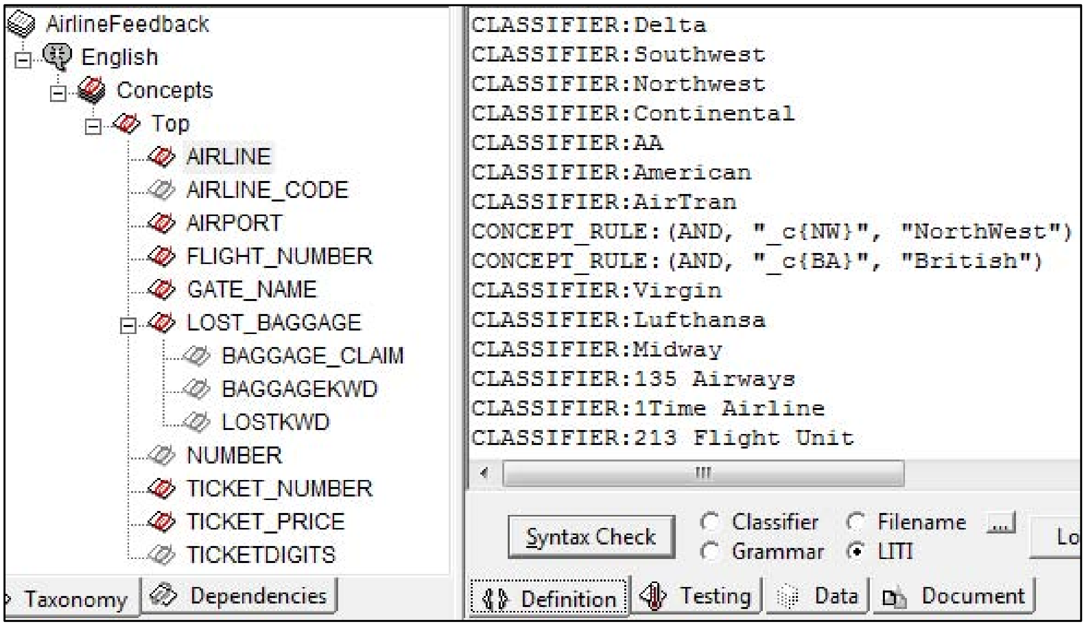
● Completed
○ Pending
○ Test Disabled

○ Case Sensitive Matching
● Case Insensitive Matching
○ Use Project Default

Description

Figure 4. Concept Settings Window

For each of the concepts that are defined in the taxonomy, you can define contextual extraction rules in the definition window as shown in Figure 5. The syntax for these rules is specific to SAS Contextual Extraction Studio. The following section provides a detailed description and syntax for these rules.



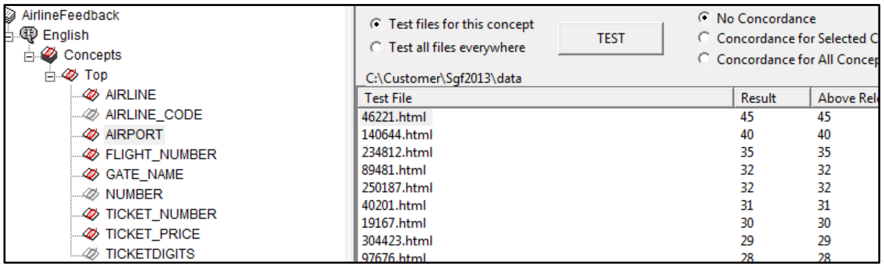
```
AirlineFeedback
  English
    Concepts
      Top
        AIRLINE
        AIRLINE_CODE
        AIRPORT
        FLIGHT_NUMBER
        GATE_NAME
        LOST_BAGGAGE
          BAGGAGE_CLAIM
          BAGGAGEKWD
          LOSTKWD
        NUMBER
        TICKET_NUMBER
        TICKET_PRICE
        TICKETDIGITS
```

```
CLASSIFIER:Delta
CLASSIFIER:Southwest
CLASSIFIER:Northwest
CLASSIFIER:Continental
CLASSIFIER:AA
CLASSIFIER:American
CLASSIFIER:AirTran
CONCEPT_RULE:(AND, "_c{NW}", "NorthWest")
CONCEPT_RULE:(AND, "_c{BA}", "British")
CLASSIFIER:Virgin
CLASSIFIER:Lufthansa
CLASSIFIER:Midway
CLASSIFIER:135 Airways
CLASSIFIER:1Time Airline
CLASSIFIER:213 Flight Unit
```

○ Classifier
○ Grammar
○ Filename
● LITI

Figure 5. Rule Definitions for the AIRLINE Concept

You compile the developed taxonomy and the associated rules for each of the concepts by selecting **Build>Compile Concepts**. A single binary file is created; it is optimized for matching a continuous stream of documents at run time. You can test and verify the prepared taxonomy against a corpus from SAS Contextual Extraction Studio by specifying the location of the corpus. You can test and verify SAS Contextual Extraction Studio against a training corpus from within SAS Contextual Extraction Studio by specifying the location of the corpus. This testing and verification enables you to make changes to the model for generalization. The testing window in Figure 6 shows the matching documents for the concept Airport in the corpus.



● Test files for this concept
○ Test all files everywhere

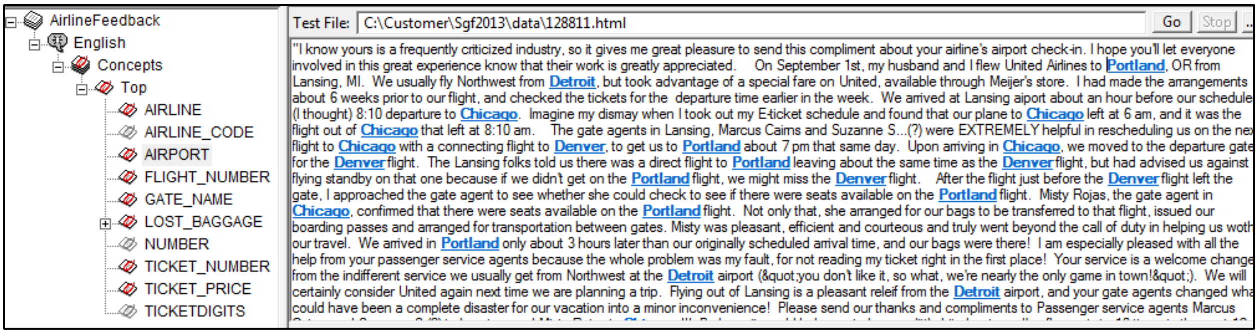
● No Concordance
○ Concordance for Selected C
○ Concordance for All Concep

C:\Customer\Sgf2013\data

Test File	Result	Above Rel
46221.html	45	45
140644.html	40	40
234812.html	35	35
89481.html	32	32
250187.html	32	32
40201.html	31	31
19167.html	30	30
304423.html	29	29
97676.html	28	28

Figure 6. Matching Documents Found in the Corpus

To verify the matches in a specific document for a particular concept or for all concepts, you can click on the specific document to see matches highlighted. As shown in Figure 7, all the matches for the concept Airport are highlighted in blue.



Test File: C:\Customer\Sgf2013\data\128811.html

"I know yours is a frequently criticized industry, so it gives me great pleasure to send this compliment about your airline's airport check-in. I hope you'll let everyone
involved in this great experience know that their work is greatly appreciated.   On September 1st, my husband and I flew United Airlines to **Portland**, OR from
Lansing, MI. We usually fly Northwest from **Detroit**, but took advantage of a special fare on United, available through Meijer's store. I had made the arrangements
about 6 weeks prior to our flight, and checked the tickets for the departure time earlier in the week. We arrived at Lansing aiport about an hour before our schedule
(I thought) 8:10 departure to **Chicago**. Imagine my dismay when I took out my E-ticket schedule and found that our plane to **Chicago** left at 6 am, and it was the
flight out of **Chicago** that left at 8:10 am.   The gate agents in Lansing, Marcus Cairns and Suzanne S...(?) were EXTREMELY helpful in rescheduling us on the nex
flight to **Chicago** with a connecting flight to **Denver**, to get us to **Portland** about 7 pm that same day. Upon arriving in **Chicago**, we moved to the departure gate
for the **Denver** flight. The Lansing folks told us there was a direct flight to **Portland** leaving about the same time as the **Denver** flight, but had advised us against
flying standby on that one because if we didn't get on the **Portland** flight, we might miss the **Denver** flight.   After the flight just before the **Denver** flight left the
gate, I approached the gate agent to see whether she could check to see if there were seats available on the **Portland** flight. Misty Rojas, the gate agent in
Chicago, confirmed that there were seats available on the **Portland** flight. Not only that, she arranged for our bags to be transferred to that flight, issued our
boarding passes and arranged for transportation between gates. Misty was pleasant, efficient and courteous and truly went beyond the call of duty in helping us woth
our travel. We arrived in **Portland** only about 3 hours later than our originally scheduled arrival time, and our bags were there! I am especially pleased with all the
help from your passenger service agents because the whole problem was my fault, for not reading my ticket right in the first place! Your service is a welcome change
from the indifferent service we usually get from Northwest at the **Detroit** airport (&quot;you don't like it, so what, we're nearly the only game in town!&quot;). We will
certainly consider United again next time we are planning a trip. Flying out of Lansing is a pleasant releif from the **Detroit** airport, and your gate agents changed wha
could have been a complete disaster for our vacation into a minor inconvenience! Please send our thanks and compliments to Passenger service agents Marcus

Figure 7. Test Results for a Document

SAS ENTERPRISE CONTENT CATEGORIZATION SYNTAX

SAS Enterprise Content Categorization supports the following functions that enable you to extract entities and other relationships between terms:

- The CLASSIFIER enables you to specify the terms to be extracted. For example, if CLASSIFIER:Northwest Airlines is defined as an entry for AIRLINE concept, whenever Northwest Airlines is found in the document, it is extracted and assigned as a match to the concept AIRLINE. See Figure 8 for an example of these types of rules.

```
CLASSIFIER:Northwest
CLASSIFIER:Continental
CLASSIFIER:AA
CLASSIFIER:American
CLASSIFIER:Airtran
CLASSIFIER:(AND, ”_c{NW}”, ”Northwest”)
CLASSIFIER:(AND, ”_c{BA}”, ”British”)
CLASSIFIER:Virgin
CLASSIFIER:Lufthansa
```

Figure 8. Sample Classifier Rules for the Airline Taxonomy

- The REGEX definition enables you to define the concept by using regular expression patterns. Because there is no fixed set of values for concepts like date, time, and ticket prices, REGEX enables you to define the patterns in terms of wildcards and characters. Two examples are shown in Figure 9.

```
REGEX:\$[0-9]+(?:\.[0-9]+)?
REGEX:[0-9]+(?:\.[0-9]+)? \s* dollars
```

Figure 9. REGEX Examples

- A CONCEPT type enables you to define a sequence of concepts, terms, syntactic classes, or a combination of them. For example, a CONCEPT: AIRLINECODE NUMBER definition extracts airline numbers like DL 234. In this example, AIRLINECODE is an intermediate concept that is defined as a list of airline codes, and NUMBER is another intermediate concept that is defined as a regular expression that extracts a sequence of digits. An example of a CONCEPT rule that uses a C_CONCEPT is shown in Figure 10.
- The C_CONCEPT definition is similar to a CONCEPT. It further enables you to define patterns such that the match for the pattern is extracted only if the surrounding context is matched.

```
CONCEPT: AIRLINE_CODE NUMBER
C_CONCEPT: \# _c{NUMBER}
C_CONCEPT: flight _c{NUMBER}
```

Figure 10. Sample CONCEPT Type and C_CONCEPT Rules for the Airline Taxonomy

- The CONCEPT_RULE definition enables you to define logical operators on top of C_CONCEPT rules. The currently defined operators are AND, OR, NOT, DIST, SENT, ORDDIST, STENT_N, PARA, SENTSTART, and SENTEND. Among other things, these operators enable you to capture when two terms or concepts are near each other, and they enable you to build entities that trigger only in context. For example, the (AND, "_c{NW}", "Northwest", "Airline") rule considers the word NW as a match for concept AIRLINE only if Northwest and Airline are found elsewhere in the same document.
- A REMOVE_ITEM definition enables you to define rules for removing false matches for concepts. For example, a REMOVE_ITEM definition such as (ALIGNED, "_c{TICKET_NUMBER}", "SKYMILES_NUMBER") enables the model to remove matches for TICKET_NUMBER if there also are matches for SKYMILES_NUMBER.
- Finally, the NO_BREAK rule enables you to define a sequence of words to be recognized as one token. For example, even though "Las Vegas" is two words, it can be recognized as a single multiword term by using this rule. This rule also avoids accidentally recognizing one of the individual terms ("Las" or "Vegas") as a match to a concept. Figure 11 provides two examples.

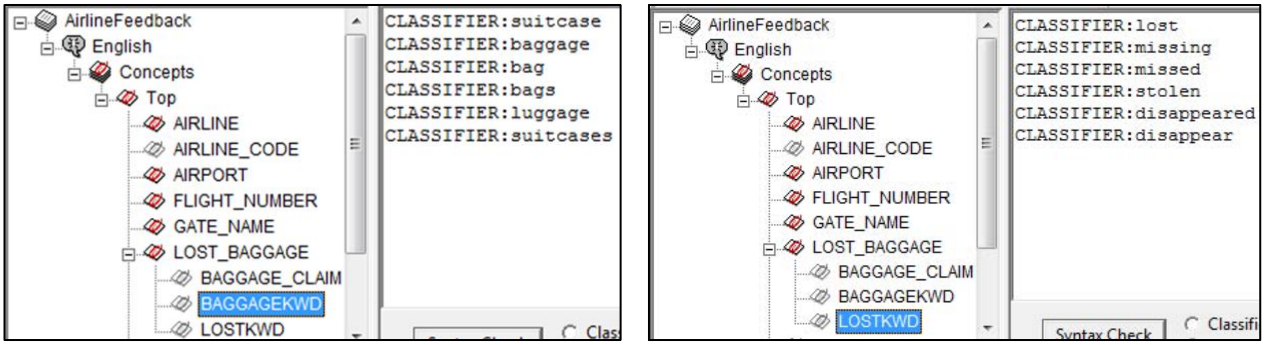
```
NO_BREAK:_c{ Las Vegas}
NO_BREAK:_c{ Delta Airlines}
```

Figure 11. NO_BREAK Examples

A LOST_BAGGAGE EXAMPLE

The syntax shown in the previous section provides very expressive power to users of SAS Enterprise Content Categorization. With experience, you can effectively create rules that extract all types of meaningful concepts. This section provides details about creating a simple but effective custom entity for SAS Text Miner that is based on relating terms to each other when they appear near each other in a document. Because the demonstration data are written by airline passengers about their travel experiences, you can define a new concept, LOST_BAGGAGE, to demonstrate more fully how you might build a particular custom entity for SAS Text Miner. The next section uses this entity to improve the models in relation to the LOST_BAGGAGE concept.

The goal for the LOST_BAGGAGE definition is to extract the occurrences of terms that are related to baggage in the context of lost or missing. Figure 12 shows a LOST_BAGGAGE concept with two supplementary concepts. The BAGGAGEKWD and LOSTKWD concepts define all possible synonyms for the terms "baggage" and "lost," respectively.



```
CLASSIFIER:suitcase
CLASSIFIER:baggage
CLASSIFIER:bag
CLASSIFIER:bags
CLASSIFIER:luggage
CLASSIFIER:suitcases
```

```
CLASSIFIER:lost
CLASSIFIER:missing
CLASSIFIER:missed
CLASSIFIER:stolen
CLASSIFIER:disappeared
CLASSIFIER:disappear
```

Figure 12. Creating Rules to Extract a LOST_BAGGAGE Relationship

After these two concepts are defined, the rule for LOST_BAGGAGE is simply as follows:

```
CONCEPT_RULE:(SENT, "_c{BAGGAGEKWD}", "LOSTKWD")
```

This rule indicates that if at least one of the terms in BAGGAGEKWD and at least one of the terms in LOSTKWD appear in the same sentence, then the term “baggage” is returned as a match for LOST_BAGGAGE.

Frequently, after evaluating the results from a few test documents, you might realize that the rule needs to be improved or refined. Because the term “baggage” is often mentioned in the context of “baggage claim,” it might be useful to refine the set of concepts that are returned for the LOST_BAGGAGE entity by removing those that also trigger a BAGGAGE_CLAIM rule by adding two steps: first, define a new rule for BAGGAGE_CLAIM, and then define a REMOVE_ITEM rule.

The BAGGAGE_CLAIM concept becomes an intermediate concept that you define with the following rule:

```
CONCEPT_RULE:(ORDDIST_3, "_c{BAGGAGEKWD}", "claim@")
```

This rule says that if the term “baggage” or its related terms appear within three words that precede variations of the term “claim,” then this incidence of “baggage” is in the baggage claim context. However, this rule does not prevent “baggage” from being classified as a match for the LOST_BAGGAGE concept. To remove these false positive matches for LOST_BAGGAGE, write the following rule:

```
REMOVE_ITEM:(ALIGNED, "_c{LOST_BAGGAGE}", "BAGGAGE_CLAIM")
```

This rule indicates that if a term matches both LOST_BAGGAGE and BAGGAGE_CLAIM, the system is not to consider it a match for LOST_BAGGAGE and is to remove it from the matches for LOST_BAGGAGE. The two rules together, which now define LOST_BAGGAGE, are shown in Figure 13.

```
CONCEPT_RULE: (SENT, “_c{BAGGAGEKWD}”, “LOSTKWD”)
REMOVE_ITEM: (ALIGNED, “_c{LOST_BAGGAGE}”, “BAGGAGE_CLAIM”)
```

Figure 13. LOST_BAGGAGE Definitions

Although this example is fairly straightforward, it demonstrates a few important components of creating useful extraction rules:

1. The example begins with basic definitions that are defined by using term lists for BAGGAGEKWD and LOSTKWD. It then uses those newly created concept names to define increasingly complex rules.

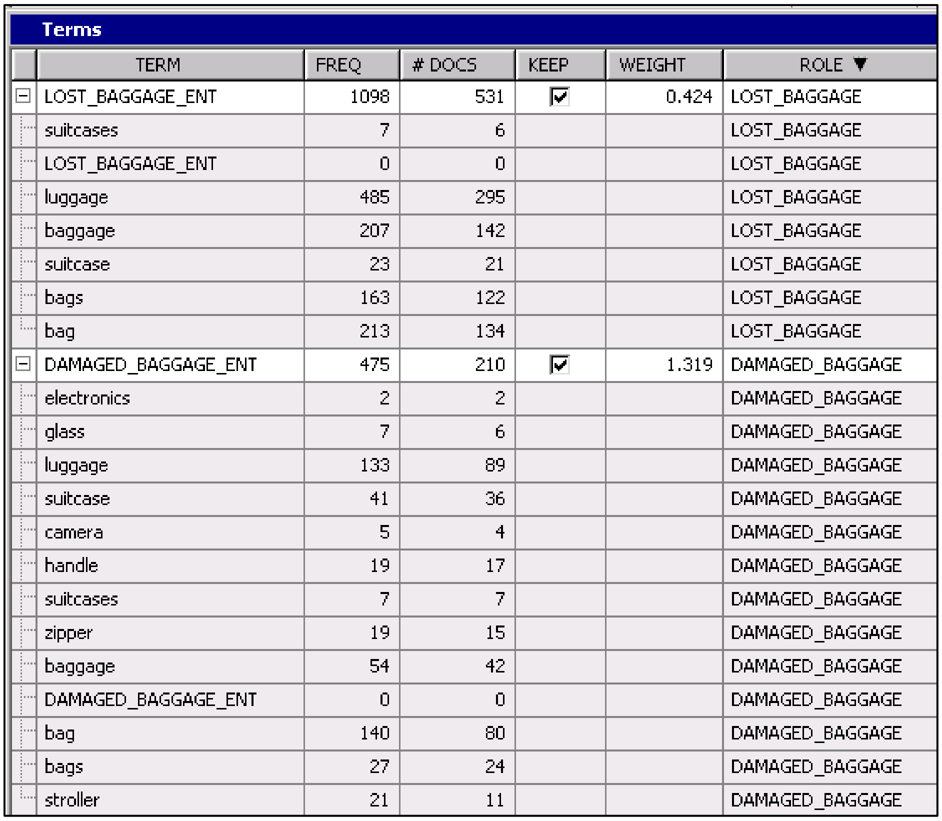
2. The example shows a rule for creating an entity that encapsulates a relationship between two terms that are near each other in your document. The SENT operator captures this relationship in the CONCEPT_RULE when the terms are in the same sentence. You can use any of the operators that are listed in the section “SAS ENTERPRISE CONTENT CATEGORIZATION SYNTAX.”

3. Finally, the example shows a refinement of a rule: using the REMOVE_ITEM definition to eliminate cases that might not be a proper match.

These components will prove useful for many of the projects in which you create custom entities for use in SAS Text Miner.

INCORPORATING CUSTOM ENTITIES IN SAS TEXT MINER MODELS

This section explores a couple of real-world tasks that demonstrate how including custom entities can help you build better models. The primary focus is on adding two custom entities: One creates strong co-occurrence patterns among reports in the data set that are related to lost baggage. The second, which was built in a similar way, captures features that relate to damaged bags. The custom entities LOST_BAGGAGE and DAMAGED_BAGGAGE are shown in Figure 14.



Terms

TERM	FREQ	# DOCS	KEEP	WEIGHT	ROLE ▼
⊟ LOST_BAGGAGE_ENT	1098	531	☑	0.424	LOST_BAGGAGE
suitcases	7	6			LOST_BAGGAGE
LOST_BAGGAGE_ENT	0	0			LOST_BAGGAGE
luggage	485	295			LOST_BAGGAGE
baggage	207	142			LOST_BAGGAGE
suitcase	23	21			LOST_BAGGAGE
bags	163	122			LOST_BAGGAGE
bag	213	134			LOST_BAGGAGE
⊟ DAMAGED_BAGGAGE_ENT	475	210	☑	1.319	DAMAGED_BAGGAGE
electronics	2	2			DAMAGED_BAGGAGE
glass	7	6			DAMAGED_BAGGAGE
luggage	133	89			DAMAGED_BAGGAGE
suitcase	41	36			DAMAGED_BAGGAGE
camera	5	4			DAMAGED_BAGGAGE
handle	19	17			DAMAGED_BAGGAGE
suitcases	7	7			DAMAGED_BAGGAGE
zipper	19	15			DAMAGED_BAGGAGE
baggage	54	42			DAMAGED_BAGGAGE
DAMAGED_BAGGAGE_ENT	0	0			DAMAGED_BAGGAGE
bag	140	80			DAMAGED_BAGGAGE
bags	27	24			DAMAGED_BAGGAGE
stroller	21	11			DAMAGED_BAGGAGE

Figure 14. The LOST_BAGGAGE and DAMAGED_BAGGAGE Entities

The DAMAGED_BAGGAGE entity is used in conjunction with the LOST_BAGGAGE entity in some of the following unsupervised experiments. The DAMAGED_BAGGAGE entity is designed to capture instances of individuals talking about their luggage (or the contents of their luggage) being damaged, broken, or torn. It also includes concepts for reports of the zipper or handle being broken.

Each of the examples in this section first runs without the preceding custom entities. Then, the example includes the custom entities to see how their inclusion affects the model. For topic or clustering models, the example is assessed subjectively by interpreting the descriptive term output of the models. In the case of supervised models, a holdout sample for testing the models is used, and precision and recall scores are provided.

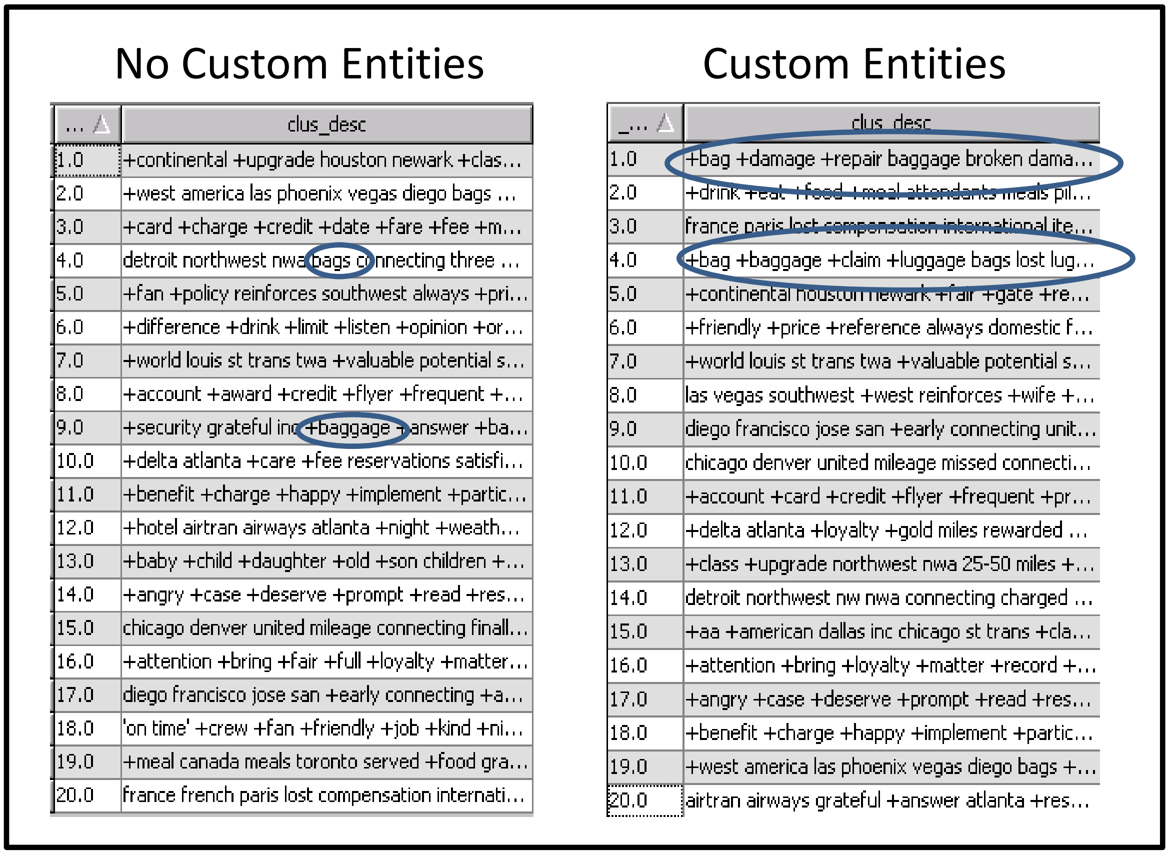
USING CUSTOM ENTITIES FOR UNSUPERVISED LEARNING

SAS Text Miner provides two nodes that focus on unsupervised techniques for exploratory text analysis: the Text Cluster node and the Text Topic node. At a high level, the main difference between the two approaches is that the Text Cluster node restricts each document to belong to exactly one cluster, whereas the Text Topic node allows documents to belong to multiple topics. In both nodes, you can specifically control the number of clusters or topics that you return.

Text Cluster Node

First, the Text Cluster node is investigated and only eight clusters are requested. Before custom entities are included, is one of the eight reported clusters is clearly about baggage of all types. That cluster description includes terms that indicate that both lost luggage and damaged luggage are contained in the same baggage cluster. However, when the LOST_BAGGAGE and DAMAGED_BAGGAGE entities are included, there is no clear baggage cluster among the eight clusters.

At first, this seems disconcerting, but the explanation for this outcome is quite simple. Including the LOST_BAGGAGE and DAMAGED_BAGGAGE entities breaks up the larger baggage cluster so that the general baggage cluster is no longer a primary cluster. You can confirm this by requesting 20 clusters instead of 8. The results are shown in Figure 15. On the left side, where custom entities are not used, a couple of terms for baggage appear, but there is no longer a clear baggage cluster of any type. However, the custom entities, on the right side, show two strong clusters. Prior to the topics being found, the entities were assigned to a common parent in the Text Filter node. (Alternatively, the entities can be assigned using the code in Appendix A.)



No Custom Entities

... △	clus_desc
1.0	+continental +upgrade houston newark +clas…
2.0	+west america las phoenix vegas diego bags …
3.0	+card +charge +credit +date +fare +fee +m…
4.0	detroit northwest nwa bags connecting three …
5.0	+fan +policy reinforces southwest always +pri…
6.0	+difference +drink +limit +listen +opinion +or…
7.0	+world louis st trans twa +valuable potential s…
8.0	+account +award +credit +flyer +frequent +…
9.0	+security grateful inc +baggage +answer +ba…
10.0	+delta atlanta +care +fee reservations satisfi…
11.0	+benefit +charge +happy +implement +partic…
12.0	+hotel airtran airways atlanta +night +weath…
13.0	+baby +child +daughter +old +son children +…
14.0	+angry +case +deserve +prompt +read +res…
15.0	chicago denver united mileage connecting finall…
16.0	+attention +bring +fair +full +loyalty +matter…
17.0	diego francisco jose san +early connecting +a…
18.0	'on time' +crew +fan +friendly +job +kind +ni…
19.0	+meal canada meals toronto served +food gra…
20.0	france french paris lost compensation internati…

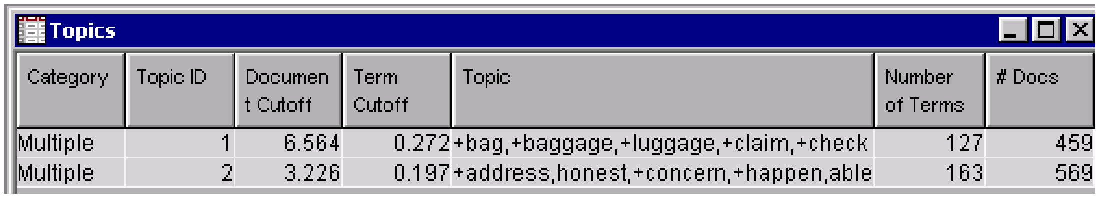
Custom Entities

_... △	clus_desc
1.0	+bag +damage +repair baggage broken dama…
2.0	+drink +eat +food +meal attendants meals pil…
3.0	france paris lost compensation international ite…
4.0	+bag +baggage +claim +luggage bags lost lug…
5.0	+continental houston newark +fair +gate +re…
6.0	+friendly +price +reference always domestic f…
7.0	+world louis st trans twa +valuable potential s…
8.0	las vegas southwest +west reinforces +wife +…
9.0	diego francisco jose san +early connecting unit…
10.0	chicago denver united mileage missed connecti…
11.0	+account +card +credit +flyer +frequent +pr…
12.0	+delta atlanta +loyalty +gold miles rewarded …
13.0	+class +upgrade northwest nwa 25-50 miles +…
14.0	detroit northwest nw nwa connecting charged …
15.0	+aa +american dallas inc chicago st trans +cla…
16.0	+attention +bring +loyalty +matter +record +…
17.0	+angry +case +deserve +prompt +read +res…
18.0	+benefit +charge +happy +implement +partic…
19.0	+west america las phoenix vegas diego bags +…
20.0	airtran airways grateful +answer atlanta +res…

Figure 15. Clustering Results with and without Custom Entities

Text Topic Node

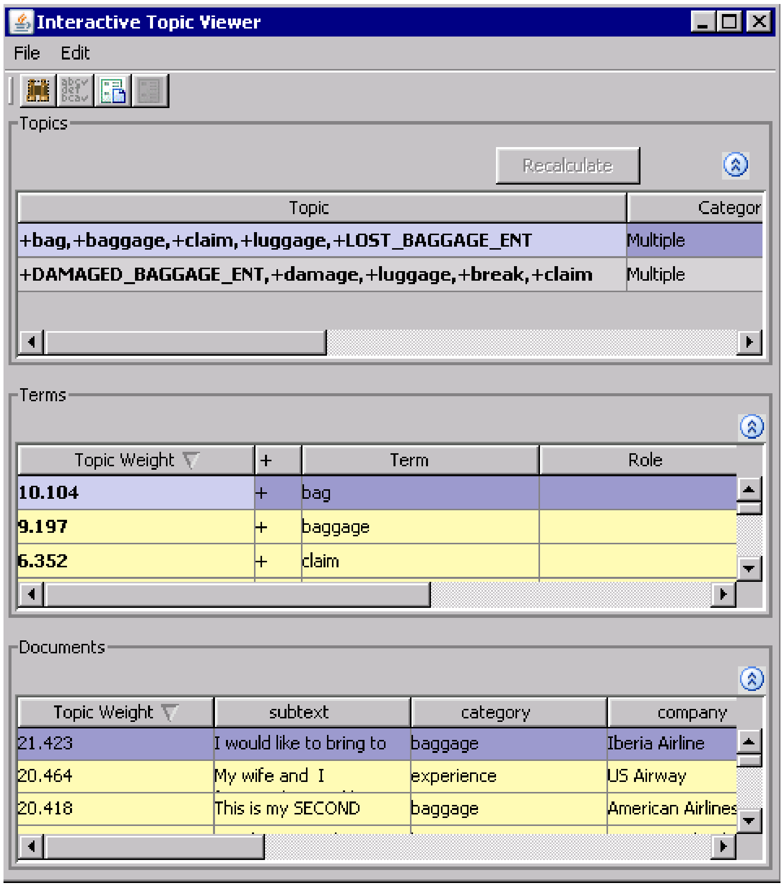
If you configure the Text Topic node to find two automatic topics (both without and with the custom entities), it shows a substantial change on the first two topics. Without the custom entities, the two requested topics are a general one about bags and baggage claim and one about addressing concerns, as shown in Figure 16.



Category	Topic ID	Document Cutoff	Term Cutoff	Topic	Number of Terms	# Docs
Multiple	1	6.564	0.272	+bag,+baggage,+luggage,+claim,+check	127	459
Multiple	2	3.226	0.197	+address,honest,+concern,+happen,able	163	569

Figure 16. Two Topics without Using Custom Entities

However, using custom entities and requesting two topics reveals exactly the results that you anticipate, as shown in Figure 17.



Topic	Categor
+bag,+baggage,+claim,+luggage,+LOST_BAGGAGE_ENT	Multiple
+DAMAGED_BAGGAGE_ENT,+damage,+luggage,+break,+claim	Multiple

Topic Weight	+	Term	Role
10.104	+	bag	
9.197	+	baggage	
6.352	+	claim	

Topic Weight	subtext	category	company
21.423	I would like to bring to	baggage	Iberia Airline
20.464	My wife and I	experience	US Airway
20.418	This is my SECOND	baggage	American Airlines

Figure 17. Two Topics When Using Custom Entities

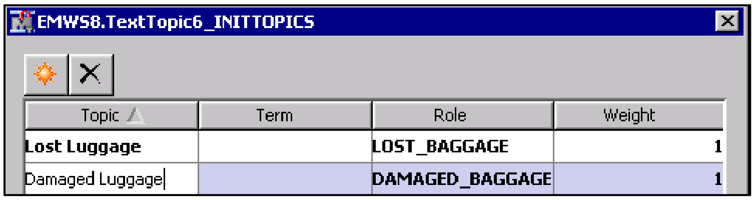
USING CUSTOM ENTITIES FOR SUPERVISED LEARNING

For supervised learning, you train a model on a set of data with the “answers,” and then you hold out a set of data in which you evaluate how well your model is doing. You label your documents according to whether they belong to the LOST_BAGGAGE category or the DAMAGED_BAGGAGE category (or both), and then you build a straightforward text mining predictive model for each of the categories.

Experiments for three different predictive models are examined in this section. Each experiment builds a model to predict the lost baggage binary target variable. All three models are based on using the Text Topic node to get the projections from the singular value decomposition (SVD) onto the document data set and then using the Regression node to learn a logistic regression model.

You can use three different sets of feature inputs to build the logistic regression model:

1. Use the SVD dimensions from the Text Topic node in which no custom entities were included.
2. Use the SVD dimensions from the Text Topic node in which custom entities were included.
3. Us the SVD dimensions from the Text Topic node, but also include the user-defined topic for the entity as a feature in the predictive model. User-defined topics is a property in the Text Topic Node. Figure 18 shows how to set the user-defined topic. When the term is left blank, all terms with the specified role are considered as part of the named topic.



EMWS8.TextTopic6_INITTOPICS

Topic	Term	Role	Weight
Lost Luggage		LOST_BAGGAGE	1
Damaged Luggage		DAMAGED_BAGGAGE	1

Figure 18. Creating User Topics from Custom Entities

In Figure 19 shows the results of the three approaches. Because most of the documents were not labeled as lost baggage in the first place, the successful classification rate was in the upper 90th percentile for all models listed. To better discriminate between the models, you can instead show the F1 score that is derived from an averaged precision and recall calculation.

	Lost Baggage
SVD-Logistic	.65
Custom Entities-SVD-Logistic	**.72**
SVD-Custom Entities-Logistic	.68

Figure 19. F Scores for Three Models

The best model is the one that includes the custom entities as a step in parsing, assigns all entities of that type to a single parent term, and then uses that parent term with the other terms to build a model. These scores might be higher if you use more training data.

SUMMARY

Custom entities give you additional control over your text mining projects by enabling you to introduce some of your own experience and background knowledge into an otherwise statistically based approach to text analytics. The custom features that you introduce can substantially alter the model by creating specific co-occurrence patterns that the bag-of-words model could not otherwise expose.

It is useful to keep in mind the following key perspectives:

- In most cases, custom entities initially expand the number of distinct features in your model. To improve co-occurrence relationship and modeling benefit, you should then explore treating all entities of the same type as if they were the same feature.
- You are likely to gain more from custom entities that are applied to longer documents than to very short documents. The strength of custom entities is in relating terms that are near one another. The longer your documents, the more this applies.
- The SVD matrix factorization is a key component in building many text mining models. It makes two documents similar, not only because of the co-occurrence patterns between those two documents, but also because of the co-occurrence patterns among those two documents and other documents. This means that you need to create custom entities that have a significant effect throughout the collection. Adding an entity rule that triggers very rarely in your collection is not likely to produce the type of change that you can observe.
- The custom entity feature in SAS Text Miner gives you the power and control to manipulate model building in a way that reflects your own perspective on the data. It enables you to insert your own background knowledge into an otherwise statistical process, enabling you to ultimately improve your models to make them more relevant, meaningful, and effective.

APPENDIX A

The following code can be placed in the Code node that immediately follows a Parse node in SAS Text Miner. You need to edit the two macro variables so that they contain the list of all entities for which you want to create a synonym.

```
/* These are entity types that you want to act as a parent */
%let entities_as_parent = LOST_BAGGAGE DAMAGED_BAGGAGE;
%let entities_as_parent_q ='LOST_BAGGAGE' 'DAMAGED_BAGGAGE';
%let entity_Term_q='LOST_BAGGAGE_ENT' 'DAMAGED_BAGGAGE_ENT';
%let prev_terms=&em_lib..textparsing2_terms;
%let prev_out=&em_lib..textParsing2_tmout;


/*Create new keys for new parent terms */
proc sql;
select max(key) into: _termsMaxId
from &prev_terms;
quit;

/* Create the parent terms*/
data parents_ent(drop= j );
  length term $256 role $256 Keep $8;
      key=&_termsMaxId;
      j=1;
      Role=scan("&entities_as_parent",j,' ');
      do until (role=' ');
         key=key+1;
         Term=strip(role)||"_ENT";
         rolestring=role;
         Keep='Y';
         _ispar=' ';
          parent_id=key;
          freq=0;
         numdocs=0;
         output;
         j=j+1;
         Role=scan("&entities_as_parent",j,' ');
     end;
run;

data terms_ent;
```

```
set &prev_terms parents_ent;
run;

/*Make a synonyms data set*/
options mprint;
data children_ent;
set terms_ent;
if role in (&entities_as_parent_q) and term not in (&entity_Term_q)
         then output;
keep key role;
run;

proc sql;
create table syns_ent as
select a.key as _termnum_, b.parent_id
from children_ent a inner join parents_ent b
on a.role=b.role;
quit;


/* Apply synonyms */
proc tmutil data=&prev_out key=terms_ent doc=&em_import_Data;
control init  release;
syn syndata=syns_ent force;
output key=term_tmutil_ent keyformat= default;
run;


data terms_ent;
set &prev_terms parents_ent;
if _isPar in (" ","+");
run;



proc sql;
create table &prev_terms as
select a.key, a.Parent, a._ispar, a.parent_id,a.freq,a.numdocs,a.keep,
            b.term, b.role, b.rolestring, b.attribute,b.attrstring
from term_tmutil_ent a left join Terms_ent b
on a.key=b.key;
quit;
```

REFERENCES

Lange, K., and Sethi, S. “What Are People Saying about Your Company, Your Products, or Your Brand?” *Proceedings of SAS Global Forum 2011Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/158- 2011.pdf

SAS Institute Inc. (2010). *Combining Knowledge and Data Mining to Understand Sentiment: A Practical Assessment of Approaches*. SAS Technical Report 2799. Cary, NC: SAS Institute Inc. Available at http://www.sas.com/reg/wp/corp/27999

ACKNOWLEDGMENTS

The authors are grateful for the comments, ideas, and insights provided by Saratendu Sethi and James Cox. The authors would also like to thank Anne Baxter for her editorial contributions.

CONTACT INFORMATION

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

Russell Albright
SAS Institute Inc.
russell.albright@sas.com

Janardhana Punuru
SAS Institute Inc.
janardhana.punuru@sas.com

Lane Surratt
SAS Institute Inc.
lane.surratt@sas.com