Paper 11775 - 2016

Using SAS® Text Miner for Automatic Categorization of Blog Posts on a Social Networking Site Dedicated to Cyclists

Heramb Joshi, Oklahoma State University; Dr. Goutam Chakraborty, Oklahoma State University

ABSTRACT

Cycling is one of the fastest growing recreational activity and sport in India. There are many NGOs (Non-government Organizations) supporting this emission free mode of transport and help protect environment from air pollution hazards. There are lot of cyclists groups in metropolitan areas which organize numerous ride events and build social networks. Though India was a bit late for joining the Internet, the social networking sites are getting popular in every Indian states and is expected to grow even faster after announcement of Digital India Project. There are many networking groups and cycling blogs sharing a plethora of information and resources across the globe. However one concern is that these blogs posts are hard to categorize according to their relevance. This makes it difficult to access required information quickly by referring to those blogs. This paper aims to provide ways to categorize the contents of these blog posts and classify them in meaningful categories for easy identification.

INTRODUCTION

Recently the world witnessed the thrill of three weeks long, prestigious Tour de France. Even this year's Ironman 70.3 world Championship, a world-wide popular triathlon which includes 56 mile bike ride out of total 70.3 mile journey, set new records of popularity. This race was marked by participation of a few Indian tri-athletes, one of them is a very well-known personality in Indian Cinemas, inspired lot of people in India to look at cycling as one of the best physical activities. The popularity and support for this environmental friendly sport is increasing day by day. There are many cycling groups (I was associated with one of them), organize various events such as Cyclothons, Velorides, Monsoon-Safari, etc. Many of them commute daily to their offices on bike inspiring others to do so and help minimize traffic issues in metropolitan cities of India. These groups make effective use of social networking sites and blogs to connect with each other. There is a lot of content shared on the Internet by these cyclists and bloggers summarizing their ride experience, advising newbies about bike selection and riding tips, promoting social cause such as tree plantation while riding to nearby City Mountains or sharing cool deals on bicycles. India being a bit new to advent of the Internet era, the content on the web is not yet categorized in meaningful groups. This makes it difficult to get the useful information at quickly. Wouldn't it be nice if whenever someone posts a blog about cycling it automatically get classified into relevant categories? This project aims to provide such a tool (in JavaScript), if incorporated in the blogging site, would take care of content categorization based on the rules generated by SAS® Text Miner.

DATA PREPARATION

The textual data for our analysis is prepared using following steps. These steps include data extraction from online cyclist's blogs, importing the textual data in SAS environment to create a SAS dataset, parsing the textual data to identify the term-document matrix and identify the linguistic terms, Text filtering to check for spelling errors using the dictionary. The detailed flow of the data preparation is explained in following steps.

DATA EXTRACTION

The textual data is collected from the cycling blogs using web-scraping. For web-scraping, BeautifulSoup4 package in python 2.7.10 is used. The web contents of the blogs are parsed using HTML parser in python. Once web content is available, the irrelevant blog contents (blog URLs, location etc.) is filtered and body of the blog post is extracted using regular expressions in python. These blog posts are extracted as independent text documents for further analysis. Following flow depicts the data extraction phase using python.

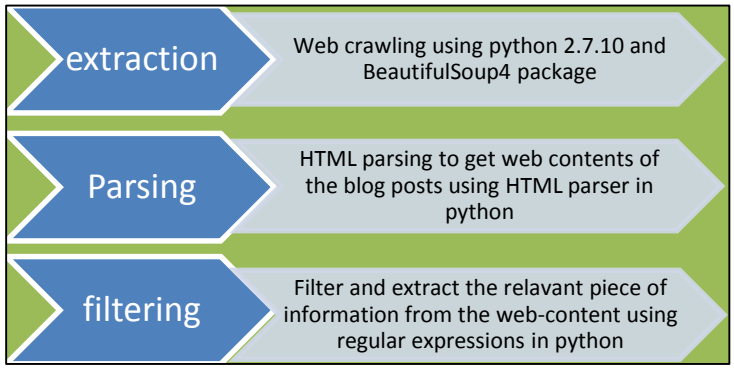


Figure 1 Data Extraction Process Flow

IMPORTING TEXTUAL DATA INTO SAS ENVIRONMENT

The textual data residing in number of blog files is imported into the SAS environment using Text import node in SAS® Text Miner. The import Source and destination directories are specified. The language used for the blogs is “English” and text size is set to 32000. This gathers the textual data spread across the documents to create a SAS dataset containing the textual data.

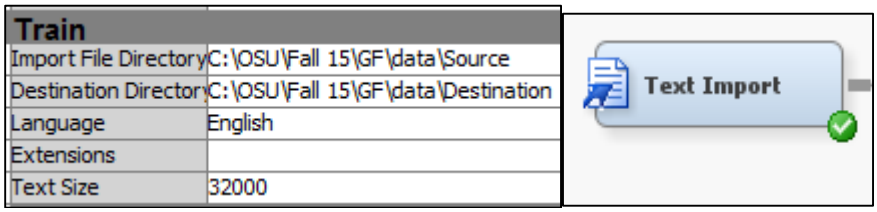
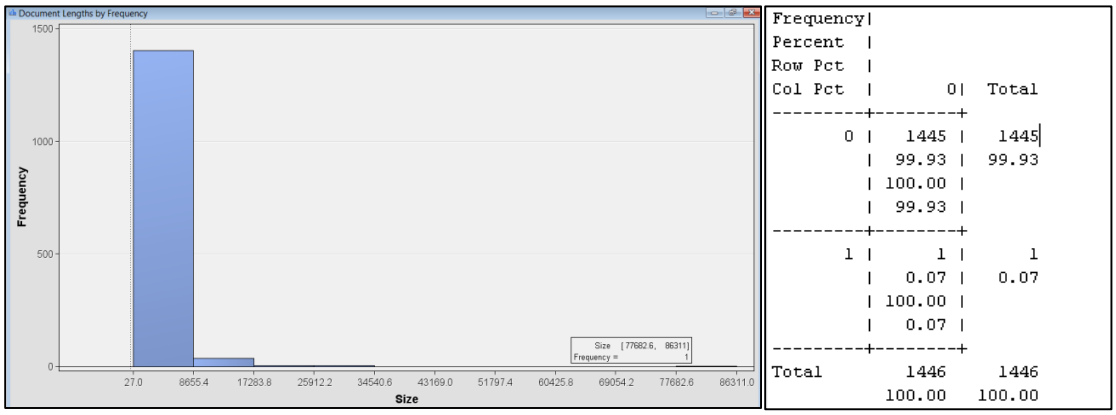


Figure 2 Text Import Node

Out of 1,446 blog posts documents, one got truncated. That blog post consist of somewhere around 78,000 characters thus got omitted. The majority of the blog posts are between 27 to around 8000 words with a few documents of higher word counts. The output of text import node is depicted as below.



Frequency Percent Row Pct Col Pct	0	Total
0	1445 99.93 100.00 99.93	1445 99.93
1	1 0.07 100.00 0.07	1 0.07
Total	1446 100.00	1446 100.00

Output 1 Text Import Node Results

METHODOLOGY

Once textual data is available in form of a SAS dataset, following two step methodology is used for text analytics [1].

1. Creating text clusters for identifying meaningful blog post categories.
2. Generating the text rules based on the categories identified in the above step.

CREATING TEXT CLUSTERS

For identifying meaningful categories of the blog posts, following text mining process flow is implemented.

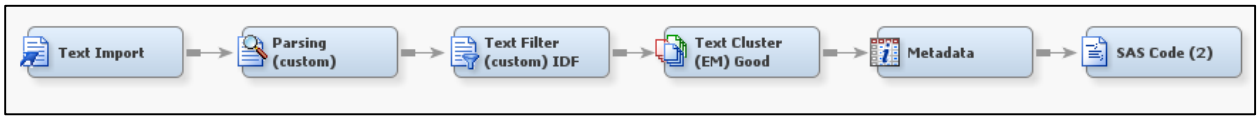
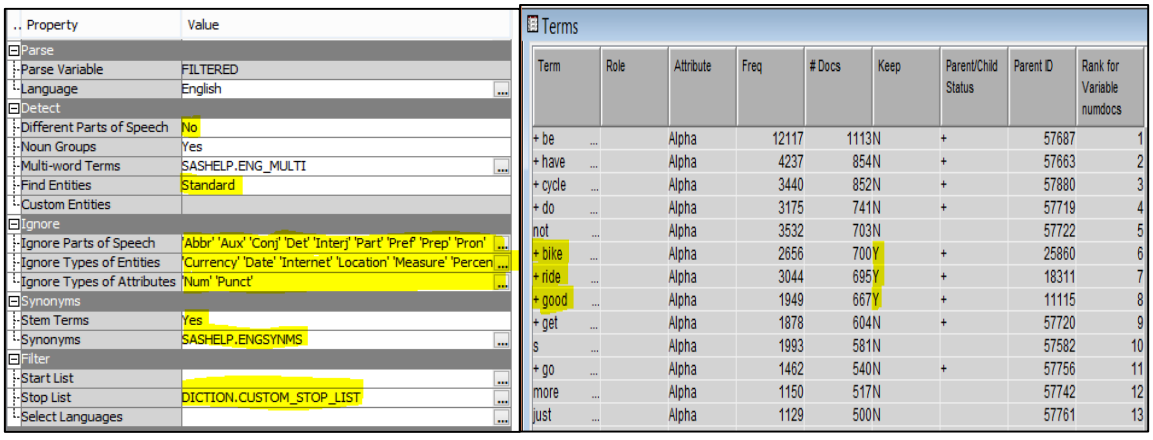


Figure 3 Modeling Diagram for Creating Text Clusters

Text parsing

The textual dataset generated by SAS® Text import node is parsed to enumerate the terms contained in the document. It identifies the word terms based on various parts of speech present in the document. Following properties are altered in properties panel of Text Parsing node.

- "Detect different parts of speech" is turned off which limits the terms with same parts of speech.
- "Find Entities" is set to "Standard".
- We have ignored following parts of speech which filters prepositions, determinants, auxiliary verbs etc. which generally contains very less information.
- We have ignored Numeric and Punctuation attributes.
- Also entities such as Currency, Internet, Measure, Person etc. are ignored.
- Even though Person entity is set to be ignored, the blog post contained Indian person names. The list of Indian names is gathered and included as a part of "Stop List" in Filter property. This significantly limited the number of terms.



Property	Value
Parse	
Parse Variable	FILTERED
Language	English
Detect	
Different Parts of Speech	No
Noun Groups	Yes
Multi-word Terms	SASHELP.ENG_MULTI
Find Entities	Standard
Custom Entities	
Ignore	
Ignore Parts of Speech	'Abbr' 'Aux' 'Conj' 'Det' 'Interj' 'Part' 'Pref' 'Prep' 'Pron'
Ignore Types of Entities	'Currency' 'Date' 'Internet' 'Location' 'Measure' 'Percen
Ignore Types of Attributes	'Num' 'Punct'
Synonyms	
Stem Terms	Yes
Synonyms	SASHELP.ENGSYNMS
Filter	
Start List	
Stop List	DICTION.CUSTOM_STOP_LIST
Select Languages	

Terms

Term	Role	Attribute	Freq	# Docs	Keep	Parent/Child Status	Parent ID	Rank for Variable numdocs
+ be		Alpha	12117	1113N		+	57687	1
+ have		Alpha	4237	854N		+	57663	2
+ cycle		Alpha	3440	852N		+	57880	3
+ do		Alpha	3175	741N		+	57719	4
not		Alpha	3532	703N			57722	5
+ bike		Alpha	2656	700Y		+	25860	6
+ ride		Alpha	3044	695Y		+	18311	7
+ good		Alpha	1949	667Y		+	11115	8
+ get		Alpha	1878	604N		+	57720	9
s		Alpha	1993	581N			57582	10
+ go		Alpha	1462	540N		+	57756	11
more		Alpha	1150	517N			57742	12
just		Alpha	1129	500N			57761	13

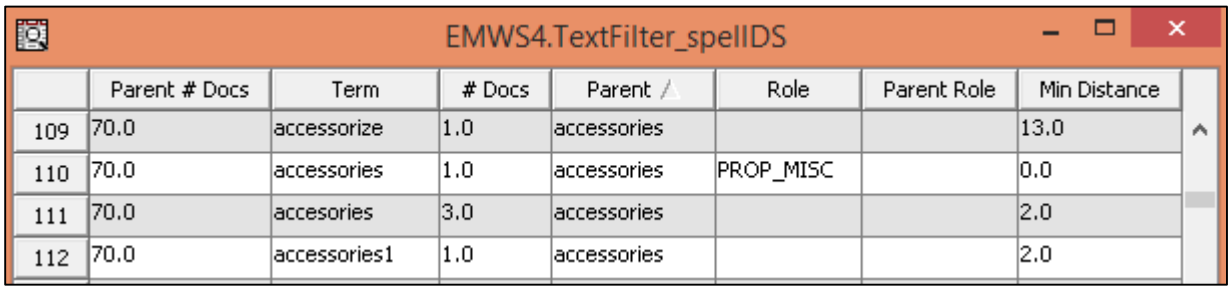
Figure 4 SAS Text Parsing Node property panel settings and Output

Some of the terms such as "cycle", "bike", "ride" etc. are the most occurring words which is obvious as we are analyzing cycling blog posts.

Text Filtering

To reduce the number of terms used in the documents, text filter node is used. We have used English dictionary to identify and correct the spell check errors; if not handled, will result in keeping similar words as different terms expanding the term document matrix. Using filter viewer, we can view which all documents contain a specific term and also create concept links based on those terms. We have used Text filtering node with term weight property as “Inverse Document Frequency”.

The “Check Spelling” option corrected wrong spellings of the word “accessories” as shown below.



	Parent # Docs	Term	# Docs	Parent	Role	Parent Role	Min Distance
109	70.0	accessorize	1.0	accessories			13.0
110	70.0	accessories	1.0	accessories	PROP_MISC		0.0
111	70.0	accesories	3.0	accessories			2.0
112	70.0	accessories1	1.0	accessories			2.0

Output 2 Text Filtering Node output

Apart from the common English word synonyms identified using Text Parsing node, we have created a few custom synonyms list treating those terms as similar terms. Using interactive filter viewer, irrelevant terms are filtered. Some of the terms that are kept for further analysis are as follows.

Terms

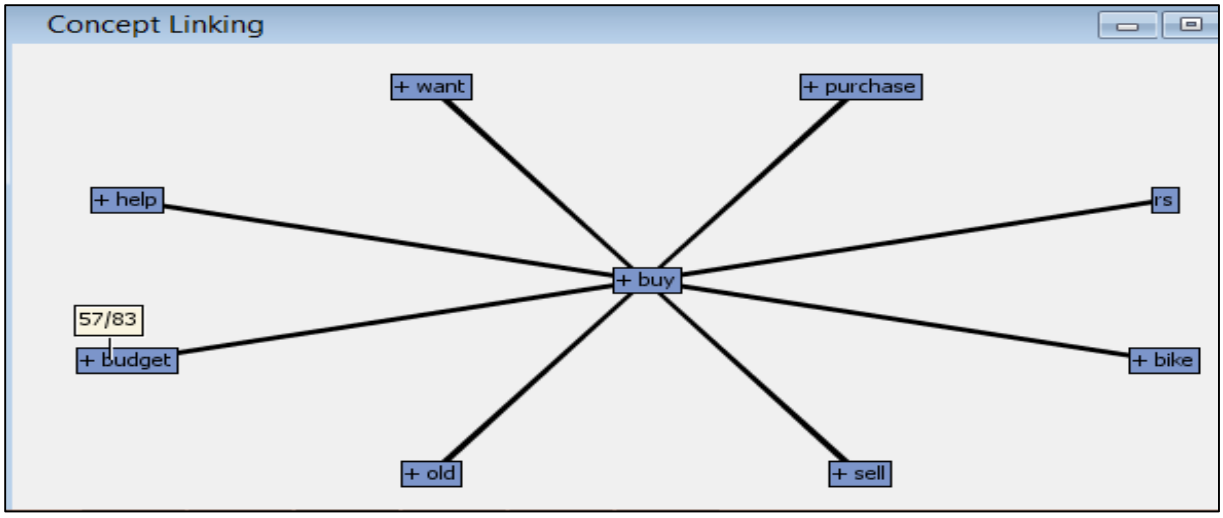
TERM	FREQ	# DOCS	KEEP	WEIGHT	ROLE	ATTRIBUTE
bike	2667	701	☑	2.037		Alpha
ride	3058	696	☑	2.047		Alpha
good	1955	667	☑	2.108		Alpha
road	1527	490	☑	2.553		Alpha
time	1266	462	☑	2.638		Alpha
day	1278	450	☑	2.676		Alpha
look	842	445	☑	2.692		Alpha
know	865	435	☑	2.725		Alpha
back	856	389	☑	2.886		Alpha
want	660	365	☑	2.978		Alpha
bicycle	1425	353	☑	3.026		Alpha
buy	664	349	☑	3.043		Alpha
help	795	348	☑	3.047		Alpha
first	665	346	☑	3.055		Alpha

Output 3 High Frequency Terms

CONCEPT LINKS

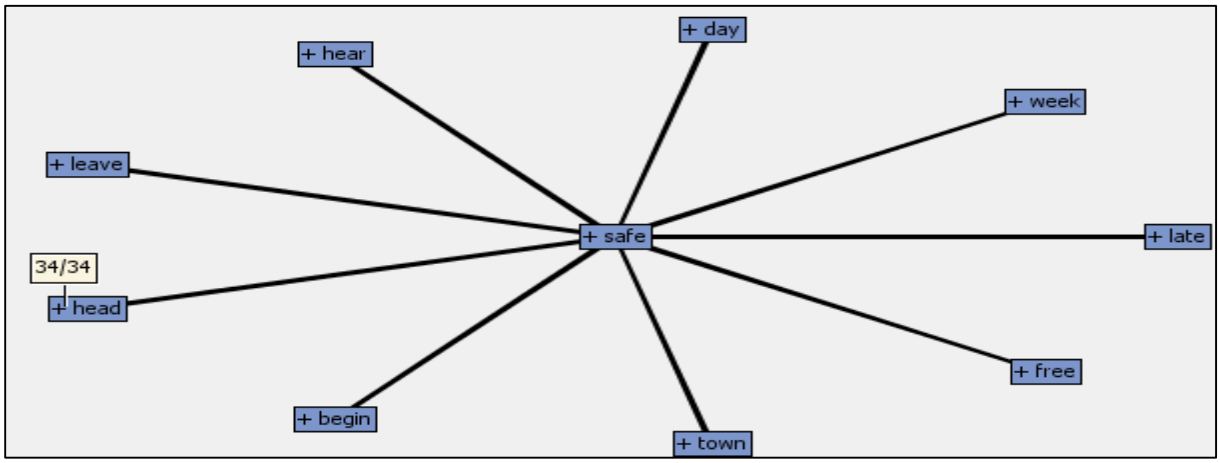
Concept links help in understanding the relationships between words based on the co-occurrence of words in the documents [1]. The hub and spoke structure of the concept link represents the association between those terms and width of the link represents the strength of association. Thicker is the link between two terms, stronger is the association between the terms [1]. Some of the important terms related to cycling blogs such as “Buy”, “Safe”, “Ride” and “Bike”, etc. are analyzed as below.

1. Buy: - As per the concept link shown below, among all the words associated with “Buy”, “Budget” is the most frequent term occurring together (57/83 i.e. 57 out of 83 documents in which the term budget occurs). This sounds reasonable, as for many people in India, budget is the most important parameter while making any purchase decisions. Also we can see, there are many people interested in buying “old” bikes and/or accessories (127/260) and discussing over the blogs about possible deals. The term “rs” represents Indian Currency, Rupees.



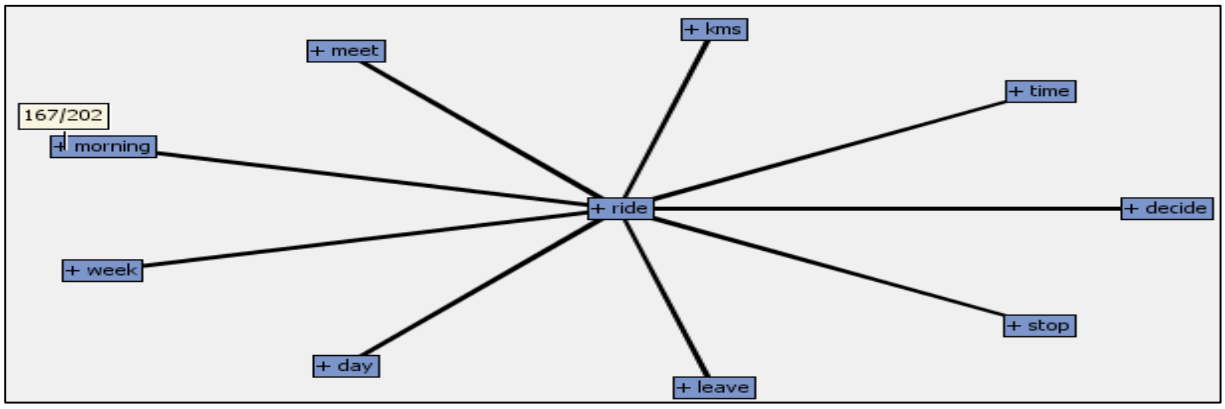
Output 4 Concept Link for the term "Buy"

2. Safe: - In the matter of traveling on Indian roads, safety is the primary concern. The head injuries due to bad road conditions are frequent. Thus, head safety is discussed more on the blogs.



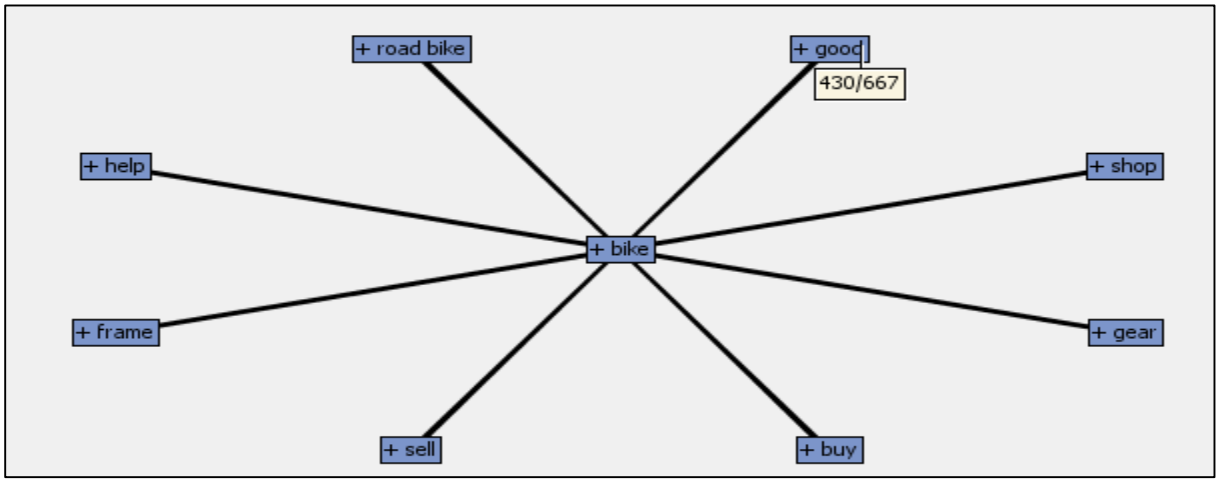
Output 5 Concept Link for the term "Safe"

3. Ride: - Also morning rides are very common in India as most of the cities in India are pretty warm during daytime.



Output 6 Concept Link for the term "Ride"

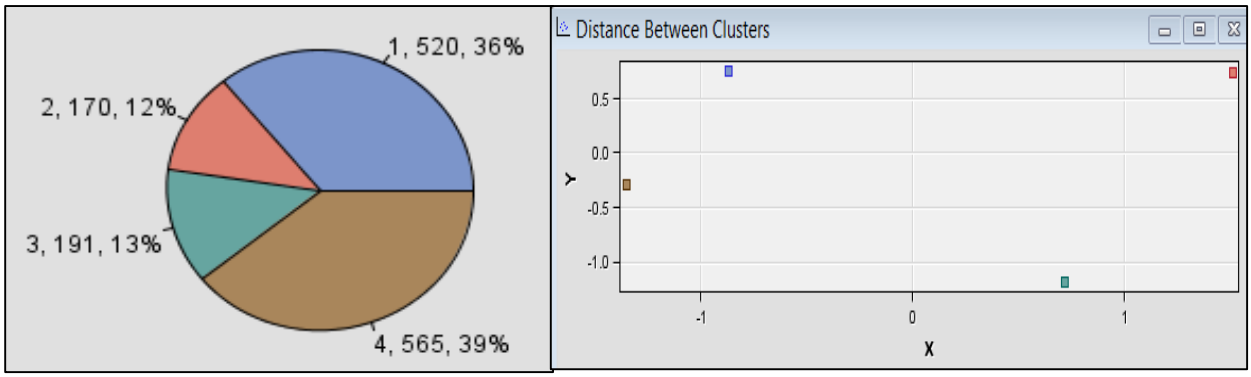
4. Bike: - The term Bike is discussed most frequently in context of “Buy” (269/349) and “Sell” (165/210). There are many blog posts which seeks “help” (243/348) in selecting suitable “bike”. Also there are frequent blog posts seeking suggestions for “good” (430/667) bikes to start. This concept link is very helpful in identifying the possible categories of the blog posts such as “Buy and Sell”, “Bike Advice” etc.



Output 7 Concept Link for the term "Bike"

Text Clustering

After filtering the irrelevant terms and combining similar terms, we used Text Cluster Node for clustering the blog posts into meaningful categories based on the terms present in them. The cluster algorithm is used is Expectation - Maximization. Initially with default properties, many clusters with overlapping terms were developed. We restricted the number of clusters in the property panel of Text Cluster Node and arrive at fewer clusters (E.g. 4, 5 and 6 clusters). Among those, 4 cluster solution is selected based on the least overlap among the descriptive terms. The cluster output is as follows.



Output 8 Text Clustering Node Output

The descriptive terms in this 4 cluster solution and their meaningful categories are as follows.

Cluster ID	Descriptive Terms	Frequency (%)	Meaningful category
1	+life +feel +bicycle +work +old +find +people +first +year +help	520 (36%)	Ride Experience
2	+bike +price +sale +sell +interest +buy +contact +condition rs +frame	170 (12%)	Buy and Sell
3	+bike +good +buy +budget +gear +suggest mtb +brand +trek +suggestion	191 (13%)	Bicycle Advice
4	+ride +start +cyclist +join +group +plan +great +day +time +back	565 (39%)	Ride Events

Table 1 Descriptive Terms for the Text Clusters

GENERATING TEXT RULES FOR CATEGORIZATION

Once the clusters are identified, the categories are assigned as mentioned above and created a new SAS dataset with the blog posts as the text variable and category as the target variable. Using this datasets, content categorization code is generated in Text Rule Builder [3]. The model diagram is as follows.

Figure 5 Modeling Diagram for generating Text Rules for Categorization

The dataset is partitioned into training (70%) and validation (30%) for honest assessment. The text parsing settings are the same as used earlier as the dataset has the same textual data we worked earlier. The text filtering properties used are also similar to the one we worked earlier except the "term weight" option now used is "Mutual Information" since now we are predicting the categories of the blog posts.

Text Rule Builder Node

Using this node, text rules for categorizing the blog posts are generated. The "Generalization Error", "Purity of Rules" and "Exhaustiveness" properties in the property panel of the node are set to "medium". This ensures optimum rule generation without overfitting the model. We have used the option "Change Target Values" in the property panel to reassign some of the categories for generating correct text rules. For example, as per below figure, the highlighted rows predict the target as "Ride Events" but the original target was "Bicycle Advice". After carefully going through the content it is found that this blog post should be categorized under bicycle buy and sell thus it is assigned a target value "Buy and Sell".

Change Target Values-WORK.TRCHANGE

Text	Data Partition	Target Variable	Original Target	Predicted Target	Why Classified	Posterior Probability	Assigned Target
hero hawk sale vashi Hi I am putting up my Hero hawk 1	Training	category	BICYCLE ADVICE	BUY AND SELL	sale & ~ride	75.1%	BUY AND SELL
Super Sunday offer at BikeShark 19-05-2013 Presenti	Training	category	BICYCLE ADVICE	RIDE EVENTS	sunday	73.7%	BUY AND SELL
Hi Guys, I am planning to buy a helmet seen in a local s	Training	category	BICYCLE ADVICE	RIDE EVENTS	route & ~second & ~look	62.1%	BICYCLE ADVICE
BSA MACH for sale; Mumbai, Juhu. i am selling off my BS	Validate	category	BICYCLE ADVICE	BUY AND SELL	sell & ~help & interest	99.1%	BUY AND SELL

Figure 6 "Change Target Values" setting of Text Rule Builder Node

Once all the changes for target values are completed, the text rule builder node is rerun to get the improved text model. The validation misclassification rate for this model is around 36%. This suggests, the text model could correctly classify 64% of the blog posts in the relevant categories. This model is fairly reasonable considering it has automatically built text rules to classify the blog posts which were scattered earlier on the blog websites.

Fit Statistics

Target	Fit Statistics	Statistics Label	Train	Validation
category	_ASE_	Average Squared Error	0.049678	0.056637
category	_DIV_	Divisor for ASE	4040	1744
category	_MAX_	Maximum Absolute Error	0.619705	0.663728
category	_NOBS_	Sum of Frequencies	1010	436
category	_RASE_	Root Average Squared Error	0.222886	0.237986
category	_SSE_	Sum of Squared Errors	200.7001	98.77549
category	_DISF_	Frequency of Classified Cases	1010	436
category	_MISC_	Misclassification Rate	0.254455	0.360092
category	_WRONG_	Number of Wrong Classifications	257	157

Output 9 Fit Statistics for Text Rule Builder Node

The Text rule builder creates simple logical rules which classifies the text into meaningful categories based on correct keywords. The text rules generated contains multiple forms of the same verb or word. This is because for analyzing the text we can used the verbs stemmed to its root but for classifying the text, different forms must be identified. A sample rule for the category “Buy and Sell” is as shown below.

```
F_category =BUY AND SELL ::
(OR
, (AND, (OR, "ineterested" , "interested" , "interesting" , "interest" , "intresting" , "inte
, "selling"
, (AND, (OR, "sales" , "sales" , "sale" ), (NOT, (OR, "ridden" , "riding" , "riden" , "riding
, "sale"
, (AND, (OR, "accessories" , "accesories" , "accessory" , "accessories" , "accessories1" ), (
```

Output 10 Content Categorization Code of Text Rule Builder Node

Some of these rules are explained as below.

1. Buy and Sell

Buy and Sell						
		(Interested, Interesting, Interest)	AND	NOT(helping,help, helped)	AND	(Selling,sold, sell)
	OR					
		(Interested, Interesting, Interest)	AND	NOT(helping,help, helped)	AND	(Buying,bought, buy)
	OR					
		Selling				
	OR					
		(Sales,sale)	AND	NOT(ridding,ridden, ride,rode)		
	OR					
		sale	AND	accessories		

Legend
Most Important Words
Must not contain these words
Either of the words must be present

Figure 7 Text Rules for the category "Buy and Sell"

When the blog user enters “Interested in selling Bike/ accessories”, the tool classifies it under the category “Buy and Sell”. The checks whether the text contains the word “help” or not. If the text contains “help” and other terms, it is most likely to seek some advice and must be categorized under “Bike Advice”.

2. Bike Advice

Bike Advice								
	Budget							
OR								
	NOT(left,leaving,leave)	AND	NOT(work,working, works)	AND	(buying,buy, buys,bought)	AND	(suggest,suggests, suggested,suggesting)	
OR								
	NOT(left,leaving,leave)	AND	(review,reviews, reviewed,reviewing)	AND	(buying,buy, buys,bought)			
OR								
	get	AND	(bikes,bike,biking)	AND	NOT(keep,keeping, kept)			
OR								
	(suggest,suggests, suggestion, suggested,suggesting)	AND	(bikes,bike,biking)					
OR								
	NOT(cyclists,cyclist)	AND	(showroom, showrooms)	AND	(bikes,bike,biking)			

Legend
Most Important Words
Must not contain these words
Either of the words must be present

Figure 8 Text Rules for the category "Bike Advice"

IMPLEMENTATION

The Text Rule Builder node identified the keywords necessary for classifying the blog posts. Further a tool is developed in JavaScript which applies these text rules for predicting the blog categories. When the blog user enters the blog content, this JavaScript tool parses the entered text, identifies the keywords and apply the text rules built earlier to classify this text. Considering that user may commit some spelling mistakes while entering the text, the spell check English dictionary is included in the tool. This automatically detects the spelling errors and provide appropriate suggestions. Consider following example in which the user enters the wrong spelling for the word “accessories”. The tool provide the appropriate suggestion and identifies the relevant category.

Figure 9 JavaScript based Implementation for classifying the blog posts

This text model needs to be extended to incorporate other variations in the blog posts as well. Due to complexities in the blog posts data, sometimes you need to go through the blog posts manually to come up with the rules that will classify them in correct categories. This will improve the accuracy of the text model further. A sample JavaScript code is as shown below.

```
if(
((text.indexOf('suggest') > -1) ||(text.indexOf('suggestion') > -1))  &&
((text.indexOf('bikes') > -1) || (text.indexOf('bike') > -1))
        )
        {
        suggestions.push('Bicycle Advice');
                }
```

Figure 10 Sample JavaScript Code Snippet

CONCLUSION

- With the validation accuracy of around 66%, the text model performance is fairly reasonable compared to the earlier scattered blog posts.
- The classifier tool is in its earlier stage. With more blog posts available for training the tool will provide dynamic set of rules for classifying other categories of blog posts such as “Bicycle hacks”, “Nutrition” etc.

REFERENCES

1. Chakraborty, Goutam; Pagolu, Murali and Garla, Satish. *Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS®*
2. SAS Documentation. “*Getting started with SAS®Text Miner 12.1”* Available at

   https://support.sas.com/documentation/onlinedoc/txtminer/12.1/tmgs.pdf
3. Chakraborty, Goutam and Pagolu, Murali. 2014 “Automatic Detection of Section Membership for SAS® Conference Paper Abstract Submissions: A Case Study” *Proceedings of the SAS Global Forum 2014 Conference*, 1746-2014, Washington, DC. : SAS Institute Inc.

   Available at https://support.sas.com/resources/papers/proceedings14/1746-2014.pdf

ACKNOWLEDGMENTS

I thank Mr. Digambar Chaudhari, Founder and CEO at ZERO Emission – Early morning riders (India) and a close friend of mine for his expert suggestions in cycling blog posts.

I also thank Dr. Goutam Chakraborty for his valuable guidance and motivation for accomplishing this paper.

CONTACT INFORMATION

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

Dr. Goutam Chakraborty
Oklahoma State University
Stillwater, OK, 74078
goutam.chakraborty@okstate.edu

Dr. Goutam Chakraborty is Ralph A. and Peggy A. Brenneman professor of marketing and founder of SAS® and OSU data mining certificate and SAS® and OSU marketing analytics certificate at Oklahoma State University. He has chaired the national conference for direct marketing educators for 2004 and 2005 and co-chaired 2007 data mining conference. He has over 25 years of experience in using SAS® for data analysis. He is also a Business Knowledge Series instructor for SAS®.

Heramb Joshi
Oklahoma State University
Stillwater, OK, 74075
heramb.joshi@okstate.edu

Heramb Joshi is a graduating Master's student in Management Information Systems from Oklahoma State University. He has a bachelor's degree in Electronics and Telecommunication engineering from Mumbai University. He worked as a graduate research assistant at CHSI Tulsa during June – August 2015. Earlier he worked as an ETL developer for 3 years managing data warehousing projects at L&T Infotech Ltd. Mumbai, India. He has two years' experience in using SAS® tools for Database Marketing and Data Mining. He is a Base SAS® 9 certified professional, SAS Certified Advanced Programmer for SAS 9 Credential, SAS Certified Statistical Business Analyst Using SAS 9: Regression and Modeling Credential, and holds the SAS and OSU Data Mining certification.