Domain-Independent Automated Processing of Free-Form Text Data in Telecom

Rajarshi Bhowmik
Department of Computer Science, Rutgers University
Piscataway, New Jersey, USA
rajarshi.bhowmik@rutgers.edu

Ahmet Akyamac
Bell Labs, Nokia
Murray Hill, New Jersey, USA
ahmet.akyamac@nokia-bell-labs.com

***Abstract*—Free-form, unstructured and semi-structured textual data has become increasingly more prevalent in the telecommunications industry, with service and equipment providers alike. Some typical examples include textual data from customer care tickets, machine logs, alarm and alerting systems, and diagnostics. There is a growing business need to rapidly and automatically understand the underlying key topics and categories of this bulk collection of text. With the present mode of operation of relying on domain experts to analyze textual data, there is a clear need to apply text analytics to automate the process. Difficulties arise due to the jargon-filled and fragmented, incomplete nature of textual data in this field. In this paper, we propose a domain-agnostic, unsupervised approach that deploys a multi-stage text processing pipeline for automatically discovering the key topics and categories from free-form text documents. Using anonymized datasets retrieved from actual customer care tickets and system logs, we show that our approach outperforms traditional text mining approaches, and performs comparably to manual categorization tasks that were undertaken by domain experts with full system knowledge.**

***Keywords*-Free-form textual data, key-phrase extraction, text processing pipeline, telecommunications industry**

I. Introduction

The use and application of unstructured and textual data has become quite prevalent in the telecommunications industry, with service and equipment providers alike. Some typical examples include textual free-form, unstructured and semi-structured data from customer care tickets, machine logs, alarm and alerting systems, and diagnostics. There is an important business need to rapidly and automatically understand the underlying key topics and topic categories in these collections of text. This information can help determine if there are specific problem areas, products, or customer issues that need to be addressed in a timely manner. Furthermore, this information can be used in lieu of the complete text corpus, and is often a key enabler of data analytics activities such as automatic machine learning and personalized data exploration. These capabilities in turn open a wide area of applications that increase service and customer quality such as predictive/proactive care, self-healing devices and applications, process work-flow improvement, etc.

Currently, methods that operate on textual data are typically viewed on a scale having two extremes. On one end of the scale, domain experts manually and extensively analyze textual data to generate a variety of particular insights; for example, to uncover systemic issues in network deployment, or to track hidden patterns in diagnostics information to debug problems, etc. This is usually a very time-consuming process, and often results in a very small set of addressed problems given the resource requirements, and can often be prone to errors. On the other end of the scale, natural language processing techniques such as semantic analysis are used to automatically summarize textual data (see references in Section II). With the advent of deep learning techniques, supervised text classification methods (e.g., [1]–[3]) have achieved high accuracy in text categorization. However, these techniques often require an extremely large training set of labeled or annotated textual data to train the models, and are often domain-specific. In the telecommunications and networking-based textual data domains, obtaining large-scale annotated training data is extremely difficult because of the proprietary nature of the data and the diversity of the domains. Additionally, textual data in these domains often contain extraneous, non-standard abbreviations, or synonymized text, and can be very fragmented, lacking correct grammatical structure.

Therefore, a need exists for an improved free-form text processing technique that is domain agnostic and automatically identifies key topics for textual data, and does not require prior training or supervision, pre-labeling or annotation of the data, nor domain expertise.

Towards this end, we introduce a text processing pipeline which is: *domain-independent*: the technique is applicable to a diverse set of free-form text data; *automated*: automatically extracted key phrases and their corresponding scores obtained by topical phrase ranking are used as features for each document, with no requirement of any human intervention (for data cleaning, for example) and/or any domain expertise (for feature engineering, for example); *unsupervised*: all stages in the text processing pipeline are unsupervised and do not require any annotated or labeled training data (thereby eliminating the need to obtain such training data which in many cases is difficult to reliably obtain); leveraging both intra-document and corpus-wide word occurrence statistics; and *improving decision making*: the text processing pipeline is able to decide whether a text document is categorical or not based on the density of the clusters generated by a

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00200

hierarchical clustering of the text documents. In other words, the text processing pipeline assigns a categorical label to a text document if it is a part of a very dense cluster, otherwise, if the cluster density is below a threshold value, the text processing pipeline assigns a topic to the text document which is identified by a set of topical key words. The cluster density is measured by the silhouette coefficient. We apply our method to anonymized datasets of customer care tickets and system logs that were previously analyzed and categorized by domain experts in the telecommunications field, and show that it performs comparably to the lengthy manual processes of categorization.

This paper is organized as follows: In Section II, we discuss some related work. We describe our methodology in Section III. We apply our algorithm to real service provider data in Section IV, and conclude in Section V.

II. Related Work

Prior text processing and categorization techniques can be broadly classified into two categories: rule-based approaches and semantic/linguistic property based approaches.

1) Rule-based Approaches: Such approaches require significant domain knowledge to extract key information from text. For example, a regular expression based pattern extraction technique for log analysis was proposed in [4] where the natural language processing (NLP) employs a natural language toolkit (NLTK) supporting part of speech (POS) pattern matching using regular expressions. In another example, in [5], regular expression based rules are generated to correct the classification of customer care tickets using a combination of traditional text mining techniques fused with input from domain experts. In a further example, in the insurance business domain, there exists an intelligent text analyzer [6] that leverages standard natural language processing tools along with knowledge engineering and machine learning for information extraction from free-form text. However, the extraction module uses if-then rules using pattern matching mechanism. Thus, these rule-based techniques are often domain specific and do not generalize well for other types of free-form text data.

2) Semantic/linguistic Property-based Approaches: These approaches leverage the semantic relations among words and phrases to extract key information. One such approach proposes using a knowledge graph-based framework for processing structured log files using semantic relationships and ontology [7]. However, this method is not suitable for unstructured text data of variable lengths and formats.

3) Key Phrase Extraction: Categorization of unstructured text data depends heavily on the extraction of keywords or phrases that entail identifying the key concept of the text. In this categorization field, automatic key-phrase extraction is an established line of approaches which can be broadly categorized into two classes: supervised and unsupervised. In several known supervised approaches (see [8]–[10]), key phrase annotated datasets are leveraged to train the models that learn to determine whether a candidate phrase is a key phrase or not.

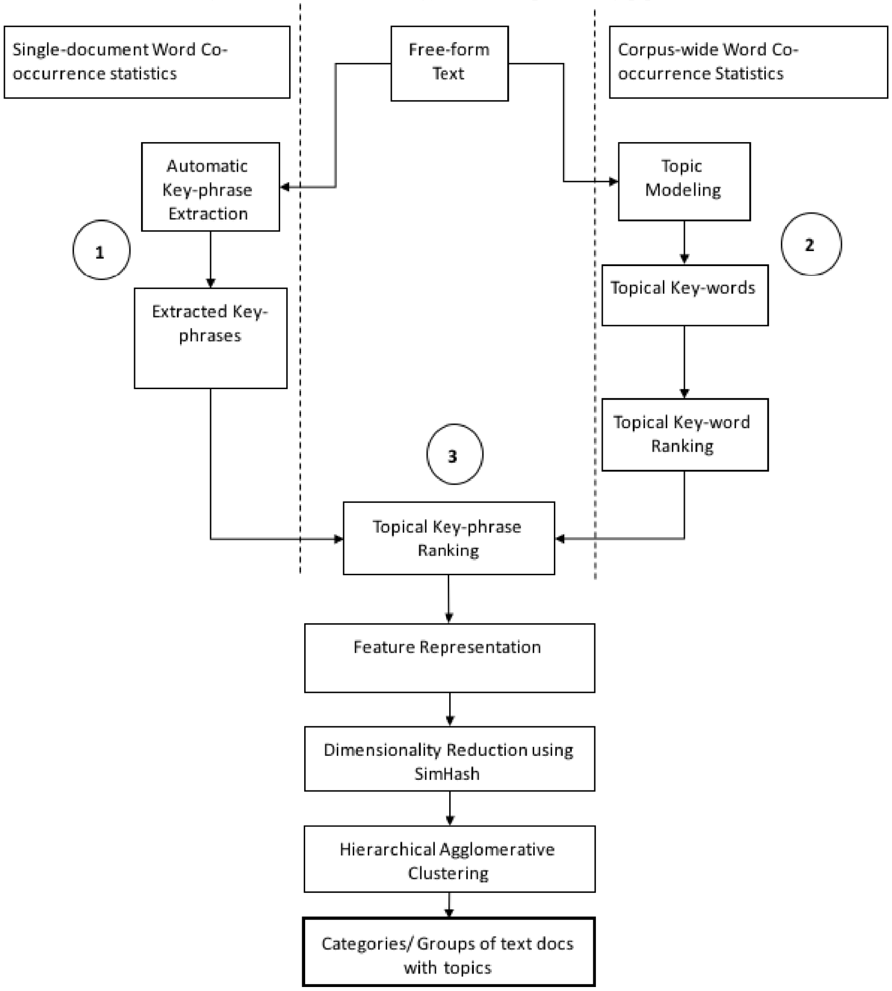
Alternatively, unsupervised approaches for key phrase extraction can be categorized into several groups (see [11] and the references therein). Mihalcea and Tarau [12] introduced TextRank in which a ranking of candidate phrases is obtained based on intra-document word co-occurrence statistics which uses a graph-based ranking algorithm to assign ranking scores to candidate key phrases. Rose et al. [13] proposed another approach which uses some heuristics to generate candidate phrases and assigns ranking scores using degree and frequency of words in the word co-occurrence matrix. In another approach corpus-wide word co-occurrence statistics are leveraged to identify relevant topics for the documents in the corpus and performs topical key phrase extraction [14]. However, these approaches all have their own drawbacks. For example, TextRank uses a fixed size window to generate candidate key phrases, imposing a limit to the length of the key phrases. Also, words with stronger connectivity (i.e. higher degree) are preferred as key words, which may not always be relevant to the main topic of the document. RAKE, on the other hand, tends to assign higher scores to longer candidate key phrases than the shorter ones, even in cases where the shorter ones may be more relevant. Topical Key Phrase Ranking is more suitable for longer text documents where each document is typically a mixture of multiple topics. For short length text document that usually entails a single topic (e.g. a customer care ticket talks about one particular problem, one line of a machine log tells about one particular event), the quality of the extracted key phrases can be very poor. Another drawback of Topical Key Phrase ranking is, lesser represented topics in the corpus are difficult to discover.

III. Methodology

In this section, we introduce our text processing pipeline for improved free-form text processing that is domain agnostic and automatically identifies key topics for textual data which does not require prior training or supervision, prelabeling or annotation of the data, or domain expertise. Figure 1 illustrates a schematic diagram of our text processing pipeline.

The text processing pipeline consists of three phases to convert the free-form text to a feature representation (as illustrated by the markers), followed by processes of dimensionality reduction and categorization. Phase 1 leverages intra-document word co-occurrence statistics to extract candidate key phrases, while the parallel Phase 2 uses corpus-wide word co-occurrence statistics for topical key word extraction and ranking. In Phase 3, topical key phrase ranking is obtained for the extracted candidate key phrases in Phase 1, using the topical key word ranking scores from

Figure 1. Our methodology - the text processing pipeline.


Single-document Word Co-occurrence statistics
Free-form Text
Corpus-wide Word Co-occurrence Statistics
Automatic Key-phrase Extraction
Topic Modeling
1
2
Topical Key-words
Extracted Key-phrases
Topical Key-word Ranking
3
Topical Key-phrase Ranking
Feature Representation
Dimensionality Reduction using SimHash
Hierarchical Agglomerative Clustering
Categories/ Groups of text docs with topics

Phase 2. In the following, we provide further details for each component of the text processing pipeline.

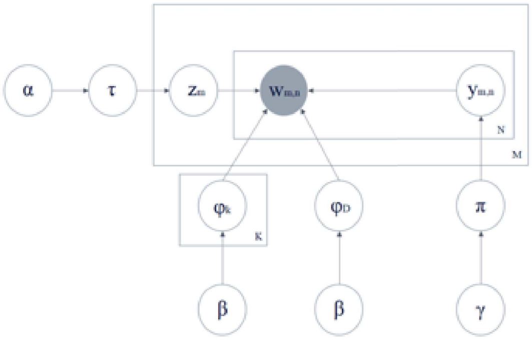
A. Key-Phrase Extraction

The key phrases are small groups of consecutive words standing together as a conceptual unit and define the uniqueness of the content of a textual document. The key phrases play an important role in document categorization. Previous works have identified key phrases in a text document by either considering the intra-document word co-occurrence statistics as in RAKE [13] or used only the corpus-wide word co-occurrence statistics as in TextRank [12]. We propose a combination of intra-document and corpus-wide co-occurrence statistics to identify key phrases of variable length.

1) Candidate Key Phrase Generation.: A single document key phrase extraction technique is applied to identify possible key phrases from each text document that rely on the intra-document word co-occurrence statistics. To elucidate, a combination of common stop words [17], lexical markers, and punctuations is used as phrase delimiters. Using such phrase delimiters, a set of candidate phrases is generated. The candidate key phrases are then split into the constituent words for constructing the phrase-based word co-occurrence matrix. The candidate phrases are then scored and ranked using this single document word co-occurrence matrix. We follow the approach used in RAKE for scoring and ranking the candidate phrases. Similar to RAKE, we use two measures - the frequency and degree of a word to compute the score of a candidate key phrase. The degree of a word is measured as the sum of its frequency in the document and the number of its co-occurrence with other words across all candidate key phrases of the text document. If a candidate key phrase has multiple words, the scores of all the words are summed up. Thus, the extracted candidate phrases can be of variable length. For example, in customer care applications, these candidate phrases might be `service outage`, `technician dispatched`, or `service restart required`, etc. However, as mentioned earlier, RAKE assigns higher scores to the longer candidate key phrases than the shorter ones. Therefore, there is a need to fine-tune these extracted candidate phrases.

2) Refining Candidate Key Phrases.: To fine tune the extracted candidate key phrases, we rely on corpus-wide word co-occurrence statistics. First, we obtain a topic distribution of the documents and a word distribution of topics. For this purpose, Latent Dirichlet Allocation (LDA) [15] could have been our de facto choice. In fact, LDA is a well-known probabilistic generative model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over the underlying set of topics. However, since traditional LDA assumes each document to be a mixture of a few topics, this is not directly applicable to our task as such an assumption does not necessarily hold true in this case. For example, each ticket, or log, or diagnostic can be directed to only one topic. Additionally, there are certain domain-specific jargons which are not associated with a specific topic. In light of these requirements, we proceed to propose a new variant of LDA as shown in Figure 2.

Figure 2. The generative model as a variant of LDA.



Algorithm 1 shows the generative process of our model. Our model distinguishes two types of words in a free-form text document: *domain-specific jargons*, denoted by $\mathcal{J}$ and general vocabulary words, which we also call as *topic words*, denoted by $\mathcal{V}$. We need such discrimination to accommodate domain-specific named entities, abbreviations etc. As shown in the plate notation of our model in Figure 2, the domain-specific jargons are assumed to have a Dirichlet prior ϕ_D parameterized by β, the topic distribution τ is assumed to have a Dirichlet prior parameterized by α, and π is the Dirichlet prior with parameter γ for choosing between topic words and domain-specific jargons. For each topic k, a topic word distribution is assumed with a Dirichlet prior ϕ_k parameterized by β. z_{d_m} denotes the topic of the m-th text document in the corpus and $y_{m,n}$ is a binary variable indicating whether the n-th word of the m-th document is a topic word or a domain-specific jargon. If the value of the random variable $y_{m,n}$ is 0, then the word $w_{m,n}$ is identified to be a domain-specific jargon and hence, is sampled from the Categorical distribution of domain-specific jargons with the Dirichlet prior ϕ_D . On the other hand, if the value of the random variable $y_{m,n}$ is 1, then the word $w_{m,n}$ is identified to be a topical word for the previously chosen topic of the document and is sampled from the Categorical distribution of the chosen topic with the Dirichlet prior ϕ_k.

The final outcome of this process is a topical probability distribution of general vocabulary words i.e. the probability of a word belonging to a particular topic which will be used for obtaining the topical keyword ranking and thereafter, for the topical key phrase ranking.

Algorithm 1 The Generative process for the text corpus.

```
procedure GENERATE CORPUS
    choose $\phi_D \sim \text{Dir}(\beta)$, $\pi \sim \text{Dir}(\gamma)$, $\tau \sim \text{Dir}(\alpha)$
    for each topic $k \in K$ do
        choose $\phi_k \sim \text{Dir}(\beta)$
    end for
    for each document $d_m$ do
        choose a topic $z_{d_m} \sim \text{Dir}(\tau)$
        for each word $w_{m,n} \in d_m$ do
            choose $y_{m,n} \sim \text{Bernoulli}(\pi)$
            if $y_{m,n}$ is 0 then
                choose $w_{m,n} \sim \text{Categorical}(\phi_D)$
            else
                choose $w_{m,n} \sim \text{Categorical}(\phi_{z_{d_m}})$
            end if
        end for
    end for
end procedure
```

3) Topical Key Word Ranking.: After obtaining the topical key words, we proceed to obtain a topical ranking for these key words. Assigning topical ranking scores to words bolster the alignment of words to topics. To this end, we deploy a PageRank-style algorithm similar to Liu et al. [14] to obtain the topical key word ranking.

Formally, for each word $w_i \in \mathcal{V}$, the topical key word ranking score for topic z is defined as

$$S_z(w_i) = \lambda \sum_{j:w_j \to w_i} \frac{e_z(w_j, w_i)}{O_z(w_j)} S_z(w_j) + (1 - \lambda) P_z(w_i) \tag{1}$$

such that $\sum_{w_i \in \mathcal{V}} P_z(w_i) = 1$ and $O_z(w_j) = \sum_{j:w_j \to w_i} e_z(w_j, w_i)$. P_z is the probability distribution of words for topic z and $e_z(w_j, w_i)$ is defined as the number of times the words w_i and w_j co-appear in a text document that belongs to topic z. λ is the damping factor ranging between 0 and 1.

4) Topical Key Phrase Ranking: For each phrase $p \in \mathcal{P}$, the set of all candidate key phrases extracted using RAKE, we obtain the topical phrase score $R_z(p)$ by the product of the candidate key phrase score $S(p)$ obtained in Phase 1 and the sum of the topical key word ranking scores of the constituent words of the phrase p. Formally,

$$R_z(p) = S(p) \sum_{w_i \in p} S_z(w_i) \tag{2}$$

Finally, we rank a phrase p of the m-th document d_m as

$$R(p) = \sum_{z=1}^{K} R_z(p) Pr(z|d_m) \tag{3}$$

where $Pr(z|d_m)$ is the probability that the document d_m belongs to topic z.

B. Feature Representation

After scoring and ranking the key phrases, we represent each text document in the corpus as a bag-of-phrases in the $|\mathcal{P}|$ dimensional feature space, where $\mathcal{P}$ is the set of all key phrases across all the documents in the corpus. The feature values are assigned to the scores obtained in the key phrase ranking task. Thus, each document is represented as a very sparse $|\mathcal{P}|$ dimensional vector.

C. Dimensionality Reduction

Since our goal is to cluster the text documents into distinct categories and each document is now represented in a high dimensional feature space, it is beneficial to perform dimensionality reduction in order to avoid the so called "curse of dimensionality" and facilitate the downstream application of clustering algorithms. To this end, we find the *SimHash* algorithm [16] to be a good fit for our requirement. Our choice is motivated by the fact that SimHash not only reduces the dimensionality of the feature space drastically but also puts the near-similar documents close to one another in the low-dimensional vector space. Note that the reduced vector representations are binary vectors.

D. Categorization

Finally, we apply the standard agglomerative hierarchical clustering technique with hamming distance as the distance measure to cluster similar text documents together. A cut-off distance, which is given as an input hyper-parameter, decides the number of clusters $|C|$. Once the documents are clustered, we define the cluster density $dense(c) : c \in C$ as the average silhouette coefficient of each document point within the cluster. If the cluster density is above a predefined threshold, the text documents are annotated to belong to that cluster, and the cluster is considered to be categorical. Otherwise, the cluster is not considered categorical. Instead, the top-k most frequent phrases from all the documents within the cluster are selected as representatives of the topic for the cluster, and the text documents are annotated to be part of this topic.

IV. Evaluation and Discussion

In this section, we report on the experimental results for two different datasets to demonstrate our model's ability to handle free-form and semi-structured textual data originating from heterogeneous domains. We compare the performance of our methodology to available baselines for the two datasets - human-annotated ground truth, and a bag-of-words based clustering model with domain expert categorization, respectively.

A. Datasets

We evaluate our model on two different types of datasets representing completely different domains in the telecommunications industry. The data was sourced from two real datasets from service providers that are anonymized to remove personally identifiable and confidential information and references. In both of these domains, it is important to understand if the free-form textual data can be separated into categories, and achieving this in an automated way is a significant improvement in terms of cost and processing time compared to having domain experts manually process the text.

The first dataset comes from the customer experience management domain, and consists of a collection of 100,000 machine-generated system logs, or syslogs. These syslogs were generated over a period of 6 days from home routers deployed at approximately 400 customer locations. Home routers typically consist of multiple sub-systems potentially manufactured by different vendors. There are multiple software processes running on the devices, performing tasks related to networking and resource allocation (memory, CPU, etc.) Most of the sub-systems and sub-processes record syslogs which, when collected together, appear as semi-structured or unstructured text as there is usually no established standard by which the logs are recorded. For this particular dataset, a domain expert with full knowledge of the sub-systems and corresponding syslog generation keywords was able to annotate the set of vendor logs into categories such as `DEBUG`, `ERROR`, `ALERT`, etc. We took the annotation of the expert as the ground truth. In general, the domain expert identified 8 different categories of syslogs, with one category being a very small one consisting of only 10 syslogs. Table 1 contains some examples of these machine-generated syslogs and their corresponding domain expert assigned categories.

The second dataset comes from the customer care domain, and consists of a collection of 195 trouble tickets at emergency and high priority severity levels raised by customers for network related issues, and subsequently resolved by engineers with appropriate domain knowledge. In addition to free form text descriptions of the problems, these tickets incorporate a total of approximately 10,000 free-form text analyses recorded by engineers (ranging from tickets with 1 analysis to tickets with 170 analyses). Thus, each ticket

Table I
SAMPLES FROM THE SYSLOG DATASET.

Log Text	**Category**
user part debug System App Took Too Long With Output rest_api ref. 24 took a long time 115 ms	DEBUG
local part error named zone localhost in getting clients: loading from master file localhost for home zone failed: file not found	ERROR
ssm part tr96 device alert cpu usage avg Alert: cpu util avg exceeded threshold u1=50: value=54.1667	ALERT

contains not only the free-form textual description of the issues raised by the customers, but also the comments and suggestions of the engineers. For this dataset, we had the results of a previous clustering study performed using bag-of-words approaches, followed by analysis and selection of the clusters by a domain expert. We show sample text from a short (few recorded analyses) and medium (moderate number of recorded analyses) ticket in Table II. For this set of tickets, analysis by the domain expert determined that the tickets belonged to 4 categories (representing high level categories), or 8 categories (representing high level categories, in addition to a first level of sub-categories).

B. Experimental Setup

For extracting the candidate key phrases using phrase delimiters, we use the standard punctuations and the freely available stop word list available in [17].

We executed our topic modeling algorithm for 100 iterations with the following hyper-parameters: $\alpha = 0.05, \beta = 0.5, \gamma = 1.0$ (described in Section III) and a default number of topics. For inference, we used a standard Markov Chain Monte Carlo (MCMC) approximation method called Gibbs Sampling.

For obtaining the topical keyword ranking, we ran our PageRank-style algorithm for 100 iterations which, in our case, turned out to be sufficient.

We specified different cut-off distances for obtaining the clusters. For the syslogs dataset and the customer care ticket dataset, we set the cut-off distance to be 0.37 and 0.2, respectively. However, the end user is free to select a different cut-off distance resulting in a different number of clusters, if necessary. Note that if the data in the reduced dimension is saved after processing, then the pipeline need not be re-started, and the user can directly experiment with different numbers of clusters.

C. Experimental Results

We first present our experimental results for the syslogs dataset. As shown by the dendrogram in Figure 3, our text processing pipeline identified 9 distinguishable clusters as depicted by the different color codes. We measured the performance of our model using the *accuracy* score as the evaluation metric. Note that these metrics are typically used for the evaluation of supervised learning methods. However, since our syslogs dataset has human annotated labels as the ground truth and correspondence between the generated clusters and the annotated labels can be identified by manual exploration, we chose to use accuracy as our metric.

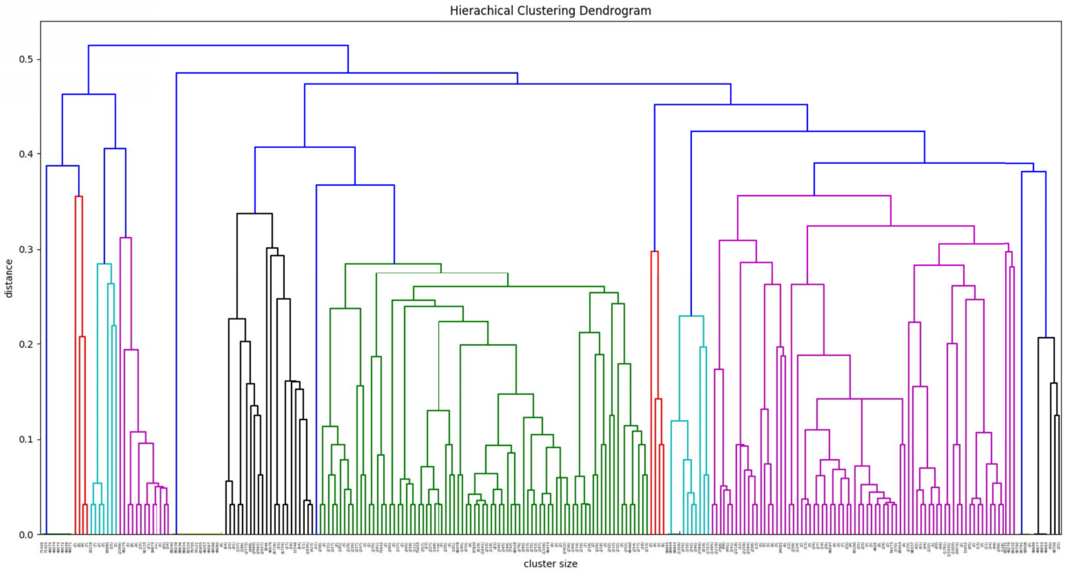
Table III shows the cluster ID and the corresponding categories as assigned by our expert analysis of the produced clusters when compared to the original syslog categories annotated by the original domain expert. For each correspondence group, the number of syslogs accurately categorized vs. mis-categorized is shown, along with the accuracy for each group. There are some interesting observations in Table III: the ALERT and STATUS categories were covered by two clusters, though the clusters did not fully split along the lines of the annotated labels (analysis showed that these two categories came from the same subsystem); the small category of 10 syslogs was not captured as a separate cluster, but included in cluster 7; another small category, WARNING, was not accurately captured due to textual similarities to the ERROR category; the INFO and ERROR categories were covered by more than one cluster, suggesting that there are potentially some well-defined sub-categories. Overall, our model produced results with 81.93% accuracy compared to the manual categorization by the domain expert with full domain knowledge.

For our second dataset of customer care tickets, we compare our method with previously applied bag-of-words models that used *cosine normalization* followed by *Euclidean distance* calculations for the clustering to account for the volume difference between the engineers' analysis text between the tickets (recall that each ticket had between 1 and 170 different analyses recorded). Two approaches were previously applied: a standard text mining approach that included the removal of stop words [17], punctuation and numbers, as well as stemming and stem completion of words; and a domain knowledge enhanced approach that involved additional trimming of the text to remove extraneous words, markers and unnecessary information such as engineer names, e-mail addresses, locations, etc. The domain knowledge also included handling of technical synonyms. Note that we did not use any domain knowledge or preprocessing while running our algorithm on the same data set. Our dimensionality reduction phase produced a binary feature vector of 32 dimensions and hence, we resorted to

Table II
SAMPLES FROM THE TROUBLE TICKET DATASET.

Ticket Length	**Sample Ticket Text**
Short	Failure in connection after replacement of last test modules to system located in park terrace. Operations Support Systems.Network and Service Management were impacted.
Medium	File system crash. During integration with system; file system crashed. Operations Support Systems.Network and Service Management.Problem and Impact Description: File systems crashed after integrating component with system. Total outage. Actions Taken and Final Solution: The local team was trying to integrate storage array with system. During the activity there was a step to restart the system. Something went wrong with restart and that corrupted the system ...(more)...

Figure 3. Hierarchical clustering dendrogram of syslogs.



hamming distance as a distance measure. We experimented for a range of clusters from $k = 4$ to 10, to compare against the domain expert's categorization. We calculated the average *silhouette coefficient* for the data points for each of the above number of clusters, and compared the metrics generated by the two bag-of-words models and our algorithm. We report the comparative results in Table IV, where the cluster sizes of $k = 4$ and 8 are specially marked with square brackets to represent the domain expert's selections of main topics and sub-topics, respectively.

As can be observed by the silhouette metrics in Table IV, this is a particularly challenging dataset to effectively cluster. This is due to the large volume of text in a small number of high priority tickets, in addition to the widely varying amount of text between the tickets. The infusion of domain knowledge appears to have had a positive influence on the bag-of-words model, especially for the number of clusters selected by the domain expert where it showed a 17% to 22% improvement. Our method, which does not rely on any domain expertise, was able to produce significantly better results than either of the bag-of-words models. As seen in Table IV, for the number of clusters selected by the domain expert, our method showed a 76% to 175% further improvement, though the absolute silhouette metrics still point to the challenges in the data. The hierarchical clustering using our method is shown in Figure 4. Based on our method, we selected 10 clusters, which also account for largest performance improvement over the bag-of-words models (marked with an asterisk in Table IV).

V. CONCLUSION

Extracting insights from textual data is an increasingly important need in the telecommunications industry, particularly for service and equipment providers that have to operate and

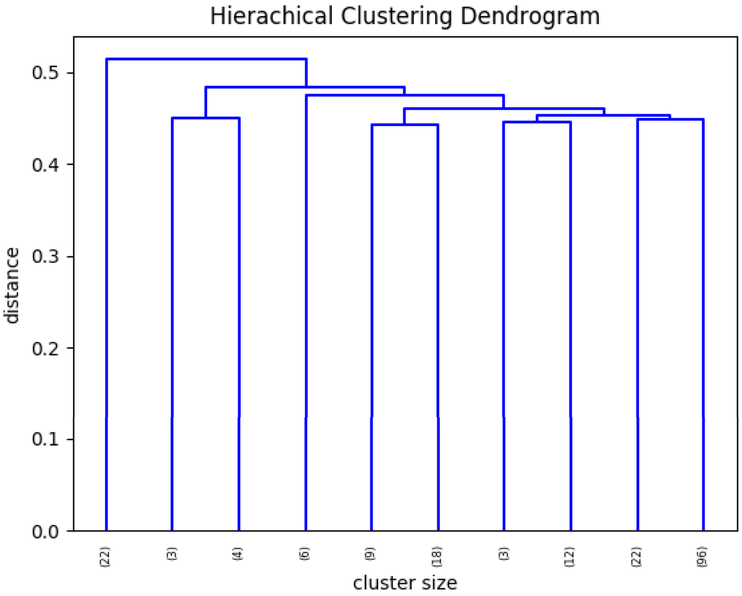
Table III
CLUSTER ASSIGNMENT COMPARED TO HUMAN ANNOTATED SYSLOGS.

Cluster ID	Annotated Label	Accurate	Inaccurate	Accuracy%
1, 2	ALERT, STATUS	34,375	10,965	75.82%
5, 8, 9	INFO	26,578	1,063	96.15%
7	NOTICE, DEBUG	15,210	4,038	79.02%
2, 3	ERROR	5,582	1,819	75.42%
4	WARNING	181	189	48.92%
Overall		81,926	18,074	**81.93%**

Table IV
AVERAGE SILHOUETTE COEFFICIENTS OF CLUSTERS.

Method	$[k=4]$	$k=5$	$k=6$	$k=7$	$[k=8]$	$k=9$	$k=10*$
Bag-of-words w.o. domain knowledge	0.035	0.031	0.030	0.030	0.028	0.034	0.036
Bag-of-words with domain knowledge	0.043	0.046	0.049	0.034	0.033	0.035	0.035
Our method	0.076	0.078	0.081	0.092	0.091	0.093	0.100

Figure 4. Hierarchical clustering dendrogram of customer care tickets.



take actions based on text from many diverse sources. The present approach of domain experts analyzing the textual data is costly and error-prone, and application of traditional text analytics methods is difficult due to the characteristics of text appearing in telecommunications data, as well as the frequent unavailability of appropriately labeled training sets. In this paper, we proposed a text processing pipeline that is domain agnostic, automated and unsupervised, to categorize free-form text data without requiring any prior domain knowledge, training or pre-processing. As such, our method is rapidly applicable to a diverse range of textual data found in the telecommunications field. We applied our method to actual anonymized service provider data sets, and demonstrated our method's ability to find interesting patterns in both semi-structured and unstructured textual data. Benchmarking against domain expert analysis of this data, we showed that our method can achieve performance that is comparable to domain expert categorization with full knowledge (for a syslog dataset), or much improved compared to traditional text mining approaches with domain expert marking (for a customer care ticket dataset).

REFERENCES

[1] Kim, Yoon 2014, ”Convolutional Neural Networks for Sentence Classification.”, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 17461751

[2] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 1. MIT Press, Cambridge, MA, USA, 649-657.

[3] Siwei Lai, Liheng Xu, Kang Liu and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In Proc. Conference of the Association for the Advancement of Artificial Intelligence (AAAI) 2015.

[4] Carasso, David: White paper available on the Splunk, Inc. website www.splunk.com (2007)

[5] Ahmet Akyamac, Chitra Phadke, Dan Kushnir and Huseyin Uzunalioglu, Predicting Home Network Problems Using Diverse Data, Proceedings of the 36th IEEE Sarnoff Symposium, Newark, NJ, Sept 2015.

[6] Barry Glasgow and Alan Mandell and Dan Binney and Lila Ghemri and David Fisher: MITA: An Information-Extraction Approach to the Analysis of Free-Form Text in Life Insurance Applications, AI Magazine 19:59 (1998), doi:10.1609/aimag.v19i1.1354

[7] Nimbalkar, P.,Mulwad, V., Puranik, N., Joshi, A., and Finin, T.: Semantic Interpretation of Structured Log Files, 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), IEEE 549-555

[8] Riloff, Ellen, and Lehnert, Wendy: Information extraction as a basis for high-precision text classification, ACM Transactions on Information Systems (TOIS) 296-333 (1994)

[9] Turney, Peter D.: Learning algorithms for keyphrase extraction. Information Retrieval (Springer) 303-336 (2000)

[10] Hulth, Anette: Improved Automatic Keyword Extraction Given More Linguistic Knowledge, Proceedings of the 2003 conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 216-223 (2003)

[11] Hasan, Kazi Saidul, and Vincent Ng.: Automatic Keyphrase Extraction: A Survey of the State of the Art, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland: Association for Computational Linguistics. 1262-1273 (2014)

[12] Mihalcea, Rada, and Paul Tarau: TextRank: Bringing Order into Text, EMNLP, 404-411 (2004)

[13] Rose, Stuart and Engel, Dave and Cramer, Nick and Cowley, Wendy. (2010). Automatic Keyword Extraction from Individual Documents. Text Mining: Applications and Theory. 1 - 20. 10.1002/9780470689646.ch1.

[14] Liu, Zhiyuan, Wenyi Huang, Yabin Zheng, and Maosong Sun, 2010, Automatic Key Phrase Extraction via Topic Decomposition, Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 366-376

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (March 2003), 993-1022.

[16] Charikar, Moses S. 2002, Similarity Estimation Techniques from Rounding Algorithms., Proceedings of the 34th Annual ACM Symposium on Theory of Computing, ACM, 380-388

[17] http://www.lextek.com/manuals/onix/stopwords2.html