Insights Exploration of Structured and Unstructured Data and Construction of Automated Knowledge Banks

Arvind Maurya, Yogesh Gupta, and Stuti Awasthi

***Abstract*—Enterprise data is in abundance in form of knowledge articles, forums, blogs and open internet. However, this data has not been tapped effectively to bring out real and differentiated values to help enterprises as well as their customers. In this paper we have described how contextual text mining can drastically improve productivity of support engineers and also enable customer to do self-resolution of commonly occurring problems. Average resolution time for user problems can range between few hours to more than a week depending on ease of availability of relevant information and knowledge of engineer handling the problem. In order to significantly reduce the response time, we approached it through automating the construction of knowledge banks based on multiple contexts present in a single source. Knowledge identification and extraction are two separate solution arcs and information flows from one arc to another to build an optimal solution using both supervised and unsupervised learning techniques. We applied this solution for network division of a technology company and the experiment demonstrated reduction in response time and thereby productivity gains for support engineers by 30% over a period of 3 months.**

***Index Terms*—Ngrams, stemming, feature extraction, stop words elimination, vector space model, naïve bayes, cosine similarity measure, canopy clustering, kmeans clustering, hadoop, mahout, mapreduce.**

I. Introduction

Today lot of thrust is there for analytics and mining information as much as possible from available data sources.

Let's take one such business problem. There is this product company A, who manufactures network products. It is very difficult, if not impossible, for the company to test the product for all the possible scenarios with all possible configurations. Company has setup internal and external forums and also listen to the information available over internet. Current SLA for resolving the issue varies from 5-15 days depending on nature of the problem and average productivity of engineer. However, in this fast changing and connected world, these

SLAs on resolution time seems to be on higher side and ends up affecting user's experience and customer satisfaction.

We tried to analyze the problem to identify where major bottleneck is. It was found that today when a problem comes to support engineer, resolution times varies depending on his/her own subject matter expertise and experience of resolving similar issue, ability to frame and write search query to utilize existing data and also kind of search engine in place to help support engineer.

So solution of the problem has to address two aspects. First, making solution to be free of human capabilities as much as possible and secondly enable context based mining of the data and make it available in a form which not only can be consumed by support engineer but also by end user as well e.g. (Fig. 1).

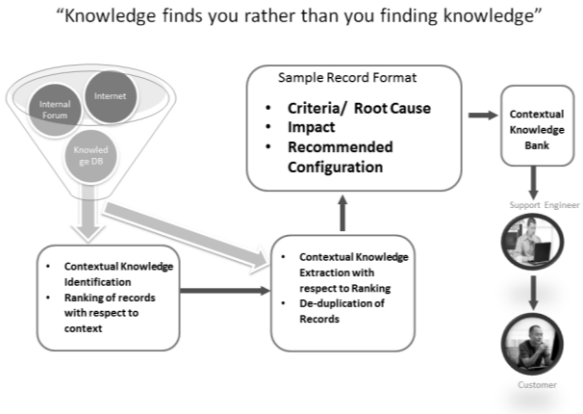

Fig. 1. Knowledge bank creation and flow of information.

The approach and solution has to be generic and should be usable across different domains. The information needs to be processed to produce knowledgeable contents stacked against problems and problems categories. Depending upon how generic the problem is, these knowledge documents can be also be supplied to customers who are yet to discover the problem but are susceptible to that and hence increasing customer satisfaction and cutting on support calls. For example healthcare domain can benefit immensely by adopting this type of solution as it deals with huge amount of unstructured data related with analysis of description of an ailment, its symptoms and recommended prescription. Knowledge banks produced through this solution can help in improving the time of cure for patient's by helping medical practitioners with upfront information about disease and what treatment helped earlier.

II. Related Work

There exists a quiet rich literature on methods and devices for the analyzing unstructured text data. Our problem is to identify, extract and group best practices related to diverse criteria or topics and suggest the best cluster which is relevant for the user. We have found various works possess few similarities with respect to some of the techniques commonly used in text mining but not following the same approach.

We found that, document clustering for search engine is

Manuscript received January 20, 2016; revised March 9, 2016. This work was supported in full by the HCL Technologies Limited.

The authors are with HCL Technologies Ltd, A-8 and 9, Sector-60, Noida - 201301, UP, India (e-mail: maurya-a@hcl.com, yogeshg@hcl.com, stutiawasthi@hcl.com).

one of the most common use case which requires similar documents to be clustered e.g. [1] build the dictionary based on keywords from entire text of document and analyze documents based on the number of occurrence of keywords in the documents. Documents are then clustered based on similar word or phrases. In contrast, our solution focus on extracting relevant information from multiple data sources at paragraph/statement level based on domain specific words and weightage techniques like CPD to categorize them and cluster.

There are experiments which are performed by extracting important phrases from the documents, ranking and clustering the documents based on phrases as performed in [2], [3] where as we have created features using BoW approach for categorization and clustering of the best practices only instead of the complete document. Some experiments were done to perform multi document summarization using centroid based clustering techniques e.g. [4]. In contrast, we have not performed any summarization on the clustered output. There are also several works on information extraction using Ontology as used in [5]. Here the information extraction of artists is performed and stored in a knowledge base which is then used to generate biographies of the artists. In our work, we have not used ontologies to extract the information rather we have used NLP techniques like POS tagging to extract the important features. Our use case required to suggest the best practices group of corresponding topics based on the issues faced by users and resolution provided to the user.

Many authors worked on Topic Modelling to identify the topics e.g. [6] where linguistic analysis is performed to extract and identify topics present in the question of interest. In our solution we used the domain specific keywords along with Canopy and K-Means clustering techniques to group the paragraph/statement for certain topics criteria's. One of the works done in [7] is also relevant in similar scenario where question answer pairs are extracted from the public forums. The authors have used classification techniques to identify the questions and graph based unsupervised techniques to link the answers to the questions. Our approach is different from this work as our solution will work on any data source and not only forums and we applied classification for categorizing the text under labels like "Description", "Impact" and "Recommendation" and further clustering them to provide the best group which may contain the description to the problem, its impact and recommendations to resolve such problems.

III. Proposed Approaches

We came up with text analytics based solution to identify and extract the knowledge based on *domain/product specific Bag of Words (BoW).* Some of the bag of words are obtained from domain expert and others are identified by applying word weightage scheme as Categorical Proportional Difference (CPD). These BoWs enables context based mining of the data (knowledge identification using Inverted Index) as well as classifying mined data in context of problem. Relevant information from classified data is then extracted from the documents by using the BoW and classified as Description, Impact or Recommendation (knowledge Extraction using Naïve Bayes algorithm). This format can be specified based on domain and form in which knowledge is required. As multiple contexts can prevail in the same information data source, hence we applied clustering using Canopy and K-Means to create groups of Best Practices for these multiple contexts.

Solution consists of set of steps (Fig. 2) and on each step algorithms (Fig. 3) have been applied to get optimum output which is suitable for further processing on advanced steps. Workflow given below explains on how the proposed system will work starting from support engineer defining problem statement to training of the model and creation of knowledge banks.

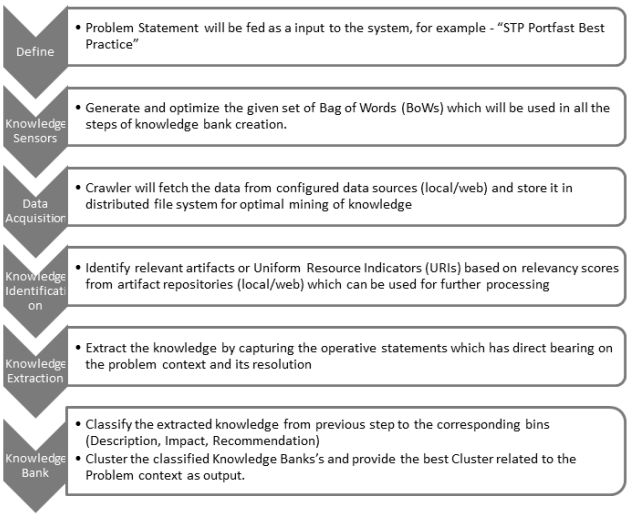

Fig. 2. Process flow of approach.

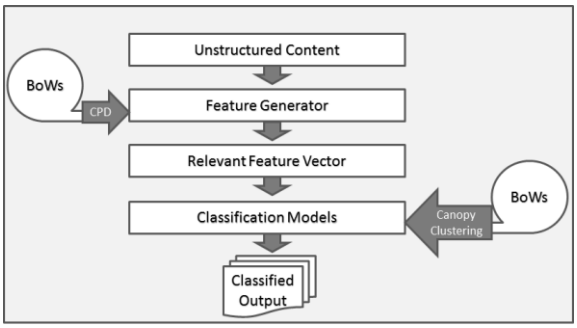

Fig. 3. Knowledge extraction approach.

IV. Methods Description

A. Content Crawling and Knowledge Identification

Data acquisition starts with populating a queue with a list of base URIs. A crawler component will be used for accessing (local) or downloading (web) artifacts repositories. Any references whether local or web will also be crawled based on defined settings. Relevant content from the crawled artifacts will be extracted using regular expressions and will be written to the distributed storage for further processing.

For crawled data, every crawled URI will be given a weighted score (WS) based on world frequencies of labels (Description, Impact and Recommendation).

Word frequencies are calculate for a single label as follows.

LC = Unique Occurrence of all the words in a Label for a single URI

TLC = Unique occurrence of all the words in a Label in all URIs

TLF = Total occurrence of all the words in a Label in all URIs

$$LWS = LC \times (TLC/TLF)$$

Weighted score for a single URI will be:

$$WS = \sum_{i=1}^{i=N} LWS_i$$

where N is total number of crawled URIs.

B. Feature Extraction Using CPD Word Weighing Scheme

We have used the BoW approach to identify the relevant paragraphs/sentences for the problem statement. Some of the bag of words are provided by the SMEs, but SMEs are not expected to provide all the possible BoWs. So we generated custom dictionary using traditional Tf-Idf approach which gave us around only 40% of accuracy in extracting the useful paragraph/sentences. We found out that due to similarity in domain based words, the weightages provided by Tf-Idf were not able to distinctly identify word and labels associations. For example the words like "bandwidth", "issue" etc. may occur in any of the 3 labels (Description, Impact and Recommendations) which make it difficult for the feature extraction to identify relevant records and also for the classification engine to model well. To improve feature extraction accuracy, we implemented CPD (Categorical Proportional Difference) as a word weightage scheme.

CPD (Categorical Proportional Difference)

CPD [8] value of a term is computed by finding the ratio of the difference between the number of documents of a category in which it appears and the number of documents in which it appears of another category, to the total number of documents in which that term appears. CPD value for a feature can be calculated by using equation

$$\mathbf{cpd} = \frac{|posD - negD|}{posD + negD} \quad (1)$$

Here, posD is the number of positive review document in which a term appears, and negD is the number of negative review documents in which that term appear. Range of CPD value is 0 to1. If CPD value of a feature is close to 1 it means that this feature is occurring dominantly in only one category of documents.

In our approach, we applied the TF (term frequency approach) to collect the raw list of eligible BoW after applying preprocessing NLP techniques stop word removal, stemming and lemmatization. After generating the eligible terms as features, we generated the CPD value for the custom BoW. The CPD values ranged from 0 to 1 based on frequency of term appearing in correct label vs term appearing in incorrect label. We analyzed the results and filtered only those word with CPD value as 1 in order to identify those BoW which can clearly distinguish between various labels. By applying this approach, we have enhance d the confidence of reliable words to be considered as features in Feature Extraction Phase. Implementation of CPD improved the Feature extraction accuracy to 78.67% which is substantial improvement of 38.67% as compared to TF-IDF approach.

C. Classification

Once the features are extracted and relevant statements/paragraphs are filtered, next step is to categorize them into 3 labels namely Description, Impact and Recommendation. We have used Naïve Bayes [9] as a multiclass classification algorithm, which considers the probability of occurrence of words in a document in modeling. We have used Naïve Bayes implementation of Mahout to build classification model. We also noticed that Classification accuracy improved by 20% when we considered CPD as weightage scheme as compared to Tf-Idf weighting scheme.

D. Clustering

The next stage after classification is grouping the categorized data in clusters and producing best cluster as knowledge bank artifact in form of best practice or problem solution. As one of the approach we applied K-means clustering on the vectorized and classified records and found that generated clusters were not good in the context of problem. We identified the cause as poor quality of Bag of Words and came to the conclusion that BoW used for classification will not work for the clustering technique. The reason being classification BoW carry all important words specific to a domain and though the same words may occur in different contexts but a clustering algorithm may group them together. This will cause the groups to contain different context in same cluster which is not required.

We tried to modify BoW using only few domain related words and using LDA (Latent Dirichlet Algorithm) to identify the topics at a granular level and group records based on topics. Topics generated from this approach also didn't suffice the requirement due to presence of common BoW from same domain, the quality of the topics were also mixed. For Example, LDA works well if we have records from say Sports and Politics as they belong to different domain but in case of LAN Switching domain Topics like "Port", "LAN" are too granular to be distinguished by the algorithm due to similarity in context.

We then approached to get the specific BoW based on problem context and again vectorizing the classified content and apply Canopy clustering [10] to get the number of clusters that should be created for the problem. The canopy clustering is fast approximate clustering technique which tries to estimate the approximate cluster centroids with distance threshold. For our experiment, we provided thresold1 ($t1$) = 0.2 and threshold2 ($t2$) =0.5 as input parameters. The output of the Canopy clustering is the cluster centroids points, which are provided to the K-means clustering as "K". As new vectorized records are very specific to problem context, the clustering output provided by K-means generated more relevant results as clusters. We observed that the first cluster contains the most relevant group of records corresponding to problem context and thus we finalized the 1st cluster as best practice artifact or solution of the problem.

This approach helped us in identifying the best cluster of relevant records based on problem definition. We extracted relevant information and discarded irrelevant information in order to provide the good clusters of information to knowledge bank.

V. Experimental Setup

We used 3 data sources (Open Internet, Internal Forum and Internal relational database containing e-mail and other communication records) from network division of the technology company as input to create and test our solution. We collected the data as follows:

- *Bag of Words (dictionary) for Classification and Clustering*: We collected few important bag of words for Clustering and Classification from subject matter expert. The additional important bag of words for the classification stage is derived using weighting scheme CPD. Final dictionary consists of both SME provided and custom generated BoW. The dictionary is generated in Sequence file format where key is the <Word> and value is the combination of <WordID> and <Word_Weight>

Key: report: Value: 4310,0.5
Key: past: Value: 4305,0.2
Key: observ: Value: 4304,1.0
Key: push: Value: 4307,1.0
Key: error: Value: 4306,0.31

- *Train Data (Labeled Data for Classification):* We collected the labeled data for Classification into Description, Impact and Recommendation labels from SME. Train data format:

/<Actual_Label>/<Unique_Record_Id>\t<Raw Paragraph Text>

- *Test Data:* Our solution crawled the data paragraph wise and stored the test data which contains text along with URI information in HDFS. Test data format:

/<Datasource_Uri>/<Unique_Record_ID>\t<Raw Paragraph Text>

Following the approaches described in section 3, analytical model was created to produce clusters. In order to test the model, test data from the identified sources was then passed to the model to extract knowledge contents and evaluated by domain experts and support engineers for accuracy and relevance.

VI. Implementation

We have implemented our solution based on Big Data framework Apache Hadoop©. We have used HDFS as a filesystem to store the input and output data.

For knowledge identification phase, custom map reduce jobs have been implemented which works on Bag of Words and Lables and as output produces count weiged score as per formula given in Section III.A. Top 20% URIs are then passed to knowldege extraction phase.

Apache Mahout©[11] is used as machine learning library on top over Hadoop and standard implementation of Naïve Bayes, K-means and Canopy clustering is used from Mahout library. We have implemented custom map reduce jobs to execute various phases of the solution.

The raw text is converted to Vector Space model by using Mahout's vector writable format in sequence file. The vectors contained the combination of <WordID>:<Weightage> for further processing as

Key: /Recommendation/99: Value: {4245:1.0,4107:0.94,2218:1.0,2527:1.0,4076:0.26,2526:1.0,839:1.0,2793:1.0,2792:1.0}

The standard Naïve Bayes implementation of Mahout is used by passing the vectorized text for both train and test data.

*$MAHOUT_HOME/bin/mahout **trainnb** -i ${train_vector_path} -el -o ${model_path} -li ${labelindex_path} -ow $c*

*$MAHOUT_HOME/bin/mahout **testnb** -i ${test_vector_path} -m ${model_path} -l ${labelindex_path} -ow -o ${output_path} $c*

Canopy clustering mahout's implementation is used to identify the number of clusters whose centroid points will be passed to the K-means to generate clustered output. The Cosine distance measure is used for similarity metric for the clustering

$MAHOUT_HOME/bin/mahout ***canopy*** *-i ${classified_testvector_path} -o ${output_path} -dm org.apache.mahout.common.distance.CosineDistanceMeasure -t1 ${threshold1} -t2 {threshold2} -ow -cl*

*ut **canopy** -i ${classified_testvector_path} -o ${output_path} -dm org.apache.mahout.common.distance.CosineDistanceMeasure -t1 ${threshold1} -t2 {threshold2} -ow -cl*

K-Means clustering mahout's implementation is used to generate final clusters but passing centroid points obtained from canopy clustering along with max iterations and centroid convergence thresholds.

$MAHOUT_HOME/bin/mahout kmeans -i ${classified_testvector_path} -c ${canopy_centroidpoints_path} -o ${kmeans_output_path} -dm org.apache.mahout.common.distance.CosineDistanceMeasure -cd ${convergence_threshold} -x ${max_iterations} -cl -ow

The output generated from the K-means clustering is dumped by using ClusterDump utility of mahout.

*$MAHOUT_HOME/bin/mahout **clusterdump** -dt sequencefile -i ${ kmeans_output_path} -d ${dictionary_path} -o ${clusterdump_output_path} -p ${clusteredpoint_dir_path} -dm org.apache.mahout.common.distance.CosineDistanceMeasure -b 100 -n 10 -of CSV*

Final cluster output was created by processing the cluster dump and storing final output of cluster in the below format <ClusterID>

/<Datasource_*url>/<Label><Text>*

VII. Results

We performed our experiment in "LAN Switching" domain

using 3 data sources namely (Open Internet, External Forums and Internal Relational Database) for network division of the technology company. The training data set consisted of only 1860 records with 620 records for each of the 3 labels. The test data consists of 70020 records. We executed our results on the test data and validated the results from the SME. Following are the results for each phase and overall turnaround time improvement by the solution.

Feature Generation: We have used accuracy as a performance metric. The accuracy of feature generation phase is 78.67%. The BoW approach for the solution works well to eliminate the true negatives (Table I).

TABLE I: FEATURE GENERATION CONFUSION MATRIX

	Predicted	
Actual	37458(True +ve)	8513(False -ve)
	6423 (False +ve)	17626(True -ve)

Total Test Records = 70020

The performance can be improved by applying semantic techniques and develop the ontology for the domain which will help to retain more true positives and reduce false negatives and false positives.

Classification: Even though Naïve Bayes is a simple classifier, it works well if the model is trained well. We classified the extracted records from the feature extraction and achieved 75.35% correctly classified results (Table II).

TABLE II: CLASSIFICATION CONFUSION MATRIX

	Description	***Impact***	***Recommendation***
Description	**260**	20	24
Impact	10	**88**	9
Recommendation	79	104	**404**

We found few reasons for this classification accuracy by the model. One of the reason is that model training done with less number of training records. Other reason is BoW which will occur in Description and Impact will most likely to be in Recommendation. Performance can be improved by using more train data and enhance classification algorithms like SVM for better classification accuracy. Improvised BoW using semantic techniques will also help to tune the model better.

	Correct %	Incorrect %
Knowledge Bank Cluster	100	0
Cluster Content	77	23
Classification	65	35

Fig. 4. Turn around time comparisons.

Clustering: The results obtained from Canopy and KMeans provided good results. All the cluster created were all relevant cluster and related with problem context. Clusters contained 77% correct content and 33% irrelevant content as per users query (Fig. 4). The clustering performance will be increased if feature extraction accuracy can be improved as all irrelevant contents will be discarded. Different topic modelling algorithms like LSA, LDA etc. can also be used to identify specific topic related clusters based on user query keywords.

Cumulative turnaround time was compared at each step between manual query resolutions by SME with our solution (Fig. 5). Manual minutes taken by an SME to resolve a query on an average is 1020 minutes which was reduced to 715 minutes by our solution providing a Productivity Gain of 30%.

Besides the overall improvement in the turnaround time, our solution provided multiple clusters and the best relevant clusters are provided to the users whereas SME usually provides only one of the solutions which may or may not work for the end user and this situation will involve multiple rounds of iteration responses between SME and end user. Our solution can save time for both the user and the SME so that multiple recommendations for the problem is provided at same time.

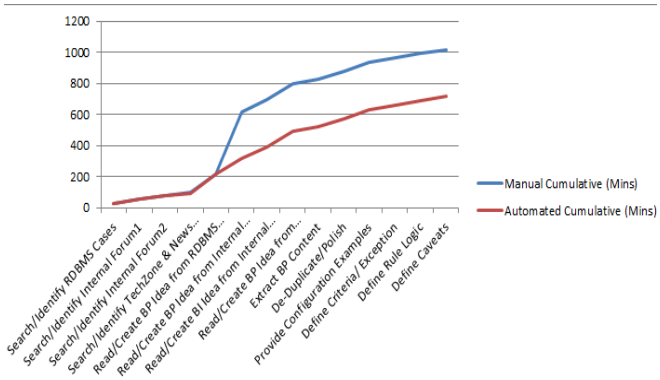

Fig. 5. Turn around time comparisons.

VIII. CONCLUSION AND FUTURE WORK

We developed a solution and evaluated it on enterprise data for user query resolution by identifying the Description, Impact and possible Recommendations. Experimental results showed that relevant information is extracted, grouped and provided to user for their query resolution.

Future works will focus on expanding the scope of the experiments by improving the feature extraction techniques by using semantic analysis and developing ontologies for better information extraction. Advanced classification algorithms like SVM will be used for improved classification accuracy. Clustering techniques will be improvised by using Topic Modelling techniques and will try to generate a single cluster with all possible solutions. Apart of improvement in accuracy, we will also work towards better structuring of the results in order to maintain information flow to the user. Deduplication and Summarization are few other objectives that we will try to achieve in our future work.

ACKNOWLEDGMENT

We wish to thank Dhanyamraju S U M Prasad, and, Prathameshwar Pratap Singh for supporting and providing required information to conduct experiments for validating hypothesis of the solution.

REFERENCES

[1] Z. Michalewicz and A. Jankowski, "System and method for analysis and clustering of documents for search engine," Publication Number US20020065857 A1.

[2] J. Thomas and K. Ramachandran, "Phrase-based document clustering with automatic phrase extraction," Publication Number US8392175 B2.

[3] A. Lynn, "Patterson: Phrase-based searching in an information retrieval system," Publication Number US 7599914 B2.
[4] D. R. Radev, H. Y. Jing, and M. Budzikowsk, Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies.
[5] A. Harith, K. Sanghee, M. E. David *et al*., "Automatic ontology-based knowledge extraction from web documents," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 14–21, 2003.
[6] G. Bierner and A. G. Zadeh, "Method and apparatus for implicit topic extraction used in an online consultation system," Publication Number US 20140114986 A1.
[7] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. H. Sun, "Finding question-answer pairs," presented at Online Forums SIGIR'08, July 20–24, 2008, Singapore.
[8] M. Simeon and R. Hilderman, "Categorical proportional difference: A feature selection method for text categorization: Australian computer society," in *Proc. Conference in Research and Practice in Information and Technology*, 2008.
[9] D. J. Hand and K. Yu, "Idiot's Bayes — not so stupid after all?" *International Statistical Review*, vol. 69, no. 3, pp. 385–399, 2001.
[10] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high dimensional data sets with application to reference matching," in *Proc. the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 169-178.
[11] S. Owen, R. Anil, T. Dunning, and E. Friedman, Mahout in action 2012 manning publications Co ISBN 9781935182689.

Arvind Maurya received his master's degree in computer science from University of Allahabad. He is currently working with HCL Technologies Ltd. India as a solution director. In the research and development division of HCL, Arvind has been involved in cloud, internet of things, machine learning, distributed computing and process automation research areas. Arvind has filled 4 patents so far in virtualization, machine learning and cloud areas.

Yogesh Gupta received his master's degree in computer science from IP University, Delhi, India. He works with HCL Technologies Limited, Noida, India as a solution architect. His current research interests include Internet of things, distributed analytics & storage engine and process automation. Yogesh has filled 7 patents so far in analytics, test automation and cloud areas.

Stuti Awasthi received her bachelor's degree in computer science from UPTU India. She is currently working with HCL Technologies India Ltd as a technical lead. Her current research interest includes IOT, machine learning, NLP, deep learning and distributed computing. Stuti has filled 2 Patents till now in the area of HealthCare analytics involving NLP and machine learning.