E-mails Mining using Generalized Addressing Patterns (GAP)

Lakshmi Sravana Grande, K. Mallikarjuna Mallu, P. Pedda Sadhu Naik

Abstract

Emails become an important medium of communication. A user may receive tens or even hundreds of emails every day. Handling these emails takes much time. Therefore, it is necessary to provide some automatic approaches to relieve the burden of processing the emails. A straightforward method is to group the similar emails by supervised classifications such as mail-id, to-mail-id, subject, message, sending-time, attachments. Email mining is a process of discovering useful patterns from emails. Clustering techniques can be applied over email data to create groups of similar emails. In our algorithm, natural language processing techniques and frequent item set mining techniques are utilized to automatically generate meaningful Generalized Addressing Patterns (GAPs) from mail-id, to-mail-id, subject, message, sending-time, attachments of emails. Then we put forward a novel unsupervised approach which treats GAPs as pseudo class labels and conduct email clustering in a supervised manner, although no human labeling is involved. Our proposed algorithm is not only expected to improve the clustering performance, it can also provide meaningful descriptions of the resulted clusters by the GAPs. Experimental results on open dataset and a personal email dataset collected by ourselves demonstrate that the proposed algorithm outperforms the K-means algorithm in terms of the popular measurement F1. Furthermore, the cluster naming readability is improved by 68.5% on the personal email dataset.

Keywords

Frequent item set, Email Clustering, Weighted Email Similarity, Email Attributes Similarity.

I. Introduction

Emails become an important medium of communication. A user may receive tens or even hundreds of emails every day. Handling these emails takes much time. Therefore, it is necessary to provide some automatic approaches to relieve the burden of processing the emails. Email communication has come up as the most effective and popular way of communication today. People are sending and receiving many messages per day, communicating with partners and friends, or exchanging files and information. E-mail data that are now becoming the dominant form of inter-and intra-organizational written communication for many companies and government departments. E-mails are the essential part of life now just likes mobile phones.

A straightforward method is to group the similar emails by supervised classifications such as mail-id, to-mail-id, subject, message, sending-time, attachments. Supervised methods need a predefined taxonomy. The taxonomy may evolve over time with the change of the users’ work, which requires the users to update the taxonomy manually. What is more, whenever the taxonomy is changed, a considerable amount of training data is indispensable for building an effective classifier. However, the preparation of training data is time-consuming and expensive. Thus, an unsupervised technique such as clustering is an attractive alternative.

Conventionally, email clustering is based on the representation of bag-of-words. This simplistic approach cannot take full advantage of valuable linguistic features inherent in the semi-structured emails, which may result in unsatisfactory performance. In this paper, we present a novel technique to cluster emails according to the sentence patterns discovered from the subject lines of the emails. In this method, each subject line is treated as a sentence and parsed through natural language processing techniques. After that, the terms in the subject lines are converted to generalized terms such as “person”, where “person” can be instantiated as different people names in different emails. Based on the generalized terms, we mine some patterns called Generalized Addressing Patterns (GAP) to indicate the overall meaning of the subject lines. An example of the GAPs is {“person”, “seminar”, “date”} which means that someone (“person”) gives a “seminar” on someday (“date”). It is clearly that the GAPs can help summarize the subjects of a large number of similar emails which results in a semantic representation of the subject lines. To mine GAPs, we utilize the existing frequent closed itemset mining techniques. However, some redundancy still exists in the set of closed GAPs. Grouping similar GAPs is a simple way to tackle this problem. GAPs in the same group will represent the same cluster. The similarity between two GAPs is defined based on their subset-superset relationship and their supports. Meanwhile, the number of GAP groups can be several times larger than the actual number of desired clusters. A heuristic rule based on the length and the support of the GAPs is applied to select GAP groups.

The main contributions of this paper can be summarized as follows:

- A novel algorithm is proposed to automatically mine the semantic knowledge from subject lines of emails in terms of generalized Addressing patterns (GAP);
- A novel clustering algorithm is proposed to leverage the discovered GAPs. Experiments on both an open email dataset and a dataset collected by ourselves show that our method achieves significant improvement as compared to the K-means clustering algorithm without using the GAPs.

A. Email as a Database

Technically an email is constituted of number of fields or attributes like from-mail-id, to-mail-id, subject, message, attachments etc. A set of emails in a mail box can be treated as number of email records with the number of attributes of an email. In this way the emails stored in the mail box can be treated as the email database.

B. Email Mining

Data mining techniques can be applied over the email databases to discover the useful and interesting knowledge pattern, which is named as email mining. There are number of applications of email mining today. Some of the interesting email mining applications are email categorization, summarization, automatic answering and spam filtering etc.
C. Clustering Emails

Clustering is a technique of creating group of similar objects. When clustering is used in email mining it is called as email clustering. Email Clustering can be explained as follows. Subject-based folders can be automatically constructed starting from a set of incoming messages. In this case, the goal is to build automatic organization systems which will analyze an inbox recognize clusters of messages with the same concept give an appropriate name to each cluster and then put all messages into their corresponding folders.

II. Previous Work and Definitions

Significant work is done in the field of email mining. This section represents a brief overview of some work done in email mining including email classification, spam detection etc.. Machine Learning and Data Mining based solutions are proposed by Katakis, Tsoumakas, and Aristotle to develop the intelligent techniques to automate the email managing tasks like automated answering, email summarization etc. A Malicious Email Filter named MEF (a freely distributed malicious binary filter) is developed by Schultz and Eskin, and Zadok. The proposed system filters multiple malicious attachments inane email by using detection models obtained from data mining over known malicious attachments. An approach for detecting self-propagating email viruses based on statistical anomaly detection is proposed by Gupta and Sekar. A spam filtering methods is proposed by Seulley and Cormack which considers the noisy user feedback as an important parameter for spam filtering [6]. Users give feedback that is often mistaken, inconsistent, or even maliciously inaccurate is considered as spams in the proposed method. Number of application area’s and techniques are highlighted in email mining by Katakis, Tsoumakas, and Vlahavas. The application includes Email classification and clustering, automatic answering, Thread Summarization and Spam Filtering etc. The issue of opinion spams or the trustworthiness of online opinions have been studied by Jindal and Liu. Based on the analysis of 5.8 million reviews and 2.14 million reviewers from amazon.com, they have shown that review spam was widespread.

A technique to classification the email is proposed by Martin, Sewani, Nelson, Chen, and Joseph. The proposed classification is used for identifying spam messages. A collection of behavioral features are identified in the proposed technique for a user’s email traffic which enables the rapid detection of abnormal email activity. A demonstration of the effectiveness of outgoing email analysis is also done in the proposed technique using an application that detects worm propagation. A hierarchical folder structure process is proposed by Frommholz as an email classification method for archive new arriving emails. New email triggers actions are generated over the classification implementation to archive the new emails. An email classification technique using Naive Bayes is proposed by Tmka. A system to group and summarize email messages is developed by Ayodele, Khusainov and Ndzi. The propose system uses the subject and content of email messages to classify emails based on users’ activities and generate summaries of each incoming message with unsupervised learning approach. The framework of proposed system solves the problem of email overload, congestion, difficulties in prioritizing and difficulties in finding previously archived messages in the mail box. A technique to map botnet membership using traces of spam email is proposed by Zhuang, Dunagan, Simon, Wang, and Tygar. To group bots into botnets multiple bots participating in the same spam email campaign are taken. Then they have applied the proposed technique against a trace of spam email mail services.

A neural network based system is proposed for automated e-mail filing into folders and anti-spam filtering by Clark, Koprinska, and Poon. Performance wise more accuracy is claimed in the proposed system than previous existing systems. Also the effects of various feature selection is investigated. A comparison of email classifiers is done by Youn and McLeod; email data was classified using four different classifiers Neural Network, SVM classifier, Naive Bayesian Classifier, and J48 classifier with different data size and different feature size. The result of the comparison was J48 classifier is more efficient for the dataset chosen. A classification engine named SpamBayes is proposed by Meyer and Whateley. The important features are discussed in the proposed technique. A Robinson, Woodhead, Peters technique of ‘tiling’ unigrams and bigrams to produce better results are also included in the proposed technique.

Data mining techniques are applied over the emails by Vakali and Pallis to discover the new social networks and communities. A new data structure named social graph is created based on the data available in email communication to discover the new social networks. A technique to create the social networks of email correspondents is proposed by Bird, Gourley, and Swaminathan. Questions relating to participation in the email are discussed in the proposed technique. The relationship of email activity and commit activity in the CVS repositories is also explained in the technique. A method for finding experts their contact details using e-mail messages is proposed by Balog and Rijke. Messages are traced on the basis of a topic, and then associated experts are discovered. An unsupervised technique is proposed for both the list of potential experts and their personal details are obtained automatically from e-mail message headers and signatures. An e-mail content mining technique for author identification or authorship attribution is proposed by Vel, Anderson, Corney and Mohay. Discrimination based technique between authors for the case of both aggregated e-mail topics as well as across different email topics is discussed.

An open source framework is proposed by Nagwani to measure the similarity between two objects, where each object is consists of number of attributes of different data types. According to the different data types of the attributes, weights are assigned for pairwise similarities of the attributes to calculate the overall similarity of a pair of objects. The proposed framework is generic in nature and can be used for clustering of any generic objects. A weighted similarity based mining model is proposed by Nagwani and Singh to discover the similar and duplicate software bugs in a software bug repositories. A software bug to object transformation is also proposed and implemented in the similar work.

A. Clustering Algorithms

The most widely used clustering algorithm in textual data is the K-Means algorithm. In order to group some points in K clusters, K-Means works in 4 basic steps:

1. Randomly choose K instances within the dataset and assign them as cluster centers
2. Assign the remaining instances to their closest cluster center
3. Find a new center for each cluster

The drawbacks of the Existing Systems is it could be incorporating the similarity of the email attachments etc. for the more accurate clustering of the emails. The other direction of the proposed work could be applying the proposed email similarity function for the more email mining operations like thread summarization,
automatic answering of the emails and classification of the emails for participating all the attributes of the emails and achieving more accurate results.

III. Proposed Model

A. Generalization of Terms in E-mail Subjects

To mine GAPs from email subjects, the first step to generalize the terms in the subjects. In this paper, a natural language parser, Microsoft’s NLPWin, is employed for this purpose. The NLPWin tool takes a sentence as input and builds a syntactic tree for the sentence. Figure 1 is an example syntactic tree generated by NLPWin tool for the sentence (email subject) “Hello James Bond.”

Fig. 1: An Example of Syntactic Tree Generated by NLPWin

NLPWin tool can generate factoids for noun phrases, e.g. person names, date and places. Part of the nodes in the syntactic tree has an attribute “FactPred” to specify their factoids predicted by the statistical language model of NLPWin. The factoid of a node is essentially a generalization of the word/phrase represented by the node, which captures its semantic meaning. The original words in the email subjects are replaced as the factoids to help mine sentence patterns as described in next subsection. For example, in the above example syntactic tree, the predicted factoids of the node “NOUN1” is “person” as shown in fig. 2.

Fig. 2: Factoids Predicted for the Node “NOUN1”

B. Mine Generalized Addressing Pattern

For each email subject, after stop words are removed, NLPWin is used to generate its syntactic tree, and the factoids of the nodes are added into the email subjects. The resultant email subjects are called as generalized sentences. For example, from the syntactic tree shown in fig. 1, {welcome, Bob, Brill, person} is the generalized sentence. The subsets of a generalized sentence are also called generalized sentences. Given two generalized sentence s1 and s2, if s1 is a subset of s2, then we say that s2 contains s1, which is similar to that of frequent itemset.

The formal definition of generalized Addressing pattern is as follows:

Definition: Generalized Addressing pattern (GAP). Given a set of generalized sentences S = {s1, s2, ..., sn} and a generalized sentence p, then a generalized Addressing pattern is defined as the maximal length of the GAPs in that group and the support of a GAP group is defined as the maximal length of the GAPs together. A parameter sp_num is used to control how many GAP groups are selected for clustering. The rationale behind this simple heuristic rule is that a longer GAP is more confident than a shorter one in deciding the membership of the emails.

C. GAPs Grouping and Selection

Although mining closed GAPs can reduce the number of generated GAPs substantially, some redundancy still exists in the closed GAPs. Grouping similar GAPs together is a simple way to tackle the above problem. A high min_conf may fail to group similar GAPs together; while a low min_conf threshold will group more GAPs together but it may introduce some noises. Experimental results showed that min_conf value between 0.5 and 0.8 were safe for all the tested data sets, and grouping similar GAPs together improves the performance substantially.

The number of GAP groups can be several times larger than the actual number of clusters. A heuristic rule based on the length and the support of the GAPs is applied to select GAP groups. First, we sort the GAP groups in descending order of length. Second, we sort them in descending order of support. Finally, we select the first sp_num GAP groups for clustering. The length of a GAP group is defined as the maximal length of the GAPs in that group and the support of a GAP group is defined as the maximal support of the GAPs in that group. A parameter sp_num is used to control how many GAP groups are selected for clustering. The rationale behind this simple heuristic rule is that a longer GAP is more confident than a shorter one in deciding the membership of the emails.

D. GAP-PCL: GAP as Pseudo Class Label

Based on GAPs, we proposed a novel clustering algorithm to form a pseudo class for the emails matching the same GAP group, and then use a discriminative variant of Classification Expectation Maximization algorithm (CEM) [2, 5] to get the final clusters. When CEM is applied to document clustering, the high-dimension usually causes the inaccurate model estimation and degrades the efficiency. Here linear SVM is used as the underlying classifier. As shown in the following pseudo code of the GAP-PCL algorithm, only the emails not matching any GAP group are classified. The algorithm would stop when it converges or the predefined iteration limit is reached. A threshold is defined to control whether an email is put into a pseudo class. Only when the maximal posterior probability of an email is greater than the given threshold, the email will be put into the class with the maximal posterior probability, otherwise the email will be put into a special class D_other.

The proposed GAP-PCL algorithm uses GAP groups to construct initial pseudo classes. SVM classifier is fed by the classification output of the previous iteration. The sp_num parameter in GAP-PCL should be no greater than the desired number of clusters k.
Ex: 3 selected GSP Groups to be pseudo classes.
1: \{[A,B] \ [B,C] \ [B,C] \}
2: \{[C,D,E,F] \ [C,D] \}
3: \{[D,E,G] \ [D,G] \}
1,2,3 will be the pseudo class label and a GSP of an email subject line will be described as a vector.
Vectors in this vector space is in the form 
\( (X_a, X_B, X_C, X_D, X_E, X_F, X_G) \) & 
\( X_a = 1 \) if this GSP contains A 
\( = 0 \) if there is not A in this GSP…etc
Ex: \([A,B] = (1,1,0,0,0,0,0)\)

IV. Experiments
To demonstrate the effectiveness of our proposed GAP-PCL clustering algorithm, we conduct several experiments on two Email datasets: the open dataset Enron email dataset and a private email dataset collected by ourselves. On the private data, we conduct a case study on clustering naming.

A. Email Datasets
In this section, we describe the open Enron email dataset and the private email dataset used in our experiments.

1. Enron Email Dataset
The Enron email dataset [8] is the archive email from many of the senior management of Enron Corporation, and is now the public record. The dataset is provide by SRI after major clean-up and removed of attachments.
The Enron email dataset used here is a subset of the original Enron email dataset, which is generated by Bekkerman, McCallum, and G. Huang [1].

Table 1: Statistics on Private Email Dataset

<table>
<thead>
<tr>
<th>User</th>
<th>#Folder</th>
<th>#Message</th>
<th>#Smallest</th>
<th>#Largest</th>
</tr>
</thead>
<tbody>
<tr>
<td>User1</td>
<td>25</td>
<td>1178</td>
<td>7</td>
<td>260</td>
</tr>
<tr>
<td>User2</td>
<td>15</td>
<td>719</td>
<td>9</td>
<td>176</td>
</tr>
<tr>
<td>User3</td>
<td>11</td>
<td>1500</td>
<td>8</td>
<td>522</td>
</tr>
<tr>
<td>User4</td>
<td>5</td>
<td>476</td>
<td>33</td>
<td>144</td>
</tr>
</tbody>
</table>

2. Private Email Dataset
Four volunteers (named as user1, user2, user3 and user4) for privacy in table 1 in our organization provide us with their personal email for experiments. Each of them has manually organized his/her emails into self-defined folders before this research work began. The similar preprocessing as Enron email dataset is conducted on this private email dataset to clean the data.

B. Evaluation Criteria
The clustering performance of the proposed GAP-PCL clustering algorithm is evaluated against the manually generated class labels based on external criteria [3].

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Same</td>
<td>Different</td>
</tr>
<tr>
<td>Same</td>
<td>SS</td>
<td>DS</td>
</tr>
<tr>
<td>Different</td>
<td>SD</td>
<td>DD</td>
</tr>
</tbody>
</table>

X belongs to one of the four possible cases as shown in Table 2. After computing the four values in Table 2, precision and recall and F1-Measure are calculated as following:

\[ P = \frac{|SS|}{|SS + SD|}, \quad R = \frac{|SS|}{|SS + DS|}, \quad F_1 = \frac{2PR}{P + R} \]

C. Clustering Performance Study
We compared the proposed GAP-PCL algorithm with the basic K-means algorithm to show its effectiveness. The basic K-means algorithm randomly selects k emails as the initial cluster centers. To alleviate the effectiveness of random initialization, we ran K-means clustering for 10 times and report the average performance in the following experiments. Meanwhile, to study the effectiveness of the pseudo class label generated by GAPs, we use the generated GAPs to initialize the K-means clusters. We call the resultant algorithm as GAP-means, which is another baseline algorithm.

1. Experimental Results on Enron Email Dataset
In the experiments conducted on Enron email dataset, the minimum support threshold (min_sup) is set as 4 to generate the GAPs and the minimum length of GAPs is restricted to 2. The cluster number k on each user’s email data is set as the folder number of each user.

Experimental results on Enron email dataset are reported as the F1 values over the seven users. As shown in Figure 3, GAP-PCL achieves consistently significant improvements on seven users compared with GAP-means and K-means clustering.

As shown in fig. 3, the clustering performance varies a lot on the seven users’ dataset, which depends on the level of complexity and homogeneity of each dataset. Such an observation is in consistent to the classification results reported by Bekkerman, McCallum, and G. Huang in their work.

Fig. 3: Performance of GAP-PCL, GAP-means and K-means on Enron Email Dataset (in terms of F1)
2. Experimental Results on Private E-mail Dataset
Experimental results on the private email dataset are shown in fig. 4, in terms of F1. Similar to the observation on Enron dataset, we can see that GAP-PCL achieves consistently significant improvements on four users compared with GAP-means and K-means clustering. Another observation from fig. 4, is that GAP-means achieves some improvements over the basic K-means algorithm, especially on user1 and user4, which proves the effectiveness and usefulness of the GAPs.

D. Cluster Naming
In GAP-PCL, cluster names are generated as follows: if emails in one cluster match one or more GAP groups, the cluster is named by the GAP with the highest support, otherwise it is named by the top five words sorted based on the scores computed as follows:

\[ \text{Score}(t_i) = \frac{\sum \log(C_i) \cdot f_i}{\ln(1 + \sum f_i)} \]  

(3)

basic K-means algorithm, clusters are named using ranked word features. Some sample cluster names generated through GAPs and word features are shown in Table 3.

Table 3: Sample Cluster Names Generated from GSPs and Ranked Word Feature

<table>
<thead>
<tr>
<th>CLASS SEMANTICS</th>
<th>GSP</th>
<th>WORD FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Interview</td>
<td>time</td>
<td>confirm, accommodate, lunch, convenient, attend</td>
</tr>
<tr>
<td>2 Talks given by some persons</td>
<td>talk person</td>
<td>venue, talk, speaker, room, Dr</td>
</tr>
<tr>
<td>3 Introducing somebody</td>
<td>Welcome</td>
<td>person, welcome, join, degree, joined, university</td>
</tr>
<tr>
<td>4 Analysis on PSS log</td>
<td>analysis</td>
<td>associations, except, divides, structuralize, surprise</td>
</tr>
<tr>
<td>5 Paper review</td>
<td>paper</td>
<td>review, paper, dear, IEEE, papers, review</td>
</tr>
</tbody>
</table>

From Table 3, we can see that GAPs capture and summarize the contents of the clusters more precisely and compactly than word features. For sample 1, 2 and 3, the GAPs contain factoids generated by NLPWin, which not only makes the GAPs much easier to understand, but also makes the discovery of such GAPs possible. For sample 1 and 4, it is difficult to understand the names generated from word features if the users do not read the email contents. In sample 2 and 5, although the ranked word features contain enough information but they also contain some noisy words.

. The cluster names generated by GAP-PCL and the basic K-means algorithm on each data set were evaluated by the owner of the dataset via readability scores from 1 (un-readable) to 3 (clear). Additionally, seven experimenters were invited to help evaluate the readability of the cluster names on all data sets.
We conducted an experiment to evaluate the cluster names generated from GAPs and ranked word features.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Owners Average</th>
<th>Experimenters Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1.65</td>
<td>1.81</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td>1.97</td>
<td>2.02</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>2.80 (+71.3%)</td>
<td>2.44 (p=0.0005782)</td>
</tr>
<tr>
<td>GSP-PCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 4</td>
<td>2.71 (+37.7%)</td>
<td>2.58 (p=0.006793)</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 5</td>
<td>1.05</td>
<td>1.73</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 6</td>
<td>2.42 (+130%)</td>
<td>2.43 (p=0.0008237)</td>
</tr>
<tr>
<td>GSP-PCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 7</td>
<td>1.72</td>
<td>1.73</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 8</td>
<td>2.31 (+34.0%)</td>
<td>2.18 (p=0.0589)</td>
</tr>
<tr>
<td>GSP-PCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvement</td>
<td>68.5%</td>
<td>32.4%</td>
</tr>
</tbody>
</table>

The seven experimenters are unaware of our algorithm. T-test is performed on the seven experimenters’ scores. The results are shown in Table 4. We can see that the names generated from GAPs are more readable in both the viewpoints of owners and other experimenters. The improvements are statistically significant according T-test results.

### V. Conclusions and Further Work

In this paper, we proposed a novel approach to automatically extract embedded knowledge from the email mail-id, to-mail-id, subject, message, sending-time, attachments to help improve email clustering. Natural language processing technique and the frequent closed itemset mining technique are employed to generate generalized Addressing patterns (GAP for short) from email subjects, which can be used to assist clustering as well as serve as good cluster descriptors. To leveraged the discovered GAPs, a novel unsupervised approach is proposed, which treats GAPs as pseudo class labels and classifies emails using a supervised learning algorithm (although no human labeling is involved). The experimental results showed that GAP-PCL obtains significant improvements both on the cluster quality and cluster name readability compared with the basic K-means algorithm.

### References


[19] Seongwook Youn, Dennis McLeod, “A Comparative Study for Email Classification”, Proceedings of International Joint Conferences on Computer, Information, System Sciences, and Engineering (CISSE’06), Bridgeport CT, December 2006.


Lakshmi Sravani Grande was born in Markapur, Prakasam Dt, Andhra Pradesh, India. She received B.Tech in C.S.E from JNT University, Kakinada, Andhra Pradesh, India. Presently, she is pursuing M.Tech in C.S.E from Samuel George College of Engineering and Technology, Markapur, Prakasam Dt, Andhra Pradesh, India. Her Research interest includes Emails Mining Using Generalized Addressing Patterns.

K Mallikarjuna mallu was born in Bestavaripeta, Prakasam Dt, Andhra Pradesh, India. He received B.Tech in IT from JNTU Hyderabad, Andhra Pradesh, India. And Through gate he completed his M.Tech C S E from JNTU Kakinada, Andhra Pradesh, India. Currently he is working as Assoc. Professor in Department of CSE at Dr. Samuel George Institute of Engineering and Technology, Markapur, Prakasam Dt, Andhra Pradesh.

P. Pedda Sadhu Naik received B.Tech (CSE) Degree from JNT University in 2003 and M.Tech (CS) Degree from JNTUCE Anapatapur in 2007. He is doing Ph D from JNTUK,Kakinada in Computer Science under the guidance of Dr T. Venu Gopal. He is Associate Professor, JNTU College of Engineering, Jagityal. Karimnagar District. He has 9 years of teaching experience. He joined as Assistant Professor in Dr.Samuel George Institute of Engineering & Technology, Markapur, India in 2003. Presently he is working as Associate Professor and Head of the Department of CSE. His Interested research areas are Image Processing and Soft Computing. He has life member of ISTE and IAENG. He organized various National level Technical Symposiums as Convener. He attended Various National and International Workshops and Conferences on image Processing.