A New Method of Text Categorization and Summarization with Fuzzy Confusion Matrix

Dr. Goutam Sarker
Senior Member IEEE,
Associate Professor
Dept. of CSE
NIT Durgapur.

Antara Pal
M. Tech Student,
Dept. of CSE
NIT Durgapur.

Saswati Das
M. Tech student,
Dept. of CSE
NIT Durgapur.

Abstract— *Present work is a technique fuzzy text categorization followed by extractive summarization of categorized texts. At the onset, the texts of different subjects are fuzzy categorized based on relative matching with index terms of corresponding subjects. After forming the categorical groups, extractive summarization is performed on each text of each category. The fuzzy categorization is evaluated with fuzzy confusion matrix. The performance evaluation of this fuzzy categorization with Holdout method in terms of accuracy, precision, recall and f-score is appreciably high. The accuracy of summarization is evaluated using human generated summary and is fair. Also the categorization and summarization time is acceptable.*

Keywords—*Fuzzy Text Categorization, Fuzzy Confusion Matrix, Extractive Summarization, Term Frequency, Inter document frequency, Sentence Weight, Clustering, OCA, Holdout Method, Accuracy, Precision, Recall, F-Score*

I. INTRODUCTION

Text Categorization is one of the promising area in the field of Data Mining to perform grouping of texts based on their attributes. Fuzzy Categorization of texts involves categorization according to degree of belongingness of texts to different categories. Text Summarization achieves the task of compression of the text size while preserving the overall meaning. Text summarization can be classified as extrinsic or extractive summarization and intrinsic or abstractive summarization. Abstractive summarization [11] requires understanding of semantics or meaning of sentences to build the summary whereas extractive summarization method selects subsets of important sentences in original texts, based on their weight and eliminates the redundant as well as irrelevant sentences to form the summary. In our proposed work, three operation viz. categorization, summarization and categorization followed by summarization have been carried out on a set of textual documents. Categorization does the task of categorizing different text documents into their respective categories. Summarization takes different text of same categories and summarizes them. The output produced is the compressed version of original text. In categorization followed by summarization, the input contains different texts for categorization, the result of which is implicitly passed to the system for summarization. This is the summary of text documents belonging to each of the defined categories.

Text categorization can be used to

1. Typically organize texts or stories according to subject categories.
2. Classification of academic or research papers by technical domains and sub domains.
3. Spam filtering, where email are classified into spam and non-spam categories.
4. Email routing, sending an email sent to a general address to a specific address or mailbox depending on topic.

Text summarization can be used in

1. Summarization of news to headlines, SMS or WAP-format for mobile phones.
2. Search engines to showcase the brief/compressed description of the search result (e.g. Google search engines) .

II.. THEORY OF OPERATION

A. Text Mining

Text Mining involves series of process in order to derive meaningful information, which are as follows:

a. Text Preprocessing
b. Text Transformation
c. Attribute Selection
d. Deriving Patterns and finally
e. Interpretation and Evaluation input data.

B. Text Preprocessing

Preprocessing is the extraction of keywords from the original input document. The transformation includes word separation, removal of links, HTML or other tags, removal of stop words, punctuations.

Stop words: frequent words that are not useful for categorization i.e. article, prepositions, conjunctions, pronouns etc.

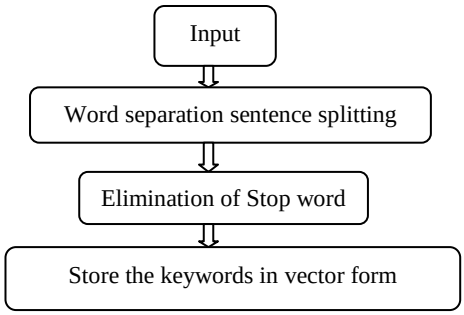

Fig 1. Schematic of preprocessing of text

C. Text Categorization

Text categorization [8,9,10] is one of the key applications in Data Mining. It involves assignment of category to unlabeled new text document.

In our approach, keyword extraction has been done to categorize text. It is a supervised learning method which relies on labeled training data to achieve accuracy in classification [3,6,7,8,9,10]. Keyword extraction is achieved through preprocessing, after which the preprocessed text uses the help of keyword set of each category to decide the category of the text. Each of preprocessed text keywords is compared with predefined categorized words for their fuzzy categorization. The text are set to be 'Highly Probable' to a certain category if its keywords matching with that very category exceeds 70%, 'Moderately Probable' if matching is between 40% - 50% and finally 'Least Probable' is less than 15%. The category to one having maximum match is inferred to be the category of the corresponding input text.

D. Summarization

Our approach to summarization [11] is extraction based. The system takes into account the existing words and phrases in original text to form the summary. The text document is summarized based on sentence weight.

The system calculates the frequency of keywords, which is used to get the weight of keyword. Both intra and inter document frequency [11] is calculated.

All these attributes are used ultimately to compute the weight of each sentence. Sentence which are redundant are identified by clustering [1]. From sentences in a cluster, a representative sentence is chosen (having maximum weight and minimum sentence length). Rest of the redundant sentence in each cluster is ignored.

Now, number of sentence is reduced to number of clusters formed (including singleton cluster).

Then, it arranges the sentences in descending order of their weight. 25% of sentences in weight-wise descending order are selected, while rest is deselected.

Then the selected sentences are compared with original text to find the right position of each sentence in the summary.

And then lastly, the machine generated summarized texts are compared with human generated summaries for those texts to evaluate the summary system in terms of accuracy.

E. Performance Evaluation

(i) Fuzzy Categorization

In our present work we have introduced fuzzy confusion matrix in which each text has been considered to be a fuzzy member of any category, if its attribute matching exceeds a certain threshold(i.e. threshold $_{cat}$ = 15%). The text are graded to be highly probable to a certain category if its match with that category exceeds 70%, moderately probable if it lies between 15% and 40% and least probable if less than 15%. A match between predicted and actual category is scored the highest (if the class predicted happens to be the actual class, then the degree of fuzziness is one among the three possibilities i.e. H, M, or L).

In our previous paper [9,10] we have formulated the rules for the construction of the fuzzy confusion matrix for the two different situations (i.e. Actual Class = Predicted Class and Actual Class ≠ Predicted Class). In the present work we are updating the rule for calculating the Incremental Score (∑) for the case Actual Class ≠ Predicted Class in the following manner.

Let m and n represents the row number and column number in the two tables respectively.

Where $1 \leq m \leq 3$, $1 \leq n \leq 3$

For Actual Class = Predicted Class

∑ = [3 - (Defuzzified Difference between Actual Class and Predicted Class)] $\forall\, m, n$.

Table I. INCREMENTAL CONSTRUCTION OF FUZZY CONFUSION MATRIX

		Actual Class		
		Highly Probable	Moderately Probable	Least Probable
Predicted Class	Highly Probable	+3	+2	+1
	Moderate Probable	+2	+3	+2
	Least Probable	+1	+2	+3

For Actual Class ≠ Predicted Class

∑ = [3 - (Defuzzified Difference between Actual Class and Predicted Class)] for m ≠ n.

∑ = [3 - (Defuzzified Difference between Actual Class and Predicted Class)] - (n-1) for m = n.

Table II. INCREMENTAL CONSTRUCTION OF FUZZY CONFUSION MATRIX

Predicted Class \ Actual Class	Highly Probable	Moderately Probable	Least Probable
Highly Probable	+1	+2	+3
Moderate Probable	+2	+1	+2
Least Probable	+3	+2	+1

Where the defuzzified values are

- Highly Probable = 3
- Moderately Probable = 2
- Least Probable = 1

(ii) Discrete Categorization

a. Accuracy: It is the ratio of records of all categories that are correctly classified with respect to total number of records.

For example, for two class classification

Predicted Class \ Actual Class	class1	class2
class1	a	b
class2	c	d

$$\text{accuracy} = \frac{a+d}{a+b+c+d}$$

Where a, b, c and d are defined in the matrix.

b. Precision: It is the ratio of the total number of records actually belonging to a category with respect to the total number of records predicted to belong to that category.

$$\text{precision} = \frac{a}{a+b}$$

c. Recall: It is the ratio of the total number of records predicted to belong to a category with respect to the total number of records actually belonging to belong to that category.

$$\text{recall} = \frac{a}{a+c}$$

d. F-score: Harmonic mean of precision and recall.

$$\text{F-score} = \frac{2*precision\ *recall}{precision\ +recall}$$

iii Summarization

Evaluation of Summarization is done by comparing the system generated summary with human generated summary which is the accuracy of summarization.

$$\text{Accuracy}_{\text{sum}} = \frac{No\ of\ matc\ hed\ sentences}{Total\ no\ of\ sentences}$$

IV. OVERVIEW

In our present work we have used extraction of keyword for categorization which is a supervised learning method relying on labeled training data. Also to build summary, we used Extractive method.

To do this, we use the following features.

1. Keyword: The remaining words after preprocessing are the keywords.

2. Term Frequency (t_f) : Denotes the frequency of keyword in the text.

3. Representative Sentence: Similar sentences (syntactically) are clustered into a group. Among these sentences, the one having highest sentence weight to length ratio, is chosen to be the representative sentence of that cluster. The rest are considered to be redundant.

4. Normalized Term Frequency (ntf_i) : The normalized term frequency of the i^{th} keyword k_i ,

$$ntf_i = \frac{t_{fi}}{(t_{f1} + t_{f2} + t_{f3} + \ldots + t_{fn})} * w_k$$

5. Inter Document Frequency (*IDF*) : The IDF of i^{th} keyword (IDF_i) is the ratio of the number of text files in which that keyword k_i is occurring i.e. (n_i) and the total number of text files(n).
Therefore, $IDF_i = \frac{n_i}{n}$

6. Weight of keyword k_i in a sentence s_i is the sum of its normalized term frequency and inter document frequency i.e. $t_f + IDF_i$.
Hence, the weight of sentence become , where there are j different keywords in i=1 sentence s_i . Now we perform clustering to remove redundant sentence and arrange the sentences in descending order of their weight.

$$w_{si} = \sum_{i=1}^{j} ntf_i + IDF_i$$

7. Clustering or Grouping of similar sentences: Each sentence in the text file is compared to all other sentences in the text file to form a cluster of syntactically similar sentences. Ruling out the singleton clusters, for all other clusters choose one of the sentences as the representative sentence of the cluster. The representative sentence is chosen on the basis of highest sentence weight to sentence length (in original text) ratio.

$$w_{si} = \frac{w_{si}}{l_{si(in\ the\ original\ \ text)}}$$

w_{si} = updated weight of sentence in a cluster

The one sentence having maximum w_{si} is selected and the rest are deselected since they are considered redundant.

8. Evaluation : Comparison of system generated summary to that of human generated summary is made to calculate the accuracy of the system.

V. ALGORITHM FOR PRESENT TECHNIQUE

a. Algorithm for Categorization

Input : Text files of different subjects, threshold$_{cat}$ = 15%.
Output : Categorized text files contained in labeled folders.

Steps :

i. Extract keywords (preprocess)

ii. Perform matching of each keyword with all the labeled set of keywords (training dataset) say D_j, for $1 \leq j \leq n$, where n = No Subjects taken.

iii. Find the degree of belongingness of each document (d_i) (where $1 \leq i \leq m$, where m is the no of input text files taken) with each of the input keyword set D_j of predefined category.

iv. Let DB[i,j] and MF[i,j] be matrices that store the percentage matching of d_i with D_j and its corresponding Membership Factor(Membership Factor can be HP, MP, LP) respectively.

v. for d_i

if DB[i,j] ≥ 70% then MF[i,j] = HP
else if 40% ≤ DB[i,j] < 70% then MF[i,j] = MP
else if 15% ≤ DB[i,j] < 40% then MF[i,j] = LP
else MF[i,j] = 0 (i.e. d_i does not exceed the threshold)

vi. Choose the category C_j (where C_j is the corresponding category of keyword set(D_j)) of the text file d_i as one which has maximum matching with the keyword set(D_j).

b. Algorithm for Multiple Text File Summarization

Input : Text files of same topic for summarization; threshold their respective summary folder.

Output : Summary files of corresponding input files contained in their respective summary folders.

Steps :

i. Extract keywords k_i.

ii. Find the term frequency of each keyword k_i.

iii. Find the normalized term frequency ntf_i of keyword k_i.

iv. Compute the Inter Document Frequency (IDF_i) i.e. the ratio of text files in which that word (n_i) is presented and total number of text files (n).

v. Compute the weight of keyword as $ntf_i + IDF_i$.

vi. Compute the weight of sentence si as

$$\sum_{i=1}^{j} ntf_i + IDF_i$$

where, j is the total number of keywords in the sentence.

vii. Perform clustering and find the representative sentence for each cluster based on the maximum sentence weight to length of sentence ratio i.e.

$$max\left(\frac{w_{si}}{li}\right)$$

w_{si} = weight of sentence si
li = length of sentence (s_i) in original text

viii. Sequentially arrange sentence according to the decreasing weights in text files. Then select first 25% sentence and each deselect others to obtain the summary.

ix. Repeat step 1 to 8 for each text files in the input folder.

c. Algorithm for Clustering

Input : Threshold , text files.
Output : A set of sentences with maximum ratio of ($\frac{w_i}{l_i}$).

Steps :
i. Extract weighted sentence, w_{si} in a text file.
ii. Group sentences based on the predefined threshold.
iii. Calculate ratio of sentence group weight to length for all s_i belonging to group G_i i.e.

$\frac{w_{si}}{l_i}$, 1<= i<= n ϵ group i

n = number of sentence in group i.

iv. Select the sentence having max($\frac{w_{si}}{l_i}$) value and deselect others.

VI. PERFORMANCE EVALUATION AND EXPERIMENTAL RESULTS

The categorization and summarization is evaluated using an Intel(R) Xeon(R) CPU E5506 @2.13GHz having 12GB RAM and Windows 7 Ultimate 64-bit OS on MATLAB 2013a platform.
The text file contains of different subjects like Computer Science, Biology, Chemistry etc is collected from the internet (en.wikipedia.org, www.textbooksonline.tn.nic.in/books, and many more).
We have used benchmark text (en.wikipedia.org, www.ncert.nic.in/ncerts/textbook/textbook) and our own created human generated summaries for the purpose of system evaluation; due to the lack of benchmark text with their corresponding summaries in the internet.

We have taken some collected sample texts in a single folder named 'text_files' containing 38 texts of different categories. The Fuzzy Categorization of same is given below:

Table 1: Fuzzy Categorization of input folder 'text_files'

Input file name And size (kb)	Highly probable (70%-100%)	Moderately probable (40%-69%)	Least probable (15%-39%)
bio1.txt(6)	biology		Civil
bio2.txt(9)		biology	Civil
bio3.txt(9)			Civil
bio4.txt(10)		biology	Civil
bio5.txt(6)			Civil
bio6.txt(7)			Biology
bio7.txt(6)			Civil
bio8.txt(6)			Chemistry
bio9.txt(14)		chemistry	Civil
bio10.txt(10)			Physics
bio11.txt(8)			Physics
bio12.txt(10)			Physics
chem1.txt(8)		chemistry	Physics
chem2.txt(11)		chemistry	Physics
chem3.txt(7)		chemistry	Civil
chem4.txt(5)			Physics
comp1.txt(9)		computer sc	Civil
comp2.txt(7)		computer sc	Civil
comp3.txt(9)		computer sc	computer sc
comp4.txt(8)			computer sc
comp5.txt(8)			computer sc
comp6.txt(8)			computer sc
comp7.txt(8)		computer sc	
comp8.txt(8)			computer sc
ce1.txt(8)		civil	computer sc
ce2.txt(8)		civil	Physics
ce3.txt(8)		civil	Biology
ce4.txt(8)		civil	Chemistry
ce5.txt(8)			Civil
ce6.txt(6)			computer sc
ce7.txt(10)			Civil
ce8.txt(10)			Civil
phy1.txt(7)			Physics
phy2.txt(12)			Physics
phy3.txt(6)			Physics
phy4.txt(7)			Physics
phy5.txt(8)			computer sc
phy6.txt(13)			Physics

Table 2: Categorization Result for Folder 'text_files'

Input file name And size (kb)	no of texts	catgory *	avg. time (sec)	accuracy (%)
bio1.txt(6)	38	bio	5.21	89.47
bio2.txt(9)		bio		
bio3.txt(9)		bio		
bio4.txt(10)		bio		
bio5.txt(6)		Bio		
bio6.txt(7)		Bio		
bio7.txt(6)		Bio		
bio8.txt(6)		Chem		
bio9.txt(14)		chem		
bio10.txt(10)		Bio		
bio11.txt(8)		Bio		
bio12.txt(10)		Bio		
chem1.txt(8)		Chem		
chem2.txt(11)		Chem		
chem3.txt(7)		Chem		
chem4.txt(5)		Civil		
comp1.txt(9)		Comp		
comp2.txt(7)		Comp		
comp3.txt(9)		Comp		
comp4.txt(8)		Comp		
comp5.txt(8)		Comp		
comp6.txt(8)		Comp		
comp7.txt(8)		Comp		
comp8.txt(8)		Comp		
ce1.txt(8)		Ce		
ce2.txt(8)		Ce		
ce3.txt(8)		Ce		
ce4.txt(8)		Ce		
ce5.txt(8)		Ce		
ce6.txt(6)		Ce		
ce7.txt(10)		Ce		
ce8.txt(10)		Ce		
phy1.txt(7)		Phy		
phy2.txt(12)		Phy		
phy3.txt(6)		Phy		
phy4.txt(7)		Comp		
phy5.txt(8)		Phy		
phy6.txt(13)		Phy		

Abbreviation Details:

bio - Biology
chem - Chemistry
cse - Computer Science and Engineering
ce - Civil Engineering
phy - Physics

For the summarization operation we have taken some folders each for a defined category.

Table 3: Folder wise Summarization results

Sl no	Folder name	Name of texts	No of texts	summ texts *	Sum folder name **	Time (sec)	Individual accuracy (%)	Avg. Accuracy(%)
1	folder1_sum	bio1.txt(6)	12	bs1	f1ss	62.6187	57.14	57.58
		bio2.txt(9)		bs2			70.00	
		bio3.txt(9)		bs3			55.55	
		bio4.txt(10)		bs4			60.00	
		bio5.txt(6)		bs5			57.14	
		bio6.txt(7)		bs6			58.33	
		bio7.txt(6)		bs7			62.50	
		bio8.txt(6)		bs8			58.33	
		Bio9.txt		Bs9			54.54	
		Bio10.txt		Bs10			60.00	
		Bio11.txt		Bs11			80.00	
		Bio12.txt		Bs12			71.42	
2	folder2_sum	ce1.txt(8)	8	ces1	f2ss	67.5873	57.14	66.21
		ce2.txt(6)		ces2			50.00	
		ce3.txt(10)		ces3			42.85	
		ce4.txt(10)		ces4			66.66	
		ce5.txt(6)		ces5			87.50	
		ce6.txt(11)		ces6			70.00	
		ce7.txt(7)		ces7			66.66	
		ce8.txt(7)		ces8			88.88	
3	folder3_sum	comp1.txt(9)	8	css1	f3ss	63.0995	50.00	54.19
		comp2.txt(7)		css2			66.66	
		comp3.txt(9)		css3			70.00	
		comp4.txt(8)		css4			20.00	
		comp5.txt(6)		css5			71.42	
		comp6.txt(7)		css6			60.00	
		comp7.txt(7)		css7			50.00	
		comp8.txt(6)		css8			45.45	
4	folder4_sum	phy1.txt(7)	4	phs1	f4ss	15.7685	60.00	60.11
		phy2.txt(12)		phs2			60.00	
		phy3.txt(6)		phs3			60.00	
		phy4.txt(7)		phs4			20.00	
		Phy5.txt		Phs5			75.00	
		Phy6.txt		Phs6			85.7	
5	folder5_sum	chem1.txt(8)	4	chs1	f3ss	15.0983	62.50	62.29
		chem2.txt(11)		chs2			66.66	
		chem3.txt(7)		chs3			60.00	
		chem3.txt(5)		chs3			60.00	

Abbreviation Details :

bio - Biology
chem - Chemistry
cse - Computer Science and Engineering
ce - Civil Engineering
phy - Physics

Table 4: categorization followed by summarization.

Input file name And size (kb)	no of texts	catgory *	summary files **	time (sec)
bio1.txt(6)	38	Bio	bs1	503.13
bio2.txt(9)		Bio	bs2	
bio3.txt(9)		Bio	bs3	
bio4.txt(10)		Bio	bs4	
bio5.txt(6)		Bio	Bs5	
bio6.txt(7)		Bio	Bs6	
bio7.txt(6)		Bio	Bs7	
bio8.txt(6)		Chem	Bs8	
bio9.txt(14)		Chem.	Bs9	
bio10.txt(10)		Bio	bs10	
bio11.txt(8)		Bio	Bs11	
bio12.txt(10)		Bio	bs12	
chem1.txt(8)		Chem	Chs1	
chem2.txt(11)		Chem	Chs2	
chem3.txt(7)		Chem	Chs3	
chem4.txt(5)		Civil	Chs4	
comp1.txt(9)		Comp	Css1	
comp2.txt(7)		Comp	css2	
comp3.txt(9)		Comp	css3	
comp4.txt(8)		Comp	css4	
comp5.txt(8)		Comp	Css5	
comp6.txt(8)		Comp	Css6	
comp7.txt(8)		Comp	Css7	
comp8.txt(8)		Comp	Css8	
ce1.txt(8)		Ce	Ces1	
ce2.txt(8)		Ce	Ces2	
ce3.txt(8)		Ce	Ces3	
ce4.txt(8)		Ce	Ces4	
ce5.txt(8)		Ce	ces5	
ce6.txt(6)		Ce	ces6	
ce7.txt(10)		Ce	ces7	
ce8.txt(10)		Ce	ces8	
phy1.txt(7)		Phy	Phs1	
phy2.txt(12)		Phy	Phs2	
phy3.txt(6)		Phy	Phs3	
phy4.txt(7)		Cse	Phs4	
phy5.txt(8)		Phy	Phs5	
phy6.txt(13)		Phy	Phs6	

Abbreviation Details :
* bio - Biology
chem - Chemistry
cse - Computer Science and Engineering
ce - Civil Engineering
phy - Physics

** Extension for all summarized text files are .txtsum.txt
bs : Summarized text files of biology
ces - Summarized text files of Civil
css - Summarized text files of Computer Science and Engineering
phs - Summarized text files of Physics
chs - Summarized text files of Chemical

Table5: FUZZY CONFUSION MATRIX

Actual→ Predicted↓		physics			chemistry			biology			civil			Computer		
		H	M	L	H	M	L	H	M	L	H	M	L	H	M	L
Physics	H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	L	0	0	5	0	2	1	0	0	3	0	1	0	0	0	0
Chemistry	H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	3	0	0	0	1	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Biology	H	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
	L	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Civil	H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
	L	0	0	0	0	1	0	1	2	4	0	0	3	0	2	0
computer	H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
	L	0	1	0	0	0	0	0	0	0	0	1	1	0	0	4

Table6: PERFORMANCE EVALUATION WITH HOLDOUT METHOD

Acuuracy = 56.06%			
category	Precision(%)	Recall(%)	f-score(%)
Physics	64.28	90	74.99
Chemistry	60	56.25	58.06
Biology	85.71	31.57	46.14
civil	45.65	70	55.26
Computer sc	82.76	85.71	84.20

Table 7: DISCRETE CONFUSION MATRIX

Predicted Class \ Actual Class	physics	Chemistry	Biology	Civil	Computer sc
physics	5	0	0	0	0
Chemistry	0	3	2	0	0
Biology	0	0	10	0	0
Civil	0	1	0	8	0
Computer sc	1	0	0	0	8

Table8: PERFORMANCE EVALUATION OF DISCRETE CONFUSION MATRIX

Acuuracy = 89.47%			
category	Precision(%)	Recall(%)	f-score(%)
Physics	100	83.33	90.90
Chemistry	60	75	66.66
Biology	100	83.33	90.90
civil	88.88	100	94.12
Computer sc	88.88	100	94.12

VIII. CONCLUSION

We have designed and developed a combined keyword based fuzzy categorization and extractive

summarization technique which categorizes input text into different category and thereafter summarizes them.

The fuzzy categorization is based on keyword extraction and subsequent matching and summarization is done based on weight of sentences. It deselects the redundant sentences through optimal clustering to produce a concise and meaningful summary.

As the summarization is not abstractive, it may consider two semantically some sentences with different keywords as different, thus slightly degrading the summary quality.

The time for fuzzy categorization and summarization is affordable. Different performance metrics for categorization and summarization is good.

X. REFERENCES

[1] Sarker, G.(2010),An Unsupervised Natural Clustering with Optimal Conceptual Affinity, Journal of Intelligent Systems,19(3), 289-300. DOI: 10.1515/JISYS.2010.19.3.289

[2] Sarker, G.(2007),A Heuristic Based Hybrid Clustering for Natural Classification, International Journal of Computer, Information Technology and Engineering (IJCITAE),1(2), 79-86.

[3] Sarker, G.(2008),A Heuristic Based Hybrid Clustering, Institution of Engineers (I), Computer Engineering Division,Vol. 89, 7- 10.

[4] Sarker, G.(2010),An Unsupervised Natural Clustering with Optimal Conceptual Affinity, Journal of Intelligent Systems,19(3), 289-300. DOI: 10.1515/JISYS.2010.19.3.289

[5] Sarker, G.(2013),An Optimal Back Propagation Network for Face Identification and Localization ,International Journal of Computers and Applications (IJCA),ACTA Press, Canada.,35(2).,DOI 10.2316 / Journal .202.2013.2.202 – 3388.

[6] Sarker, G., Dhua, S., Besra, M. (2015),A Learning Based Handwritten Text Categorization,2015 International Conference on Advances in Computer Engineering and Applications, (ICACEA – 2015). ISSN: 978-1-4673-6910-7/15/$31.00 © 2015 IEEE.

[7] Sarker, G., Besra, M., Dhua, S.(2015),A Programming Based Handwritten Text Identification. 2015 International Conference on Advances in Computer Engineering and Applications, (ICACEA – 2015). ISSN: 978-1-4673-6910-7/15/$31.00 © 2015 IEEE.

[8] Sarker, G., Besra, M., Dhua, S.(2015),A Malsburg Learning BP Network Combination for Handwritten Alpha Numeral Recognition,2015 International Conference on Advances in Computer Engineering and Applications, (ICACEA – 2015). ISSN: 978-1-4673-6911-4/15/$31.00 © 2015 IEEE.

[9] Sarker, G., Dhua, S., Besra, M. (2015), An Optimal Clustering for Fuzzy Categorization of Cursive Handwritten Text with Weight Learning in Textual Attributes, 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS – 2015) held at Jadavpur University Kolkata INDIA 9-11 July, 2015, ISSN: 978-1-4799-8349-0/15/$31.00 ©2015 IEEE, pp. 6-11.

[10] Sarker, G., A Weight Learning Technique for Cursive Handwritten Text Categorization with Fuzzy Confusion Matrix – 2016 International Conference on Control, Instrumentation, Energy & Communication (CIEC), held at Kolkata, 978-1-5090-0035-7/16/$31.00 © 2016 IEEE,Jan. 20-30, 2016, pp 188-192.

[11] Sarker G., A New Technique For Extraction Based Text Summarization – 31stIndian Engineering Congress, 15-18 December, Kolkata 2016, The Institution of Engineers (India), pp 99-104.

[12] Sarker, G., Besra, M., Dhua, S.(2015),A Malsburg Learning BP Network Combination for Handwritten Alpha Numeral Recognition,2015 International Conference on Advances in Computer Engineering and Applications, (ICACEA – 2015). ISSN: 978-1-4673-6911-4/15/$31.00 © 2015 IEEE.

[13] S. Mori, C.Y. Suen and K. Kamamoto, "Historical review of OCR research and development" Proc. of IEEE, Vol. 80, pp. 1029–1058, July 1992.

[14] S. Impedovo, L. Ottaviano and S. Occhinegro, "Optical character recognition", International Journal Pattern Recognition and Artificial Intelligence, Vol. 5(1-2), pp. 1–24, 1991

[15] C. L. Liu and H. Fujisawa, "Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems", Int. Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, 2005.

[16] R. Plamondon and S. N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, Jan 2000.

[17] F. V. D. Heijden, "Edge and line Feature Extraction Based on Covariance Model", IEEE Transaction On Pattern Analtysis And Machine Intelligence Vol.11, No.1, January 1995.

[18] J.Pradeep, E.Srinivasan, and S.Himavathi, "Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No. 1, Feb 2011.

[19] O. Rohlik, P. Mautner, V. Matousek and J. Kemph, "HMM Based Handwritten Text Recognition Using Biometrical Data Acquisition Pen", Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation July 16-20,
2003, Kobe, Japan.

[20] H. Cao, R. Prasad and P. Natarajan, "Handwritten and Typewritten Text Identification and Recognition using Hidden Markov Models", International Conference on Document Analysis and Recognition, 2011