Pattern Taxonomy Mining for Text Categorization

Neeraj Kesavan
M.Tech CSE
SCOPE, VIT University
Vellore, TN, India

N. Jaisankar, PhD
Professor
SCOPE, VIT University
Vellore, TN, India

Ramani S.
Assistant Professor (Senior)
SCOPE, VIT University
Vellore, TN, India

ABSTRACT

Most of the text mining methods use term-based mining. All those methods are affected by common problems such as synonymy and polysemy. Mining of patterns have more advantage than other term based methods. Pattern Taxonomy Mining can be used to increase the effectiveness in the discovery of useful patterns. In addition to solving the common problems in term based mining, this paper tries to address the low occurring problems as well. Algorithms to deploy patterns and to evolve inner pattern are used to improve the effectiveness of pattern discovery. RCV1 text collection is used for experiments in this paper. Performance and execution of text categorization have significantly enhanced without any lose in the accuracy rate.

Keywords

Pattern Mining, Text Categorization, Inner Patterns, Pattern Taxonomy, Useful Information.

1. INTRODUCTION

In the recent years, relevance of data and its mining has increased. This is because of the huge amount of data generated both online and offline. Handling large collections of data is difficult. This difficulty is due to the fact that these data collections contain both relevant and irrelevant information. This is where data mining comes to rescue. Using data mining, we can mine the entire data and extract only the relevant information, which can be used for further processes.

Many types of data are available, such as image, text, audio, video etc. Among these, Textual data is the most used or generated data. Text Mining can be used to get the required subset from the textual datasets. This subset gives the collective meaning of the entire dataset. Text mining is to mine the collection of Textual Data to get knowledge or interesting information. Most of the text mining techniques are term based methods such as kNN, Rocchio, NB, SVM [6] etc. This paper elaborates on pattern based mining instead of term based. Patterns contain more meaning on semantic context. Patterns are the sequence of terms that are present in the text data. The effective usage and discovery of these meaningful patterns and its further formulations after mining is still under research.

Text mining based on terms has the advantage of reduced computational complexity. But has some problems too. Synonymy and Polysemy are the main problems that the term based methods suffer from. These can be rectified to a great extent by using pattern based mining methods. Many datasets are available now for text categorization studies, such as Reuters-21756, RCV1, TREC [1] [4] [10] etc.

The remaining section of the paper is structured into two portions: first portion describes the basic concept and related studies and the second portion gives details about system architecture and proposed system, followed by result of implementation and conclusion.

2. CONCEPT AND RELATED STUDIES

The term based text mining faces the problems of polysemy: one word can have more than one meaning according to the different situation in which it is used, and synonymy: more than one word can have the same meaning depends upon the situation in which they are used. So by simply considering each term we cannot conclude the meaning of the document. Which means the meaning of document that we are summarizing from the terms can be completely different from what it actually means.

This can be solved by using the patterns. Since the patterns are combination of more than one word, it can give more accurate meaning than other methods in the context of semantic meaning. For pattern based mining, PTM (Pattern Taxonomy Model) [12] [13] [14] can be used. But this still have problems such as lesser frequency of the terms i.e., the number of occurrence of some of the particular words which have higher importance in giving meaning to the document. This will affect the process of effectively getting meaning of the documents, because the terms’ weights are calculated and used in this method.

A collection of documents contain both positive and negative documents [5]. Positive in the sense it belongs to a particular category which the collection of documents represents and negative documents in the sense it does not belong to that particular category. So while making patterns we need to think about both of these documents.

Some documents may act like it belongs to a particular domain but it actually does not. These kind of documents need to be found out and have to be removed from the pattern base. At the same time the documents which have some desired patterns and is falsely determined to a negative type, need to be considered while pattern mining.

Categorization of document [1] is the main thread of this paper and this should be done with good accuracy and better performance. A collection of documents needs to be identified whether it belongs to a particular domain or not. For this, first thing that has to be done is to effectively discover some pattern base from documents which strictly belongs to that particular domain. And the next thing is to make the system learn these patterns. That is, the system has to be trained with the patterns; the true patterns and the false ones. True patterns are those set of patterns that essentially contains some patterns from strictly positive documents, and the false patterns are those set of patterns from documents that does not actually belongs to the particular domain but may have been falsely determined as a positive document. So the system has to be made aware of both kinds of patterns. Thus the dataset collection is divided in to positive and negative documents [5]. And the system is trained with both of these.

Once the training is done we can test for accuracy and then if appropriate accuracy is achieved, then we can go with the categorization process. The proposed system and its implementation are explained in the following sections.

3. ARCHITECTURE

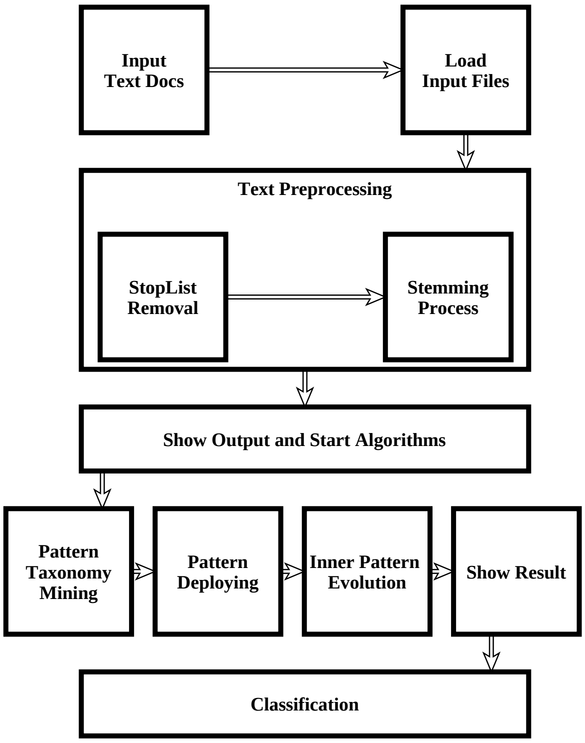

Fig 1: Proposed Architecture

4. PROPOSED SYSTEM

In pattern taxonomy method, patterns are represented as a tree like structure; the tree starts with single word and goes on increasing its height with the increase in the length of patterns. In the earlier [14] PTM the tree height is till the highest pattern that is derived from all unique words in the document. This is one of the hectic portion in which the execution and the performance is affected. Actually by restricting the height of the taxonomy tree, we can achieve a better execution and performance improvements without any lose in the accuracy. For this the algorithm needs to be modified.

4.1 Pattern Taxonomy

Patterns are discovered from the documents. All possible frequent and closed patterns are discovered [8] [13]. These patterns are made in to a taxonomy format or in to a tree like structure.

To start with, we first need to load the dataset and pre-process it. In the pre-processing step we are actually converting the documents in to separate tokens. For this stop-list removal and stemming is done [9]. In stop-list removal, unwanted and meaningless things such as article, punctuations, prepositions, symbols etc. are removed. In the stemming process, all the prefixes and suffixes from the words are removed and the words are retained in its normal form.

Document files may contain many paragraphs, P {p1, p2, p3, etc...}. Each of these paragraphs needs to be considered as separate documents in order to find patterns from that document. From each of the paragraphs, P = {p1, p2, p3...}, patterns are collected and formed from terms T {tm1, tm2, tm3, etc...}. For example

Table. 1 Sequence Terms in each of the paragraphs

Paragraph	Sequence Terms
P1	tm2 tm5 tm4
P2	tm1 tm3
P3	tm1 tm3 tm4 tm7
P4	tm2 tm5 tm6 tm4
P5	tm2 tm5 tm6 tm4
P6	tm1 tm3 tm4 tm7

From this we have to find the common or frequent patterns which are closed [7]. Patterns are called closed when they have a set to cover. Covering sets for the above are:

Table. 2 Covering set for the term sequences

Term Sequence	Covering Set
{tm2 tm5 tm4}	{p1,p4,p5}
{tm2 tm5}	{p1,p4,p5}
{tm2 tm4}	{p1,p4,p5}
{tm5 tm4}	{p1,p4,p5}
{tm2}	{p1,p4,p5}
{tm5}	{p1,p4,p5}
{tm1 tm3}	{p2,p3,p6}
{tm1}	{p2,p3,p6}
{tm3}	{p2,p3,p6}
{tm4}	{p1,p3,p4,p5,p6}

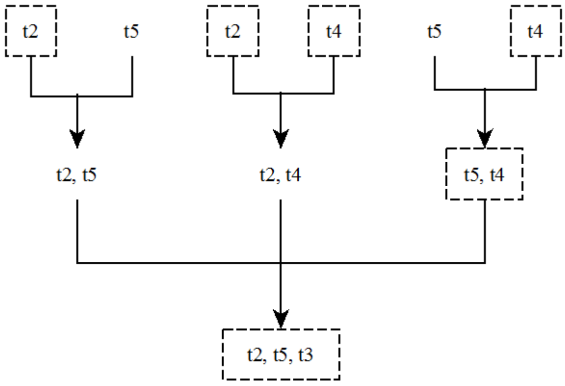

Fig 2: Pattern Taxonomy

Fig. 2 illustrates a good example of the pattern taxonomy for the recurrent patterns in Table 2, where in fact the nodes represent repeated Patterns and their covering packages; non-closed Patterns can be pruned; the sides are "is-a" connection. After pruning, some immediate "is-a" retaliations might be modified [14].

The pattern structure is how the patterns are made and it has a key role in the effectiveness of the pattern taxonomy model. The below given example shows how the patterns are discovered.

(data, mining) (mining, extraction)→(data, mining, extraction) new pattern is made.

(data, mining) (text, extraction)→Discarded

While comparing two term-sets the last and the first terms in the set are considered. If they are same, new pattern or term-set is made or else it is discarded. This is how the sequential patterns are discovered from the documents [7] [14].

4.2 Pattern Taxonomy Mining

In other methods, the evaluation and analysis of patterns are inappropriate. This leads to problems with the usage of the discovered patterns by mining methods. PTM tries to address this problem. Instead of using the patterns as it is, patterns are mapped to a general hypothesis space. So the re-evaluation and emphasizing of precise pattern can be achieved.

By reducing and simplifying the feature space and by using the term-weights to represent the significance level, low occurrence problem can be handled [2]. This in turn improves the efficiency and effectiveness of the pattern based system to discover knowledge.

The algorithm used for PTM is given below. In this, a set of sequential patterns returned from the SP_Mining methods is taken and each of the patterns is converted to an expanded form and merged. This gives deployed patterns ie, a term-weight pair set for each of the documents.

Algorithm 1: PTMining (Doc_List, Min_Supp)

```
Start
Pattern Vector Pv = null
For each of the documents in the Doc_List
Extract the frequent patterns FP
SeqP = SP_Mining(FP,Min_Supp)
initial pattern vector iPv = null
For each of the patterns p in SeqP
add expanded form of p to iPv
endFor
Pv = Pv U iPv
10.endFor
End
```

4.3 Sequential Pattern Mining

A pruning criterion is used in the SP_Mining for elimination of non-covered patterns during the discovery process of sequential patterns. This is a recursive algorithm, which repeats until there is no pattern left to discover. So output of SP_Mining is a set of closed sequential patterns which has support greater than or equal to the given minimum support [14].

Algorithm 2: SP_Mining (P_List, Min_Supp)

```
Start
SeqP contains patterns with length and support
constraints – pruning. SeqP←SeqP← {Pa ϵ| ∃ Pb
ϵP_List such as length (Pa) =length (Pb)-1∧Pa c Pb ∧
support (Pa) = support (Pb)}}
Store the patterns, SeqP = SeqP U P_List
For  each of the patterns p in P_List generate p-
projected database PDb
For each of the freq.Terms t in PDb P'=p join t
if support(P')>=Min_Supp then
PL'= PL' U P'
endFor
endFor
if Patterns are still left in PL' call
SP_Mining(PL',Min_Supp)
Return SeqP
End
```

4.4 Pattern Deploying Method

In PTM the input is a document set, but in this case a set of sequential patterns is given as input for deployment process. Here pattern deployment is done based on the support of the patterns in the sequential pattern set. As the pattern support is used for the re-evaluation of the patterns in PDeploying, the difference is reduced while estimating the values of significance.

Algorithm 3: PDeploying (SeqP)

```
Start
sum=0, Pvector=null
For each of the patterns p in SeqP
sum+=support(p)
endFor
For each of the p in SeqP
f=support(p)/(sum*length(p))
P'=null
For each of the terms in p
P'= P' U {(t,f)}
endfor
add P' to Pvector
endFor
End
```

4.5 Evolution of Inner Patterns

Deployed pattern evolution is done by i-Pattern algorithm. A set of d-patterns, ie deployed patterns and a set of documents (positive and negative) are given as input to this algorithm [14]. This algorithm returns a set of term-weight pair [2]. And this can be used in the testing. This algorithm is used to discover all the offenders of negative type documents i.e, to collect all those deployed patterns that have similar pattern structure with negative documents. The algorithm SufflePattern is called with the collected offenders to perform the main functionality of i-Pattern algorithm.

Algorithm 4: i-Patterns (DP_List, Docs)

```
Start
tw_pair=null
t= thrshold(Docs)
For each doc di in Docs
if threshold(di)>t
dp= {Ap in Docs|termset(Ap) ∩ di!=Null }
Ap=ShufflePattern(di,dp)
endif
For each of the deployed patterns in di in Docs
tw_pair+=Ap //append operation
endFor
endFor
return tw_pair
End
```

4.6 Shuffling Method

The distribution of term weight in the deployed pattern is tuned using the shuffling method. Two types of offender are considered here, complete offenders and partial offenders. The deployed patterns are removed from its set if it is a complete offender. For partial offenders, an offering and base value is calculated and based on these values the term weights are tuned.

Algorithm 5: ShufflePattern(Doc,dp)

```
Start
    For each of th deployed pattern d in dp
    if termset(d)⊆Doc //Complete offender
    DP_list-={Ap}
    else //Partial Offender
    Calculate offering value
    Calculate base value
    For each term ti in termset(Ap)
    if ti in d //Shrink the offenders weight
    ti.wt*=1/μ
    else //Shuffling the weights
    ti.wt*=(1+offering/base)
    endif
    endFor
    endif
    endFor
    return DP_List
End
```

4.7 Limiting Patten Taxonomy Height

In the earlier work, entire length of pattern that is possible to make from a document is used for the tree like pattern taxonomy [14]. This is wasting the execution time of the entire system as it is not giving any significant advantage to the system. At lower levels of the taxonomy lesser pattern length can achieve the output with more performance and less execution period. Pattern_Limiter in this algorithm is limiting the pattern length and the height of taxonomy.

Algorithm 6: Pattern_Limiter()

```
Start
    Set Pattern Length pl // No.of words in each pattern
    for each 0 to pl Call SP_Mining (term_set)
    ts =Initial term_set
    for each combinations of words in ts
    Call SP_Mining (term_set)
    end for
    end for
End
```

5. RESULTS AND DISCUSSION

The implementation of the proposed system is successfully completed and is tested for its working with dataset inputs. Performance can be measured using several measuring standards based on precision and recall [11]. A criterion that can be used for performance evaluation is, F_{β}-measure, where β is a parameter that attributes degree of importance to precision and recall [3]. To attribute equal importance to precision and recall, β=1 is used in the experiments [14]. When β=1, F_{β}-measure can be expressed as.

$$F_{\beta=1} = \frac{2*Precision*Recall}{Precision+Recall}$$

The test results with dataset show significant improvement in the performance of the system and the execution time is reduced noticeably. And the result comparison with other methods also shows the improvement achieved by this method. The result comparison table is given below.

Table 3. Result Evaluation with Dataset RCV1

Data Set RCV1	
Topic: Corn	
Train-Set: 182	**Test-Set:** 57
F1 Values of different methods	
Methods	**F1 Value**
NB	65
kNN	78
Trees	92
SVM	90
PTM	98.18

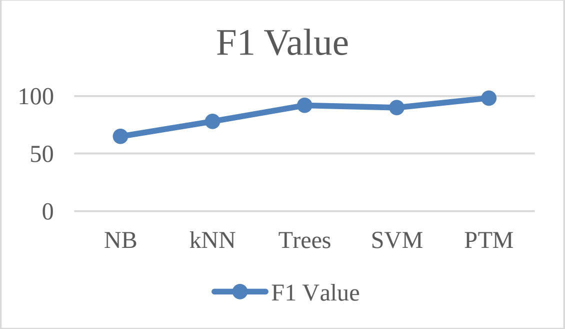

Fig 3. Comparison of F $_{\beta=1}$ values to highlight performance.

6. CONCLUSION AND FUTURE WORK

Amongst various text mining techniques present today, most of them are methods based on the terms. All of these suffer from key problem such as misinterpretation of the meaning of the word term, i.e. word terms may possess various meaning according to the context in which they are used, and more than a single word may project same meaning in a given situation. Even though term based methods give faster outcome the aforementioned problems are actually major concerning factors in terms of the accuracy of the system while doing the categorization of document files on a set of large collection. As a solution to these problems, pattern based method of mining is used since patterns give semantic type meaning. So this will help to understand what actually a certain document means from a set of patterns from that document text file. The key problem with the effective usages of the patterns is that the pattern making and its further deploying will take more time as of now. So PTM is used with the modification in the basic setup to achieve better performance. The existing PTM uses entire lengthy patterns which take comparatively more time to process. Low rate of occurrence of patterns is also a problem. The proposed solution is capable of addressing low occurrence by making patterns from all combinations of words in the text file. To reiterate, by limiting the height of the taxonomy tree and by creating all possible combination of patterns, performance improvement and execution efficiency are achieved.

Effective usage and management of the term and pattern space can be considered as further area of improvement. If a data model for the training patterns can be defined and loaded in to the system, the performance can be improved by avoiding repeated pattern discovery. This has a scope for future work.

7. REFERENCES

[1] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical Report 941, Norwegian Computing Center, June 1999.

[2] S.T. Dumais, "Improving the Retrieval of Information from External Sources," Behavior Research Methods, Instruments & Computers, 1991, 23(2), pp. 229-236.

[3] David D. Lewis. 1995. Evaluating and optimizing autonomous text classification systems. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '95), Edward A. Fox, Peter Ingwersen, and Raya Fidel (Eds.). ACM, New York, NY, USA, 246-254.

[4] Lewis, D.D., (2004), The LYRL2004 Distribution of the RCV1-v2 Text Categorization, http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm, Accessed on June 2016.

[5] Yuefeng Li, Abdulmohsen Algarni, and Ning Zhong. 2010. Mining positive and negative patterns for relevance feature discovery. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10). ACM, New York, NY, USA, 753-762.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

[7] Manan Parikh, Bharat Chaudhari and Chetna Chand. "A Comparative Study of Sequential Pattern Mining Algorithms," International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 2, February 2013, pp. 103-109.

[8] J. Pei, J. Han, and R. Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets," Proceedings 2000 ACM-SIGMOD International Workshop Data Mining and Knowledge Discovery (DMKD '00), pp. 11-20, May 2000.

[9] M.F. Porter, "An Algorithm for Suffix Stripping," Program, Automated Library and Information Systems, vol. 14, no. 3, pp. 130-137, 1980.

[10] S. Robertson and I. Soboroff. The TREC 2002 filtering track report. In Proceedings of TREC'02, 2002.

[11] Wikipedia, (2007) Precision and Recall, https://en.wikipedia.org/wiki/Precision_and_recall, Accessed on June 2016.

[12] S. t. Wu, Y. Li and Y. Xu, "Deploying Approaches for Pattern Refinement in Text Mining," Sixth International Conference on Data Mining (ICDM'06), Hong Kong, 2006, pp. 1157-1161.

[13] Sheng-Tang Wu, Yuefeng Li, Yue Xu, Binh Pham and Phoebe Chen, "Automatic Pattern-Taxonomy Extraction for Web Mining," Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on, 2004, pp. 242-248.

[14] N. Zhong, Y. Li and S. T. Wu, "Effective Pattern Discovery for Text Mining," in IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 1, pp. 30-44, Jan. 2012.