Mining a Data Reasoning Model for Personalized Text Classification

Luepol Pipanmaekaporn* and Yuefeng Li†
Computer Science Discipline, Faculty of Science and Technology
Queensland University of Technology, Brisbane, QLD 4001, Australia
Email: luepol.p@gmail.com* and y2.li@qut.edu.au†

***Abstract*—It is a big challenge to acquire correct user profiles for personalized text classification since users may be unsure in providing their interests. Traditional approaches to user profiling adopt machine learning (ML) to automatically discover classification knowledge from explicit user feedback in describing personal interests. However, the accuracy of ML-based methods cannot be significantly improved in many cases due to the term independence assumption and uncertainties associated with them.**

This paper presents a novel relevance feedback approach for personalized text classification. It basically applies data mining to discover knowledge from relevant and non-relevant text and constraints specific knowledge by reasoning rules to eliminate some conflicting information. We also developed a Dempster-Shafer (DS) approach as the means to utilise the specific knowledge to build high-quality data models for classification. The experimental results conducted on Reuters Corpus Volume 1 and TREC topics support that the proposed technique achieves encouraging performance in comparing with the state-of-the-art relevance feedback models.

***Index Terms*—Personalized text Classification, User Profiles, Relevance Feedback, Reasoning Model, and Data Mining**

I. Introduction

As the vast amount of online information available causes information overloading, the demand for personalized approaches for information access increases. One of the key techniques for personalized information access is personalized text classification [1], [4], where a system is able to retrieve or filter contents according to personal interests [9]. As for personalization, a user profile is used to represent user interests and perferences.

It is not uncommon that hand-coding user profiles is impractical since users may be unsure of their interests or not have any technical knowledge to describe their profile. It is hence preferable to directly learn classifiers from examples. A common user profiles acquiring approach is to explore relevance feedback (RF). In particular, a user is given to express his/her opinions by deciding which documents are *relevant* or *non-relevant* to the user. By using the explicit feedback, machine learning (ML) techniques could be adopted to learn a text classifier that represents the user interest [23], [24] or search intent [40]. For example, Rocchio [12], [20] and SVMs [13], [14] are two effective learning algorithms in this literature.

Nevertheless, the performance of ML-based approaches to RF often cannot significantly improve. This is since the nature of ML techniques that require a large training set to achieve good performance whereas in fact the number of feedback documents given by a user is small. Furthermore, ML-based approaches typically deal with training documents with the term independence assumption and ignore any syntactic and semantic information of correlations between terms. As a result, they may miss some useful terms that are added into user profiling models [3], [6], [21].

Data mining (DM) based approaches to relevance feedback have recently given great interests [34], [35], [39]. These approaches basically discover frequent patterns that capture frequent terms and their relationships in text and consequently utilise the discovered patterns for constructing relevance models. In [35], the authors adopted data mining to mine relevant documents in order to discover a set of sequential patterns. A document evaluation function is formed by those patterns to score new documents. A *deploying* method was proposed [34] to solve the problem of low-frequency occurrence of patterns in text. Instead of patterns, a weighted vector of terms is generated by discovered patterns and used for building the relevance model. Some deploying-based approaches (e.g. IPE [39]) attempt to improve the quality of relevance model by using negative feedback. Although experimental results conducted on RCV1 data collection illustrate the usefulness of frequent patterns for personalized text search, we believe that the existing approaches may not be able to obtain high-quality relevance feedback models. Firstly, these approaches focus on building relevance models that use patterns, but ignore the attempt to select a small set of high-quality patterns. Furthermore, it is still not clear how to effectively deal with the result of pattern mining to improve the effectiveness of relevance feedback models.

Motivated by these issues, this paper presents a novel relevance feedback approach for discovering user profiles from text using data mining and reasoning techniques. In specific terms, it discovers features (frequent patterns) from relevant and non-relevant text and constraints specific ones by reasoning rules to eliminate some conflicting information. To construct the user profile model, we developed a Dempster-Shafer (DS) approach that allows to establish the connection between patterns and terms. It also allows to incorporate the uncertain nature of text features (i.e., terms and patterns) for modelling user's interests. The experimental results conducted on RCV1 and TREC text collections [22] support that the data reasoning approach achieves encouraging performance as compared to the state-of-the-art techniques.

TABLE I
A SET OF PARAGRAPHS

$Paragraph$	$Terms$
dp_1	t_1 t_2
dp_2	t_3 t_4 t_6
dp_3	t_3 t_4 t_5 t_6
dp_4	t_3 t_4 t_5 t_6
dp_5	t_1 t_2 t_6 t_7
dp_6	t_1 t_2 t_6 t_7

In summary, our contributions include

- We propose a novel relevance feedback approach for personalized text categorization.
- We analysis text patterns by observing their semantic relationships and devising reasoning rules that investigate specific patterns to describe user interests.
- We propose a novel method for constructing user profiles using frequent patterns in text to improve performance of text categorization.

The rest of the paper is organized as follows: Section 2 gives some basic definitions of frequent patterns in text. In section 3, we provide a data mining framework for discovering features in relevant and non-relevant text. We also describe a novel feature selection method based on the investigation of reasoning rules. Section 4 presents how Dempster-Shafer approach facilitate the utilisation of the discovered patterns for constructing the user profile model. Extensive experimental results are presented in Section 5 and related work is discussed in Section 6, following by conclusions in Section 7.

II. BACKGROUND

Let D be a training set of documents, including a set of positive (relevant) documents, D^+, and a set of negative (irrelevant) ones, D^-. When splitting a document into paragraphs, a document d can also be represented by a set of paragraphs $PS(d)$.

A. Frequent and Closed Patterns

Let $T = \{t_1, t_2, \dots, t_m\}$ be a set of terms which are extracted from D^+. Given X be a set of terms (called a *termset*) in document d, $coverset(X)$ denotes the covering set of X for d, which includes all paragraphs $dp \in PS(d)$ where $X \subseteq dp$, i.e., $coverset(X) = \{dp | dp \in PS(d), X \subseteq dp\}$. The *absolute* support of X is the number of occurrences of X in $PS(d)$: $sup_a(X) = |coverset(X)|$. The*relative* support of X is the fraction of the paragraphs that contain the pattern: $sup_r(X) = \frac{|coverset(X)|}{|PS(d)|}$. A termset X called*frequent pattern* if its sup_a (or sup_r) $\geq min\_sup$, a minimum support.

Table I lists six paragraphs for a given document d, where $PS(d) = \{dp_1, dp_2, \dots, dp_6\}$, and duplicate terms are removed. Assume $min\_sup = 3$, ten frequent patterns would be extracted as shown in Table II.

Given a set of paragraphs $Y \subseteq PS(d)$, we can define its *termset*, which satisfies

$$termset(Y) = \{t | \forall dp \in Y \Rightarrow t \in dp\}$$

By defining the closure of X as:

$$Cls(X) = termset(coverset(X))$$

TABLE II
SEQUENTIAL PATTERNS AND COVERING SETS

$Frequent\ Pattern$	$Covering\ Set$
$\{\mathbf{t_3, t_4, t_6}\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{\mathbf{t_1, t_2}\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_1\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{\mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4, dp_5, dp_5\}$

a pattern (or termset) X is *closed* if and only if $X = Cls(X)$. Let X be a closed pattern. We have

$$sup_a(X_1) < sup_a(X) \quad (1)$$

for all patterns $X_1 \supset X$.

B. Closed Sequential Patterns

A *sequential pattern* $X =< t_1, \dots, t_r >$ $(t_i \in T)$ is an ordered list of terms, where its $sup_r \geq min\_sup$. A sequence $s_1 =< x_1, \dots, x_i >$ is a *sub-sequence* of another sequence $s_2 =< y_1, \dots, y_j >$, denoted by $s_1 \sqsubseteq s_2$, iff $\exists j_1, \dots, j_i$ such that $1 \leq j_1 < j_2 \dots < j_i \leq j$ and $x_1 = y_{j_1}, x_2 = y_{j_2}, \dots, x_i = y_{j_i}$. Given $s_1 \sqsubseteq s_2$, we usually say s_1 is a *sub-pattern* of s_2, and s_2 is a *super-pattern* of s_1. To simplify the explanation, we refer to sequential patterns as patterns.

As the same as those defined of normal patterns, we define the *absolute support* and *relative support* for a pattern (an ordered termset) X in d. We also denote the covering set of X as $coverset(X)$, which includes all paragraphs $ps \in PS(d)$ such that $X \sqsubseteq ps$, i.e., $coverset(X) = \{ps | ps \in PS(d), X \sqsubseteq ps\}$. X is then called a $frequentpattern$ if $sup_r(X) \geq min\_sup$. By using Eq. (1), a frequent sequential pattern X is *cloesd* if not $\exists$ any super-pattern X_1 of X such that $sup_a(X_1) = sup_a(X)$.

To improve the efficiency of finding all closed sequential patterns from training documents, an algorithm, $SPMining(D^+, min\_sup)$, was introduced by [35]. The *SPMining* algorithm used well-known *Apriori* property to narrow down the searching space.

C. Dempster-Shafer theory

Dempster-Shafer (hereafter DS) [37] is a statistically based technique for combining evidence. It can be considered a generalization of Bayesian theory as it allows assignment of probability to uncertain events, offering a way to represent ignorance or uncertainty. A beneficial characteristic of DS is the ability to use partial knowledge over propositions and represent uncertainty as part of a modelling process. Recently, there are an increasing number of developments and applications using DS approach. In particular, generalized evidence theory [10], [18], Data Fusion [32], Machine Learning [7], [8], association rules mining [17], Information Retrieval [26], [30], Web mining models [15], [36].

In general terms, DS deals with a finite set of exclusive and exhaustive propositions, called the *frame of discernment* (denoted by Ω). All the subsets of Ω belong to the power set of Ω, denoted by 2^{Ω}. A strength of subset of elements in Ω is given by the definition of a mass function $m : 2^{\Omega} \rightarrow [0, 1]$, which provides a measure of uncertainty, applied over all the subsets of elements in the frame of discernment. The mass function also satisfies the following properties:

(1) $m(\emptyset) = 0$ and

(2) $\sum_{A \in 2^{\Omega}} m(A) = 1$

DS provides a rule, known as the Dempster's rule of combination [25], for combining evidence, possibly originating from different sources of data (e.g. Sensors). The combination yields a probability mass assigned to a subset of Ω, given a subset of propositions A, characterized by a mass distribution m_1 and subset of propositions B, characterized by a mass distribution m_2. The normalized version of the combination rule is the following:

$$m_1 \oplus m_2(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{\sum_{B \cap C \neq \emptyset} m_1(B) m_2(C)} \tag{2}$$

for all $A \in 2^{\Omega}$, where $m_1 \oplus m_2(A)$ denotes the combine evidence.

In the DS, probability masses applied over all the subsets of elements in the frame of discernment can be used to infer the mass for the single elements as the means to make decisions. The masses are represented by probability functions called *Pignistic* probabilities [27]. The pignistic probability is defined as:

$$BetP(A) = \sum_{B \subseteq \Omega} \frac{|A \cap B|}{|B|} \frac{m(B)}{(1 - m(\emptyset))} \tag{3}$$

for all subsets $A \subseteq \Omega$. A shortcoming of DS is related to the use of masses instead of probability measures and high involvements for users to explicitly provide values for the mass functions.

A shortcoming of DS is related to the use of masses instead of probability measures and difficulties in coming up with these values for the mass functions [37].

III. Relevant Feature Discovery

Vector Space Model (VSM) is the popular choice for representing information in text documents since it is efficient and effective for text processing. However, it fails to capture semantic information which is often represented by relations between terms (i.e., syntactic or semantic phrases). Finding phrases in text is related to mining frequent subsequences in sequence collections. We hence apply data mining to discover useful features available in relevant text. Mining sequential patterns offers to generate both low-level features (terms) and high-level ones (phrases) in sentences or paragraphs w.r.t. frequency. They enjoy statistical properties since they are frequent. Furthermore, many noisy patterns could be removed w.r.t. the minimum support constraint. We adopt $SPMining$ algorithm [35] (also used in [34] [39]) to discover frequent subsequences (hereafter patterns) in paragraphs of positive documents D^{+}. For all positive documents $d_i \in D^{+}$, the *SPMining* algorithm finds a set of patterns based on a given min_sup to obtain the following vector:

$$\overrightarrow{d_i} = \langle (p_{i_1}, f_{i_1}), (p_{i_2}, f_{i_2}), \ldots, (p_{i_m}, f_{i_m}) \rangle \tag{4}$$

where p_j in pair (p_j, f_j) denotes a pattern and f_j is its frequency in d_i. The result of this algorithm is a set of document vectors, which can be expressed as follows.

$$\eta = \left\{ \overrightarrow{d_1}, \overrightarrow{d_2}, \ldots, \overrightarrow{d_n} \right\}$$

where $n = |D^{+}|$.

A. A weighted combination operator

For each vector $\overrightarrow{d_i} \in \eta$, the frequency of pattern can imply the pattern's significance in the context of document. As training documents may contain a pattern more than once, it is important to determine which patterns are significant in aspects of information use. However, existing data mining algorithms usually exclude the local support information by only considering their binary presence and absence in training documents. As a result, they lose in the local pattern's significance that may provide some insights. For example, considering two patterns p and q that occur 20 times and 2 times in the same document with equal importance can be incorrect.

To achieve this, we apply the idea of data fusion to effectively combine multiple sets of patterns in different documents into a single one. In information retrieval, data fusion has been used to combine results from different retrieval models, different document representations, different query representations and so on, to improve effectiveness [30], [33].

We first define a score function ρ_i that assigns a score to a pattern p_j based on its frequency in a document d_i as the following equation:

$$\rho_i(p_j) = \begin{cases} \frac{f_j}{\sum_{p_k \in \overrightarrow{d_i}} f_k} & ; p_j \in \overrightarrow{d_i} \\ 0 \quad ; otherwise \end{cases} \tag{5}$$

where f_j denotes the absolute support value (Sup_a) of pattern p_j in document d_i. Given two score functions ρ_a and ρ_b belonging to document d_a and d_b respectively, we define a weighted linear combination operator $\oplus$ to compose the two score functions into the combined score for pattern p_j. This operator can be found as the following equation:

$$\rho_a \oplus \rho_b(p_j) = \frac{1}{K} \times \begin{cases} w_a \times \rho_a(p_j) + w_b \times \rho_b(p_j) & ; p_j \in \overrightarrow{d_a} \cap \overrightarrow{d_b} \\ w_a \times \rho_a(p_j) \quad ; p_j \notin \overrightarrow{d_b} \\ w_b \times \rho_b(p_j) \quad ; p_j \notin \overrightarrow{d_a} \\ 0 \quad ; p_j \notin \overrightarrow{d_a} \cup \overrightarrow{d_b} \end{cases} \tag{6}$$

where w_a and w_b be a user-defined weight associated with document d_a and d_b respectively and $K = w_a + w_b$, which is a weight normalisation. The weights reflect the importance of feedback documents, which can be the document's length or the degree of perceived relevance given by a user or IR system. If all the documents are equally weighted, then the combined score of pattern is fairly averaged.

Let SP^+ be a set of sequential patterns collected from all the relevant training documents in a collection D^+, i.e., $SP^+ = Sp_1 \cup Sp_2 \cup \cdots \cup SP_{|D^+|} = \bigcup_{i=1}^{|D^+|} Sp_i$. For each pattern $p_j \in SP^+$, the score assigned to the pattern p_j can calculated by combining all the score functions of documents in a document collection as the following equation:

$$\rho_c(p_j) = \bigoplus_{i=1}^{|D^+|} \rho_i = \frac{1}{K} \sum_{i=1..|D^+|} w_i \times \rho_i(p_j) \qquad (7)$$

where $\rho_c(p_j)$ returns the combined support given to the pattern p_j, w_i is a weight associated with document d_i and $K = w_1 + w_2 + \cdots + w_{|D^+|}$. Since we only know which documents are positive or negative, but not which one is more important, in this paper all training documents are equally treated (i.e., 1).

B. From relevant features to specific features

Although patterns provide highly detailed descriptors for document representation, their large number of generated patterns may hinder their effective use. This is since many of these patterns are redundant and conflict. Adding such patterns can harm the classification accuracy due to the overfitting effect; however, it is very difficult to identify which patterns are noisy since they depend on users' perspectives for their information needs [39].

To this end, we propose a novel method to detect and eliminate patterns that are conflict. The idea is to check which patterns have been used in the context of the non-relevant data. We first define two kinds of errors: *total conflict* error and *partial conflict* error.

Definition 1 (total conflcit). *Given a pattern $p \in R$, p is called total conflict with a category $\overline{R}$ if $\exists q \in \overline{R}$ and $termset(p) \subseteq termset(q)$.*

Definition 2 (partial conflcit). *Given a pattern $p \in R$, p is called partial conflict with a category $\overline{R}$ if $\exists q \in \overline{R}$ and $termset(p) \cap termset(q) \neq \emptyset$.*

To apply this idea, we discover patterns from negative documents in D^- and consequently fuse them into a single collection, defined as SP^-, as we did in positive documents. Based on the above definitions, we identify all patterns in response to the following rules:

$$\begin{array}{l} S^+ = \{p | p \in SP^+, \forall q \in SP^- \Rightarrow p \not\subseteq q\} \\ S^- = \{q | q \in SP^-, \exists p \in SP^+ \Rightarrow q \cap p \neq \emptyset\} \\ N = (SP^+ \cup SP^-) - S^+ - S^- \end{array}$$

where $S^+ \cap S^- \cap N = \emptyset$. SP^+ and SP^- are two categories of patterns in the relevant and non-relevant data respectively. For the relevant category, non-conflict patterns contain termsets that are specific to user's interests or a user because they never overlap with any patterns from the non-relevant category while partial conflict ones are termsets that share a part with some of those patterns. All these patterns are classified into S^+. On the other hand, total conflict patterns in the relevant category are classifed into N since they contain termsets that may frequently occur in the context of non-relevant data.

A collection S^- consists of all conflict patterns in the non-relevant category. Such patterns are useful to identify noisy terms in the relevant data. Also, non-conflict patterns in the non-relevant category is classified into N because they are irrelevant data.

Once patterns were classified, we store all patterns in S^+ and S^- and remove patterns contained in the collection N.

IV. User Profile Construction

In this section, we describe our approach to construct the user profile model by using the specific knowlege.

A. Mapping patterns to belief functions

The initial user profile is first built based on two category of patterns S^+ and S^-. Let Ω consists of n terms extracted from all patterns in the two pattern collections, i.e., $\Omega = \{t_1, t_2, \ldots, t_n\}$. We define a set-valued mapping $\psi :: S^+ \cup S^- \rightarrow 2^\Omega$ to associate the relationship between patterns and the term space in Ω to generate mass functions.

Based on this mapping, we define a mass function $m^+ : 2^\Omega \rightarrow [0, 1]$ on Ω, the set of terms, called *positive mass* function, which satisfies

$$m^+(A) = \begin{cases} 0 \quad if \quad A = \emptyset; \\ \frac{\rho_c(\{p | p \in S^+, \Gamma(p) = A\})}{\sum_{B \subseteq \Omega} \rho_c(\{q | q \in S^+, \Gamma(q) = B\})} \quad , otherwise \end{cases} \qquad (8)$$

for all $A \subseteq \Omega$, where $\rho_c(p)$ returns the combined support of pattern p obtained by Eq.(5) and B is a subset on Ω space.

As we did in the positive data, patterns from non-relevant data (i.e., S^-) can be used to generate mass functions, defined as *negative mass* functions ($m^-(A)$). A positive mass function $m^+(A)$ represents the strength that supports set A, a set of terms, whereas a negative mass function $m^-(A)$ means the contrary.

Figure 1 illustrates an example of mapping knowledge (patterns) to mass functions on Ω space.

A main advantage of representing the discovered knowledge with belief functions is that uncertainties associated with text features (i.e., frequent terms and patterns) can be represented.

B. Weight assignment by belief functions

In order to reason with the derived mass functions, we present the new idea to assign weights of terms in the profile vector. The main advantage of the weight assignment method is that it takes uncertainties represented by mass functions in estimating term weights.

For each term $t_i \in \Omega$, we first transfer positive mass functions into a *pignistic probability* [28] as the following functions:

$$Pr_{m^+}(t_i) = \sum\nolimits_{\emptyset \neq A \subseteq \Omega, t_i \in A} \frac{m^+(A)}{|A|} \qquad (9)$$

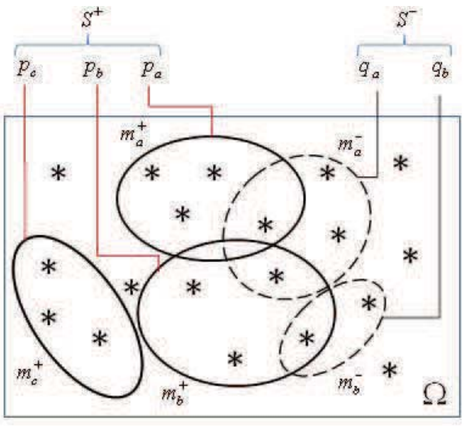

Fig. 1. An example of mass functions generated by discovered patterns

where A denotes a subset of elements in Ω and $|A|$ is the number of elements in A. The pignistic value represents the expected probability assigned to single elements in the frame of discernment for betting [28]. In our case, we use the resulting probability as the means to score terms in the profile vector corresponding to their distribution in the term dependency data (i.e, positive and negative data). The high value assigned to a term represents the high importance of the term in the underlying data. The pignisitic probability assigned to a term t_i with negative mass functions can be estimated by

$$Pr_{m^-}(t_i) = \sum_{\emptyset \neq A \subseteq \Omega, t_i \in A} m^-(A)/|A|.$$

Finally, the two probability functions are combined to estimate the weight of each term t_i in the user profile Ω as the following equation:

$$w(t_i) = \frac{Pr_{m^+}(t_i) \times (1 - Pr_{m^-}(t_i))}{1 - \min\{Pr_{m^+}(t_i), Pr_{m^-}(t_i)\}} \tag{10}$$

The term's weight measures the term's importance in respect to the user's interests. When the pignistic value of a term given by positive mass functions ($Pr_{m^+}(t_i)$) is high, the term tends to be a good identifier for identify relevant documents. As a result, the term weight value tends to be high. Conversely, the pignistic probability with negative mass functions ($Pr_{m^-}(t_i)$) is supposed to be negatively correlated with the user's topic of interest. When this value is high indicating that the term tend to be used in describing other topics, the important weight given to the term is reduced as a consequence.

A document evaluation function is built for the use of user profile in document filtering. Given a new document d, the relevance score given to the document d can be calculated as the following function:

$$R(d) = \sum_{t \in d} \frac{tf(t)}{\sum_{t_j \in d} tf(t_j)} \times support(t) \tag{11}$$

where $support(t) = w(t)$ if $t \in \Omega$; otherwise $support(t) = 0$ and $tf(t)$ denotes the term frequency of term t in document d.

It is easy to apply a threshold strategy to the document evaluation function for making a binary decision, aiming to predict the class labels of document d into relevant and non-relevant to a user. Given a threshold value ζ, if $r(d) \geq \zeta$ then the document d is *relevant*; otherwise it is *non-relevant*. The best value of ζ can emperically estimated.

V. Experimental Evaluation

In this section, we evaluate the proposed approach. We conduct experiments on RCV1 data collection and TREC topics. We also discuss the testing environment including the data collection, baseline models, and evaluation methods. The data reasoning model (afterhere *DRM*) is a supervised approach that needs a training set including both positive (relevant) documents and negative (non-relevant) documents from an individual user.

A. RCV1 data collection and TREC topics

Reuter Corpus Volume 1 (RCV1) is used to test the effectiveness of the proposed model. RCV1 corpus consists of all and only English language stories proposed by Reuter's journalists between August 20,1996, and August 19,1997, a total of 806,791 documents that cover very large topics and information [22]. For each topic, some documents in RCV1 data collection are divided into a training set and a testing set. TREC(2002) has developed and provided 50 assessor topics for the filtering track, aiming at building a robust filtering system [29]. The relevance judgements of documents in the assessor topics have been made by human assessors of the National Institute of Standards and Technology (NIST),i.e., assessor topics. According to [29], the justification of enough using the 50 assessor topics for evaluating robust IF systems was given. In this study, we use only the 50 assessor topics for performance evaluation in the proposed model.

All documents in RCV1 are marked in XML. To avoid bias in the experiments, all the meta-data information in the collection have been ignored. The documents are treated as plain text documents by preprocessing the documents. The tasks of removing stop-words according to a given stop-words list and stemming terms by applying the Porter Stemming algorithm are applied.

B. Baseline Models and Settings

We group baseline models into two main categories. The first category includes a number of data mining (DM) based methods for IF (i.e., PTM [35], PDS [34] and IPE [39] while the second category includes two effective machine learning models in text categorization and filtering (i.e. Rocchio [12] and SVM [13]). DM-based models were discussed in the section Related work.

1) DM-based models:: Both PTM and PDS models use only positive features (i.e, patterns for the case of PTM and terms for the case PDS) from training relevant documents to generate user profiling models while IPE uses both positive and negative features. For data mining models, the minimum support threshold (min_sup) is an important parameter and is sensitive for a specified data set. We set $min\_sup = 0.2$ (20% of number of paragraphs in a document) for all baseline models in the category (also DRM) since this value was recommended best for this data collection [34], [35], [39].

2) Machine Learning based models:: The Rocchio algorithm has been widely adopted in text categorization and filtering [12], [24]. The Rocchio builds a Centroid for representing user profiles. The centroid $\vec{c}$ of a topic can be generated as follows:

$$\alpha\frac{1}{|D^+|}\sum_{\vec{d}\in D^+}\frac{\vec{d}}{||\vec{d}||}-\beta\frac{1}{|D^-|}\sum_{\vec{d}\in D^-}\frac{\vec{d}}{||\vec{d}||} \qquad (12)$$

where $||\vec{d}||$ be normalized vector for document d. α and β be a control parameter for the effect of relevant and nonrelevant data respectively. According to [5], [12], there are two recommendations for setting the two parameters: $\alpha = 16$ and $\beta = 4$; and $\alpha = \beta = 1.0$. We have tested both accommodations on assessor topics and found the latter recommendation was the best one. Therefore, we let $\alpha = \beta = 1.0$.

SVM is a state-of-the-art classifier [13]. In our experiments, we used the linear kernel since it has been proved to be as powerful as other kernels when tested on the Reuters-21578 data colleciton for text classification [24]. We hence used the following decision function in SVM:

$$h(x) = sign(w.x + b) = \begin{cases} +1 & if(w.x + b) > 0 \\ -1 & otherwise \end{cases}$$

where x is the input object; $b \in R$ is a threshold and $w = \sum_{i=1}^{l} y_i\alpha_i x_i$ for the given training data:$(x_i, y_i), \ldots, (x_l, y_l)$, where $x_i \in R^n$ and $y_i = +1(-1)$, if document x_i is labeled positive (negative). $\alpha_i \in R$ is the weight of the sample x_i and satisfies the constraint:

$$\forall_i : \quad \alpha_i \geq 0 \quad and \quad \sum_{i=1}^{l}\alpha_i y_i = 0 \qquad (13)$$

The SVM here is used to rank documents rather than to make a binary decision, and it only uses terms based features extracted from training documents. For this purpose, threshold b can be ignored. For the documents in a training set, we know only what are positive (negative), but not which one is important. We assign the same α_i value (i.e., 1) to each positive document first, and then determine the same α_i (i.e., α') value to each negative document based on the Eq. (11). Therefore, a testing documents d is scored by the function $r(d) = w.d$ where . means *inner products*; d is the term vector of the testing document; and

$$w = \left(\sum_{d_i\in D^+} d_i\right) + \left(\sum_{d_j\in D^-} d_j\alpha'\right) \qquad (14)$$

For each assessor topic, we choose 150 terms in the positive documents, based on *tf*idf* values for all ML-based methods.

C. Results

The effectiveness is determined by five different measures commonly used in Information Retrieval (IR): The average precision of the top 20 documents ($top - 20$), F_1 measure, Mean Average Precision (MAP), the break-even point (b/p), and Interpolated Average Precision (IAP) on 11−points. Precision (p), Recall (r), and F_1 are calculated by the following functions:

$$p = \frac{TP}{TP + FP}, r = \frac{TP}{TP + FN}, F_1 = \frac{2 * p * r}{p + r}$$

where TP is the number of documents the system correctly identifies as positives; FP is the number of documents the system falsely identifies as positives; FN is the number of relevant documents the system fails to identify. The larger a $top-20$, MAP, b/p, F_1 measure score is, the better the system performance. 11−points measure is also used to compare the performance of different systems by averaging precisions at 11 standard recall values (i.e., recall = 0.0, 0.1,...,1.0).

DRM is firstly compared with all data mining based models. We also compare DRM with the state-of-the-art machine learning based models underpinned by Rocchio and SVM for each measuring variable over all the 50 assessing topics.

TABLE III
COMPARISON RESULTS OF DRM WITH ALL DM-BASED METHODS ON ALL ASSESSOR TOPICS

Model	*top-20*	*MAP*	*b/p*	$F_{\beta=1}$
DRM	**0.549**	**0.484**	**0.470**	**0.466**
PTM (IPE) [39]	0.493	0.441	0.429	0.440
PTM (PDS) [34]	0.496	0.444	0.430	0.439
PTM (Closed Seq. ptns) [35]	0.406	0.364	0.353	0.390
%chg	+11.35	+9.75	+9.55	+5.90

TABLE IV
COMPARISON RESULTS OF DRM WITH ALL ML-BASED METHODS ON ALL ASSESSOR TOPICS

Model	*top-20*	*MAP*	*b/p*	$F_{\beta=1}$
DRM	**0.549**	**0.484**	**0.470**	**0.466**
Rocchio [12]	0.474	0.431	0.420	0.430
SVM	0.453	0.408	0.421	0.409
%chg	+15.82%	+12.29%	+11.90%	+8.37%

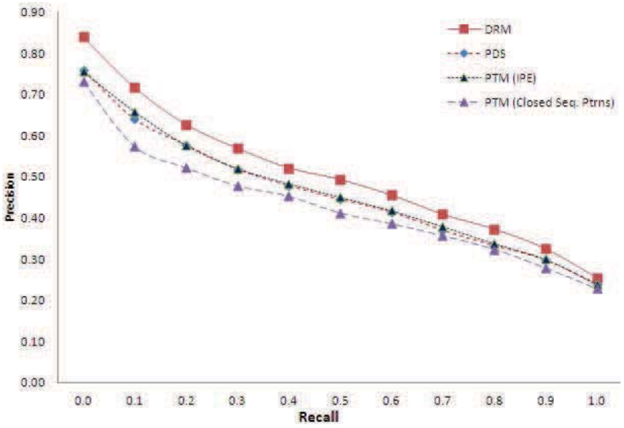

Fig. 2. Comparison results of DRM with all DM-based methods in IAP 11−points

1) DRM vs data mining-based models: The results of overall comparisons between DRM and all DM based models have shown in Table III. The most important findings revealed in this table are that both PDS and IPE models outperforms

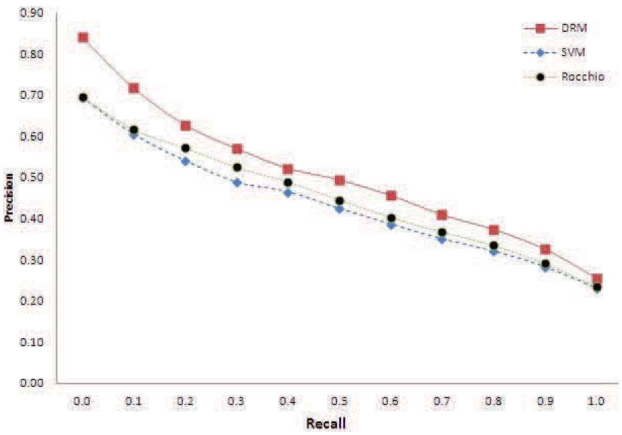

Fig. 3. Comparison results of DRM with all ML-based methods in IAP $11-$points

PTM model over all the standard measures while the slight increase in IPE as compared to PDS. The results support the effective use of patterns in text for user profling.

We also compare DRM with IPE. As seen in Table III, DRM significantly incrases for all the evaluation measures with $+9.14\%$ (max $+11.35\%$ on $top-20$ and min $+5.90\%$ on F_1) in percentage change on average over the standard measures. The encouraging improvments of DRM is also consistent and significant on $11-$points as shown in Figure 1. The results illustrate the highlights of the DS approach to reduce uncertainties involved in estimating term weights.

2) DRM vs machine learning-based models: As shown in Table IV, both Rocchio and SVM models that are based on keyword-based models perform over PTM model, excepting for PDS and IPE. This illustrates keywords remain the very effective concept for text retrieval. However, the results compared between the ML-based models and IPE (also PDS) support that patterns are much effective to select useful terms.

In comparisons with Rocchio and SVM, DRM performs better than Rocchio with $+12.09\%$ increasing in average (max $+15.82\%$ on $top-20$ and min $+8.37\%$ on F_1). The excellent performance of DRM is also obtained as compared to SVM.

VI. Related Work

The frequent pattern-based text classification has been explored by many studies. Earlier approaches are related to associative classification, such as ARC-BC [2], SPAM [11], and HARMONY [31], which mines predictive association rules from a training collection of documents and builds a rule-based text classifier. The results in [2], [38] showed that ARC-BC can perform well on ten most populated Reuters categories as compared to well-known text classifiers, including C4.5, Rocchio, Naive Bayes, excepting for SVMs. In [11], SPAM built by sequential patterns instead of frequent ones showed that it outperformed SVMS in some text collections. HARMOMY [31] focuses on selecting the highest-confidence rules for each training instance to build the text classifier. The objective of our work is different because we are mainly interested in using frequent patterns to build a global model for text classification.

Recently, the focus was more on using frequent patterns to construct new features to improve the quality of text classifier. In [35], a centroid-based text classifier, called PTM, is built by weighted sequential patterns discovered from a relevant text collection. Instead of a full set of features, the closed set is applied to reduce the number of generated patterns. In [19], the authors focused to select top-k discriminative patterns for each training instance from a set of size-1 and size-2 frequent patterns to improve the quality of text classifier. The experimental results showed in [19] highlight the importance of selecting a subset of high-quality patterns.

Nevertheless, the usefulness of frequent patterns is limited by the fact that many mined patterns are never used, especially long patterns. A deploying method for the effective use of patterns in text was proposed in [34], called PDS. It builds a weighted vector of terms from a set of sequential patterns to score new documents corresponding to a relevant category. The result in [16], [34] showed that PDS can largely improve the performance as compared with state-of-the-art text classifiers. Some deploying-based approaches, i.e., IPE [39], focused to improve the classification accuracy by incorporating negative feedback to reduce the effect of noisy terms in relevant documents.

Our work is different from the proposed approaches in the following aspects: (1) we focus on selecting specific patterns from sets of sequential patterns in relevant and non-relevant text collections that describe a target category, where such patterns are investigated by specifying reasoning rules; and (2) we provide a new solution to deal with the set of specific features for text classification. It adopts Dempster-Shafer theory that allows to build the relationship between patterns and terms in estimating weights of terms used in a text classifier.

VII. Conclusions

The paper presents a data reasoning approach for Web user profiling. We have presented a unified model for representing and reasoning about user preference data to construct a correct user profile. We discover patterns from the user data and show how to utilise the patterns for profile construction. We also developed a Dempster-Shafer approach as the means to reduce uncertainties included in text features. Many experiments are conducted on TREC standard text collections and compare the proposed approach with the-state-of-the-art information filtering models. The experiment results illustrate that our proposed approach can improve the system performance.

In the future direction, we are working on reasoning with representations of co-occurrence relations patterns to improve the performance of the data reasoning model.

References

[1] I. Antonellis, C. Bouras, and V. Poulopoulos. Personalized news categorization through scalable text classification. *Frontiers of WWW Research and Development-APWeb 2006*, pages 391–401, 2006.

[2] Maria-Luiza Antonie and Osmar R. Zaïane. Text document categorization by term association. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 19–, Washington, DC, USA, 2002. IEEE Computer Society.

[3] Jing Bai, Jian-Yun Nie, Guihong Cao, and Hugues Bouchard. Using query contexts in information retrieval. In *Proceedings of the 30th international ACM SIGIR Conf.*, pages 15–22, 2007.

[4] A. Baruzzo, A. Dattolo, N. Pudota, and C. Tasso. A general framework for personalized text classification and annotation. *Adaptation and Personalization for Web 2.0*, page 31, 2009.

[5] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *ACM SIGIR 17th International Conf.*, pages 292–300, 1994.

[6] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st international ACM SIGIR Conf.*, pages 243–250. ACM, 2008.

[7] T. Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 737–760, 2008.

[8] Z. Elouedi, K. Mellouli, and P. Smets. Belief decision trees: theoretical foundations. *International Journal of Approximate Reasoning*, 28(2-3):91–124, 2001.

[9] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. *The Adaptive Web*, pages 54–89, 2007.

[10] J.W. Guan and D.A. Bell. Evidence theory and its applications, vol. 2. 1992.

[11] S. Jaillet, A. Laurent, and M. Teisseire. Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3):199–214, 2006.

[12] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the 4th International Conf. on Machine Learning*, ICML '97, pages 143–151, 1997.

[13] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998.

[14] T. Joachims. Transductive inference for text classification using support vector machines. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 200–209. MORGAN KAUFMANN PUBLISHERS, INC., 1999.

[15] Y. Li and N. Zhong. Web mining model and its applications for information gathering. *Knowledge-Based Systems*, 17(5-6):207–217, 2004.

[16] Yuefeng Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *Proceeding of 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 753–762, 2010.

[17] B. Liu, W. Hsu, S. Chen, and Y. Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, pages 47–55, 2000.

[18] Dayou Liu and Yuefeng Li. The interpretation of generalized evidence theory. *Chinese Journal of Computers*, 20(2):158–164, 1997.

[19] H.H. Malik and J.R. Kender. Classifying High-Dimensional Text and Web Data using Very Short Patterns. In *8th IEEE ICDM International Conference on Data Mining*, pages 923–928, 2008.

[20] Y.Q. Miao and M. Kamel. Pairwise optimized rocchio algorithm for text categorization. *Pattern Recognition Letters*, 2010.

[21] Nikolaos Nanas and Manolis Vavalis. A ”bag” or a ”window” of words for information filtering? In *Proceedings of the 5th Hellenic Conf. on Artificial Intelligence*, pages 182–193. Springer-Verlag, 2008.

[22] T.G. Rose, M. Stevenson, and M. Whitehead. The reuters corpus volume 1-from yesterdays news to tomorrows language resources. In *3th International Conf. on Language Resources and Evaluation*, pages 29–31, 2002.

[23] S. Schiaffino and A. Amandi. Intelligent user profiling. In *Artificial intelligence*, pages 193–216, 2009.

[24] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March 2002.

[25] K. Sentz and S. Ferson. Combination of evidence in dempster-shafer theory. Technical report, Citeseer, 2002.

[26] L. Shi, J.Y. Nie, and G. Cao. Relating dependent indexes using dempster-shafer theory. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 429–438. ACM, 2008.

[27] P. Smets. Constructing the pignistic probability function in a context of uncertainty. In *Uncertainty in artificial intelligence*, volume 5, pages 29–39. Elsevier, 1990.

[28] P. Smets. Decision making in a context where uncertainty is represented by belief functions. *Belief functions in business decisions*, 17:61, 2002.

[29] I. Soboroff and S. Robertson. Building a filtering test collection for trec 2002. In *Proceedings of the 26th international ACM SIGIR conference*, page 250. ACM, 2003.

[30] T. Tsikrika and M. Lalmas. Combining evidence for relevance criteria: a framework and experiments in web retrieval. *Advances in Information Retrieval*, pages 481–493, 2007.

[31] J. Wang and G. Karypis. On mining instance-centric classification rules. *IEEE transactions on knowledge and data engineering*, pages 1497–1511, 2006.

[32] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang. Sensor fusion using dempster-shafer theory. In *IEEE Instrumentation and Measurement Technology Conference Proceedings*, volume 1, pages 7–12. Citeseer, 2002.

[33] S. Wu and S. McClean. Performance prediction of data fusion for information retrieval. *Information processing & management*, 42(4):899–915, 2006.

[34] S.T. Wu, Y. Li, and Y. Xu. Deploying approaches for pattern refinement in text mining. In *6th IEEE ICDM International Conf. on Data Mining*, pages 1157–1161, 2006.

[35] S.T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen. Automatic pattern-taxonomy extraction for web mining. In *3th IEEE/WIC/ACM WI International Conf. on Web Intelligence*, pages 242–248, 2004.

[36] Y. Xie and V.V. Phoha. Web user clustering from access log using belief function. In *Proceedings of the 1st international conference on Knowledge capture*, pages 202–208. ACM, 2001.

[37] R.R. Yager and L. Liu. *Classic works on the Dempster-Shafer theory of belief functions*. Springer Verlag, 2008.

[38] Osmar R. Zaïane and Maria-Luiza Antonie. Classifying text documents by associating terms with text categories. In *Proceedings of the 13th Australasian database conference - Volume 5*, ADC '02, pages 215–222, 2002.

[39] N. Zhong, Y. Li, and S.T. Wu. Effective pattern discovery for text mining. *IEEE Transactions on Knowledge and Data Engineering*, DOI: http://doi.ieeecomputersociety.org/10.1109/TKDE.2010.211.

[40] X. Zhou, S.T. Wu, Y. Li andY. Xu, R.Y.K. Lau, and P. Bruza. Utilizing search intent in topic ontology-based user profile for web mining. In *Proceeding of 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 558–561, 2006.