行政院國家科學委員會專題研究計畫　成果報告

運用場景辨識於文件過濾與分類

計畫類別：個別型計畫
計畫編號：NSC93-2213-E-216-010-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：中華大學資訊管理學系

計畫主持人：劉瑞瓏

計畫參與人員：楊怡雰、林文盟、林靜婉

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 94 年 7 月 22 日

行政院國家科學委員會專題研究計畫成果報告

運用場景辨識於文件過濾與分類

Context Recognition for Document Filtering and Classification

計畫編號：NSC 93-2213-E-216-010
執行期限：93 年 8 月 1 日至 94 年 7 月 31 日

主持人：劉瑞瓏　中華大學資訊管理學系

1. 中文摘要

現今大量的文件資訊常是以階層式樹狀架構分門別類，以方便資訊的瀏覽、檢索、訂閱、與分送。在實務上，由於文件內涵包羅萬象，新文件可在任一時間到達，但往往僅有極少部份的文件適合被分類至特定社群(或主題) 之類別樹中。因此，文件之過濾與分類應被整合考量，以便建構一個可以將適當資訊自動分類至適當類別之資訊處理中心，確實讓適當資訊在適當時機為需要的人所用。此整合式文件過濾與分類之主要挑戰是精準地估算文章於各類別的符合度。不精準的符合度估算會將大量的文件分類至不合適的類別，進而錯亂了後續資訊的瀏覽、檢索、訂閱、與分送。為因應此挑戰，本計畫研究開發一個整合式文件過濾與分類之機制 CR4IFC。此機制是藉由辨識文件與類別之討論場景來提昇文件符合度估算之準確度。場景之辨識有兩個最主要的挑戰: 各類別特徵之探勘及場景符合度門檻之釐定。除了突破這些挑戰之外，我們並進行以真實資料為基礎之實驗，以完整驗證 CR4IFC 在不同環境下進行整合式文件過濾與分類之效能與穩定度。此研究除了對資訊檢索相關領域深具意義之外，亦對資訊共享而言具實務價值，可讓大量多變之資訊於社群使用者間更快速精確地暢其流。

關鍵詞：樹狀文件類別架構、文件過濾、文件分類、討論場景之辨識、類別特徵之探勘、場景符合度門檻之釐定

Abstract

Much information has been hierarchically organized to facilitate information browsing, retrieval, and dissemination. In practice, much information may be entered at any time, but only a small subset of the information may be classified into some categories in a hierarchy. Therefore, achieving document filtering (DF) in the course of document classification (DC) is an essential basis to develop an information center, which classifies suitable documents into suitable categories, reducing information overload while facilitating information sharing. In this project, we develop a technique CR4IFC, which conducts DF and DC by recognizing the context of discussion (COD) of each document and category. Experiments on real-world data show that, through COD recognition, the performance of CR4IFC may be significantly better. The results are of both theoretical and practical significance. They may serve as an essential basis to develop an information center for a user community, which organizes and shares a hierarchy of textual information.

2. Introduction

Information is often hierarchically organized as a text hierarchy to facilitate browsing, dissemination, and retrieval (e.g. the information hierarchies of individual users, businesses, libraries, and Internet search engines). In practice, a text hierarchy is often designed for a specific application, and hence lots of documents in the real world may be entered at any time, but only a small subset of them may be classified into some categories in the hierarchy. Therefore, document filtering (DF) and document classification (DC) should be integrated together to classify *suitable documents* into *suitable categories*. It aims to reduce information overload while facilitating information sharing. A document is suitable for a hierarchy if the hierarchy contains at least one category that shares enough semantics with

the document. Only suitable documents are classified into suitable categories. Unsuitable documents should be filtered out of the hierarchy.

For each input document *d* and category *c*, integrated DF and DC consists of three steps: (1) estimating the extent to which *d* shares semantics with *c*, (2) based on the estimation, deciding whether *d* may be classified into *c*, and (3) if the decision is uncertain to a certain extent, interacting with the user to confirm the decision. For the first step, previous studies often estimated the degree of acceptance (DOA, e.g. similarity) of *d* with respect to *c*. For the second step, previous studies often employed a threshold to make a binary decision (i.e. accept or reject) for *c*. If the DOA of *d* with respect to *c* is higher than or equal to the threshold of *c*, *d* is classified into *c*; otherwise it is rejected by *c*. As to the third step, the system needs to produce comprehensible results for the user to confirm. It aims to promote the quality of DF and DC decisions by a limited amount of comprehensible system-user interactions. Unfortunately, previous studies seldom devoted efforts to the step.

In this project, we explore how the recognition of the *context of discussion* (COD) of each document and category may contribute to integrated DF and DC in a text hierarchy. More specially, we explore the way and the impact of introducing COD recognition into the above three steps for integrated DF and DC. The basic rationale is: a document could be classified into a category only if its COD matches the category's COD, which depends on the profiles of the category's *ancestors*. For example, suppose a text hierarchy contains two categories about decision support systems (DSS): (1) "Root → Manufacturing Management → DSS," and (2) "Root → Financial Management → DSS". If a document talks about DSS *and* its COD is about the usage of DSS in manufacturing (finance), it should be classified into the first (second) category; otherwise it should be filtered out, no matter how and to what extent it talks about DSS, manufacturing, and finance, *individually*. This is the main contribution of COD recognition.

COD recognition introduces three challenges: (1) mining category profiles for COD recognition, (2) making proper DF and DC decisions by COD recognition, and (3) producing comprehensible results for user confirmation. The first challenge identifies those features that are both *content-indicative* and *generality-indicative* for individual categories, while the second challenge identifies *how* and *to what extent* a document's COD should be matched with a category's COD before the document may be classified into the category. The third challenge identifies *when* and *how* comprehensible results should be produced for confirmation. The three challenges were not simultaneously tackled by previous studies.

We thus develop a text mining technique CR4IFC, which performs integrated DF and DC by COD recognition. Empirical results show that COD recognition may successfully help CR4IFC to make both more accurate and comprehensible decisions, which are essential in supporting high-quality information browsing, retrieval, management, and dissemination.

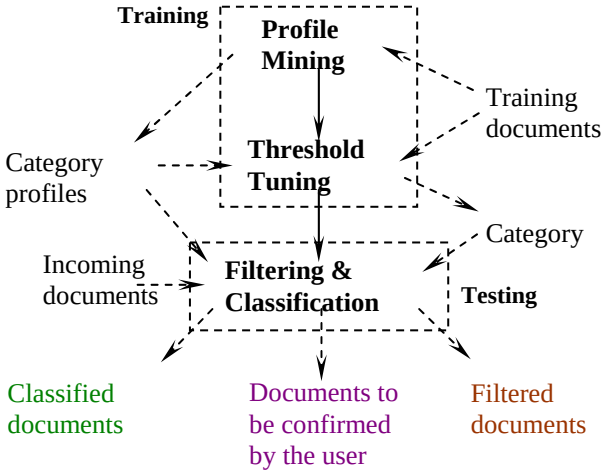

Figure 1. Process flow of CR4IFC

3. Result and Discussion

Figure 1 outlines the process flow of CR4IFC, which consists of three components: a profile miner, a COD threshold tuner, and a filtering classifier. The former two components are triggered in the training phase, while the third component is triggered once a document is entered. The profile miner identifies content-indicative and generality-indicative

features for each category under the root of the hierarchy. Based on the profiles mined, the COD threshold tuner estimates the DOA values of all training documents, and accordingly tunes a threshold for each category under the root. Once a document is entered to the system, the filtering classifier consults the profiles and the thresholds to make DF and DC decisions. For those decisions with low confidence, the filtering classifier produces comprehensible results for the user to confirm.

3.1 The profile miner

To build the profiles, CR4IFC invokes *ProfileMining*(*root*), which builds the profiles in a top-down manner. The profile of a category x is a set of 3-tuples $<w, s_{w,x}, g_{w,x}>$, where w is a term serving as a profile term (a feature) for x, $s_{w,x}$ is the support of w under x (i.e. including x and its descendents), and $g_{w,x}$ is the strength of w in distinguishing x from *siblings* of x. That is, $s_{w,x}$ and $g_{w,x}$ consider the distributions of w under x and siblings of x, respectively. A term w is likely to be a good profile term for a category x if (1) it occurs frequently under x (i.e. having a higher $s_{w,x}$), *and* (2) it occurs relatively infrequently under siblings of x (i.e. having a higher $g_{w,x}$).

More specially, $s_{w,x}$ is estimated by $P(w|x)$, which is equal to $TF(w,x)$ / $Size(x)$, where $TF(w,x)$ is the times of occurrences of w in documents under x (i.e. the documents in x and all descendants of x), and $Size(x)$ is the total number of terms occurring in the documents under x. On the other hand, $g_{w,x}$ is estimated by $P(w|x) \times (B_x/\Sigma_i P(w|x_i))$, where B_x is one plus the number of siblings of x (i.e. the summation of $P(w|x_i)$ is conducted over x and its siblings). Thus, for example, w may get a higher $g_{w,x}$ if its support under x (i.e. $P(w|x)$) is higher than its average support under x and siblings of x (i.e. $\Sigma_i P(w|x_i)/B_x$). In that case, w may be good in distinguishing x from siblings of x. Obviously, $0 \le g_{w,x} \le B_x$, and when w only occurs in x, $g_{w,x} = B_x$ and $g_{w,b} = 0$ for each sibling b of x. If $P(w|x)$ is higher than $\Sigma_i P(w|x_i)/B_x$ (i.e. the average $P(w|x_i)$), $g_{w,x} > 1$; otherwise $g_{w,x} \le 1$.

Therefore, the profile miner actually estimates the strengths of profile terms. All non-stop words occurring under a category x may serve as profile terms for x. The point here is that, a profile term w may get a stronger strength in x only if it is both content-indicative (i.e. $s_{w,x}$ is high) and generality-indicative (i.e. $g_{w,x}$ is high). Those terms with higher strengths may be good profile terms for x, which may be good COD indicators for the descendents of x.

3.2 The COD threshold tuner

For each leaf category x, CR4IFC invokes *CODThresholdTuning*(x) to derive a set of thresholds, which includes a threshold for x and a threshold for each ancestor a of x (for governing the COD of x). The basic idea is that the thresholds for the ancestors should reflect the *minimum* DOA values of those documents that may be classified into x. After setting such thresholds, ancestors of x may work together to check the COD of input documents, making DF and DC decisions on x more proper.

More specially, for an ancestor a of a leaf category x, its threshold is set to make *all* training documents belonging to x able to pass the test of a. Obviously, there might still be documents not belonging to x but able to pass all the tests of the ancestors of x. Let Q be the set of these documents. To tune a threshold for x per se, the set Q and the set of those documents really belonging to x (i.e. P) are used. The threshold is simply the DOA value of some document in P that may maximize the system's performance on P and Q with respect to a given criterion (e.g. the F-measure).

The contribution of basing thresholding on the set Q deserves discussion. Each leaf category x actually relies on its ancestors to identify Q, which consists of those documents that really deserve consideration in the thresholding process for x. Those documents that cannot pass the tests of the ancestors should be noises in thresholding. This is because the documents cannot pass the COD tests, making their DOA values with respect to x no longer meaningful, even though the DOA values are high. Noise reduction may make thresholding more reliable.

3.3 The filtering classifier

Given a document d, the filtering classifier returns a set of categories to which d may be classified (S_1) and a set of potential categories for the user to confirm (S_2). If both

sets are empty, *d* is actually filtered out of the hierarchy. The basic idea is: the system may confidently classify *d* into a category *c* only if it may pass all the tests of *c* and *c*'s ancestors under the root. The test of *c* is for matching the contents of *c* and *d*, while the tests of the ancestors are for matching the COD of *c* and *d*. If *d* can pass all but one of the tests, *c* could only be a potential category for the user to confirm (and hence would be put into S_2).

It is interesting to note that, through profile mining and COD recognition, CR4IFC may make the system-user interaction more comprehensible, which is an effective way to promote the quality of DF and DC. The interaction is conducted only there are potential categories (i.e. the set S_2) for a document. By checking the ancestors of the categories, the user may check how the document's COD matches the category's COD, making it easier for the user to make a decision (i.e. 'Accept' or 'Reject').

3.4 Experimental evaluation

Experiments on a real-world document database were conducted to evaluate CR4IFC. We aimed to empirically investigate the contributions of COD recognition to integrated DF and DC. The results showed that, through COD recognition, the performances of CR4IFC were both significantly better and more stable.

To facilitate objective evaluation and cross-validation, experimental data was extracted from a public database of Yahoo (http://www.yahoo.com). We extracted categories under 5 first-level categories: "science," "computers and Internet," "society and culture," "business and economy," and "Government". The text hierarchy contained 507 categories among which there were 211 leaf categories, which totally contained 3612 documents. Its height was 8.

The amount and distribution of the documents deserve discussion as well. In the hierarchy, the largest (smallest) leaf categories contained 150 (3) documents. We believed that such an environment was common for many applications in which users could not provide much data, and some of the categories (folders) contain very few documents. Actually the problem of sparse and skewed data was often identified as a practical problem to which considerable effort was devoted. The text hierarchy may thus facilitate the measurement of the contributions of COD recognition under such a common environment.

There should be two types of experimental data: *in-space data* and *out-space data*. The former was for training the systems and testing DC performances, while the latter was for testing DF performances (since it should be filtered out). Therefore, we randomly and comprehensively removed 20 leaf categories from the text hierarchy (i.e. about 10% of the leaf categories). That is, the documents in these 20 categories served as the out-space data, while the final text hierarchy contained 191 leaf categories (211-20), which served as the in-space data.

DC and DF require different evaluation criteria. For DC, we employed precision (P) and recall (R), which were common evaluation criteria in previous studies. To integrate P and R into a single measure, the well-known F-measure was employed as well: $F_\beta = [(\beta^2+1)PR] / [\beta^2P+R]$, where β is a parameter governing the relative importance of P and R. As in many studies, we set β to 1 (i.e. the F_1 measure), placing the same emphasis on P and R.

On the other hand, to evaluate DF, we employed two criteria: filtering ratio (FR) and average number of misclassifications for misclassified out-space documents (AM). FR was estimated by [number of out-space documents filtered out / number of out-space documents], while AM was estimated by [total number of misclassifications / number of out-space documents misclassified into the text hierarchy]. A better system should reject more out-space documents (i.e. higher FR) and avoid misclassifying out-space documents into many categories (i.e. lower AM). The in-space criteria (i.e. P, R, and F_1) and the out-space criteria (i.e. FR and AM) together could help to precisely identify those systems that are really better in real-world environments in which both in-space and out-space documents could be entered at any time

We also implemented four baselines for performance comparison with respect to CR4IFC. For objective comparison, no user

was allowed to interact with CR4IFC (i.e. documents were automatically rejected by potential categories, ref. the set S_2). The baselines were employed to represent those filtering classifiers that do not conduct COD recognition, since previous hierarchical classifiers required extensive revisions in order to perform integrated DF and DC by COD recognition. The baselines may thus help to justify the contributions of COD recognition to DF and DC.

The baselines were NB+FixedT, NB+T, RO+T, and HS+T, which could comprehensively represent various related methodologies. From the viewpoint of classification methodology, NB+FixedT and NB+T employed the probabilistic Naive Bayes method (NB), RO+T employed the vector-based Rocchio method (RO), while HS+T employed the hierarchical shrinkage method (HS). On the other hand, from the viewpoint of thresholding, NB+FixedT set a fixed threshold of 0.5 for each category, while the other three baselines employed thresholding to set a relative threshold for each category by analyzing DOA scores of documents, which were estimated by the classifiers. As in many studies, all training documents were used to tune the thresholds. The thresholds were tuned in the hope to optimize the system's performance with respect to F_1, which was commonly employed in many previous studies as well.

Moreover, all the baselines required a fixed (predefined) feature set, which was built using the training documents. The features were selected according to their weights, which were estimated by the χ^2 (chi-square) weighting technique. The technique was shown to be more promising than others. As noted above, there is no perfect way to determine the size of the feature set. Therefore, for each baseline, we explore the possible range of feature set sizes: 5000, 20000, 40000, and 80000 (almost equal to the total number of different terms in the in-space data).

The results showed that by employing COD recognition, CR4IFC achieved both significantly better and more stable performances in both DF and DC.

4. Evaluation

Preliminary results of the project have been published in [1]. The contributions of the research project are of theoretical significance, since this is the first attempt to introduce COD recognition to interactive DF and DC. The contributions are of practical significance as well, since much information has been organized into text hierarchies and may be entered at any time. The results may be applied to various applications in which information and knowledge are processed, managed, and shared among a community of users.

5. Reference

[1] Rey-Long Liu, 2005, "Mining for Context Recognition in Document Filtering and Classification," *Proc. of the Fourth Annual ACIS International Conference on Computer and Information Science*, Jeju Island, South Korea (14-16 July 2005).