Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2007 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2007

Preserving User Preferences in Document-Category Management: An Ontology-based Evolution Approach

Yen-Hsien Lee
National Chiayi Univ. Chiayi, Taiwan, yhlee@mail.ncyu.edu.tw

Chih-Ping Wei
National Tsing Hua Univ. Hsinchu, Taiwan, cpwei@mx.nthu.edu.tw

Paul Jen-Hwa Hu
University of Utah, USA, actph@business.utah.edu

96. Preserving User Preferences in Document-Category Management: An Ontology-based Evolution Approach

Yen-Hsien Lee
Department of MIS
National Chiayi Univ.
Chiayi, Taiwan, ROC
yhlee@mail.ncyu.edu.tw

Chih-Ping Wei
Inst. of Tech. Management
National Tsing Hua Univ.
Hsinchu, Taiwan, R.O.C.
cpwei@mx.nthu.edu.tw

Paul Jen-Hwa Hu
Acct. and Info. Systems
University of Utah, USA
actph@business.utah.edu

Abstract

Preserving the user's preference in document-category management is essential because it affects his/her search efficiency, cognitive processing load, and satisfaction. Prior research has investigated automated document category evolution by using lexicon-based document-category evolution techniques which take into account the document categories previously created by the user. However, comparing documents at the lexical level cannot solve word mismatch or ambiguity problems effectively. To address such problems inherent to the lexicon-based approach, we propose an ONtology-based Category Evolution (ONCE) technique, which uses an appropriate ontology to support document-category evolution at the conceptual level rather than at the lexical level. Specifically, we develop an Ontology Enrichment (OE) technique for automatic leaning of concept descriptors in the adopted ontology. We empirically evaluate the effectiveness of the proposed ONCE technique, using a lexicon-based document-category evolution technique (i.e., CE2) and the hierarchical agglomerative clustering (HAC) technique for benchmark purposes. According to our empirical results, ONCE appears more effective than CE2 and HAC, and achieves higher clustering recall and precision.

Keywords: Document-category management, Ontology-based category evolution, Category evolution, Concept descriptor learning, Ontology enrichment

Introduction

The advances and proliferation of information technology have fostered rapid creation and dissemination of information, typically in the form of textual documents, on a massive scale. Analysis of the current practices suggests the common use of document category by individuals and organizations to support users' information search in the ever-increasing corpora. As new documents arrive over time and are assigned to the previously created categories, the appropriateness or cohesiveness of the existing categories may deteriorate because the new documents bring about significant changes in the category contents and therefore adversely affect category coherence and distinction. Understandably, this necessitates category re-organization and may require new category creation. New documents often arrive in great frequency and enormous quantity and therefore make document category management increasingly challenging. When not properly managed, document categories will evolve in an ad hoc manner, and document assignments can become inconsistent.

The sheer volume of new documents and the likelihood of their assignments to inappropriate categories make the manual document-category management approach prohibitively tedious and ineffective. Hence, automated document-category management represents an appealing alternative and can be greatly supported by appropriate text mining techniques. Of particular importance is document clustering, which partitions a collection of documents into distinct groups in which the documents in each group share substantial similarity and collectively reveal a theme concealed in the underling document corpus (Boley et al. 1999; El-Hamdouchi and Willett 1986; Larsen and Aone 1999; Pantel and Lin 2002; Wei et al. 2006). Previous research has examined the use of document clustering for automated document-category management, with a predominant focus on a discovery or total re-discovery approach which aggregates all documents, both existing and new, in the analysis to create a new set of document categories.

However, this discovery-based approach may not be effective because it does not preserve the user's perspective or preference manifested in the document categories he or she created previously. According to the context theory of classification (Barreau 1991; Case 1991; Kwasnik 1991; Lakoff 1987; Quiroga 2004), an individual's document grouping behavior not only requires document content analysis but also involves the context (e.g., user role, task) that prompts his or her use of a particular perspective in document grouping. Document grouping behavior is an intentional act that reflects an individual's perspective or preference with respect to semantic coherency or document categorization (Rucker and Polanco 1997). In situations where the adequacy of existing document categories (previously created by a user) has deteriorated as they include influxes of new documents over time, the categories, to some extent, still reflect the user's document grouping preference. The discovery-based approach, supported by traditional document clustering techniques, may be adequate for document grouping from a pure content analysis perspective but does not preserve the user's document grouping preferences.

Preserving the user's preference in document-category management is essential because it affects the user's search efficiency, cognitive processing load, and satisfaction. People are habitual and can obtain considerable efficiency gain in their document searches through repetitions; e.g., the power law of practice (Johnson et al. 2003). Such habitual behaviors demand desirable continuity in managing the evolution of document categories, the absence of which can greatly hinder the user's search effectiveness and efficiency and lead to frustration. All else being equal, a user can locate relevant or target documents faster when searching from a familiar set of document categories than from categories completely new to him or her. The benefits of preserving the user's preference in document grouping also can be explained by the expectation-disconfirmation theory (Oliver 1908). According to this theory, individuals have some expectations about their use of current document categories formed on the basis of their experiences with the previous created categories; they can become dissatisfied or even frustrated if such expectations are disconfirmed by their use of the current document categories.

To preserve a user's document grouping preference, Wei et al. (2002; 2005) investigate document category evolution and propose the CE and CE2 techniques for re-organizing document categories while taking into account the document categories previously created by the user through the processes of category decomposition and category amalgamation. Both

CE and CE2 adopt the lexicon-based approach that measures document similarity on the basis of the overlap between or among the feature vectors representing individual documents. While preliminary evaluations of CE and CE2 are encouraging, comparing documents at the lexical level cannot solve word mismatch or ambiguity problems effectively.[9] In response, we propose an ONtology-based Category Evolution (ONCE) technique, which uses an ontology to support document-category evolution at the conceptual rather than lexical level. In general, an ontology offers a shared, common understanding of a domain that can be easily communicated between or among humans as well as application systems (Fensel 2000). Specifically, on the basis of concepts defined in the ontology, ONCE transforms each document originally described at the lexical level into the concept-based representation and supports automated document-category evolution in a more appropriate manner.

Although many professional associations have created their domain ontologies (or concept hierarchies, to be more specific), but few of them have concept descriptors readily available. In this study, we assume that a document corpus is associated to an ontology (i.e., each concept in the ontology is associated with a subset of documents), but concept descriptors are not available for an ontology. In response, we also propose an Ontology Enrichment (OE) technique for automated extracting from the precategorized documents a representative set of concept descriptors for each concept in an ontology.

The remainder of this paper is organized as follows: In Section 2, we discuss existing document-category evolution techniques and provide an overview of ontology. In Section 3, we detail our proposed ONCE technique for document-category evolution. Section 4 describes our evaluation design and highlight key comparative analysis results. This paper is concluded in Section 5 with a summary and some future research directions.

Literature Review

Overview of Lexicon-based Category Evolution Techniques

To address the evolving nature of document categories, Wei et al. (2002) first propose the CE technique that takes into consideration the user's document grouping preference. In essence, CE takes as inputs all the documents and existing document categories previously created by the user to generate new document categories, each of which contains documents of increasing similarity or coherence.

Broadly, CE consists of category decomposition and category amalgamation stages. In category decomposition, CE splits an existing document category into multiple new subcategories, each of which contains documents that are increasingly cohesive or germane to a fine-grained topic. For each existing category, CE first selects a set of representative features for each category and uses them to represent all documents in the category. CE then tentatively splits an existing category into two subsets, each containing documents that exhibit maximal similarity to those in the same subset and share minimal similarity with documents in the other subset. For each existing category, CE evaluates the disjointedness between the resultant subsets, and an existing category gets decomposed when its intracategory disjointedness exceeds a prespecified threshold. The CE technique adopts the PAM algorithm (Kaufman and Rousseeuw 1990) to decompose an existing category into multiple subcategories and uses the silhouette coefficient measure to determine the optimal number of subcategories to be created. Subsequently, all the documents in the original

[9] Word mismatch refers to the phenomenon where different words are used to describe the same concept or object, whereas word ambiguity refers to the phenomenon where a word is used to describe different concepts or objects.

category are assigned to appropriate subcategories generated by this decomposition process.

In the category amalgamation stage, similar document categories or subcategories that result from the decomposition stage are merged to form more general categories, each of which contains documents that pertain to a topic of broader scope. Specifically, CE reselects features for each category or subcategory, and then uses the resulting features to represent individual documents in the respective categories or subcategories. When completing feature reselection and document rerepresentation, CE examines the similarity of the categories (or subcategories), and performs category coalescence. It starts with as many clusters as there are categories or subcategories that result from the category decomposition stage; that is, each category or subcategory forms a cluster initially. To estimate the similarity between clusters, CE uses the complete link method, which measures intercluster similarity on the basis of the minimum similarity (i.e., intercategory overlapping) between all intercluster pairs of categories or subcategories. Two most similar clusters are then merged to form a new cluster. This merging process continues until no intercluster similarity is greater than the prespecified merging threshold. At the completion of the category amalgamation stage, CE generates a set of new categories, which in effect have evolved from those previously created by the user, and reassigns the documents to appropriate resulting categories accordingly.

Wei et al. (2005) then propose CE2 to address the inherent limitations of CE. Specifically, CE2 replaces the intra-category disjointedness measure based on the collective feature set with document-based category cohesion that measures the average similarity of all the document pairs in a category. This similarity measure is more effective for assessing the adequacy of a document category. In addition, CE2 mitigates CE's limitation in category decomposition by distinguishing the most dissimilar documents from those in a document category and then decomposing that category accordingly. CE2 applies this process to all document categories and sub-categories until the coherence of each category exceeds the specified threshold. CE2 uses document-based inter-category similarity to address the ineffective inter-category overlap measure, particularly in situations where the distribution of documents in different categories or sub-categories is asymmetric.

As with CE, CE2 comprises category decomposition and category amalgamation. In the category decomposition stage, CE2 splits an existing category, when appropriate, into multiple sub-categories; each resulting subcategory contains similar documents pertinent to a fine-grained topic. Specifically, CE2 employs the hierarchical divisive clustering (HDC) algorithm (Kaufman and Rousseeuw 1990) to decompose an existing category into a set of subcategories. For each existing document category, the HDC algorithm starts by placing all documents in one cluster and then subdivides the category into two smaller clusters until the average similarity of all document pairs (i.e., category cohesion) in each cluster exceeds a predefined similarity threshold (α_s).

In the category amalgamation stage, CE2 merges multiple categories or sub-categories generated from the category decomposition stage to create a more general document category, which contains related documents pertinent to a topic of broader scope. CE2 re-performs feature selection across all the categories or subcategories to create a global dictionary that comprises a universal feature set for all document categories or subcategories. For document re-representation, CE2 adopts a binary scheme, identical to the document representation method used in category decomposition. CE2 then performs category coalescence to merge similar categories or subcategories on the basis of their intercategory similarity, thus creating more general categories. To avoid inconsistent processing between category decomposition and amalgamation, CE2 prohibits a direct merge of two subcategories that have been created from the same category during the category decomposition stage. Subsequently, CE2 extends the HAC algorithm (Voorhees 1986) for merging similar categories or sub-categories until the similarity of the permissible merge under examination is lower than the prespecified similarity threshold α_m. After completing the category coalescence, CE2 generates a set of categories that, in effect, have evolved from those previously established by the user.

Overview of Ontology

Ontology refers to a systematic account of existence. Philosophically, ontology entails explicit, formal specifications of how to represent objects, concepts, and other entities (including the relationships among them) commonly assumed to exist in a domain of interest. Computationally, ontology can be used to define a common vocabulary that formally represents knowledge and facilitates its sharing and reuse. In this connection, ontology describes the specification of a representational vocabulary for a shared domain of discourse, such as definitions of class, relations, functions, and other objects (Gruber 1993).

Typically, an ontology consists of a set of related concepts, relations, instances and axioms (e.g., constraints) (Keet 2004). A concept represents an abstract or generic idea derived or inferred from particular instances. A relation describes a relationship (e.g., taxonomic or associative) between or among concepts, properties of concept, or functions that relate a concept to a set of terms or descriptors (hereafter referred to as concept descriptors). Taxonomic (i.e., is-a) relationships are the most eminent relations that organize multiple concepts into a concept taxonomy or hierarchy, whereas associative relations relate concepts across a concept hierarchy. In addition, axioms (e.g., constraints) describe the boundary of or restrictions on the value of an instance of a particular concept. Axioms can represent knowledge effectively and support its inferencing.

Many professional associations have created their domain ontologies (or concept hierarchies, to be more specific), but few of them have concept descriptors readily available. For example, the Association for Computing Machinery (ACM) develops the computing classification system (CCS)[10] that provides a general structure for computing. The ACM CCS hierarchy is primarily used as an indexing scheme for organizing articles published in various ACM periodicals and therefore dos not define concept descriptors within the hierarchy. The lack of such concept descriptors has greatly restrained the potential applications of an established concept hierarchy. Hence, we consider in this research a populated concept hierarchy as an ontology. Given a concept hierarchy and the relative, well-classified documents of the respective concepts, our proposing OE technique can be applied to discover important concept descriptors, and thus deriving a set of representative features for all of the targeted concepts. The design of OE technique will be detailed in the following section.

Design of OE and ONCE

In this section, we first discuss the design of the proposed Ontology Enrichment (OE) technique, which is used in our research to generate a set of representative descriptors for each concept in the ontology. Subsequently, we detail our proposed Ontology-based Category Evolution (ONCE) technique for document-category evolution.

Ontology Enrichment (OE) Technique

[10] http://www.acm.org/class/1998/.

As shown in Figure 1, the overall process of the OE technique consists of three phases. In the feature extraction phase, all nouns and noun phrases are extracted as features from each document. We measure the features within each concept their relative importance to the sibling concepts, and select the k_{cd} most important features as the set of concept descriptor of the specific concept. To maintain the representativeness and distinctiveness of a concept, the features that appear frequently in multiple concepts are excluded from the list of concept descriptors as in the phase of concept refinement. The detailed design of each phase of the proposed OE technique is as follows:

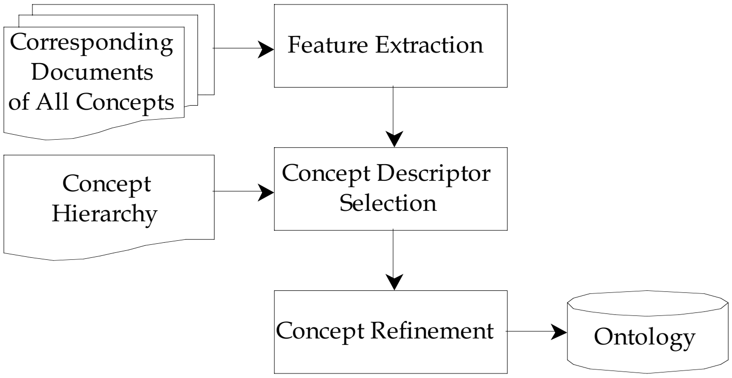

Corresponding Documents of All Concepts
Feature Extraction
Concept Hierarchy
Concept Descriptor Selection
Concept Refinement
Ontology

Figure 1: Overall Process of Learning of Concept Descriptors

Feature Extraction: The nouns and noun phrases, which are considered as the features of the respective document, are extracted from the documents. We first used the rule-based part of speech tagger proposed by Brill (1992, 1994) to tag each word in each of the documents. We then implemented a noun phrase parser suggested by Voutilainen (1993) to extract the nouns and noun phrases from each document.

Concept Descriptor Selection: We select a set of weighted features (i.e., concept descriptors), which collectively describes a concept in the concept hierarchy. For example, assuming a concept hierarchy (as illustrated in Figure 2), the concept descriptor selection is applied to select the important descriptors of each concept; i.e., A, B, and C (the children of the ROOT node). The importance of a feature in a concept (e.g., concept A) is then evaluated in relation to its sibling concepts (e.g., concepts B and C), regardless of other concepts at any other levels or subtrees (e.g., concept A.1, B.2). Similarly, when measuring the weight of a feature in relation to the concept A.1, we only consider its relative importance to the siblings of concept A.1; e.g., concept A.2 and A.3.

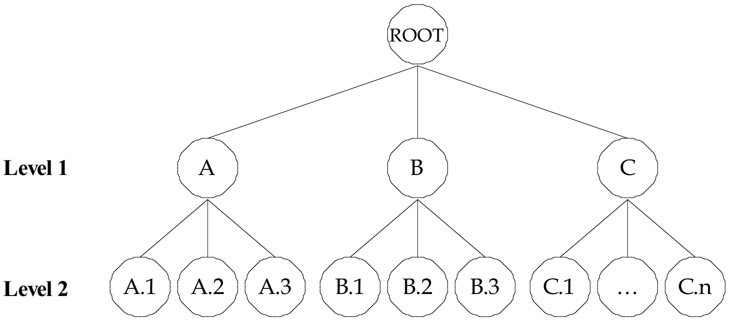


Figure 2: An Example of A Concept Hierarchy

In addition to be able to distinguish from its siblings, the descriptors of a concept should also appear frequently and evenly in the documents pertaining to that concept. Specifically, we determine the weight of a feature f_i in a concept o_j by considering the term frequency of f_i appearing in o_j, the percentage of documents pertaining to o_j that contain f_i, and the specificity of f_i among o_j and its siblings. The weighting function of f_i in o_j is defined as:

$$w_c(f_i, o_j) = TF(f_i, o_j) \times pd_{ij} \times \left[\log_2 s - \left(- \sum_{h=1}^{s} \left(\frac{pd_{ih}}{\sum_{r=1}^{s} pd_{ir}} \times \log_2 \frac{pd_{ih}}{\sum_{r=1}^{s} pd_{ir}} \right) \right) \right],$$

where $TF(f_i, o_j)$ denotes the term frequency of f_i in o_j, pd_{ij} is the number of documents that contain f_i in o_j over the total number of documents in o_j, and s is the number of siblings of o_j plus 1 (i.e., including o_j itself).

We measure the specificity of the feature f_i by the entropy function, based on the respective percentage of documents in each investigated concept (i.e., concept o_j and its siblings) that f_i appears in. Specifically, we measure the specificity of f_i by taking the difference between the derived entropy value and its theoretical maximum. Besides, a specific concept (i.e., those at lower levels of the hierarchy) requires a large number of descriptors to depict its diverse specializations; as the level descends along the hierarchy, the number of descriptors increases. Accordingly, we select k_{cd} features as the descriptors of a target concept at level one and select $(k_{cd} + (n-1)\times\delta_{cd})$ descriptors for a concept at the level n.

Concept Refinement: Our concept descriptor weighting function only takes into consideration the importance of a feature in a concept comparative to its sibling concepts. The selected descriptors should be representative of the target concept and, at the same time, discriminative to its siblings. This function does not consider all the concepts in the hierarchy; therefore, some of the selected descriptors of a concept may not effectively discriminate the concepts other than its siblings. This reduces their representativeness for the target concept. In the concept refinement phase, a feature which is commonly selected as the descriptor of other concepts will be removed. We use a pre-determined commonality threshold α_p to remove a descriptor if it appears in more than α_p percent of the concepts in the hierarchy. Upon completing concept refinement, each concept in the hierarchy is represented by a feature vector in which the weight of a feature is estimated by the concept descriptor weighting metric. The resulting feature vectors, together with the concept hierarchy, represent the ontology which, in turn, serve as the input to the proposed ONCE technique.

Design of Ontology-based Category Evolution (ONCE)

We propose the ONtology-based Category Evolution (ONCE) technique which employs a domain ontology to address the limitations inherent to the lexicon-based category evolution techniques, such as CE and CE2. With conventional category-evolution techniques, a categorized document is represented as a set of features. Given a domain ontology (consisting of a concept hierarchy and the corresponding concept descriptors), ONCE transforms each categorized document into a set of weighted concepts. Subsequently, ONCE evolves document categories by performing category decomposition and category amalgamation using these categorized concept-based documents. As shown in Figure 3, the process of ONCE consists of *document transformation*, *category decomposition*, and *category amalgamation*, each of which is detailed as follows.

Document Transformation: In this phase, important features are extracted and concepts are mapped to the categorized documents. We use the rule-based part of speech tagger proposed by Brill (1992, 1994) to tag each word in a document, followed by implementing a noun phrase parser proposed by Voutilainen (1993) for extracting nouns and noun phrases from each tagged document. In the subsequent concept mapping, we measure the degree of relevance between a categorized document and each concept in the hierarchy and then transform the document from a set of features into a set of weighted concepts. The weighting function of relevance degree between a document d_i and a concept o_j is defined as $w_m(d_i, o_j) = \left(\sum_{f_k \in o_j} \left(w_c(f_k, o_j) \times TF(f_k, d_i) \right) \right) \times pf_{ij}$, where f_k is one of the descriptors of the concept o_j, $w_c(f_k, o_j)$ is the weight of the descriptor f_k in the concept o_j, $TF(f_k, d_i)$ is the within-document term frequency of the descriptor f_k in the document d_i, and pf_{ij} is the percentage of the number of descriptors in the concept o_j that appears in the document d_i. According to our proposed weighting function, we sum the product of the weight of each descriptor in the concept o_j and its respective term frequency in the document d_i. The more descriptors of a concept appear in a document, the greater the confidence that the document embraces that concept. The relevance degree is then adjusted by multiplying the percentage of the number of descriptors in the concept o_j that appears in the document d_i. After measuring the degree of relevance between each document and each concept in the hierarchy, each document is represented as a set of weighted concepts.

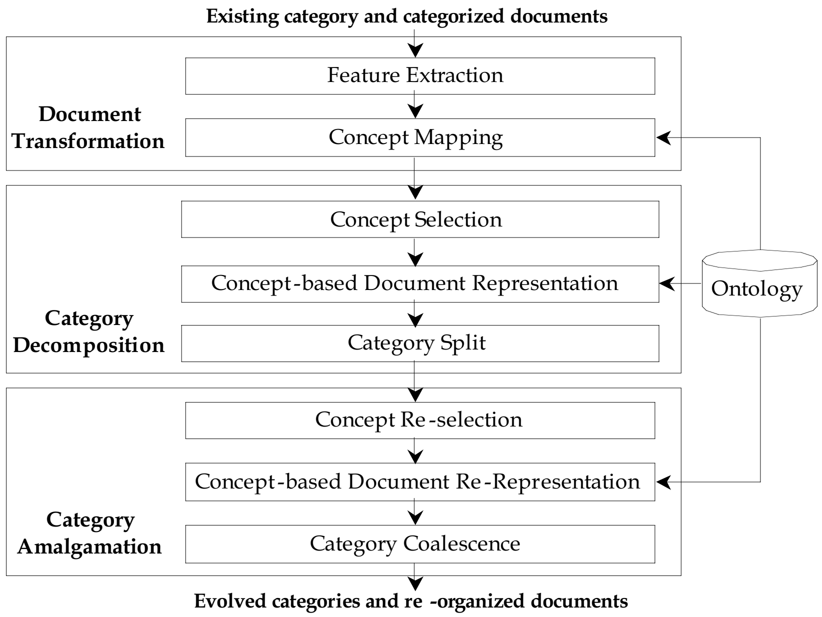


Figure 3: Overall Process of ONtology-based Category Evolution (ONCE) Technique

Category Decomposition: In this phase, ONCE assesses the cohesiveness of the documents in each category and splits the category until the cohesiveness of each resulting (decomposed) category exceeds the pre-specified threshold. For category decomposition, ONCE performs three tasks: *concept selection*, *concept-based document representation*, and *category split*. Upon transforming each document into a set of weighted concepts, concept selection proceeds, thereby selecting the concepts most representative of the documents in each (existing) category. To measure the importance of a concept in relation to a specific category (e.g., category C_k), we propose a revised TF×IDF measure by replacing the TF value of a concept o_j with the summation of the degree of relevance between the concept and all the documents in C_k. In addition, we incorporate a small fraction to IDF (i.e., 0.01). The revised TF×IDF weighting function for the concept o_j in the category C_k is thus defined as

$w_d(C_k, o_j) = \left(\sum_{d_i \in C_k} w_m(d_i, o_j)\right) \times (\log_2 \frac{n_k}{n_{kj}} + 0.01)$, where d_i is a document in C_k, $w_m(d_i, o_j)$ is the relevance degree of d_i and the concept o_j, n_k is the number of documents in C_k, and n_{kj} is the number of documents in C_k that contain o_j. Finally, the k_s concepts with the highest TF×IDF scores are then selected as the local dictionary for C_k and used to represent each document of the category.

After selecting the concepts of a category, ONCE represents each document using a concept vector; i.e., the concept-based document representation. We adopt a weighting scheme to represent each concept in a document by assigning a particular weight to each concept on the basis of its importance in the document. When determining the weight of a concept, we consider not only the relevance degree between a document and the concept, but also that between the document and other relevant concepts. Two concepts may be relevant when they locate closely in the hierarchy. In this study, we measure the similarity of two distinct concepts by examining their distance in the hierarchy. Given a concept hierarchy of l-level hierarchical structure, we define $(1/2)^{l-(k-1)}$ as the concept similarity at the level k in relation to its parent concept, and $(1/2)^{l-(k-1)} \times (1/2)^{l-((k-1)-1)}$ as the similarity in relation to its grandparent node. The similarity of two concepts can then be calculated using the product of the similarity of the respective concepts in relation to their closest common ancestor. In situations where the closest common ancestor of two concepts is the root node of the hierarchy, the similarity of these concepts is set to 0. In addition, we define the similarity between a concept and itself to be 1. Let O be the set of concepts selected for an existing category C_k. We use the concept similarity measure defined above to estimate the weight of a concept o_j in a document d_i, assuming d_i belonging to C_k, the maximal product of the degree of relevance of o_h in d_i, and the similarity between o_h and o_j for every $o_h \in O$. Using the matrix representation, we formally define the weight of each concept in a document in the existing category C_k as $P_{|C_k|\times|O|} \otimes Q_{|O|\times|O|} = R_{|C_k|\times|O|}$, where $P_{m\times n}$ is the document-concept matrix in which each element $p_{ij} = w_m(d_i, o_j)$ is the degree of relevance of d_i and o_j, $Q_{n\times n}$ is the similarity matrix of concepts in which each element q_{ij} is the similarity between the concepts o_i and o_j, $R_{m\times n}$ is the document-concept matrix in which each element r_{ij} is the weight of the concept o_j in d_i and defined as $r_{ij} = \mathop{\mathrm{Max}}_{k=1}^{|O|} (p_{ik} \times q_{kj})$, $|C_k|$ is the number of documents in C_k, and $|O|$ is the number of concepts selected for C_k.

To perform category split, ONCE uses the hierarchical divisive clustering (HDC) algorithm to decompose a category into multiple sub-categories as necessary. Specifically, ONCE uses the cosine similarity to measure the similarity between two concept vectors and adopts a pre-defined similarity threshold (α_s) as the termination condition of the hierarchical divisive clustering. Upon the completion of category split, each existing document category is decomposed into one or more sub-categories. The decomposed sub-categories resulting from all the existing document categories then serve as the inputs to category amalgamation.

Category Amalgamation: It involves three tasks: *concept re-selection*, *concept-based document re-representation*, and *category coalescence*. Concept re-selection selects (or re-selects) concepts representative of the entire sub-categories that result from category decomposition. The selected concepts are then used to represent each document as a concept vector (i.e., concept-based document re-representation). Finally, similar sub-categories are merged to form a category of a broader scope in category coalescence. Design details of ONCE for concept re-selection, concept-based document re-representation, and category coalescence are as follows.

ONCE performs concept re-selection across all categories to generate a global dictionary for the entire sub-categories. This establishes an equal basis for comparing similarity of different pairs of sub-categories, which is required by the subsequent category coalescence. We use the revised TF×IDF metric in concept re-selection, thereby including the k_m concepts with the highest TF×IDF scores in the global dictionary to represent the documents in each sub-category. We replace the TF value of the concept o_j with the summation of the degree of relevance between a document and each concept, across all sub-categories. The revised TF×IDF measure for o_j is defined as $w_a(o_j) = \left(\sum_{d_i \in D} w_m(d_i, o_j)\right) \times (\log_2 \frac{N_c}{n_j} + 0.01)$ where $w_m(d_i, o_j)$ is the relevance degree of the document d_i and the concept o_j, D is the target collection of documents over all subcategories, N_c is the number of subcategories derived from the category decomposition phase, and n_j is the number of subcategories whose documents contain the concept o_j.

In concept-based document re-representation, ONCE adopts the weighting method as defined in the category decomposition phase to measure the weight of a concept in a document. ONCE performs category coalescence to merge similar categories or sub-categories, thus creating more general categories. ONCE adopts the same category coalescence method as CE2 (i.e., an extended hierarchical agglomerative clustering (HAC) algorithm) to merge sub-categories. Upon completing the category coalescence task, ONCE generates a set of categories which, in effect, have evolved from the document categories previously created by the user or the provider.

Evaluation Design and Results

We empirically evaluate the effectiveness of the proposed ONCE technique, using CE2 and HAC (a traditional hierarchical clustering technique) for benchmark purposes. The following describes our evaluation design (including evaluation document corpus, procedure and metrics) and our experimental evaluation results.

Evaluation Design

Evaluation Document Corpus: For the purpose of concept descriptor learning, we obtained source documents from ACM and used the ACM CCS classification structure as the concept hierarchy for learning concept descriptors. In our evaluation, we removed the first two level-one nodes, A (i.e., General Literature) and B (i.e., Hardware), and their child nodes from the concept hierarchy because of their irrelevance to the documents used in our evaluation experiment. Furthermore, the General and Miscellaneous nodes at level-two and level-three do not depict concrete concepts and therefore were excluded from the hierarchy used in our evaluation. To discover important concept descriptors, we randomly selected a total of 14,729 abstracts of research articles from the ACM digital library. Each article is indexed by one or more designations to indicate its subject area(s) within the CCS classification structure. We removed these nodes in which had only one abstract and their child nodes from the hierarchy because the number of documents is not sufficient for generating descriptors representative of such nodes. The nodes which do not have siblings were also removed from the hierarchy, because we cannot measure the relative importance of the features in our concept descriptor weighting function. As a result, a total of 1,032 nodes were retained in the hierarchy, including 9 nodes at level one, 49 at level two, 263 at level three, and 711 at level four.

For the evaluation purpose, we collected 433 research articles in information systems and technology from a digital library website that specializes in the science literature.[11] Choice of our document corpus is appropriate because most standard document sets, including Reuters RCV1 and Reuters 21578, do not support the use of an established ontology, a distinct focus of our evaluation. A senior faculty of Management Information Systems reviewed all the selected articles and classified them into 17 categories. To maintain a comparable number of categories, we chose from those classified 12 categories, each of which has a minimum of 10 documents. As a result, a total of 400 articles were used in our evaluation, spanning across 12 categories and having an average of 138 words in an article. For each article (document) in the corpus, we used only its title, abstract and keywords in the evaluation.

Evaluation Procedure: For each document, we consider the category specified in the document corpus to be accurate; i.e., true category. To create document categories that simulate influxes of new documents inappropriately assigned to the existing categories, we randomly select some documents from a category and re-assigned them to other categories. Following a particular Gaussian probability distribution, we first split each true category into a dominant subset and multiple minor subsets. Table 1 summarizes the specific evaluation scenarios in which the number of minor subsets under examination ranges from 2 to 6; i.e., from *Gaussian-3* to *Gaussian-6* distributions.

Table 1: Evaluation Scenarios – by Gaussian Distributions

Scenario	Dominant	Minor-1	Minor-2	Minor-3	Minor-4	Minor-5
Gaussian-3	86.6%	13.1%	0.3%			
Gaussian-4	68.2%	27.2%	4.3%	0.3%		
Gaussian-5	54.7%	31.9%	10.9%	2.2%	0.3%	
Gaussian-6	45.1%	31.8%	15.8%	5.6%	1.4%	0.3%

For each evaluation scenario, all dominant subsets remain in their true categories, while each minor subset is randomly merged with the dominant subset from another true category. That is, each minor subset is combined with the dominant subset of a different true category. To minimize potential biases resulting from the randomization process for generating a synthetic dataset, we randomly sample 80% of the documents from the true categories to create a synthetic dataset for an evaluation scenario and repeat the randomization process 30 times. Each evaluation scenario is then to be evolved by ONCE as well as the benchmark techniques; i.e., CE2 and HAC. We evaluate the effectiveness of each investigated technique using its average performances across the 30 random trials.

Evaluation Metrics: We evaluate the effectiveness of each investigated technique in terms of cluster recall and cluster precision, both of which anchor the analysis of the association of a document pair that pertains to the same cluster (Roussinov and Chen 1999; Wei et al. 2005).

[11] CiteSeer Scientific Literature Digital Library, http://citeseer.nj.nec.com/.

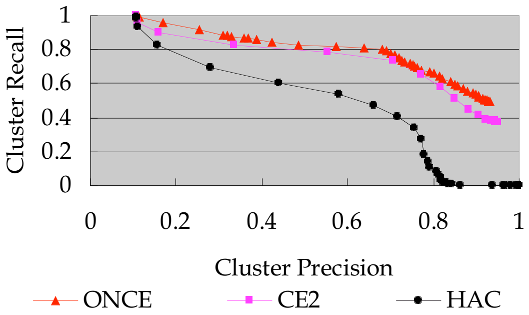
To assess the inevitable tradeoff between cluster precision and cluster recall, we analyze the precision/recall trade-off (PRT) curve which depicts the effectiveness of an investigated technique under different merging thresholds; i.e., inter-cluster similarity threshold for HAC and category coalescence merging threshold for both ONCE and CE2. In this study, we examine the merging threshold for each technique over the range of 0 and 1, in increments of 0.02. Evidently, PRT curves closer to the upper-right corner are more desirable than those closer to the point of origin.

Evaluation Results

Prior to our comparative evaluation, we take a computational approach to tune parameters critical to the proposed OE and ONCE techniques. Three parameters need to be determined their appropriate values in the OE technique, including the number of descriptors for each concept at level one (k_{cd}), the increment of descriptors for each concept at the next level (δ_{cd}), and a pre-specified commonality threshold required in concept refinement (α_p). On the other hand, the ONCE technique has to tune three parameters, including the intra-category similarity threshold (α_s), the number of features for category decomposition (k_s), and the number of features for category amalgamation (k_m). According to our experimental tuning results, we set k_{cd} at 20, δ_{cd} at 10, α_p at 10%, α_s at 0.45, k_s at 100 and k_m at 40 in the subsequent evaluation experiments.

Using the parameter values selected based on our parameter-tuning analyses, we design and conduct evaluations to compare the effectiveness of ONCE, CE2, and HAC. Our evaluation involves different category-evolution scenarios. As shown in Figure 4-A, both ONCE and CE2 noticeably outperform HAC in the *Gaussian-3* scenario. Moreover, the cluster recall of ONCE is 5% higher than that of CE2 across all the different levels of clustering precision attained by both techniques. Overall, ONCE appears more effective than CE2 in the *Gaussian-3* scenario.

As shown in Figure 4-B, 4-C and 4-D, ONCE and CE2 become less effective when the quality of (existing) document categories deteriorates; i.e., from the *Gaussian-4* scenario to the *Gaussian-6* scenario. The decrease in performance is not considerable; both techniques remain advantageous over HAC in these evaluation scenarios. Overall, ONCE appeared more effective than CE2, particularly when the required cluster recall is higher, suggesting the number of evolving document categories is smaller than or closer to the number of true categories in the document corpus; i.e., 12. At average cluster recall levels, ONCE and CE2 are largely comparable in effectiveness. ONCE appears to be more effective than CE2 at higher levels of cluster precision, in scenarios characterized by *Gaussian-4, Gaussian-5 and Gaussian-6* in our evaluation. Overall, our comparative analysis results suggest that ONCE is more effective than HAC, and outperforms CE2 in all the evaluation scenarios investigated. The effectiveness of ONCE seems more robust than that of CE2 over the range of document-category quality examined.



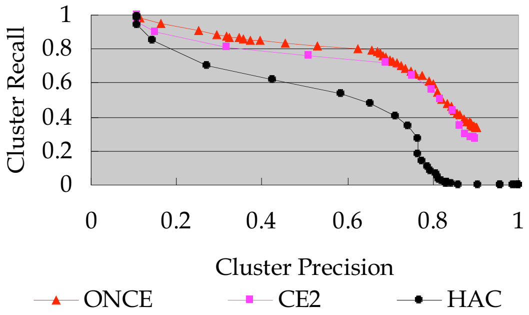
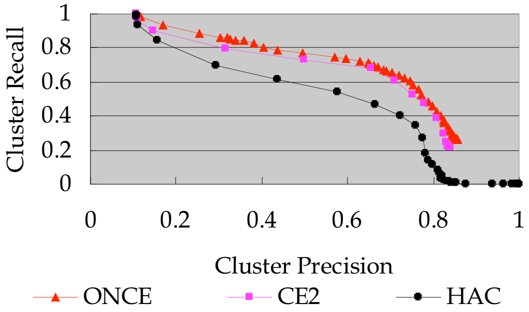


Figure 4-A: ***Gaussian-3*** **Distribution Scenario**

Figure 4-B: ***Gaussian-4*** **Distribution Scenario**



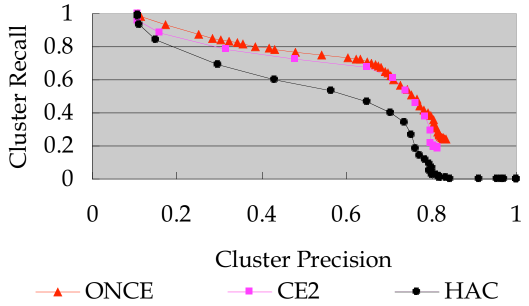


Figure 4-C: ***Gaussian-5*** **Distribution Scenario**

Figure 4-D: ***Gaussian-6*** **Distribution Scenario**

Conclusion and Future Research Directions

Continually, an existing category has to adapt to the changes in its document collection. Most previous document clustering research has taken a full or complete discovery approach; i.e., discovering categories from ground zero. As a consequence, this approach creates a single set of categories for all users without taking into account individuals' preferences or prior grouping behaviors and therefore is not likely to support personalization. Furthermore, the resultant categories may significantly deviate from those expected by or familiar to the user; thus, demanding an increased cognitive load on the part of the user when browsing through the new categories.

Though the proposed CE2 technique in a prior research has attained to a satisfactory effectiveness in preserving personal preference, it has several inherent limitations that need to be addressed. Motivated by the need of more effective and advanced document management approach, we propose ONCE by addressing the problems of word mismatch and ambiguity arisen when performing category evolution on the lexical level. ONCE evolves document categories based on the belonging concepts of documents rather than the frequency of features. We empirically evaluate the effectiveness of ONCE using the synthetic document sets. Our empirical evaluation results reveal that the effectiveness of ONCE outperforms its benchmarks (i.e., CE2 and the discovery-based document clustering technique, HAC) across all investigated scenarios.

This study has several limitations that deserve our future research attentions. First, our evaluation used simulated rather than real-world scenarios. To mediate this limitation, we are currently designing further evaluations that involve human subjects and use real-world document-management contexts. Second, this study focuses on single-category documents. Understandably, a document may simultaneously pertain to multiple categories (to equal or differential degrees). In turn, this requires effective document category management capable of dealing with multi-category documents. Moreover, our ontology enrichment technique assumes the availability of a concept hierarchy. While this assumption is generally reasonable in many domains, the creation and maintainability of such a concept hierarchy is knowledge intensive and time consuming. Hence, an effective approach for learning concept hierarchies from document corpora will extend our proposed ONCE technique to any application domains and would have a profound impact on ontology creation, evolution, and maintenance. Finally, the proposed ONCE technique provides a basis for continued ontology-based document management research. The development and evaluation of advanced ontology-based techniques for text categorization and document clustering represent interesting and essential future research directions.

References

Barreau, D. K., "Context as A Factor in Personal Information Management Systems," *Journal of the American Society for Information Science,* (46:5), June 1991, pp. 327-339.

Brill, E., "A Simple Rule-based Part of Speech Tagger," *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, Association for Computational Linguistics, 1992, pp. 152-155.

Brill, E., "Some Advances in Rule-based Part of Speech Tagging," *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, AAAI Press, 1994, pp. 722-727.

Case, D. O., "Conceptual Organization and Retrieval of Text by Historians: The Role of Memory and Metaphor," *Journal of the American Society for Information Science,* Vol. (42:9), October 1991, pp. 657-668.

El-Hamdouchi, A. and Willett, P., "Hierarchical Document Clustering Using Ward's Method," *Proceedings of ACM Conference on Research and Development in Information Retrieval*, 1986, pp. 149-156.

Fensel, D., *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, 2000.

Gruber, T. R., "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, (5), 1993, pp. 199-220.

Johnson, E. J., Bellman, S., and Lohse, G. L., "Cognitive Lock-in and The Power Law of Practice," *Journal of Marketing,* (67:2), April 2003, pp. 62-75.

Kaufman, L. and Rousseeuw, P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, 1990.

Keet, C. M., *Aspects of Ontology Integration*, The PhD proposal, School of Computing, Napier University, Scotland, 2004.

Kwasnik, B. H., "The Importance of Factors That Are Not Document Attributes in The Organization of Personal Documents," *Journal of Documentation,* (47), 1991, pp. 389-398.

Lakoff, G., *Women, Fire and Dangerous Things: What Categories Reveal about the Mind.* Chicago: University of Chicago Press, 1987.

Larsen, B. and Aone, C., "Fast and Effective Text Mining Using Linear-time Document Clustering," *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 16-22.

Oliver, R. L., "A Cognitive Model for the Antecedents and Consequences of Satisfaction," *Journal of Marketing Research,* (17:4), November 1980, pp. 460-469.

Pantel, P. and Lin, D., "Document Clustering With Committees," *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland: ACM Press, 2002, pp. 199-206.

Quiroga, L. M., "Crosby, M. E., and Iding, M. K. Reducing Cognitive Load," *Proceedings of the 37th Hawaii International Conference on Systems Sciences*, 2004.

Roussinov, D. and Chen, H., "Document Clustering for Electronic Meetings: An Experimental Comparison of Two Techniques," *Decision Support Systems,* (27:1), 1999, pp. 67-79.

Rucker, J. and Polanco, M. J., "Siteseer: Personalized Navigation for the Web," *Communications of the ACM*, (40:3), March 1997, pp. 73-75.

Voorhees, E. M., "Implementing Agglomerative Hierarchical Clustering Algorithms for Use in Document Retrieval," *Information Processing and Management*, (22), 1986, pp. 465-476.

Voutilainen, A., "NPtool: A Detector of English Noun Phrases," *Proceedings of Workshop on Very Large Corpora*, 1993.

Wei, C. and Hu, P. J., and Dong, Y. X., "Managing Document Categories in E-commerce Environments: An Evolution-based Approach," *European Journal of Information Systems*, (11:3), 2002, pp. 208-222.

Wei, C. P., Hu, P., and Lee, Y. H., "An Evolution-based Approach to Preserving User Preferences in Document Category Management," *Proceedings of 9th Pacific-Asia Conference on Information Systems (PACIS)*, July 2005.

Wei, C., Yang, C. S., Hsiao, H. W., and Cheng, T. H., "Combining Preference- and Content-based Approaches for Improving Document Clustering Effectiveness," *Information Processing and Management,* (42:2), March 2006, pp. 350-372.