ELSEVIER

Automation in Construction 12 (2003) 395–406

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

Automating hierarchical document classification for construction management information systems

Carlos H. Caldas, Lucio Soibelman*

Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Newmark CE Lab. MC 250, 205 North Mathews Avenue, Urbana, IL 61801, USA

Accepted 8 January 2003

Abstract

The widespread use of information technologies for construction is considerably increasing the number of electronic text documents stored in construction management information systems. Consequently, automated methods for organizing and improving the access to the information contained in these types of documents become essential to construction information management. This paper describes a methodology developed to improve information organization and access in construction management information systems based on automatic hierarchical classification of construction project documents according to project components. A prototype system for document classification is presented, as well as the experiments conducted to verify the feasibility of the proposed approach.

Keywords: Construction management; Classification systems; Information management; Information systems; Text/data mining

1. Introduction

The use of communications and information technologies in the construction industry is creating new opportunities for collaboration, coordination, and information exchange among organizations that work on a construction project. Inter-organizational construction management information systems are increasingly being used for this purpose. They comprise a set of interrelated components that collect, retrieve, process, store, and distribute data to support planning, control, and decision-making among project organizations. In the distributed and dynamic construction environment, the ability to exchange and integrate data from different sources and in different formats becomes crucial to the development of the construction processes supported by these management information systems. Furthermore, the data collected in these systems provide a valuable source for data mining [11,28]. Discovered knowledge can be used to increase the performance of future activities and projects.

Given that a large percentage of the project documents is generated in text format, methods for organizing and improving access to the information contained in these types of documents become essential to construction information management. Construction information classification systems (CICSs) can be used to support this information management process. The classification structure in a construction information

* Corresponding author. Tel.: +1-217-333-4759; fax: +1-217-333-9464.

E-mail addresses: caldas@uiuc.edu (C.H. Caldas), soibelma@uiuc.edu (L. Soibelman).

0926-5805/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0926-5805(03)00004-9

classification system (CICS) defines concept hierarchies that can be used for document classification, providing a common framework for document organization and management among project organizations. These classification frameworks can be embedded in inter-organizational information systems, like project websites, project management software, and document management systems. Examples of CICSs include: the CSI MasterFormat [17], CSI UniFormat [33], CI/SfB, Uniclass, and the Overall Construction Classification System [20].

One limitation of the existing inter-organizational information systems is the reliance on manual classification methods conducted by human experts. With the growth in the use of information technologies by construction companies, the increasing availability of electronic documents, and the development of model-based systems, manual classification becomes impractical. One example of the limitations of manual classification is the time and effort that would be required to classify all documents created in a construction project (contracts, specifications, meeting minutes, change orders, field reports, and requests for information, among others), according to all components of a CICS.

Another limitation of the current systems is the consideration of documents as single units for the purpose of classification and retrieval. Many construction documents, including specifications and meeting minutes, should clearly be divided and then assigned to more than one item of a CICS. This limitation can be illustrated by the case in which a project manager wants to access information contained in meeting minutes regarding a specific CSI MasterFormat item in order to solve an issue. Using current technologies, the project manager would need to manually search and analyze each document individually in order to obtain the desired information.

A third problem that exists in available systems is the lack of support for differences in vocabularies and naming conventions. This problem can be illustrated by the case in which an architect gives a name for a particular object in a project model. Since there is usually no standard vocabulary among organizations that participate in a construction project, references to that particular object in project documents are often done using different names. Using current technologies, project managers would need to map the model object's name to the terms being used in the different construction documents.

The previously mentioned limitations and the push towards fully integrated and automated project processes justify the need for the development of automated classification methods for construction project documents that can explore the internal characteristics of these documents and adapt to different classification frameworks.

This paper presents a unique way to improve information organization and access in inter-organizational construction management systems based on methods for automated hierarchical classification of construction project documents according to CICSs items. In order to accomplish this goal, a combination of techniques from the areas of information retrieval and text mining was explored. As a result, a methodology for automated hierarchical document classification was devised and implemented. A prototype of a construction document classification system was also developed to provide easy deployment and scalability to the classification process. The developed prototype automated all steps of the text classification process. Experiments were conducted to validate the results and demonstrate the applicability of the implemented techniques.

2. Construction management information systems

The escalating globalization and complexity of construction projects have increased the participation of companies from diverse locations in project teams [3]. In this environment, effective inter-organizational construction management information systems able to minimize time and distance constraints are necessary. Examples of such systems are described extensively in literature [18,19,22,27,32,34,39]. In the distributed and dynamic construction environment, the ability to exchange and integrate information from different sources and in different data formats becomes crucial to the improvement of the construction processes supported by these systems. Simoff and Maher [26] argue that a key issue in managing construction information is the diversity of data types, including:

- structured data files, stored in database management systems or specific applications, such as data

warehousing, enterprise resource planning, cost estimating, scheduling, payroll, finance, and accounting;
- semi-structured data files, such as HyperText Markup Language (HTML), Extensible Markup Language (XML), or Standardized General Markup Languages (SGML) files;
- unstructured text data files, such as contracts, specifications, catalogs, change orders, requests for information, field reports, and meeting minutes;
- unstructured graphic files stored in binary format, such as 2D and 3D drawings; and
- unstructured multimedia files, such as pictures, audio, and video files.

For instance, let us consider a typical construction situation where a construction manager wants to find all available information about one construction activity, say, placing concrete in a slab. He/she will probably find the drawings in computer-aided design (CAD) files, the cost estimates in files produced by cost estimation systems, the schedule in files generated by project management software, the specifications and contracts in text documents, the communications among project members in e-mail files, and price quotes in files collected from different websites. A major task is how to retrieve, classify, and integrate information in these different file formats, especially considering that the files can also be stored in different organizations, computers, or file systems.

Information integration methodologies have been investigated worldwide in order to improve information organization and access in inter-organizational construction management information systems. Teicholz [31] argues that project information should be integrated in three dimensions: "(1) horizontal integration of multiple disciplines that take part in a construction project; (2) vertical integration of multiple stages in the life cycle of a facility; and (3) longitudinal integration over time, which is also related with the capture of knowledge that allows improved performance or better decisions in the future."

Fisher and Kunz [8] argue that technical and managerial strategies have been used to improve information integration. On the technical side, there are four approaches to achieve integration [21,40]: "(i) communication between applications; (ii) knowledge-based interfaces linking multiple applications and multiple databases; (iii) integration through geometry; and (iv) integration through a shared project model holding all the information relating to a project according to a common infrastructure model."

The technical integration through a shared project model can be based on the creation of model-based systems using 3D/4D CAD [1] or on the use of distributed software architectures to facilitate the integration of decentralized project information [29,32]. The adoption of data standards can support these integration approaches. Examples of initiatives in this area are presented by Eastman [7], and include the ISO-STEP, the Industry Foundation Classes (IFC) created by the International Alliance for Interoperability [12], and the aecXML specification [2].

Currently, the majority of the architecture, engineering, construction, and facilities management (AEC/FM) information integration initiatives focus on structured data types. Nevertheless, Soibelman and Caldas [27] argue that a large percentage of the construction data is stored on semi-structured and unstructured files. Recent research work addressed some of the issues related with unstructured data integration. Fruchter [9] describes tools to capture, share, and reuse project information. Garrett et al. [10] explore the use of text analysis for building up classifications of regulation sections. Wood [38] describes an approach to extracting concepts from textual design documentation. Brű̋ggemann et al. [4] proposed the use of arbitrarily structured metadata to markup documents. Scherer and Reul [24] use text clustering techniques to group similar documents and retrieve project knowledge from heterogeneous AEC/FM documents. Yang et al. [35] and Kosovac et al. [16] proposed the use of controlled vocabularies (thesauri) to integrate heterogeneous data representations. Since a great percentage of AEC/FM information is exchanged using text data files, the management of the information contained in these types of documents becomes crucial to construction information management.

3. Construction information classification systems

Construction management information systems generate a significant quantity of data that needs to be organized, stored, accessed, and used by all project organizations. The increase in the amount and types of information generated and the construction industry's subsequent reliance on it motivated the creation of classification standards that can comprehend the full scope of construction information. These standards enable the organization of project information and facilitate the communication between project organizations throughout the project's life cycle.

The information classification standards created by the AEC/FM industry are called construction information classification systems [13]. They can be defined as a standard representation of construction project information. According to Kang and Paulson [13,14], a construction information classification system provides a common method for improving organization and coordination of information in construction projects. Examples of CICSs include the CSI Masterformat [17], the CSI Uniformat [33], and the Overall Construction Classification System [20], and Uniclass [14]. For instance, in OCCS project facilities, constructed entities, spaces, elements, work results, products, process phases, process services, process participants, process aids, process information, and attributes are all defined in a standard manner. Therefore, CICSs provide a common framework for information organization and access in construction management information systems as well as knowledge dissemination, being an essential component in the integration of construction project information.

4. Automated hierarchical construction document classification

From the observations and problems presented in Sections 1 and 2, we can infer that information integration, organization, and access should be considered in construction management. Since a great percentage of the information exchanged among construction organizations is stored in text data files, the management of the information contained in these types of documents becomes essential. In order to improve the management of text-based information, an automated document classification method was devised and implemented. The method was designed according to the construction document classification process developed by the authors and described in Ref. [6]. The importance of this study is that automated document classification methods can be used to improve information organization and access in current information management systems as well as being a foundation for integration of construction documents in emerging model-based systems.

Experiments were conducted in order to evaluate the alternative methods that could be applied in each of the phases of the document classification process. The database selected for this evaluation was the Sweet's Product Marketplace [30]. This database stores data from over 10,700 manufacturers and 61,300 products for the construction industry. Construction products are classified using the hierarchical structure of CSI MasterFormat [17] in this database.

The experiments were conducted using 3030 randomly selected documents from the Sweet's database. The goal was to verify the classification accuracy of the proposed automated document classification method, using the classification decisions already defined in the Sweet's database as a benchmark. The selected documents were originally classified in the database according to a subset of 121 CSI MasterFormat items. These items were distributed according to the CSI MasterFormat classification hierarchy and were composed of 16 items on level one, 52 items on level two, and 53 items on level three.

The activity diagram of the proposed document classification process is presented in Fig. 1. The definition of the classes and the selection of the training positive, training negative, testing positive, and testing negative documents that will be used to create the classification model and verify their accuracy are the initial activities that should be conducted.

The documents used to create the classification models as well as the new documents to be classified are usually stored in different data formats including: word processor, spreadsheet, e-mail, HTML, XML, PostScript (PS), and Portable Document Format (PDF) files. In order to apply the classification algorithms, these files need to be converted to text file format. This is usually done using file converter systems in order to create a text version of each document, while keeping the original documents in their native formats and locations. The text versions can then be used in the remaining activities of the classification process.

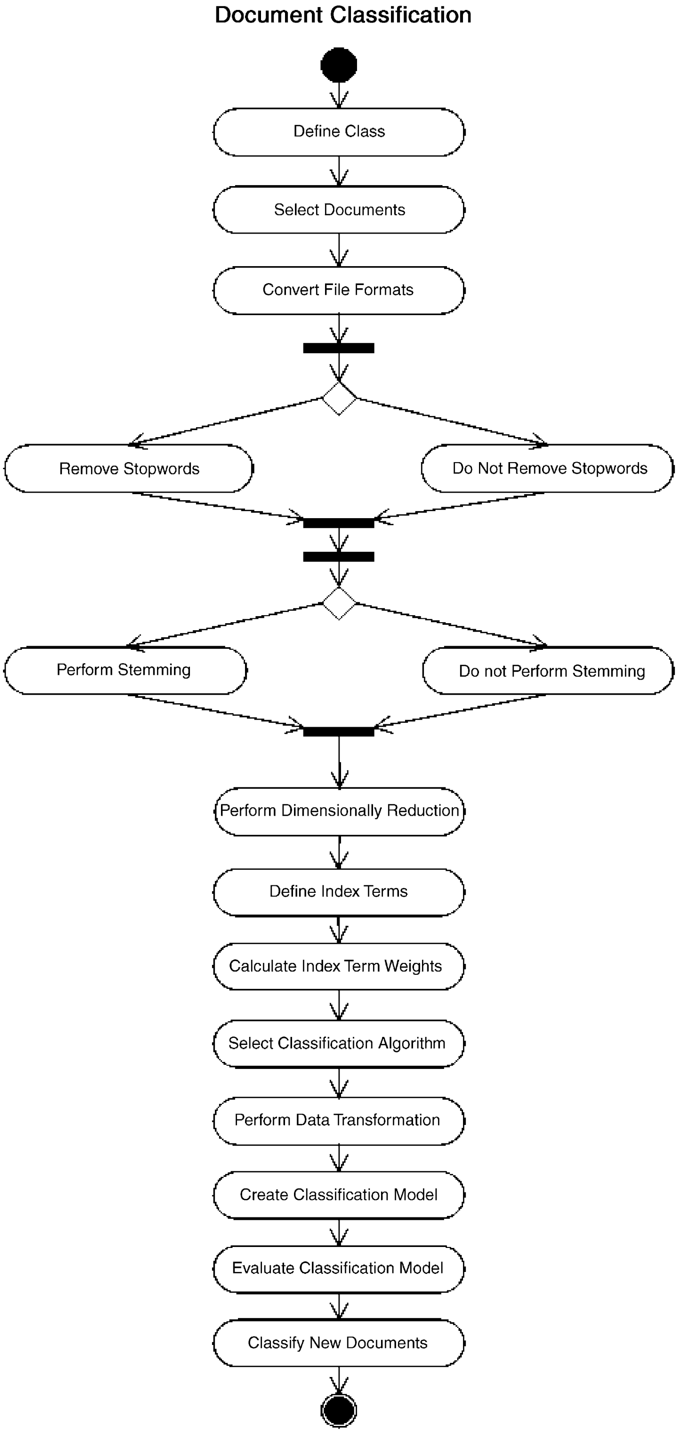

Fig. 1. UML activity diagram of CDCS.

The next two steps require decisions regarding removal of stopwords and stemming. Stopwords are frequent words that do not carry information relevant to text classification like conjunctions, prepositions, and pronouns. Stemming is the process of prefix and/or suffix removal to generate word stems. This is done to group words that have the same conceptual meaning. Our experiments revealed that the removal of stopwords, as well as the use of stemming algorithms improves classification accuracy in most of the cases. The index terms were obtained in one of the steps of the document classification process. Therefore, predefined index terms were not used in the process.

According to Sebastiani [25], a major characteristic, or difficulty of text classification problems is the high dimensionality of the feature space. Many classification algorithms cannot deal with such a large feature set, since processing is extremely costly in computational terms. Hence, in many cases, there is a need to reduce the original feature set, which is commonly known as dimensionality reduction (DR) or attribute selection in the pattern recognition literature.

Various DR methods have been tested in this research. These methods are grounded on concepts from the areas of information theory and linear algebra [36]. In our experiments, the information gain method gave satisfactory results. In the information gain method, the expected reduction in entropy caused by selecting a term that will be used to classify the documents is calculated for all terms that occur in the documents belonging to each class. Terms with highest information gain are selected. The information gain is calculated using the following formula:

$$\begin{aligned}\mathrm{Gain}(\mathrm{I},\mathrm{C}) = {} & \mathrm{Entropy}(\mathrm{T},\mathrm{C}) - (N_{\mathrm{hasT}}/N_{\mathrm{total}}) \\ & \times \mathrm{Entropy}(\mathrm{T},\mathrm{C}_{\mathrm{hasT}}) - (N_{\mathrm{noT}}/N_{\mathrm{total}}) \\ & \times \mathrm{Entropy}(\mathrm{T},\mathrm{C}_{\mathrm{noT}}),\end{aligned}$$

where: Gain(T,C) = Information gain for term T in class C; Entropy(T,C) = $-(N_{\mathrm{pos}}/N_{\mathrm{total}}) \times \log_2 (N_{\mathrm{pos}}/N_{\mathrm{total}}) - (N_{\mathrm{neg}}/N_{\mathrm{total}}) \times \log_2 (N_{\mathrm{neg}}/N_{\mathrm{total}})$; Entropy(T,$\mathrm{C}_{\mathrm{hasT}}$) = $-(N_{\mathrm{poshasT}}/N_{\mathrm{hasT}}) \times \log_2 (N_{\mathrm{poshasT}}/N_{\mathrm{hasT}}) - (N_{\mathrm{neghasT}}/N_{\mathrm{hasT}}) \times \log_2 (N_{\mathrm{neghasT}}/N_{\mathrm{hasT}})$; Entropy(T,$\mathrm{C}_{\mathrm{noT}}$) = $-(N_{\mathrm{posnoT}}/N_{\mathrm{noT}}) \times \log_2 (N_{\mathrm{posnoT}}/N_{\mathrm{noT}}) - (N_{\mathrm{negnoT}}/N_{\mathrm{noT}}) \times \log_2 (N_{\mathrm{negnoT}}/N_{\mathrm{noT}})$; N_{total} = Total number of training documents in class C; N_{pos} = Total number of positive training documents in class C; N_{neg} = Total number of negative training documents in class C; N_{hasT} = Total number of training documents in class C that has term T; N_{noT} = Total number of training documents in class C that does not have term T; N_{poshasT} = Total number of positive training documents in class C that has term T; N_{neghasT} = Total number of negative training documents in class C that has term T; N_{posnoT} = Total number of positive training documents in class C that does not have term T; N_{negnoT} = Total number of negative training documents in class C that does not have term T.

The research demonstrated that the effectiveness of DR methods depends on the classification method used. For instance, the results for support vector machines [15] without dimensionality reduction were slightly better than when dimensionality reduction was used. Table 1 presents the classification accuracy results for support vector machines in different CSI MasterFormat levels without dimensionality reduction, as well as the best classification result obtained from the test cases where dimensionality reduction was used.

Classification algorithms cannot directly interpret text documents. For this reason, a preparation and indexing procedure that maps a text document into a compact representation of its content needs to be uniformly applied to training and test documents. The vector space model was selected for document representation because the resulting model can be uniformly applied to the different classification algorithms analyzed. In the vector space model, vectors represent documents. The collection of documents is represented by an $m \times n$ term-by-document weighted frequency matrix $\mathbf{A}=\{a_{ij}\}$, where a_{ij} was defined as the weight of a term i in document j. Each of the m

Table 1
Effect of dimensionality reduction on classification accuracy using SVM

CSI MasterFormat level	Classification accuracy	
	Dimensionality reduction	
	Without (%)	With (%)
Level 1	95.88	94.33
Level 2	91.53	88.64
Level 3	86.37	83.17
All levels	92.05	89.53

unique terms in the document collection is assigned a row in the matrix, while each of the n documents in the collection is assigned a column in the matrix. A non-zero element, a_{ij}, indicates not only that term i occurred in document j, but also the number of times the term appears in that document or its relative weight. Since the number of terms in a given document is typically far less than the number of terms in the entire document collection, the matrix **A** is usually very sparse. For each class (defined here as a CICS item), only the terms selected after the dimensionality reduction step are used to create the vector space model. An independent vector space model needs to be created for each class.

Several ways of determining the weights a_{ij} were investigated, including: Boolean weighting, absolute frequency, term frequency-inverse document frequency (tf-idf) weighting, and normalized term frequency-inverse document frequency (tfc) weighting [23]. These approaches were originally developed based on two empirical observations regarding text documents: (i) the more times a word occurs in a document, the more relevant it is to the subject of the document, and (ii) the more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

In Boolean weighting, a value of 1 is given to each cell, a_{ij}, in which the term i occurred in document j. In absolute frequency weighting, the cell a_{ij} value is given by the absolute frequency of the term i in document j. $\mathrm{tf-idf}$ weighting uses the following formula to calculate the cell values:

$$\mathrm{tf-idf}_{ki} = f_{ki} \times \log_2(N/d_k),$$

where: tf-idf$_{ki}$ = the tf-idf weight of term k in document i; f_{ki} = the absolute frequency of term k in document i; N = the number of documents in the collection; d_k = the number of documents containing term k.

The reasoning behind the tf-idf weighting is that if the term occurs in many of the documents in the collection, then it does not serve well as a document identifier and should be given a low weight as a potential index term. In tfc weighting, the values for each cell a_{ij} is calculated by the formula:

$$\mathrm{tfc}_{ki} = \mathrm{tf-idf}_{ki} \Big/ \sqrt{\sum_{s=1}^{T} (\mathrm{tf-idf}_{si})^2},$$

where: tfc$_{ki}$ = the tfc weight of term k in document i; tf-idf$_{ki}$ = the tf-idf weight of term k in document i; tf-idf$_{si}$ = the tf-idf weight of term s in document i; T = set of all terms that occurs at least once in the collection.

In tfc weighting, the values of tf-idf weighting are normalized to minimize the effect of length differences among documents. Our experiments demonstrated that these different weighting schemes have different classification accuracies. Table 2 presents the accuracy results in different CSI MasterFormat levels, using the index weighting methods previously described.

The machine learning algorithms used to create the classification models have their own data input format and requirements. Usually, their data input is made using text files containing the data that will be processed. The data transformation step aims to create the data input files required by the classification algorithms. Basically, the information from the vector space model is converted into the appropriate text file format.

Pattern classification algorithms are used to create the classification models. In this case, the classes are represented by the items of a Construction Information Classification System. Hence, construction document classification is defined as the task of assigning a Boolean value to each pair $\{d_j, c_i\} \in D \times C$, where D is a domain of project documents and C is a set of CICS items (classes). A value of T (true) assigned to $\{d_j, c_i\}$ indicates a decision that document d_j is related with item c_i, while a value of F (false) indicates that d_j is not related with item c_i.

Several algorithms were tested, including: naive Bayes, k-nearest neighbors, Rocchio, and support vector machines (SVM). Table 3 presents the classification accuracy results in different CSI MasterFor-

Table 2
Effect of the index weighting methods on classification accuracy

CSI MasterFormat level	Classification accuracy			
	Index weighting method			
	Boolean (%)	Abs. frequency (%)	tf-idf (%)	tfc (%)
Level 1	89.11	81.48	82.98	95.88
Level 2	78.89	65.12	64.70	91.53
Level 3	69.49	50.05	50.32	86.37
All levels	80.58	67.83	68.30	92.05

Table 3
Effect of the classification method on classification accuracy

CSI MasterFormat level	Classification accuracy without dimensionality reduction			
	Classification method			
	Naive Bayes (%)	k-nn (%)	Rocchio (%)	SVM (%)
Level 1	94.18	81.80	93.81	95.88
Level 2	87.87	68.47	88.35	91.53
Level 3	81.93	58.19	84.48	86.37
All levels	88.88	71.15	89.53	92.05

mat levels using different classification algorithms. Since SVM outperformed the other classification methods in this experiment, and was also the method with best performance in other experiments conducted by the authors and reported in Ref. [6], a support vector machine [15] was the algorithm selected for the implementation of the automated hierarchical document classification process.

By using a SVM classifier, a classification model can be created for each class by observing the characteristics of a set of documents that have previously been classified manually by a domain expert. This approach relies on the existence of an initial corpus of documents previously classified according to their relevance to a set of project components. A document d_j is called a positive example of c_i if $\{d_j, c_i\} = T$ and a negative example of c_i if $\{d_j, c_i\} = F$.

Since each construction document can belong to more than one class (one individual document can be related to more than one CICS item), the classification process was designed to handle multiple binary classifications. In this case, each document is compared with each class. For each class, a binary decision is made in order to define whether the document is related or not with that particular class (CICS item). The large number of classes that usually need to be defined in order to classify construction documents imposes another challenge on the classification task. For multiple binary classifications, a classification model has to be created for each of the existing classes.

In support vector machines, each model is defined by a specific multidimensional space composed of all training document vectors for that class. The SVM classifier aims to find a decision surface that best separates the positive and negative training document vectors for each class in a high dimensional feature space. Each dimension in this feature space is represented by an index term, and the coordinate for each dimension is defined by the corresponding index term weight. In its simplest linear separable case, SVM finds a hyperplane that separates the set of positive examples from the set of negative examples with maximum margin. Fig. 2 illustrates the linear separating hyperplane. The points $\boldsymbol{x}$ which lie on the hyperplane satisfy $\boldsymbol{w}\cdot\boldsymbol{x}+b=0$, where w is normal to the hyperplane, $b/\|\boldsymbol{w}\|$ is the perpendicular distance from the hyperplane to the origin, and $\|\boldsymbol{w}\|$ is the Euclidian norm of $\boldsymbol{w}$ [5].

This problem can be solved using constrained quadratic programming optimization methods in which the margin, given by $2/\|\boldsymbol{w}\|$, is maximized subject to the constraints $\mathrm{y}_{\mathrm{i}}*(\boldsymbol{w}\cdot\boldsymbol{x}_{\mathrm{i}}+b)\geq 1$, where $\boldsymbol{x}_{\mathrm{i}}$ represents each individual training document vector for the class being considered and y_{i} corresponds to classification decision ($+1$ for positive documents and -1 for negative documents) for document vector $\boldsymbol{x}_{\mathrm{i}}$. Data about all multidimensional spaces and hyperplanes (support vectors) need to be stored efficiently since these data will be required in order to classify new/unseen documents.

After generating the classification model, its effectiveness is evaluated. The alternative adopted for this evaluation was to randomly split the initial collection of documents into two sets.

- Training set: set of documents that were used to create the classification model.
- Test set: set of documents that were used for testing the effectiveness of the classifier.

In our experiments, the random selection of training and testing sets was repeated 10 times and the results were averaged in order to calculate the accuracy of each classification model.

In each trial, the documents in the test set did not participate in the training set. If this condition was not satisfied, then the experimental results obtained would probably be unrealistically good. The definition of the size of the training set was also crucial to avoid overfitting. This happens when the classifier performs with few errors on the training set and does not generalize to the new test cases.

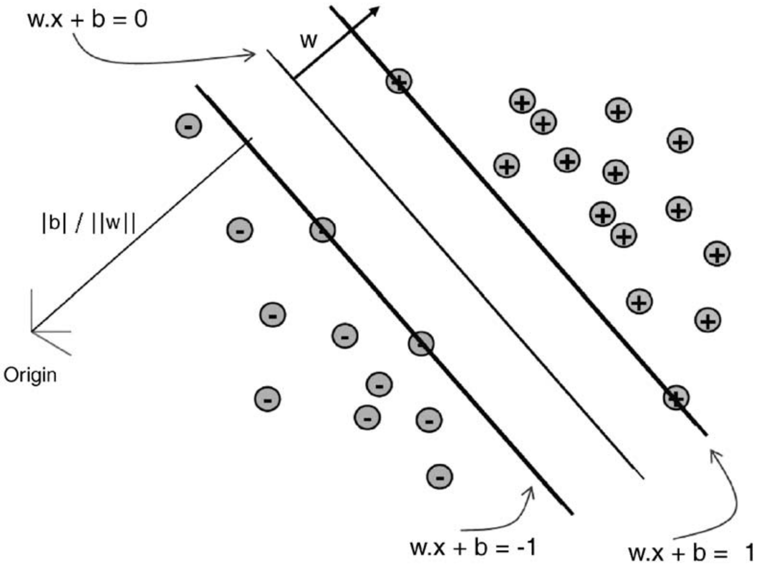

Fig. 2. SVM Classification.

Whenever a new document needs to be classified, it must be projected into the multidimensional space of each of the existing classes considering the same data preparation options (e.g.: use of the stemmer, index term weighting method). This projection is conducted very carefully since the index terms in the document to be classified need to match the right multidimensional space dimensions. Considering that the new document vector is $\boldsymbol{x}_{\text{new}}$, the classification decision for a new document for a given class is given by the sign of $(\boldsymbol{w}\cdot\boldsymbol{x}_{\text{new}}+b)$. A positive value means that the new document is related to this class. A negative value means that the new document is not a member of this class.

Since, there are several classification models (one for each class), the new document needs to be projected in several multidimensional spaces. Therefore, this process needs to be repeated for each of the existing classification models.

5. Implementing automated hierarchical document classification

A prototype system, called the Construction Document Classification System (CDCS), was implemented in order to test the feasibility of the proposed approach. The system enables the classification of construction documents according to the specific classification items found in construction information classification systems. CDCS automates the steps involved in the document classification process previously described. It is currently composed of seven main modules: data selection, data conversion, dimensionality reduction, data preparation, data transformation, learning, and classification. The system was implemented in the programming language Java and uses Java Database Connectivity (JDBC) to communicate with a database management system (SQL Server). This database stores the data generated during the creation of the classification models; this data will also be used in the classification of new documents.

In CDCS, the classification structure can be defined according to a hierarchy of classes. For instance, considering the CSI MasterFormat [17] as the classification structure, the document is initially classified according to each element of the first level (CSI MasterFormat level one-Divisions). For the elements in the first level in which the classification decision was true (meaning that the document was related with that particular CSI MasterFormat level one item-Division), the binary classification can then be conducted for the second hierarchical level (CSI MasterFormat level two). Following the same process, for

Table 4
Hierarchical classification results (level one)

CSI MasterFormat code	Class name	Classification accuracy (%)
01000	General Requirements	93.90
02000	Site Construction	95.23
03000	Concrete	91.13
04000	Masonry	95.40
05000	Metals	90.51
06000	Wood and Plastics	94.87
07000	Thermal and Moisture Protection	96.04
08000	Doors and Windows	97.39
09000	Finishes	96.27
10000	Specialties	96.81
11000	Equipment	99.34
12000	Furnishings	93.96
13000	Special Construction	96.53
14000	Conveying Systems	98.41
15000	Mechanical	99.19
16000	Electrical	97.61
	Level 1	95.88

the elements in the second level in which the classification decision was true (meaning that the document was related with that particular CSI MasterFormat level two item), the binary classification can then be conducted for the third hierarchical level (CSI MasterFormat level three).

Tests using CDCS were conducted to evaluate the performance of the proposed automated hierarchical classification method. Hierarchical classification is more challenging than flat classification because the accuracy tends to reduce in the lower hierarchical levels. This usually happens because it is more difficult to differentiate the classes at the lower levels since they contain fewer training documents and the documents are more similar.

Preliminary results indicated that the highest classification accuracy was achieved using SVM as the classification algorithm, tfc, as the index weighting method, and no dimensionality reduction. This configuration achieved an average accuracy of 95.88% for the first hierarchical level, 91.53% for the second level, and 86.37% for the third. The average classification accuracy for SVMs, considering the tests conducted in all class levels, was 92.05%, which is comparable to human performance in similar manual document classification tasks [37]. Table 4 and Fig. 3 present the hierarchical classification accuracy results for this case.

At first, the fact that the results using dimensionality results were slightly lower than when no dimensionality reduction method was used seems surprising. However, according to Joachims [15], this happens because in text classification there are only very few irrelevant features (index terms). He demonstrated that even features ranked lowest still contain considerable information and that aggressive dimensionality reduction may result in a loss of information. Similar behavior occurred in our experiments. The tfc indexing method considered both the frequency of the index term in the document and in the project collection, and used a normalization method to minimize document vector length differences. Support vector machines performed well because of the high dimensionality of the feature space, composed of document vectors that had only few entries that were not zero. This happens because each document contained only some of the index terms that occurred in the project collection.

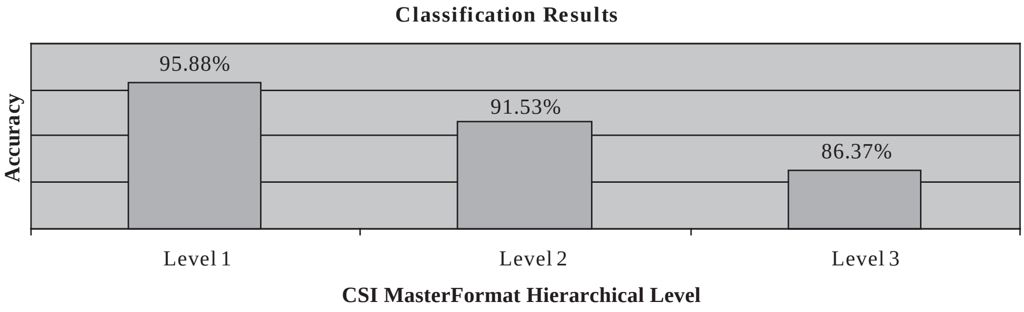

Fig. 3. Hierarchical classification results (Average by Level).

The proposed methodology can also be used to improve the organization and access to more unstructured text documents. It has been successfully tested in other types of construction documents, such as meeting minutes, requests for information, change orders, and design review documents.

6. Conclusions

In this paper, a methodology for automated hierarchical document classification was described and evaluated. Automatic hierarchical classification is part of an ongoing research project that aims to improve the organization and access of unstructured text documents in construction management information systems and facilitate the integration of such documents in model-based systems. This is a very important issue for construction information management because a large percentage of project information is stored in text documents and these documents contain valuable information for decision-making, data analysis, and knowledge discovery.

The methodology supports the generation of classification models based on project information classification structures, such as construction information classification systems or project model objects. After creating these classification models, new construction documents can be effectively classified. The main characteristics of the proposed methodology are:

- It does not require the manual assignment of metadata (keywords or index terms) to all documents in the information system. Manual assignment of metadata is a tedious task. It is also hard to achieve consistency when a large number of users from different organizations are adding documents to the system.
- It does not need the utilization of a controlled vocabulary that would only be effective if it was accepted as a standard by the AEC/FM organizations and adopted by all users of a construction management information system.
- It uses already existing AEC/FM standards to define the categories that will be used for classification; and
- It facilitates the creation of automated mapping mechanisms from documents to project components.

Experiments were conducted to verify the classification accuracy for hierarchical classification structures. A construction products' database, originally classified according to a hierarchical structure, was used in this analysis. The results demonstrated the effectiveness and applicability of automated document classification methods for construction management information systems. Examples of other problems that can benefit from the proposed automated classification method include: analysis of construction project documentation, organization of multimedia project inspection files based on their description, facilitation of automated access to project specifications in proactive project controls systems, identification of problem areas and potential causes of delays, cost overruns, or quality deviations, and generation of lessons learned that could be applied in future activities and projects.

Acknowledgements

The authors would like to thank the National Science Foundation for the support under the grant number 0201299.

References

[1] F.B. Aalami, M. Fischer, J.C. Kunz, AEC 4D-CAD production model: definition and automated generation. CIFE WP 052, 1998.

[2] aecXML, < http://www.iai-na.org/domains/aecxml/about/aecxml_about.html> (Aug 28, 2002).

[3] C.J. Anumba, N.F.O. Evbuomwan, A taxonomy for communication facets in concurrent life-cycle design and construction, Computer-Aided Civil and Infrastructure Engineering 14 (1999) 37–44.

[4] B.M. Brűggemann, K. Holz, F. Molkenthin, Semantic documentation in engineering, Proceedings of the ICCCBE-VIII, Palo Alto, CA, ASCE, Reston, VA, August, 2000, pp. 828–835.

[5] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.

[6] C.H. Caldas, L. Soibelman, J. Han, Automated classification of construction project documents, Journal of Computing in Civil Engineering, 2002 (October) 16 (4), pp. 234–243.

[7] C.M. Eastman, Building Product Models: Computer Environments Supporting Design and Construction, CRC Press, Boca Raton, FL, USA, 1999.

[8] M. Fischer, J. Kunz, The circle: architecture for integrating software, Journal of Computing in Civil Engineering 9 (2) (1995) 122–133.

[9] R. Fruchter, A/E/C teamwork: a collaborative design and learning space, Journal of Computing in Civil Engineering 13 (4) (1999) 261–269.

[10] J.H. Garrett Jr., S.J. Fenves, D.M. Stasiak, A WWW-based regulation broker, CIB Proceedings Publication 198: Construction on the Information Highway, CIB, Rottedam, 1996, pp. 219–230.

[11] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, 2001.

[12] IAI, < http://www.iai-international.org/iai_international/> (Aug 28, 2002).

[13] L.S. Kang, B.C. Paulson, Adaptability of information classification systems for civil works, Journal of Construction Engineering and Management 123 (4) (1997) 419–426.

[14] L.S. Kang, B.C. Paulson, Information classification for civil engineering projects by Uniclass, Journal of Construction Engineering and Management 126 (2) (2000) 158–167.

[15] T. Joachims, Text categorization with support vector machines: learning with many relevant features, Proceedings of ECML-98, Chemnitz, Germany, Springer, Berlin, 1998, pp. 137–142.

[16] B. Kosovac, T. Froese, D. Vanier, Integrating heterogeneous data representations in model-based AEC/FM systems, Proceedings of CIT 2000, Reykjavik, Iceland, CIB, Rotterdam, vol. 1, 2000, pp. 556–566.

[17] MasterFormat, MasterFormat 1995 Edition, Construction Specifications Institute, Alexandria, VA, 1995.

[18] W.J. O'Brien, Implementation issues in project web-sites: a practitioner's viewpoint, Journal of Management in Engineering 16 (3) (2000) 34–39.

[19] OSMOS, Open System for Inter-enterprise Information Management in Dynamic Virtual Environments-OSMOS Project, < http://cic.vtt.fi/projects/osmos/index.html> (Aug 28, 2002).

[20] OCCS, Overall Construction Classification System, < http://www.occsnet.org> (Aug 28, 2002).

[21] Y. Rezgui, Y. Brown, G. Cooper, J. Yip, P. Brandon, J. Kirkham, An information management model for concurrent construction engineering, Journal of Automation in Construction 5 (4) (1996) 343–355.

[22] E.M. Rojas, A.D. Songer, Web-centric systems: a new paradigm for collaborative engineering, Journal of Management in Engineering 15 (1) (1999) 39–45.

[23] G. Salton, C. Buckley, Term weighting approaches in automatic text retrieval, Information Processing and Management 2 (5) (1988) 513–523.

[24] R.J. Scherer, S. Reul, Retrieval of project knowledge from heterogeneous AEC documents, Proceedings of the ICCCBE-VIII, Palo Alto, CA, ASCE, Reston, VA, August, 2000, pp. 812–819.

[25] F. Sebastiani, Machine learning in automated text categorisation. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, CNR, Pisa, Italy, 1999.

[26] S.J. Simoff, M.L. Maher, Ontology-based multimedia data mining for design information retrieval, Proc. of Computing in Civil Engineering, ASCE, Reston, VA, 1998, pp. 212–223.

[27] L. Soibelman, C. Caldas, Project extranets for construction management: the American experience, Proceedings of Entac-2000, May, 2000, Salvador, Brazil.

[28] L. Soibelman, H. Kim, Generating construction knowledge with knowledge discovery in databases, Journal of Computing in Civil Engineering, vol. 16 (1), ASCE, 2002, pp. 39–48.

[29] L. Soibelman, F. Peña-Mora, A distributed multi-reasoning mechanism to support the conceptual phase of structural design, Journal of Structural Engineering 126 (6) (2000) 733–742.

[30] Sweet's.Sweet's Product Marketplace, < http://sweets.construction.com/default.jsp> (Aug 28, 2002).

[31] P. Teicholz, Vision of future practice, Berkeley-Stanford Workshop on Defining a Research Agenda for AEC Process/Product Development in 2000 and Beyond, Stanford, CA, 1999.

[32] ToCEE-Towards a Concurrent Engineering Environment Project, The ToCEE client-server system for concurrent engineering. Final Report-ESPRIT Project No. 20587, 2000.

[33] UniFormat, UniFormat 1998 Edition, 9Construction Specifications Institute, Alexandria, VA, 1998.

[34] VEGA, Virtual Enterprises using Groupware Tools and Distributed Architecture-VEGA Project < http://cic.cstb.fr/ILC/ecprojec/vega/home.htm> (Aug 28, 2002).

[35] M.C. Yang, W.H. Wood, M.R. Cutkosky, Data mining for thesaurus generation in informal design information retrieval, Proceedings of the International Computing Congress, ASCE, Reston, VA, 1998, pp. 189–200.

[36] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, Proceedings of ICML-97, 1997, pp. 412–420, Nashville, TN.

[37] S.A. Weiss, S. Kasif, E. Brill, Text Classification in USENET Newsgroups: A Progress Report, Department of Computer Science, The Johns Hopkins University, Baltimore, MD, 1997 (April).

[38] W.H. Wood, The development of modes in textual design data, Proceedings of the ICCCBE-VIII, Palo Alto, CA, CESE, Reston, CA, 2000 (August), pp. 882–889.

[39] A. Zarli, Y. Rezgui, A survey of internet-oriented technologies for document-driven applications in construction open dynamic virtual environments, Proceedings of CIT 2000-International Conf., vol. 1, Construction Information Technology, Reykjavik, Iceland, 2000, pp. 1089–1101.

[40] Y. Zhu, R.R. Issa, Web-based construction document processing via malleable frame, Journal of Computing in Civil Engineering 15 (3) (2001) 157–169.