A MACHINE LEARNING APPROACH FOR AUTOMATED DOCUMENT CLASSIFICATION: A COMPARISON BETWEEN SVM AND LSA PERFORMANCES

Tarek Mahfouz, Ball State University; James Jones, Ball State University; Amr Kandil, Purdue University

Abstract

The increasing sophistication and complexity of construction projects mandates extensive coordination between different parties and produces massive amounts of documents in diversified formats. The efficient use of these documents has become inevitably needed. The first step in making these documents effectively usable is to create efficient classification methods. This paper proposes automated models for construction document classification using Machine Learning (ML) methodology. To that end, Support Vector Machines (SVM) and Latent Semantic Analysis (LSA) were utilized for the development of the aforementioned models. The developed models were validated through overall classification accuracy and were compared against the Gold Standard of human agreement measures. The adopted research methodology generated 16 SVM and 16 LSA models, out of which the four with the highest accuracy were chosen. These models attained relatively better results than previous models in the surveyed literature. Overall accuracy ranged from 71% to 91%.

Introduction

The U.S. Census data showed that the total construction spending in 2007 was about $14 trillion [1]. This considerable amount of expenditure is constantly at risk due to the dynamic nature of the construction industry and the increasing sophistication and complexity of construction projects. These characteristics created a requirement for an extensive amount of coordination between the different parties, and the production of a massive amount of documents in diversified formats. In an effort to facilitate use and re-use of knowledge included in construction documents, Artificial Intelligence (AI) is being used to address Knowledge Management (KM) practices. It has been extensively utilized to enhance information models, document integration, and expert systems [2]. A wide range of studies were carried out using AI techniques to develop automated and semi-automated tools to enable the utilization of textual data expressed in natural language through text mining, document clustering, controlled vocabularies, and web-based models [3-7]. Although those studies resulted in significant contributions, none of them 1) investigated the development of a generic model for unstructured document automated classification; and 2) utilized Latent Semantic Analysis (LSA).

Therefore, in an attempt to provide a robust document classification methodology for the construction industry, the authors developed automated classifiers through Support Vector Machines (SVM) and Latent Semantic Analysis (LSA). The analyses and models developed in this study focused on two groups of construction documents. The first is made up of documents with high variation in words like transmittals, correspondences, and meeting minutes. The second group relates to documents of low word variations like construction claims and legal documents. To that end, the adopted research methodology 1) investigated SVM and LSA algorithms; 2) developed full and truncated feature spaces for the document sets used; 3) developed 16 SVM and 16 LSA automated classification models; 4) developed two C++ algorithms to facilitate assigning a document to a class prediction; and 5) tested and validated the developed models. It was conjectured that this research stream would help to relieve the negative consequences associated with the lengthy process of analyzing textual documents in the construction industry. In addition, the achieved outcomes of this research highlight the possibility of these techniques being adopted for automated decision support in the construction industry.

The rest of this paper details the following:

- Literature Review: This section of the paper introduces the reader to previous research performed in the field of construction decision support. It also provides background information about the concepts of Support Vector Machines (SVM) and Latent semantic analysis (LSA);
- Methodology: This section highlights the details of the adopted research methodology and describes the different steps of implementation of SVM and LSA model development;
- Results and discussion: This section highlights the results attained and discusses their implications for the current research; and
- Conclusion.

Literature Review

Over the last decade, researchers focused on developing construction information integration tools that were designed to work with structured data like CAD models and scheduling databases. However, a major portion of significant construction information is produced in semi-structured or unstructured formats like contract documents, change orders and meeting minutes, all of which are normally stored as text files [3], [4], [8]. Consequently, facilitating the use of these documents through integrated methods has become a necessity to enhance project control, performance, and data reuse. A number of research studies have addressed this issue. Ioannou and Liu [5] proposed a computerized database for classifying, documenting, storing and retrieving documents on rising construction technologies. Kosovac et al. [9] investigated the use of controlled vocabularies for the representation of unstructured data. Scherer and Reul [10] utilized text-mining techniques to classify structured project documents. Caldas et al. [3] and Caldas and Soibelman [4] used information retrieval via text mining techniques to facilitate information management and permit knowledge discovery through automated categorization of various construction documents according to their associated project component. In addition, Caldas et. al. [11] proposed a methodology for incorporating construction project documents into project management information systems using semi-automated support integration to improve overall project control. Ng et al. [6] implemented Knowledge Discovery in Databases (KDD) through a text-mining algorithm to define the relationships between type and location of different university facilities, and the nature of the required maintenance reported in the Facility Condition Assessment database.

Support Vector Machines (SVM):

The following is a descriptive background of the Support Vector Machines concept. SVM classification aims to find a surface that best separates a set of training data points into classes in a high dimensional space. In the current research, it aims to define the construction subject pertinent to each of the training documents based on the word representation in its content. In its simplest linear form, a SVM finds a hyperplane that separates a set of positive examples (documents belonging to a construction subject) from the set of negative examples (documents not belonging to the same construction subject) with a maximum margin. Binary classification is performed by using a real-valued hypothesis function, equation 1, where input x (document) is assigned to the positive class (Specific Subject) if $f(x) \geq 0$; otherwise, it is assigned to the negative class.

$$Y = \langle w.x \rangle + b \quad (1)$$

For a binary linear separation problem, a hyper-plane is assigned to be $f(x) = 0$. With respect to equation 1, the vectors w (weight vector) and b (functional bias) are the parameters that control the function of the separation hyper-plan (refer to Figure 1). In addition, x is the feature vector, which may have different representations based on the nature of the problem. Within the context of the current research, the input feature space X constitutes the training documents that are defined by the vectors x and o in Figure 1.

In the development of the proposed SVM models, a problem emerges if the data are not linearly separable. Assigning an unstructured document to a specific class cannot be represented by a simple linear combination of its content words. Consequently, a more sophisticated, higher dimension space is needed for the representation of the current problem in order for it to be linearly separable. As the literature in this field suggests, Kernel representations provides a solution to this problem by transforming the data into a higher dimensional feature space to enhance the computational power of linear machine learning [12]. As shown in equation 1, the representation of a case in the feature space for linear machine learning is achieved as a dot product of the legal factors, vector (x), and the weight vector, (w). By introducing the appropriate Kernel function, cases are mapped to higher feature space (equation 2 and Figure 1) transforming the prediction problem from a linearly inseparable to a linearly separable one. In this manner, the input space X is mapped into a new higher feature space, $F=\{\emptyset(x)|x$, where $\emptyset$ is the kernel transformation function.

$$x=(X_1, \ldots, X_n) \rightarrow \Phi(X)=(\Phi_1 X_1, \ldots, \Phi_n X_n) \quad (2)$$

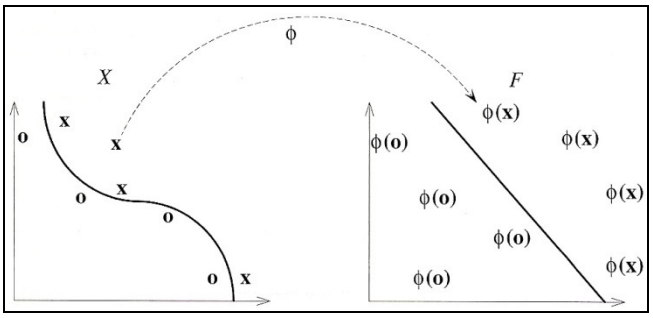


Figure 1. SVM Kernel Transformation and Classification

Latent Semantic Analysis (LSA):

Unlike SVM, "Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the meaning of words" [13]. It attempts to model the mechanism of exactly how words and passage meanings can be constructed from experience with language. A corpus of related text imposes constraints on the meaning and semantic similarities of a word. For example, a word like "bank" can mean "river

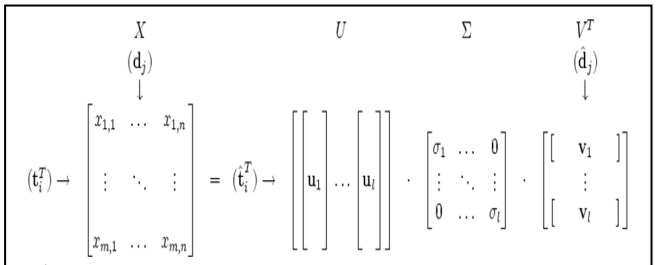
side" or "institution for financial transactions", based on the constraints imposed by the rest of words within a body of text. The theory of LSA hypothesizes that the meaning of a text is conveyed by the words from which it is composed. Therefore, it is based on determining the meaning of a word by solving these constraints in a mathematical form by utilizing linear algebra, particularly Singular Value Decomposition (SVD).

LSA is based on the concept of Vector Space Model implemented by SVM. However, the main advantage of LSA is that it utilizes a truncated space in which the number of features is decreased. LSA represents word and passage meanings in a form of mathematical averages. Word meanings are formulated as an average of the meaning of all the passages in which it appears, and the meaning of a passage as an average of the meaning of all the words it contains. LSA methodology applies SVD for the reduction of dimensionality in which all of the local word context relations are simultaneously represented. Unlike many other methods, LSA employs a preprocessing step in which the overall distribution of a word, over its usage contexts, is first taken into account independent of its correlations with other words. It then implements three well-defined steps.

First, text documents within a training corpus are represented in the form of maa trix (Figure 2). Each row of the developed matrix demonstrates a specific word in the training corpus. Each column of the matrix stands for a text document. Each cell contains the frequency with which the word of its row appears in the passage denoted by its column [13]. Often, the number of features, m, is much higher than the number of documents, n, within the collection. The developed m by n matrix will contain zero and nonzero elements. Generally, a weighing function is applied to nonzero elements to give lower weights to high-frequency features that occur in many documents and higher weights to features that occur in some documents but not all [14]. Weighing functions are of two types; namely, local and global. The former relates to increasing or decreasing a nonzero element with respect to each document. The latter relates to increasing or decreasing a nonzero element across the whole collection of documents.

Second, SVD is applied to the developed matrix to achieve an equivalent representation in a smaller dimension space [15]. With SVD, a rectangular matrix is decomposed into the product of three other matrices (Figure 2). One component matrix describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed [16].

Third, the number of features adopted for analysis is determined (Truncation). Since the singular value matrix is organized in ascending order based on the weight of each term, it is easy to decide on a threshold singular value below which term significance is negligible (refer to Figure 3) [17]. For an original matrix A with rank k, a newly truncated matrix Ak can be formulated by the dot product illustrated in equation 3.



where:

- The term-document matrix *X* represents how important a term is in a given document.
- Σ is a diagonal matrix representing the weights of the concepts. Usually, the SVD algorithm produces a Σ with $\sigma_1 > \sigma_2 > ... > \sigma_k > 0$.
- The columns of *U* are the terms in concept space and the rows of *V* are the documents in the concept space. Since SVD is used, *U* and *V* are unitary matrices; that is, the rows of *U* and the columns of *V* are of unit length and are pairwise orthogonal.
- Not all concepts are necessarily used. Only those with sufficiently large singluar values (i.e. σ-s) may be selected. The resulting matrix is the best low-rank approximation to *X* in Frobenius norm.

Figure 2. Matrix Representation in LSA [18]

$$A_i = \sum_{i=1}^{k} u_i \sigma_i v_i^T \rightarrow A_k = U_k \Sigma_k V_k^T \quad (3)$$

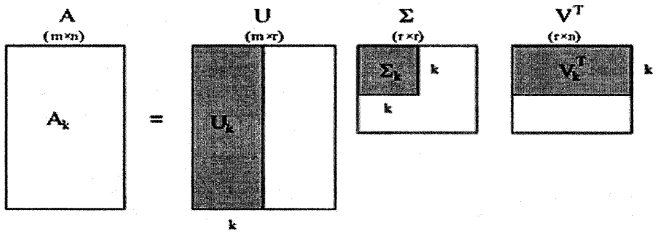


Figure 3. K Dimensional Space Representation in LSA [18]

By representing any document in the generated concept space, it is then possible to calculate "distance" on the set of such document representations, thus computing whether two

such representations are close, which usually implies that the documents themselves are related.

Methodology

The following sections of the paper describe the different steps of developing, implementing, and validating the machine-learning models. The adopted research methodology under the current task was composed of four main stages, as illustrated in Figures 4 and 5. These stages wer defined as 1) Corpus Development; 2) Feature Space Development for LSA and SVM; 3) Model Design and Implementation; and 4) Model Testing and Validation.

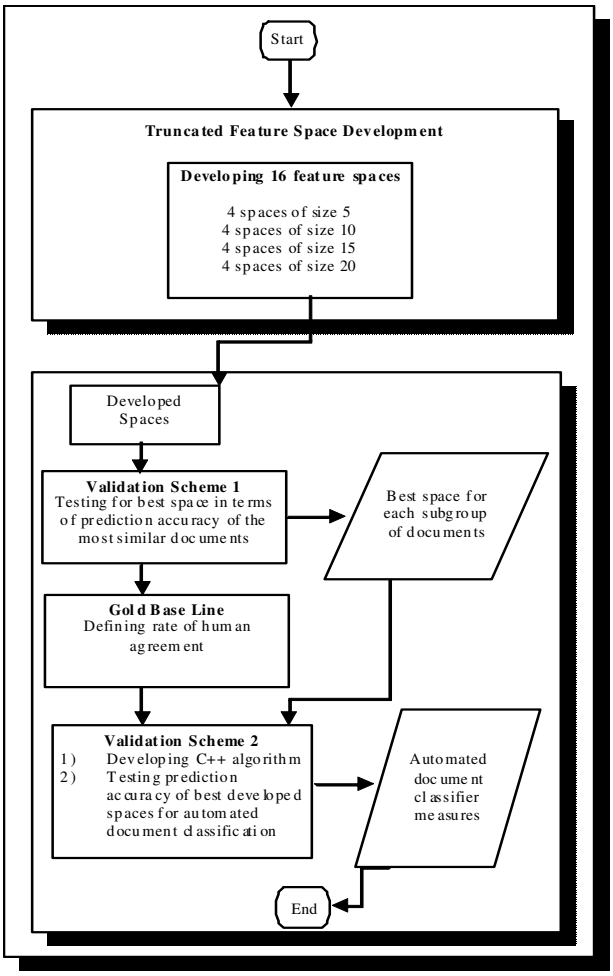


Figure 4. LSA Research Methodology

Corpus Development:

The current research task was concerned with two types of unstructured documents of high and low word variations. To that end, the first group was made up of two subgroups; the first included 300 correspondences and the second was made up of 150 meeting minutes. Furthermore, the second group had 25 claims and 300 DSC cases as its first and second subgroups, respectively. The documents pertinent to the correspondences, meeting minutes, and claims were gathered from a number of projects that were performed around the world. However, the DSC cases were gathered from the Federal Court in New York due to the abundance of cases. They were compiled using LexisNexis, a web legal retrieval system.

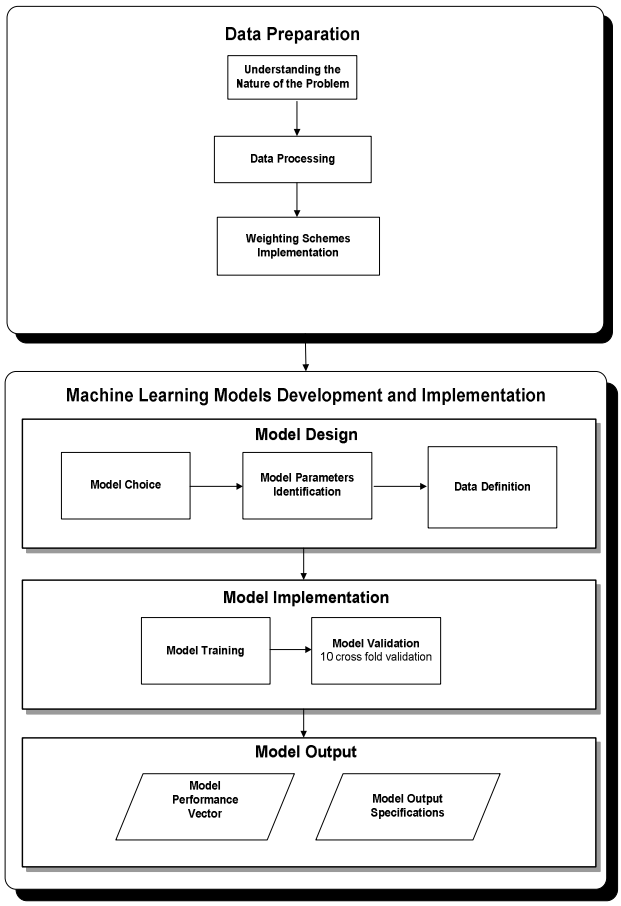


Figure 5. SVM Research Methodology

LSA Feature Space Development:

Feature space in LSA was defined by the number of features that are used to represent a document as a vector. Under the current research task, features were defined as words contained within a document. The literature related to LSA truncated feature-space development highlights that for dispersed datasets, large feature-space sizes of 100 to 500 were appropriate [15]. On the other hand, a smaller feature space of 7 to 10 would be appropriate for a set of documents con-

cerning closely related topics [19]. In earlier work by the authors, it was found that a feature space of size 10 was the best for representing a corpus of Differing Site Conditions (DCS) cases [20], [21]. The present research developed four truncated feature spaces utilizing 5, 10, 15, and 20 features. To that end, the four feature spaces were developed and tested for each subgroup of documents, yielding 16 models.

SVM Feature Space Development:

The computational capability of SVM allows for the development of feature spaces without implementing truncation. Consequently, feature spaces were developed for each of the groups tested. Although each document implicitly included the required knowledge in the form of words and phrases to perform the classification analysis, it also included textual representations that were not related to the topic. Including these words in the analysis hinders the performance of the SVM classifier. As a consequence, an initial preparation step was needed. This step involved preparing the collected dataset in an appropriate manner to enhance the analysis. The processing step included data cleaning, data integration, and data reduction [6]. For more illustrations, textual representation of documents might include frequent words that carry no meaning, misspelled words, outliers, noise, and inconsistent data. While data processing was performed on each textual case representation separately, data integration was performed over the entire dataset. In this step, the entire processed dataset was stored in a coherent manner that would facilitate their use for further analysis. While the integrated data might be very large, data reduction can decrease the data size by aggregating and eliminating redundant features.

To perform the aforementioned sub-steps, an algorithm was developed and implemented in C++. The basic principle of the developed program was to represent each document as a vector of certain weighted word frequencies. The parsing and extraction steps implemented by the algorithm are as follows: 1) Extract all words in a document; 2) Eliminate non-content-bearing words, also known as stopwords [10]; 3) Reduce each word to its "root" or "stem", eliminating plurals, tenses, prefixes, and suffixes; 4) For each document, count the number of occurrences of each word; and 5) Eliminate low-frequency words [6], [14]. Low-frequency words are those that were repeated less than three times in a document. The output of the implementation of this algorithm was w unique words remaining in d unique documents; a unique identifier was assigned between 1 and w to each remaining word, and a unique identifier between 1 and d to each document, resulting in a term-frequency (tf) matrix.

However, mere representation of significant words in the form of (tf) was not sufficient to accurately extract the required knowledge from the document corpus. For example, a word like "construction" might exist in all processed documents in high (tf). However, a decision must be made about whether this word would help assign the topic to a specific subject or not. Consequently, an appropriate weighting mechanism had to be implemented in order to create a representative matrix of these documents within the entire dataset. Literature in the field of ML and text mining illustrated the effectiveness of alternate term weighting schemes like logarithmic term frequency (ltf), augmented weighted term frequency (atf), and term frequency inverse document frequency (tf.idf) (equation 4) [3], [22]. Earlier research by the authors illustrated the superiority of the tf.idf weighing scheme over the others [20], [21], [23], [24]. As a result, tf.idf weighing was adopted for the current research. This C++ algorithm implemented the required calculations, as per equation 4, to formulate the final matrix of each set of documents.

$$tf.idf_{i,d} = \left(1 + log(tf.idf)\right) \times log\left(\frac{N}{df_i}\right) \; if \; tf_{i,d} > 0 \quad (4)$$

LSA Model Design and Implementation:

The following is a description of the steps of the LSA algorithm implemented for development of the automated document classifier. The algorithm starts with an argument, or filename, which is the name of the file or directory to be parsed. The algorithm moves sequentially through each document, extracting relevant features or words, and excluding irrelevant ones which are included in a predefined list of words and characters. It converts all letters to lower case. It is worth mentioning that features that are not included in the common word list are considered to be relevant only if they are comprised of more than two characters. In addition, features of more than 20 characters are truncated to a maximum size of 20 characters.

After extracting relevant features and associating each one with the document it was extracted from, the algorithm begins calculating term weights. The global weights of the terms are computed over the collection of documents. By default, only a local weight is assigned and this is simply the frequency with which the term appears in a document. The algorithm implements two thresholds for term frequencies: Global and Local [25]. The implementation parameters of the algorithm are defined so that the global and local thresholds are both 2. A term must appear more than two times in the entire collection and in more than two documents in the collection before it will be weighted. Next, the local weights of the features are computed. Each word weight is the product of a local weight times a global weight. Next, the algo-

rithm creates a term-by-document matrix using the Harwell-Boeing sparse matrix format. The algorithm finally performs SVD decomposition.

Sixteen truncated feature spaces were generated. Each truncated feature space was generated with a local threshold of Log function and a global threshold of Entropy function. The Log function (equation 5) decreases the effect of large differences in term frequencies [13]. The entropy function (equation 6), on the other hand, assigns lower weights to words repeated frequently over the entire document collection, as well as taking into consideration the distribution of each word frequency over the documents [13]. These thresholds were adopted for the current analysis due to their success over other types of threshold combinations and in earlier research performed by the authors [20]. Dumais illustrated that the log-entropy threshold combination attained 40% higher retrieval precision over other threshold combinations [17].

$$ltf_{i,d} = 1 + \log(tf_{i,d});\ tf_{i,d} > 0 \quad (5)$$

$$\sum_i \frac{P_{ij}\log_2(P_{ij})}{\log_2 n} \text{ where } P_{ij} = \frac{tf_{ij}}{gf_i} \quad (6)$$

where tf_{ij} is the word frequency of word i in document j, and gfi is the total number of times that the word i appears in the entire collection of n documents. General Text Parser (GTP) windows version, developed by Stefen Howard, Haibin Tang, Dian Martin, Justin Giles, Kevin Heinrich, Barry Britt, and Michael W. Berry, was utilized for the implementation of LSA feature spaces development. GTP is a general-purpose text parser with a matrix decomposition option which can be used for generating vector-space information retrieval models.

SVM Model Design and Implementation:

The proposed research methodology developed and compared the outputs of 16 SVM models as follows: 1) four 1st-degree polynomial kernel SVM models; 2) four 2nd-degree polynomial kernel SVM models; 3) four 3rd-degree polynomial kernel SVM models; and 4) four Radial Base Function (RBF) SVM models. Validation of the best developed model was based on prediction accuracy. Since the analysis was aimed at automatically classifying each document as a specific topic, each model was developed as a multiple classifier. In other words, each document was tagged with a known topic. In the training stage, the SVM classifier learns the latent relation between the existing word matrix and the tagged topic. The learning process is then performed on a 10-fold cross-validation mechanism. For more elaboration, the set of training data is divided into 10% and 90% portions in each fold. The model is trained on the 90% and tested on the other 10% cases. The process is done in an iterative manner until the model is trained and tested over the whole set of cases. The prediction accuracy of the model is developed as the average accuracy attained among all folds and the Kappa as the measure of agreement between all folds.

LSA Model Testing and Validation:

The developed LSA models were tested and validated based on correctly predicting the subject matter of newly introduced documents from each subgroup. A C++ algorithm was developed to perform the validation. The implementation of the algorithm performed four steps. First, each document in the feature space was tagged with a subject matter. The algorithm iterates sequentially through the documents storing the document number and its corresponding subject matter. Second, the LSA algorithm was implemented to extract the closest set of documents to the newly tested one. A prediction accuracy threshold of 97% was considered. In other words, any document retrieved at a similarity measure of less than 0.95 was disregarded. The algorithm was set to retrieve a document number and similarity measure (prediction accuracy %). Third, the algorithm read through the document number attained from the LSA implementation and retrieved the subject matter of each document. Fourth, it reported the subject matter of the newly tested documents by two means. The first was reported as the most repeated subject matter. The second was reported as subject matter and weights, which are calculated based on the frequency of repetition of each subject among the retrieved documents. The reported outputs were compared against manual tagging of the newly tested document to decide on the most accurate method.

SVM Model Testing and Validation:

Each of the developed SVM models was tested with a newly un-encountered set of documents from each subgroup. The testing and validation was performed in three steps. First, the new documents were converted to a tf.idf matrix as if they had been part of the original training corpus. This step was essential to allow for an accurate representation of the documents in the developed feature spaces. A C++ algorithm was developed to perform this step. Second, the trained models were run utilizing the newly developed matrixes. Third, automated assignment of topics or prediction was reported by the models. Similar to LSA testing and validation, the output of the models was compared to manual tagging of the newly introduced set of documents for each group.

Results and Discussion

The outcomes of the implementation of the aforementioned methodology are illustrated in Tables 1 and 2, and Figure 6. The discussion of these results in the following sections of the paper relates first to defining the complexity of the problem in hand based on understanding human performance in similar situations; second, to comparing the prediction accuracy of the LSA models to that of humans and derives their strengths and weaknesses; and third, to comparing the results of the developed SVM models in a similar manner to LSA.

Golden Standard:

The first step under this subtask was to establish a Golden Standard of human agreement to which the performance of this new model was to be compared. To that end, a set of eight volunteers, comprised of assistant professors, graduate students, and undergraduate students in construction engineering and management programs, was utilized to set the base level of human agreement. It was assumed by virtue of the occupations of the participating volunteers that they possed enough knowledge about construction practices and documents to be valid selectors. Each volunteer was provided a set of documents from each subgroup and asked to classify them according to similarities under related topics of his\her determination. A document was considered to be classified correctly under a specific topic if three or more persons agreed on the document's topic [8]. The average agreements between participating members in regards to each set of documents are illustrated in the third column of Table 1. It is evident from the table that the lowest human agreements were attained in relation to meeting minutes and claims. This was attributed to that fact that these documents are usually comprised of a set of aspects that could not be defined under a specific title. A construction claim, for example, might include different causes of disputes that might not be related in nature.

LSA:

The best results among all developed models are reported in Table 1. The fourth column defines the best achieved prediction accuracy of the model. The accuracy was defined as the percentage of correct predictions attained. The fifth column defines the truncated feature-space size corresponding to the best accomplished prediction accuracy.

Table 1. Golden Baseline of Human Agreement and LSA Results

Document Type		Average Agreement Between Humans	Prediction Accuracy of LSA Models	LSA Truncated Feature Space Size
Group 1	Correspondence	97%	91%	20
	Meeting Minutes	89%	80%	20
Group 2	Claims	91%	85%	10
	DSC Cases	94%	87%	10

As can be noted from Table 1:

1. The highest precision accuracy of the developed truncated feature spaces was 91% attained with respect to the classification of correspondences.
2. The lowest precision accuracy of the developed truncated feature spaces was 87% attained with respect to the classification of meeting minutes.
3. The prediction accuracy of the developed feature spaces was 6% to 9% lower than the human-agreement levels (refer to Figure 6).

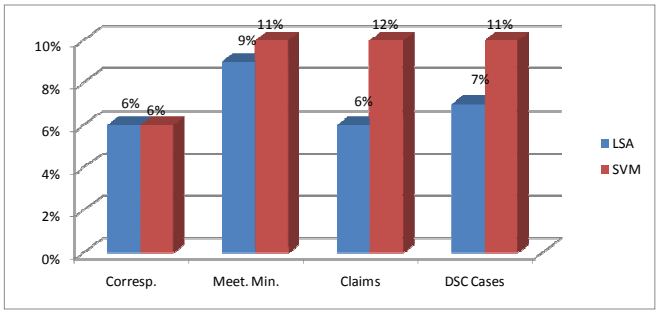


Figure 6. Variation Between LSA and SVM Prediction Models and Human Agreements

It could be deduced from the above results that the developed model was consistent with the human prediction. Both attained the highest prediction and agreements in regards to correspondences and the lowest in regards to meeting minutes. The results were attributed to the fact that the first group of analyzed documents was comprised of highly variable terms. Correspondences and meeting minutes do not follow well-standardized sets of guidelines like legal terms. On the other hand, they were comprised of natural-language representations of human thoughts, meanings and intentions that were represented in the form of words. In addition, meeting minutes usually address a variety of topics and issues, which increases the complexity of the analysis. Consequently, a larger truncated feature space is required to cap-

ture the variability in the linguistic representations. Additionally, the second group of analyzed documents was made up of words that follow structured and standardized formats, like legal and formulated engineering terms,which decreased the complexity of the analysis and yielded better prediction accuracy at a lower truncated feature space.

SVM:

Table 2 illustrates the results of the SVM models. A closer examination of the results shows that the performance of the SVM models was consistent with average human agreement and LSA models. The highest predictions were achieved in relation to correspondences and the lowest in relation to meeting minutes. Such results could be attributed to the complexity of the documents analyzed as mentioned earlier. It can also be seen that the performance of the developed SVM models was less than the LSA models. This outcome was attributed to the computational capacity of SVM classifiers. Support Vector Machine (SVM) is a state-of-the-art classification and regression algorithm, which implements strong regularization techniques; that is, the optimization procedure maximizes predictive accuracy while automatically avoiding over-fitting of the training data [26]. Furthermore, the transformation of the data into a higher dimension space through Kernel estimation provides the strength of the SVM model in solving this complex problem. On the other hand, the analysis utilizes sets of documents ranging from 25 to 300 documents and considered a large number of features of more than 2000 terms. The fact that the number of cases is less than twice the number of features deteriorates the active learning feature of SVM. "Active learning forces the SVM algorithm to restrict learning to the most informative training examples and not to attempt to use the entire body of data" [27].

Table 2. SVM Prediction Accuracy

Document Type		1st Degree Poly. Kernel SVM	2nd Degree Poly. Kernel SVM	3rd Degree Poly. Kernel SVM	Radial Base Function (RBF)
Group 1	Correspondences	84%	89%	91%	87%
	Meeting Minutes	71%	72%	78%	72%
Group 2	Claims	71%	74%	79%	72%
	DSC Cases	80%	83%	83%	79%

Conclusion

In this study, the authors proposed a methodology for automating document classification for the construction industry through machine learning. To that end, the adopted research methodology tested the suitability of Support Vector Machines (SVM) and Latent Semantic Analysis (LSA) for the task. Sixteen SVM and 16 LSA models were developed out of which the models with the best prediction accuracy were adopted. The set of documents utilized for model development, testing, and validation included 300 correspondences, 150 meeting minutes, 25 claims, and 300 DSC cases. The outcomes of this research highlight the following:

- The task at hand was a complex research task. Human agreements about document classification to specific topics ranged from 89% to 97%.
- The results of the SVM and LSA models were consistent with human agreements.
- LSA prediction accuracy ranged from 87% to 91%, whereas a range of 83% to 91% was achieved using SVM.
- The LSA analysis showed that a truncated feature space of size 10 to 20 features was suitable for the current research task, since the documents were closely related.
- Due to the complexity of the task, a 3rd–polynomial degree kernel SVM model was the most suitable.
- The outcomes of the SVM models were lower than LSA models due to their active learning feature.

The outcomes discussed in this paper illustrate the potential of these ML techniques to be adopted for automated document classification. It was conjectured that this research line will help in relieving the negative consequences associated with the lengthy analysis and classification of documents in the construction industry.

References

[1] US Census Bureau, < http://www.census.gov/const/www/c30index.html> (Accessed 2010).

[2] Labidi, S. (1997). "Managing multi-expertise design of effective cooperative knowledge-based system." Proc., 1997 IEEE Knowledge & Data Engineering Exchange Workshop, IEEE, Piscataway, NJ, 10-18.

[3] Caldas, C. H., Soibelman, L., and Han, J. (2002). "Automated classification of construction project documents." Journal of Computing in Civil Engineering, 16(4), 234-243.

[4] Caldas, C. H., and Soibelman, L. (2003). "Automating hierarchical document classification for construction management information systems." Automation in Construction, 12(4), 395-406.

[5] Ioannou, P. G., and Liu, L. Y. (1993). "Advanced construction technology system—ACTS." Journal of Construction Engineering and Management, 119(2), 288-306.

[6] Ng, H. S., Toukourou, A., and Soibelman, L. (2006). "Knowledge discovery in a facility condition assessment database using text clustering." Journal of Computing in Civil Engineering, 12(1), 50-59.

[7] Yang, M. C., Wood, W. H., and Cutkosky, M. R. (1998). "Data mining for thesaurus generation in informal design information retrieval" Proceedings of the International Computing Congress, ASCE, Reston, Va., 189-200.

[8] Al Qady, M. and Kandil, A. (2009). "Techniques for evaluating automated knowledge acquisition from contract documents." In Proceedings of the Construction Research Congress, ASCE, Reston, Va., 1479-1488.

[9] Kosovac, B., Froese, T., and Vanier, D. (2000). "Integrating heterogeneous data representations in model-based AEC/FM systems." Proceedings CIT 2000, Reykjavik, Iceland, 1, 556-566.

[10] Scherer, R. J., and Reul, S. (2000). "Retrieval of project knowledge from heterogeneous AEC documents." Proceedings of the Eight International Conference on Computer in Civil and Building Engineering, Palo Alto, Calif., 812-819.

[11] Caldas, C. H., Soibelman, L., and Gasser, L. (2005). "Methodology for the integration of project documents in model-based information systems." Journal of Computing in Civil Engineering, 19(1), 25-33.

[12] Mangasarian, L. and Musicant, D. (1999). "Massive support vector machine regression." NIPS*99 Workshop on Learning with Support Vectors: Theory and Applications December 3, 1999.

[13] Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W. (2007). "Handbook of latent semantic analysis." Lawrence Erlbaum Associates, London.

[14] Salton, G., and Buckley, C. (1991). "Automatic text structuring and retrieval – experiment in automatic encyclopedia searching." Proceeding of the 14th Annual International ACM SIGIR Conference on Research and Development in information Retrieval, 21-30.

[15] Choi, F. Y. Y., Wiemer-Hastings, P., and Moore, J. (2001). "Latent semantic analysis for text segmentation." In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing, Seattle, WA, 109–117.

[16] Hofmann, T. (1999). "Probabilistic latent semantic indexing." Proceedings of the National Academy of Science, 101, 5228-5235.

[17] Dumais, S. (1990). "Improving the information retrieval from external sources." Behavior Research Methods and Computers, 23, 229-236.

[18] Dumais, S. (1991). "Improving the retrieval of information from external sources." Behavior Research Methods, Instruments, and Computers, 23(2), 229-236.

[19] Koll, M. (1979). "An approach to concept-based information retrieval." ACM SIGIR Forum, XIII, 32-50.

[20] Mahfouz, T. (2009). "Construction legal support for differing site conditions (DSC) through statistical modeling and machine learning (ML)"Ph. D. thesis, Department of Civil, Construction, and Environmental Engineering, Iowa State Univ., Ames, IA.

[21] Mahfouz, T., and Kandil, A. (2010). "Unstructured construction document classification model through latent semantic analysis (LSA)." Proceeding of the 27th International Conference on Applications of IT in the AEC Industry (CIB-W78 2010), Cairo, Egypt.

[22] Manning, C. and Scheutze, H. (1999). "Foundations of statistical natural language processing." Cambridge: MIT Press.

[23] Mahfouz, T., and Kandil, A. (2010). "Construction legal decision support using support vector machine (SVM)." Proceeding of the Construction Research Congress 2010: Innovation for Reshaping Construction), Banff, Canada.

[24] Mahfouz, T., and Kandil, A. (2010). "Automated outcome prediction model for differing site conditions through support vector machines." Proceeding of the International Conference on Computing in Civil and Building engineering, Nottingham, United Kingdom.

[25] GTP. < http://www.cs.utk.edu/~lsi/soft.html> (Accessed 2008).

[26] Cannon, E. O., Amini, A., Bender, A., Sternberg, M. J. E., Muggleton, S. H., Glen, R. C., and Mitchel, J. B. O. (2007). "Support vector inductive logic programming outperforms the Naïve Bayes classifier and inductive logic programming for the classification of bioactive chemical compounds." J. of Comput Aided Mol., 21, 269-280.

[27] Oracle. < http://www.oracle.com/index.html> (Accessed 2009).

Biographies

TAREK MAHFOUZ is Assistant Professor of Construction Management at the Department of Technology, Ball

State University. He earned his B.Sc. (Construction Engineering, 2000) from the American University in Cairo (AUC), MS (Construction Engineering, 2005) from the American University in Cairo (AUC), and a Doctoral degree (Civil Engineering with specialization in Construction Engineering and Management, 2009) from Iowa State University. Dr. Mahfouz's main research interests are in the use of information technology, machine learning, and statistical modeling for knowledge management and decision support in the construction industry. He is a ASCE affiliated member, Construction Research Congress (CRC) committee member, ASCE Data Sensing and analysis (DSA) committee member, and ASCE Technical Council for Computing and Information Technology (TCCIT) committee member.

JAMES JONES is Assistant Professor of Construction Management at the Department of Technology, Ball State University. He earned his B.Sc. (Civil Engineering, 1992) from Purdue University, M. Eng. (Engineering Management, 2000) from the University of Idaho, M.A. (Adult and Community Education, 2004) from Ball State University, and Ed.D. (Adult, Community, and Higher Education Cognate: Executive Development for Public Service, 2008) from Ball State University.

AMR KANDIL is an Assistant Professor of Construction Engineering in the School of Civil Engineering at Purdue University. Dr. Kandil holds a B.Sc. degree in construction engineering from the American University in Cairo (AUC), MS in construction engineering from the American University in Cairo (AUC), and a Doctoral degree from the University of Illinois at Urbana-Champaign. Dr. Kandil previously was with the Department of Civil, Construction and Environmental Engineering at Iowa State University as an assistant professor of construction engineering. His main research interests are in the use of information technology applications for decision support in large-scale infrastructure projects and in sustainable infrastructure construction.