Management of Constructability Knowledge for Design Integration using Textual Latent Semantic Analysis

Vincent Kuo, vincent.kuo@aalto.fi
Aalto University, Finland

Abstract
Constructability problems are common on the construction site, due to lack of construction experience in the design team and absence of tools to address constructability. This study investigates how tacit semantic constructability knowledge/experience can be mined from typical project documentation and made dynamically and meaningfully accessible for gauging future designs. The method employs singular value decomposition (SVD), dimensionality reduction and similarity measurements, used in textual latent semantic analysis (LSA). It has been found that with limited training data, in the form of unstructured textual descriptions of ad hoc constructability problems, intuitive semantic correlations can be inferred between design features and associated constructability issues. The method poses an adaptive, predictive approach to addressing constructability problems and emulates the human reasoning process of recognizing potential problem cases from past experience upon inquiry or design review. The system is yet a conceptual prototype; the development for practical implementation is an ongoing endeavor.

Keywords: Constructability, Latent Semantic Analysis, Knowledge Management

1 Introduction

It is well known that closer collaboration between practitioners from different phases of the product lifecycle can lead to better designs, higher production efficiency, and improved customer or user experience. Such human collaboration allows complex knowledge from subsequent phases of the lifecycle to be integrated early in the design phase, towards further optimization of the design beyond mechanical parameters. This principle is directly applicable in the architecture, engineering and construction (AEC) industry, which suffers from severe knowledge fragmentation leading to designs that are rife with buildability/constructability problems when executed on site. Concerns on the disparate phases in building development was raised as early as the 1960s when a series of studies, such as Emmerson & Emmerson (1962) and Banwell (1964), were carried out in the UK. The Construction Industry Research and Information Association (CIRIA 1983) in the UK introduced the concept of “buildability”, defined as “the extent to which the design of a building facilitates ease of construction, subject to the overall requirements for the completed building”. Later the Construction Industry Institute (CII 1986) in the US developed the notion of “constructability”, defined as “the optimum integration of construction knowledge and experience in planning, engineering, procurement and field operations to achieve overall project objectives”.

The challenge of disseminating constructability knowledge lies in its abstract nature. Because of this, it requires expert tacit understandings before improvements can be realized (Wong et al. 2007). Consequently, this limits the capacity of explicated or codified constructability guidelines as they are often unable to cover the myriad unique factors in each design and/or construction case. On the other hand, the prediction of specific constructability problems upon design review is readily achievable through high-level human cognition and tacit knowledge embodied in so-called engineering experience/intuition (Kuo & Wium 2014).

This study alludes to knowledge management, however, from machine learning perspective. The proposed method investigates how commonly available organisational

textual data can be mined and processed so that accumulated past constructability knowledge, latent in textual format, can be meaningfully accessed and systematically integrated into future designs through dynamic queries by the designer. Since Latent Semantic Analysis (LSA), and in particular Singular Value Decomposition (SVD), has been shown to infer patterns and semantic correlations (Landauer 2002) in unstructured data, they pose as pertinent approaches for handling fuzziness and uncertainty inherent in complex constructability problems. From the correlations inferred between design feature descriptors and associated constructability problems, predictions can be made by querying future cases. The aim of this paper is to demonstrate the potential of the proposed linear algebra method in augmenting organisational constructability knowledge. This adaptive method may well be used by AEC firms to document problem cases/experiences comprehensively, such that they can be relevantly retrieved upon query and implemented by future designers. Moreover, it should be comparable to the way whereby a human recalls past experiences, as an act of recognizing potential problems upon inquiry or during a review process.

Section 2 is a brief introduction to LSA, SVD and related applications. In Section 3 the methodology of the analysis is described covering the textual parsing steps and mathematical background behind LSA and SVD. In Section 4 a working example is presented in the context of constructability to illustrate the method of analysis. Section 5 constitutes discussions and interpretations of the results. Section 6 concludes the paper and elaborates on the potential future research, as well as other avenues of application for the system.

2 LSA and SVD applications

Latent semantic analysis (LSA) was introduced to aid in information retrieval and search optimization. The main idea behind LSA is to collect all of the contexts within which words appear, and to establish common factors that represent underlying concepts. Extensive research in psychology suggests that LSA simulates the way the human brain distils meaning from text. (Deerwester et al. 1990; Dumais 2004) LSA was shown to mathematically model synonyms (Landauer 2002), metaphors (Kintsch & Bowles 2002), and explain various psychological phenomena (Wolfe & Goldman 2003). It uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, morphologies, or the like, and it takes as its input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs (Landauer et al. 1998). Word and passage meaning representations derived by LSA have been found capable of simulating a variety of human cognitive phenomena, ranging from developmental acquisition of recognizing vocabulary to word-categorization, sentence-word semantic priming, discourse comprehension, and judgments of essay quality. It is important to note from the start that the similarity estimates derived by LSA are not simply frequencies, co-occurrence counts, or correlations in usage, but depend on a mathematical analysis that is capable of correctly inferring deeper relations (thus the phrase “latent semantic”), and as a consequence, they are often much better predictors of human meaning-based judgments and performance (Landauer et al. 1998).

At the heart of LSA is the linear algebra operation Singular Value Decomposition (SVD), which has been used in many different applications. For instance, in digital image processing (Kalman 1996; Strang 2003) SVD is used as a mathematical tool to identify pattern redundancy in image data for compression of images, and image recognition. The same is applied in digital signal processing for noise reduction and compression (De Lathauwer et al. 2000). SVD is closely related to principle component analysis (PCA) and factor analysis used commonly in bioinformatics and micro-array analysis of gene data (Wall et al. 2003). These are non-exhaustive examples of SVD applications and show its capabilities in pattern recognition, inferring latent semantic correlations, typically thought possible through human reasoning, amidst unstructured explicit data. LSA and SVD thus pose pertinent approaches to deal with the management of tacit semantic knowledge in constructability context, traditionally regarded as an exclusively human ability, and non-explicable computationally.

From an information retrieval perspective, the analogy to the human semantic memory process can be made with LSA. One may regard a construction expert to have the relevant

experience enabling recognition and predictions of constructability problems. This is done upon inquiry inputs, for instance, by means of drawings (e.g. during design review), textual descriptions (e.g. in an email or report), audio signals (e.g. telephone), or through human discourse (e.g. in a meeting or discussion). The human is able to interpret a multitude of different knowledge formats, upon which he/she can cognitively create the semantic links between these query elements and some entities of constructability problems experienced in the past. The human thus retrieves the most relevant not through matching data, but matching "meaning". LSA exhibits this quality, allowing different knowledge formats as inputs – be it text, signals, pixels, genetic codes, bio-molecular descriptors etc. – as long they can be represented in a relationship- or occurrence-matrix between the row and column spaces.

3 Methodology and mathematical background

The input textual data is gathered from a mixed group of contractors with extensive construction experience spanning a wide variety of projects in different sectors of civil engineering. This constitutes free-form natural language descriptions of constructability problems experienced. The textual data is then parsed and processed using latent semantic analysis techniques as described briefly in the following subsections.

3.1 Text parsing and constructing a term-by-document matrix

First, the corpus of textual data is converted into a term-by-document matrix, where the rows represent words, and columns represent documents, in which words appear. The contents of the matrix are the occurrences of each respective word *i* in the respective document *j*. Thus the matrix can be regarded as an occurrence matrix of all the terms in all the documents of the corpus, as depicted in Figure 1.

Document *j*

Term *i*	Document 1	Document 2	Document 3	Document 4	Document 5
Term 1	0	0	1	3	0
Term 2	3	6	0	0	4
Term 3	5	0	0	1	0
Term 4	1	0	3	1	0
Term 5	3	1	0	5	1
Term 6	0	2	0	0	0
Term 7	1	0	2	0	3
Term 8	2	0	2	1	2

Figure 1 Example of a term-document matrix, populated by occurrences of term *i* in document *j*.

There are a number of features of text in human language that can make them difficult to process automatically. Some pre-processing or filtering needs to be done to allow the automatic compilation of an effective term-by-document matrix. The considerations taken to process the constructability input text are briefly described below:

Uppercase and lowercase. Questions should be raised about how to treat capitalization. In particular, if we have two words that are identical except that one has certain letters in uppercase. For many cases, one would like to treat for instance "THE" and "The" as the same. However in some circumstances, there may be proper nouns which would need to be distinguished from the totally lowercase counter-part, e.g. "Brown" as a name and "brown" as a colour. Different heuristic methods can be applied e.g. by converting the whole corpus into lowercase, or lowering cases that only appear in the beginning of sentences. For this analysis, the input text is simply converted to lowercase.

Tokenization. An early step of processing is to divide the input text into units called tokens, which are typically words, but can also be a number or punctuation mark. These tokens ultimately form the terms in the term-document matrix. There are many linguistic dilemmas as to what constitutes a word or token, especially when alphanumerical characters are involved or graphical words (e.g. M1cr0$0ft), nevertheless the predominant method used in English is merely occurrence of whitespace – a space or tab – which is carried out for this study.

Morphological stemming. Similar to the discussion regarding capitalization the question here is whether to distinguish between word forms such as "sit", "sits", and "sat". Stemming is referred to in literature as the removal of the affixes and to preserve the root of the word.

Stop-words removal. These are considered words of high occurrence that do not contribute semantically to the meaning of documents or passages. In other words they are function words that can be ignored in information retrieval without significant effect on retrieval accuracy. For instance prepositions, conjunctions, certain pronouns. A stop list includes words like: also, an, and, as, can, could, for, from, he, her, those, these, this, until, while etc.

There are many more additional pre-processing considerations in statistical natural language processing that can be taken – e.g. getting rid of typographical errors, differences in stop-lists, amount of tuning, tagging of proper nouns or parts of speech etc. Nevertheless, the results do not appear to show substantial difference (Landauer et al. 1998). There has been many focus studies on such pre-processing methods, which will not be covered in this paper. Using the few steps mentioned above, the term-document matrix is automatically populated with the raw occurrences of each term (row) and document (column) for the input text corpus.

3.2 TF-IDF

The occurrence matrix then needs to undergo a weighting transformation (Han & Kamber 2006), commonly referred to as TF-IDF (term frequency, inverse document frequency). The raw occurrence values in Figure 1 in the original matrix are replaced by the product $\boldsymbol{w_{ij} = tf_{ij} * idf_i}$, where $\boldsymbol{idf_i = log_2(N/n_i) + 1}$. $\boldsymbol{tf_{ij}}$ is the normalized term occurrence in each document, i.e. term frequency of term *i* in document *j*, $\boldsymbol{N}$ is the total number of documents in the whole corpus, $\boldsymbol{n_i}$ is the term occurrence of term *i* in the entire collection of documents, and $\boldsymbol{idf_i}$ is the weighting factor known as the inverse document frequency. This weighting factor serves as a way to discount the weight of high occurrence non-stop-words and promote semantic weights of the rarer terms.

3.3 Singular Value Decomposition and Dimensionality Reduction

Singular value decomposition (SVD) is carried out on the term-document matrix after the TF-IFD weighting. In essence SVD takes any general rectangular matrix **A** with *m* rows and *n* columns and decomposes it into a product of three matrices, so that $\mathbf{A = USV}^T$, where $\boldsymbol{U}$ (*m*x*m*) and $\mathbf{V}^T$ (*n*x*n*) are the left and right orthonormal matrices and $\boldsymbol{S}$ (*m*x*n*) is a rectangular matrix with non-negative singular values along the diagonal in order of decreasing magnitude. The dimension of the **S** matrix is the rank of **A**. The columns of $\boldsymbol{U}$ are eigenvectors of $\mathbf{AA}^T$, and the columns of $\boldsymbol{V}$ are eigenvectors of $\mathbf{A}^T\mathbf{A}$. The *r* singular values along the diagonal of **S** are the square roots of the non-zero eigenvalues of both $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$. In mathematical terms (Strang 2003), the row space of $\boldsymbol{A}$ is *r*-dimensional and inside $\mathbf{R}^n$, and the column space of **A** is *r*-dimensional and inside $\mathbf{R}^m$. Special orthonormal bases $\mathbf{V} = (v_1, v_2, \ldots v_r)$ are chosen for the row space, and $\mathbf{U} = (u_1, u_2, \ldots u_r)$ for the column space, such that $\mathbf{A}v_i$ is in the same direction as u_i, and s_i acts as the scaling factor such that $\mathbf{A}v_i = s_i u_i$. In matrix form, this is equivalent to $\mathbf{AV}=\mathbf{US}$, thus $\mathbf{A}=\mathbf{USV}^T$. **A** is thus the linear transformation that carries orthonormal basis v_i from space $\mathbf{R}^n$ to orthonormal basis u_i in space $\mathbf{R}^m$. In other words, the relationship between the row space and column space is reconciled, and a general rectangular matrix **A** is diagonalized into 2 independent spaces described by orthonormal bases in **U** and **V**, and related to each other by a factor of the decreasing singular values in **S**. Thus, considering matrix **A** as a term-document matrix, the matrices **US** and $\mathbf{SV}^T$ describe the scaled association patterns (of the term and document spaces respectively) governing the contents of the matrix **A**.

Essentially, any rectangular matrix that captures a relationship between two concepts or entities in a domain field can be subjected to SVD. The result is that the original matrix can be re-represented as a set of independent concept semantic vectors (columns of **U** and **V**), which can be linearly combined using **S** to produce the original data, or approximations of it at a lower dimensionality.

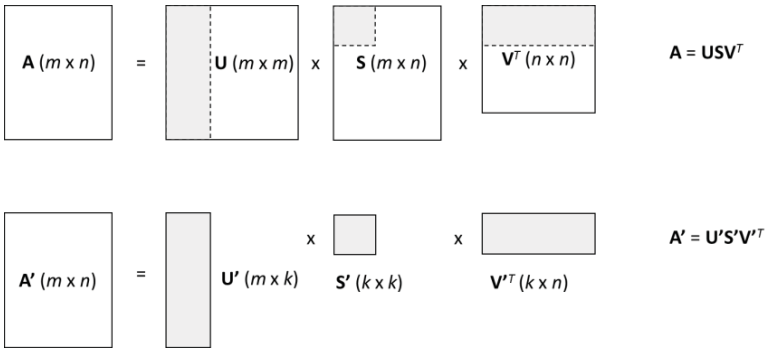

Figure 2 Graphical depiction of Singular Value Decomposition of matrix **A** into matrices **U**, **S** and **V**. The grey areas depict dimensionality reduction, by choosing a *k* number of singular values to keep, the truncated matrices U', S' and $\mathbf{V'}^T$ yields an approximate matrix **A'** with less noise.

Consequently, dimensionality reduction is performed by keeping the first *k* singular values of **S** and produce a *k*-reduced approximation of **A** as **A'**(*m*x*n*) = **U'**(*m*x*k*)*$\mathbf{S'}(k\text{x}k)*\mathbf{V'}^T(k\text{x}n)$. Figure 2 depicts dimensionality reduction to obtain a truncated matrix **A'**. The reduction of the dimensionality (choosing a *k* smaller than *n*) essentially implies the removal of least important patterns, which can be considered as noise, and preserving the most significant patterns, thereby reconstructing an approximate truncated matrix **A'** (Figure 2). The truncated matrix is based on the most important latent underlying structures within the original matrix **A** in the association of terms and documents, at the same time removing the noise or variability in word usage that plagues word-based retrieval methods.

The number of *k* dimensions to keep is not self-evident as it depends on how much of the original variance is desired to be kept without loss of relevant data. In linguistic terms, it is difficult to interpret intuitively. However, a very common technique in natural language processing is to plot the singular values of the **S** matrix against the number of dimensions (Sidorova et al. 2008). One may choose the number of dimensions corresponding to where the singular values decrease substantially (elbow of the graph) indicating the point where the patterns become insignificant. Generally, the more noise is removed (dimensions reduced), the clearer are the semantic relationships revealed in the reconstructed approximate matrix.

3.4 Cosine Similarity and Query

From SVD and dimensionality reduction we obtained the *k*-reduced **U'** and **V'** matrices, the row vectors of which represent the term- and document-spaces respectively in the same semantic space (i.e. with the same dimensionality). Essentially, the row vectors of the scaled **U'** and **V'**, thus **U'S'** and $\mathbf{S'V'}^T$ matrices, are semantic descriptors of the terms and documents respectively. This allows for term- and document-vectors in the semantic space to be compared to one another using a similarity function. Usually, similarity in text matching is addressed using cosine similarity – calculating the cosine of the angle between two vectors of the same dimensionality as in equation (1), where *a* and *b* are two vectors of the same dimensionality, and θ the angle between them. The cosine is acquired by dividing the dot product of *a* and *b*, by the product of their magnitudes.

$$cos(\theta) = \frac{a \,.\, b}{\|a\|\|b\|} \qquad (1)$$

Thus, if the two vectors are identical, the cosine similarity would yield 1 since the angle between them would be 0. Likewise, vectors that are totally uncorrelated has a cosine

similarity of 0, as they are orientated 90 degrees with respect to each other. Theoretically, if the cosine similarity is -1, then the two vectors are semantic inverses of each other. Using cosine similarity, the semantic associations between terms and terms, terms and documents, and documents and documents are obtainable explicitly.

It is now convenient to be able to use the similarity function to retrieve semantically related terms and/or documents upon an open query input by the user. A query is done as a collection of terms, forming a vertical pseudo-document vector, which undergoes TF-IDF weighting, then is transformed to the semantic space, so that its similarity can be gauged with the row vectors of **U'S'** and **S'V'**^{T} (Figure 4). The transformation to the semantic space involves multiplying the pseudo-document vector (q in equation (2)) with the k-reduced **U'** matrix to obtain q_k, which is a horizontal row vector of the query terms as a pseudo-document in semantic space.

$$q_k = q^T . U' \quad (2)$$

The semantic query vector q_k is then compared to all the row vectors in **U'S'** and **S'V'**^{T} (term and document semantic vectors respectively) using cosine similarity.

4 Constructability Data Analysis: Working Example

The corpus of text used in this investigation consists of unstructured written accounts of specific constructability issues experienced on site by construction professionals. A total of 63 problem cases are collected in the form of separate textual passages (documents). Table 1 shows a few examples to give an idea of the types of texts acquired. The extracted terms subjected to aforementioned textual pre-processing steps are shown in the adjacent column.

Table 1 A few examples of constructability cases as documents and the tokenized terms

Index	Documents	Terms
Doc5	On the majority of projects the information flow to the contractor is late, resulting in delays to the project. The design companies are normally not at fault, because they are waiting for details from vendors which must first be employed by the client, before they will supply details required by the designer to do the foundation designs. The overall design is not done timeously, allowing as little interference to the construction works as possible.	major, project, inform, flow, contractor, late, result, delai, project, design, company, normal, fault, wait, detail, vendor, employ, client, suppli, require, foundat, overall, timeous, allow, interfere, construct, work
Doc17	Changed the shape and size of pre-cast bridge beams. This was a contractor suggestion.	chang, shape, size, pre, cast, bridg, beam, contractor, suggest
Doc20	Fast track project ring beams designed as concrete. Engineer should have specified boxed steel beams for speed	fast, track, project, ring, beam, design, concret, engine, specifi, box, steel, speed
Doc22	Numerous concrete shear walls	numer, concret, shear, wall
Doc38	Rebar sizes too big or spacing too little for concrete to be vibrated in beams and columns.	rebar, size, big, space, concret, vibrat, beam, column
Doc42	Design drawings not showing enough details i.e. sectional details. Especially for structural drawings	design, draw, show, detail, section, especi, structur

The input documents vary greatly in length, register, and style – as any natural language text. There is no standard labelling so different words may be used to describe similar concepts and vice versa. There could also be spelling mistakes and grammatical and punctuation inconsistencies. The corpus, albeit small, captures a wide range of specific constructability problems, spanning many different types of design and construction elements and attributes, many of which are unrelated. Despite these challenges, the goal is to demonstrate the extent to which meaningful semantic relationships between terms and documents can be inferred from a limited amount of such unstructured textual data – 63 separate textual documents, constituting 14884 characters in total. The natural texts are kept unaltered, thus making the analysis relevant for raw data abundant within company documentation, such as project meeting minutes, emails, archives, reports etc., and thus are readily obtainable to be processed for the LSA analysis.

First, the documents are automatically pre-processed through tokenizing, removal of stop-words, and stemming, as described in Section 3.1. The result is a total 541 terms extracted from the corpus of 63 documents, thus a 541 x 63 term-document matrix is established. The occurrences of term *i* in document *j* are weighted using TF-IDF, before SVD is carried out. The dimensionality reduction is carried out by plotting the singular values of the **S** matrix against its dimensions (total 63). As seen in Figure 3, the elbow suggests where the significance of the patterns drops. The number of *k* singular values to keep is chosen to be 20.

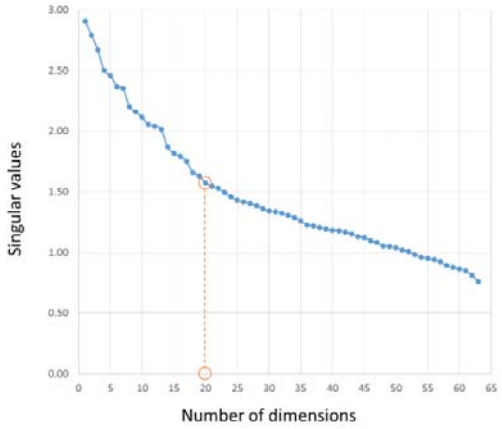

Figure 3 Plot of singular values along the dimensions

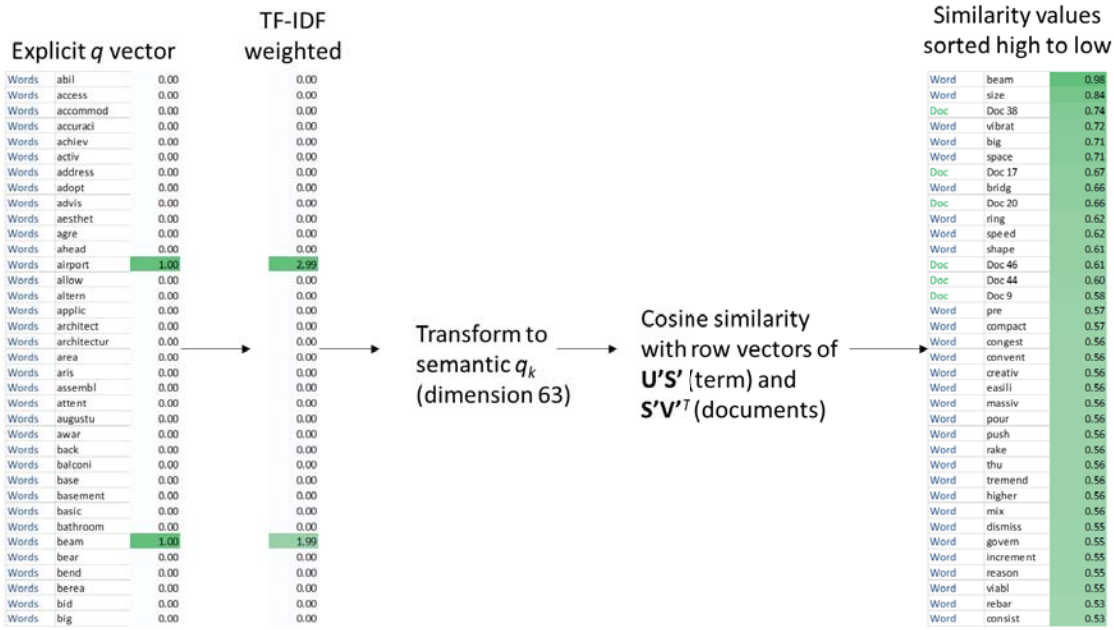

Figure 4 Graphic representation of the query process. Note that only the top parts of the vectors are shown.

After SVD and dimensionality reduction, the scaled reduced matrices, **U'S'** and **S'V'**^{T}, can be computed. Next, a query function is set up by first weighting the explicit query pseudo-document vector *q* with TF-IDF, then mapping the weighted vector *q* to the semantic space to q_k as per equation (2). Lastly the cosine similarities between q_k and each row vector of **U'S'** and **S'V'**^{T} are calculated, to yield the measures of semantic similarity between the query and all the terms and documents. Based on these similarity measures, one can then sort all the terms and documents from highest to lowest similarities with respect to the query, or decide on a threshold below which the semantic association is deemed irrelevant. For this study, the former is used so as to allow the user more flexibility in interpreting the semantic associations

returned by the system. A simple example illustrating the process is shown in Figure 4, using two terms ("airport" and "beam") as input query. Note that only the top parts of the vectors are shown, as the full vectors are too long to display.

5 Results and discussions

Theoretically, the cosine similarity values between the term and document vectors are dependent on the nature of the original term-document matrix **A**, in particular, how related terms occur together within the same document. The fundamental principle of semantic inference in human text is that similar/related terms would have much higher co-occurrence than unrelated terms. For instance, as a simple illustrative example regarding 3 words: "constructability", "concrete", and "banana". It is certainly intuitive that in a hypothetical corpus containing all of human knowledge, "constructability" and "concrete" have a much higher probability of occurring together within a document, than "constructability" and "banana". This characteristic seems to be evident in all human languages and is exploited in textual processing. Of course, natural language is rife with inconsistencies, which are computationally regarded as "noise". This challenge is addressed through the use of singular value decomposition, and subsequently dimensionality reduction, which allows less significant patterns to be removed at will. The result is a set of approximate semantic vectors corresponding to each term and document, which are of the same dimensionality and thus comparable among one another using cosine similarity.

Given the limited length of the paper, it is not possible to present too much data for interpretation and discussion. However, some examples can be shown to illustrate the response of the system. Table 2 tabulates the results of 6 single-term queries. The queries are chosen as rudimentary building elements for easy interpretation and the retrieved terms, in the context of constructability problems, are listed in decreasing order of semantic similarity. The top 10 retrieved terms, as well as the corresponding cosine similarity values, are shown.

Table 2 Different query inputs with the top 10 semantically related terms retrieved and corresponding cosine similarities

"slab"		"beam"		"basement"		"column"		"roof"		"wall"	
situ	0.95	size	0.88	lower	0.90	diamet	0.66	access	1.00	compromis	0.83
servic	0.94	vibrat	0.73	area	0.89	dramat	0.66	riski	0.99	destroi	0.83
accommod	0.92	space	0.72	damp	0.89	effici	0.66	soft	0.99	elect	0.83
ceil	0.91	big	0.72	drain	0.89	form	0.66	timber	0.99	masonri	0.83
void	0.91	bridg	0.67	flode	0.89	kg	0.66	consum	0.99	pipe	0.83
rc	0.91	shape	0.66	lynx	0.89	slip	0.66	time	0.91	stabil	0.83
suspend	0.63	ring	0.64	offic	0.89	stiff	0.66	bigger	0.79	stabl	0.83
cure	0.63	speed	0.64	rise	0.89	increas	0.66	assembl	0.79	upstand	0.83
date	0.63	pre	0.61	soil	0.89	tall	0.65	escal	0.79	shutter	0.81
manner	0.63	suggest	0.58	sub	0.89	slender	0.62	forc	0.79	finish	0.81

As seen, the terms can consist of many different aspects of the constructability problem description, however, still exhibit intuitive semantic associations. For instance, "slab" is related to elements like "ceiling" and "void" (building elements); "in situ", "RC" (reinforced concrete), and "suspended" (qualifiers of "slab"); "cure", "service", "date" (concepts potentially relating to the problem). For "basement", retrieved terms "lower", "office", "soil", "sub" are obviously intuitive; while "damp", "drainage", "flood" etc. are qualifiers of problems related to "basement". One can also see that substantial problems associated with "wall" regard their "stability". Affected elements as "electric" and "pipe" terms suggest issues regarding chasing of "masonry" walls. Same interpretations can be made with other term queries. It is worthy to note that the queries can be any number of terms, since it's essentially a pseudo-document. In the example, only single term queries are shown for simplicity. Therefore, natural language queries can also be used – for instance, inputting a detailed textual description of a design feature for the system to return related terms that are more specific. It is remarkable that such intuitive inferences can be made with only a set of 63 documents as input, as if all that the system "knows" is exclusively embedded within these 63 text passages. The performance of

the system will improve given more input training data and with more specific and standardized methods of documentation, for instance, re-work records, change cases, meeting minutes, risk reports etc. which are common in organizations and succumb to some standard template.

Likewise, documents are also retrieved based on their semantic similarity with respect to query. In the example visualization in Figure 5 only document numbers are shown, since full documents are too long to include in the paper for discussion and interpretation.

"slab"		"beam"		"basement"		"column"		"roof"		"wall"	
Doc 51	0.93	Doc 38	0.75	Doc 53	0.93	Doc 47	0.71	Doc 19	0.99	Doc 12	0.89
Doc 4	0.68	Doc 17	0.71	Doc 48	0.88	Doc 31	0.67	Doc 29	0.73	Doc 40	0.85
Doc 57	0.57	Doc 20	0.66	Doc 62	0.59	Doc 39	0.66	Doc 45	0.66	Doc 30	0.70
Doc 58	0.56	Doc 46	0.59	Doc 13	0.21	Doc 54	0.57	Doc 24	0.41	Doc 13	0.60
Doc 36	0.51	Doc 44	0.57	Doc 29	0.21	Doc 40	0.55	Doc 37	0.38	Doc 22	0.58
Doc 45	0.48	Doc 9	0.52	Doc 57	0.17	Doc 38	0.52	Doc 34	0.35	Doc 35	0.52
Doc 56	0.40	Doc 15	0.41	Doc 34	0.14	Doc 15	0.36	Doc 1	0.34	Doc 15	0.52
Doc 24	0.39	Doc 21	0.37	Doc 2	0.09	Doc 46	0.36	Doc 33	0.34	Doc 57	0.40
Doc 25	0.38	Doc 16	0.35	Doc 28	0.07	Doc 12	0.34	Doc 28	0.34	Doc 39	0.38
Doc 10	0.37	Doc 26	0.31	Doc 46	0.07	Doc 30	0.31	Doc 36	0.32	Doc 23	0.37

Figure 5 Semantically related documents (constructability problem cases) and cosine similarities for each query.

Given any query length, the system returns the most semantically related constructability problem cases. This allows the user to retrieve and further explore the details of documents. This is useful especially when the implication of retrieved terms are not immediately clear (e.g. the aforementioned “wall” terms). Another instance, “beam” returned terms that are somewhat abstract (Table 2), but briefly looking at the top *Docs* returned for “beam” (Figure 5 and Table 1). *Doc38* reveals that the problem relates to rebar “sizes”, which causes “vibration” issues due to too little “spacing”. *Doc17* alludes to changes in “shapes” for “pre-” cast “bridge” beams, while *Doc20* describes “ring” beams, which could have been steel instead of concrete for faster “speed”. Exploring such cases offers the designer an enriched perception of useful experiences, from a repository of past cases, semantically related to the query, some of which may not even explicitly contain the query term, but are still semantically similar.

The system also allows the user to quickly gauge all relevant concepts in the knowledge base, as in the simple visualization in Figure 5. For instance, it is clear that there are fewer constructability problems related to “basement” than for instance “wall”. Furthermore, the specificity of the concepts among the documents are also evident – e.g. problem cases associated with “roof” are few but very specific and relevant (high cosine similarity), whereas problems regarding “column” are wide-spread and less conceptually focused, thus relating also to other terms. It is worthy to note that the “rationality” of the system is inherently bounded by its input training data. Thus, for instance, if very few “beam” problems were provided, then “beam” related issues will be reflected as less probable. This may pose an analogy of how humans accumulate experience, where repeated experiences gain weight in memory.

In general, the system is a knowledge base, where semantic query and retrieval enables past cases to be revealed to the user, who may investigate further by reading each case in detail. This process, albeit non-deterministic, improves the discoverability of past knowledge/experience and aims to emulate recommendations from a human, though in this case, with the augmented stochastic integrity of the entire knowledge base (of any size and scope), not limited to one person’s subjective exposure. It can be applied to many other avenues, for instance, if the input training data consists of documented risk cases and associated mitigating actions, the query of a new risk case can be used to retrieve similar past risk cases, of which associated mitigation solutions may well apply effectively to the new risk case, thus resembling an intelligent case-based expert recommender system.

6 Conclusions

This study investigates how complex semantic constructability knowledge as natural language can be management so that potential future problems can be systematically and stochastically predicted given some known parameters as query – reminiscent of how a human expert may

recall past experience upon inquiry. The proposed method uses data mining, simple textual processing and singular value decomposition to show how relevant knowledge, supposedly latent in textual documentation, can be extracted and used by future designers flexibly, in an adaptive recommender- or knowledge-discovery-system. Naturally a large input corpus covering the knowledge of a required problem domain is ideal for more accurate queries within the scope of the domain. The proposed system is shown to improve the discoverability of specific pertinent knowledge amidst large masses of unstructured constructability data. The method is automated and thus poses a dynamic, predictive system whereby newly generated data can be constantly added to the knowledge base to update the latent semantic structure. This allows the system to infer semantic correlations adaptively, as circumstances may change with time (i.e. change of industry trends in technologies, processes or regulations).

Further research can be done with larger or more specific data (e.g. rework/change/risk reports), and include advanced text processing techniques (e.g. parts-of-speech tagging), to gauge improvements in system capabilities. Furthermore, it can be implemented in problem-specific pilot projects, where the impact of improved knowledge management enabled by the system, can be objectively measured and linked to project performance metrics. In the long term, such studies need to be conferred with empirical, or even qualitative studies from other fields, in order to advance theories of knowledge/competence management, learning, and cognition, both philosophically and analytically, within, as well as beyond, engineering fields.

References

Banwell, H., 1964. *The Placing and Management of Contracts for Building and Civil Engineering Work: Report of the Committee*,

CII (Construction Industry Institute), 1986. *Constructability: A Primer*, Austin, Texas.

CIRIA (Construction Industry Research and Information Association), 1983. *Buildability: An assessment*, London.

Deerwester, S. et al., 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), pp.391–407.

Dumais, S.T., 2004. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1), pp.189–230.

Emmerson, H. & Emmerson, S.H.C., 1962. *Survey of Problems Before the Construction Industries: Report Prepared for the Minister of Works*, HM Stationery Office.

Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques*, Waltham, USA: Morgan Kaufmann.

Kalman, D., 1996. A Singularly Valuable Decomposition: The SVD of a Matrix. *The College Mathematics Journal*, 27(1), p.2.

Kintsch, W. & Bowles, A.R., 2002. Metaphor Comprehension: What Makes a Metaphor Difficult to Understand? *Metaphor and Symbol*, 17(4), pp.249–262.

Kuo, V. & Wium, J.A., 2014. The management of constructability knowledge in the building industry through lessons learnt programmes. *Journal of the South African Institution of Civil Engineering*, 56(1), pp.20–27.

Landauer, T.K., 2002. On the computational basis of learning and cognition: Arguments from LSA. *Psychology of Learning and Motivation*, 41, pp.43–84.

Landauer, T.K., Foltz, P.W. & Laham, D., 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3), pp.259–284.

De Lathauwer, L., De Moor, B. & Vandewalle, J., 2000. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4), pp.1253–1278.

Sidorova, A. et al., 2008. Uncovering the Intellectual Core of the Information Systems Discipline. *MIS Quarterly*, 32(3), pp.467–A20.

Strang, G., 2003. Introduction to Linear Algebra. *Mathematics of Computation*, 18(1), p.510.

Wall, M., Rechtsteiner, A. & Rocha, L., 2003. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*. pp. 91–109.

Wolfe, M.B.W. & Goldman, S.R., 2003. Use of latent semantic analysis for predicting psychological phenomena: two issues and proposed solutions. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*, 35(1), pp.22–31.

Wong, F.W.H. et al., 2007. A study of measures to improve constructability. *International Journal of Quality & Reliability Management*, 24(6), pp.586–601.