The Topics of CAAD
A Machine's Perspective

Ziga Turk, Tomo Cerovsek, Bob Martens
University of Ljubljana, Slovenia.
Vienna University of Technology, Austria.

Key words: CAAD, machine learning, clustering, pattern recognition

Abstract: Ontology of a scientific field typically includes a taxonomy that breaks up the field into several topics. The break-up is present in the organisation of information in books, libraries and on the Web. An on-line database of papers related to CAAD called CUMINCAD was created and it includes over 3000 papers with abstracts. They are available through the search interface - one knows an author or a keyword and can find the papers where such keyword or author's name appears. Alternative interface would be through browsing papers topic by topic. The papers, however, are not categorised. In this paper, we present the efforts to use the machine learning and data mining techniques to automatically group the papers into clusters and create a set of keywords that would label a cluster. The hypothesis was that an algorithm would create clusters of papers automatically and that the clusters would be similar to the groupings a human would have made. We investigated several algorithms for doing an analysis like that but were unable to prove the original hypothesis. We conclude that it requires more than objective statistical analysis of the words in abstracts to create an ontology of CAAD.

1. INTRODUCTION

An important attribute of any scientific discipline are its framework, vocabulary, methodology, paradigms and structure - as Kuhn (1962) calls them, the "conceptual boxes ... into which the scientists, by a rather a strenuous and devoted attempt to force nature into". According to Seni and Hodges (1997), a field of science is actually defined with the following attributes:
- Axiology defines a value system in the field.
Ontology defines "what exists" and (formally) specifies the conceptualisation of the field.

Epistemology specifies what constitutes appropriate knowledge in the field, where is it and how it can be represented and transferred.

Methodology specifies the appropriate rules of inquiry.

Ontology typically includes taxonomy or some other way of breaking up the field into several topics. Each scientist involved in a field, such as CAAD, could sketch such taxonomy. Implicitly, it is present in the organisations of conference sessions, textbook sections, keywords systems used to tag scientific papers and in the portals on the World Wide Web. While the first uses may be scientifically intriguing and could provide some introspective into the CAAD community, our practical interest lies in the latter.

1.1 Motivation

A comprehensive database of CAAD papers called CUMINCAD was created (Martens and Turk, 1999) on the World Wide Web (http://itc.fgg.uni-lj.si/cumincad/). Currently it includes over 3000 papers. Each paper includes all vital bibliographic information such as title, authors, publication data etc. (Figure 1). Full abstracts are available as well. Some 500 papers from the eCAADe conferences include full texts in .pdf format.

Currently the papers are available through the search interface: if one knows an author or a keyword or any word or words that appear in the abstract, one can find the papers, where the search term appears.

Searching, however, implies that the user knows what she is looking for. Another approach to access the data is through browsing a structured collection of records. The structure should tell the user what items are similar, which are different, and how. The simplest structures of this kind are clusters or groups of similar data items. These groups can be ordered into a hierarchy. A well-known example of hierarchical interface to the content of the Web is the portal Yahoo. The groups may be predefined as classes. In this case we talk about classification and not about clustering.

The users of CUMINAD would like to (1) browse through the papers by the topics of CAAD, or (2), given a paper or a group of papers, find the papers that are similar to those. Because of the size of the database and the fact that the work has been done on a shoestring budget, manual classification of the papers has been ruled out.
CUMINCAD database is handled by a Web database tool Woda (Turk, 1998) and the goal of the first author is also to create a generic solution that could be applied to any database handled by Woda.

Ideally we would use machine learning and data mining techniques to automatically group the papers into clusters and create a set of keywords that would label a cluster. Based on such clusters, categories would be defined and the entire database would be organised accordingly.

### 1.2 Hypothesis

The hypothesis was that it is possible to write an algorithm that would create clusters of papers automatically and that the clusters would be similar to the human understanding of the field. If we name the similarity of papers $p_1$ and $p_2$ with $\sigma(p_1, p_2)$ than we believed that, given a paper $p$, we could find a set of similar papers $s_{[1..n]}$ for which is true that

$$\sigma(p, s_i) >> t$$

and another set $d_{[1..m]}$ where

$$\sigma(p, d_i) << t$$

where $t$ is some threshold value. Depicted graphically, the similarity of a given paper to other papers should look as shown in Figure 2.
Set of papers $s$ would belong to the same cluster because the similarity is much higher than a threshold value. All other would be in set $d$ where the similarity is much lower.

1.3 Paper structure

In Section "Methodology" we first present the related work then the various algorithms used to compute the similarity of papers, which is the basis for categorisation and clustering. In Section "Analysis of the CUMINCAD data" we present the results of the analysis of the CUMINCAD dataset. In Section "Conclusions" we evaluate how well did the analysis meet the stated hypothesis.

2. METHODOLOGY

2.1 Related work

Various AI technologies for analysing text databases are known since the late 1970s (van Rijsbergen 1979). They became particularly popular after the explosion of the World Wide Web, when the search engines were looking for the technologies to increase the relevance of the searches (Zamir and Etzioni, 1999) or build some intelligence into Web browsing (Mladenic, 1999). An example of an "intelligent" interface to bibliographic data is for example the www.researchindex.com. Several Websites implement such technology, for example AltaVista and Google (Bring and Lawrence, 1998).

In the area of architecture, engineering and construction not much work has been done in this direction, particularly because engineering information is not predominantly text, but drawings and product data. Maher and Simoff...
(1998) applied those techniques to searching archives of old projects and to discover new knowledge out of them. Christiansson (1998) used them to simplify access to knowledge bases. The importance of data mining technologies was also acknowledged by Hovestad (1993) in an attempt to manage unstructured design data.

2.2 Algorithms

![Diagram of Document Clustering Process]

In the 1970s and 1980s several algorithms analysing text, comparing documents and extracting most important phrases or sentences out of the text were developed (overview in Willett, 1988, and Zamir and Etzioni 1998). Quite a few programs were written that are freely available.

Both classifications of documents and clustering rely on being able to compute similarities between two documents. The overall procedure (see Figure 5) consists of the following steps: (1) selection of features, (2) extraction of features, (3) vectorisation of documents (4) computing similarities between documents and finally (5) clustering or classification itself.
2.2.1 Selection of features

The parts of the documents or data records considered relevant are selected. For example in the case of our study of bibliographic information, we selected only the title, abstract and keywords and omitted the author and publication information.

2.2.2 Extraction of features

Relevant data is extracted out of the selected data and converted into a format suitable for further processing. Typical first steps in the extraction include the elimination of stop words and stemming. Stop words are words that appear too many times to be of relevance, or words that are known not to be able to contribute to the computation of similarities between records. For example both "the" and "cad" are very frequent. The first would appear in the available stop-word libraries for the English language. The second was found very frequently in the CUMINCAD database.

Stemming is a procedure that discovers roots of the terms and makes sure that for example "mouse" and "mice", "house" and "houses" are treated as a single term.

2.2.3 Vectorisation of documents

This step is usually described as a part of similarity computation, however, in the section Conclusions we explain why we find it important, to discuss it outside that scope. Documents need to be converted to a vector of numbers. The simplest form is to create a term frequency matrix $[\omega_r]$ where the rows are records and columns the terms.

$$[\omega_r] = \begin{bmatrix}
\omega_{r11} & \omega_{r12} & \omega_{r13} & \ldots & \omega_{r1T} \\
\omega_{r21} & \omega_{r22} & \omega_{r23} & \ldots & \omega_{r2T} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega_{rN1} & \omega_{rN2} & \omega_{rN3} & \ldots & \omega_{rNT}
\end{bmatrix}$$

$T$ is total number of different terms in all records and $N$ is the number of records. $\omega_{r12}$ tells us, how many times does term number 2 appear in record 1.

2.2.4 Computing similarities between documents

The algorithms for the computing of similarities between documents differ in how they calculate the weight they give to words appearing in a document. One approach might only count if a
word is present or not; another, how many times the word is repeated and how many words are there in total etc. etc. Perhaps the most often used approach is the naive Bayes. As an illustration, we present the CYBERMAP (Salton, 1989).

2.2.5 Document clustering

There are two types of document clustering algorithms. The bottom-up algorithms start with each document as one cluster and then group them to form bigger clusters of documents. Top down algorithms work top down, iteratively, so that they create two groups of documents that are as different to each other as possible, than each of the groups is further split into two etc. Clustering can result in a linear or hierarchical set of clusters. For a broader overview see Willett (1988).

3. ANALYSIS OF THE CUMINCAD DATA

Table 1: Collection statistics (terms related statistics in general and vector specific)

<table>
<thead>
<tr>
<th>Number of all records (in whole collection)</th>
<th>Number of stem terms (in whole collection)</th>
<th>Average term usage (in whole collection)</th>
<th>Average term vector length</th>
<th>Standard deviation of vector length</th>
<th>Average frequency of terms in vector</th>
<th>% of terms with freq. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3042</td>
<td>11762</td>
<td>14.76</td>
<td>57.39</td>
<td>30.32</td>
<td>1.39</td>
<td>78.66</td>
</tr>
</tbody>
</table>
To enhance CUMINCAD, the described procedures will be used on the entire 3000 papers. To compare various approaches and parameters we extracted only the 103 papers from the eCAADe 1999 conference in Liverpool. The structure of the proceedings, done by the organisers, provides a chance to compare the results of the machine made clusters to those made by a human editor of the proceedings. Figure 4 illustrates the most frequent words. Detailed statistical analysis of the whole collection is summarized in Table 1.

3.1 Similarity measures

Figure 5 shows part of the similarity matrix of the Liverpool data set.

![Figure 5. Visualisation of similarity matrix based on the Liverpool eCAADe papers. If a row for one given paper would be extracted and sorted a threshold value of as in Figure 2 would not be evident (Figure 6).](image)

On the other hand, the graph of records most similar to Tweed (1999) is depicted in Figure 6. This is quite unlike the graph in the hypothesis. If there is no clear boundary for a single record, can we hope to find clusters with any degree of relevance?
3.2 Clustering with Clustifier

Developed at NRC Canada, Clustifier (Scott 1998) uses representation based on semantic as well as syntactic linguistic knowledge. The results of the clustering are shown in Figure and Figure 7.

| Cluster 0: caad education, computer aided architectural design, experience, caad research, design process, techniques, world wide web, presents |
| Cluster 1: spatial, methods, teaching, architectural education |
| Cluster 2: architecture, learning, virtual reality, experience, architectural design, based design, built environment, design process, design studio, caad |
| Cluster 3: virtual design studios, design process, virtual reality, approach, collaborative design |
| Cluster 4: information technology, urban planning, research projects, techniques, architects, visualisation, 1998, presents, analysis, computer technology |
| Cluster 5: computer aided architectural design, architectural computing, models, computer graphics, architectural representation, analysis, design process, techniques, year 2000 |
| Cluster 6: computer aided architectural design, learning, information technology |
| Cluster 7: 86 architectural education, virtual environments, architectural design education, modelling, information technology, computer aided architectural |

*Figure*: Clusters titles as recognised by Clustifier. Note that this tool identifies phrases!
Cluster 0: caad education, computer aided architectural design, experience, caad research, design process, techniques, world wide web, presents


Cluster 1: spatial, methods, teaching, architectural education


Cluster 2: architecture, learning, virtual reality, experience, architectural design, based design, built environment, design process, design studio, caad


Figure 7. Some clusters as generated by the Clustifier program and related papers.

Figure 8. Topic of CAAD as recognised by the Crossbow program. Characteristic terms are listed on the section headings. Note that the words are stemmed.
3.3 Clustering with Crossbow

Crossbow (McCallum 1996) is a classification tool that uses a hierarchical, top down, EM (expectation, maximisation) clustering technique. After over 150 iterations it produced the result as shown in Figure 8 and Figure 9.

Cluster 1: learn reflect research program institut lectur interfac histor base offer

Cluster 5: system model design form idea dynam analyt comput reason behaviour

Figure 9. Content of some clusters generated by Crossbow.

3.4 Clustering by a human

Andy Brown, Michael Knight and Philip Berridge (1999), the editors of the 1999 eCAADe proceedings organised the proceedings into the following clusters:
4. CONCLUSIONS

It certainly is possible to find some sense in the clusters created by the machine and these clusters can be used to suggest a user of CUMINCAD to look at a few other papers, in addition to the ones she found. In this way she would examine more works and stay on the site longer. The functionality will be made part of CUMINCAD during the first half of 2001.

However, the machine made clusters are much different to the clusters created by the humans. It seems hard for a machine without the background deep knowledge to cluster a topic on its own and, in this way autonomously define what the topics of CAAD are. Indeed the presented algorithms have numerous parameters by which the results can be tweaked, but, after extensive tweaking by humans, how “machine generated” would the results then be?

This confirms our belief that the way we understand the topics of CAAD, and into which this or that paper belongs to, is subjective and based on one's the current interests and perspectives. What defines a scientific community is, that its members, to a large extent, also share a similar deep understanding of the topic.

A machine can learn this shared perspective in several ways. One is through parameter tweaking of the presented algorithms – initial tests show that best results are achieved by manually expanding the list of stop words.
Through this approach, however, the machine would only learn a view of the tweaker.

4.1 Future work

Our future work will be dedicated to learning about the shared perspective in the same way as the humans did – through interaction with the members of the scientific community. For example, the machine can learn from the humans that use a database by observing what papers a single user in a certain time frame looks at - it is very likely that these papers belong to the same topic. Similarly, man made clusters e.g. from conference proceedings, could be used as a learning example through which the machine could learn to classify in the same way, as that conference organisers did. In this way, the machine can learn the communities' view and pass on this view onto the new members of the community. By observing the users, browsing through the classes, the machine could establish which classifications are correct and which not.

While the learning from raw data failed in our case, we believe that in collaborative Web based applications, the real potential of machine learning is in the learning from the users. The implementation is rather simple - in the next iterations of the clustering we plan to add one extra field to the extracted data set – the name of the reader who demonstrated interest in a particular record. This data is readily available from the server log files.

4.2 Future research topics

The presented methods are not relevant only in the context of scientific papers but with any other text databases related to engineering, such as building codes, historical project data, best practice guides etc. Same techniques, as presented, can also be used to ease the navigation and increase the relevance of the search results.

Text, however, is not the format in which architects and engineers would encode most information. We are used to work with drawings and product data. Computer scientists developed numerous algorithms for text learning, clustering and classification. All these algorithms work with text. However, as we have shown in Section 2 only initial steps in the algorithms parse text. Afterward only numbers are crunched. To take advantage of such technologies, CAD community needs to develop algorithms for selecting features, extracting features and vectorisation of drawings and product data.
5. REFERENCES


