A Simple Approach towards Visualizing and Evaluating Complexity of Textual Case Bases

Sutanu Chakraborti, Ulises Cerviño Beresi, Nirmalie Wiratunga, Stewart Massie, Robert Lothian and Stuart Watt

School of Computing,
The Robert Gordon University
Aberdeen AB25 1HG, Scotland, UK
Email: {sc|ucb|nw|sm|rml|sw}@comp.rgu.ac.uk

Abstract We present an approach to visualize textual case bases by "stacking" similar cases and features close to each other in an image derived from the case-feature matrix. We propose a complexity measure called GAME that exploits regularities in stacked images to evaluate the alignment between problem and solution components of cases. $GAME_{class}$, a counterpart of GAME in classification domains, shows a strong correspondence with accuracies reported by standard classifiers over classification tasks of varying complexity.

1 Introduction

This paper presents a novel approach to visualizing textual case bases, and evaluating their complexity. Visualization is useful in the Textual CBR (TCBR) context for the following reasons:

1. easing knowledge acquisition from human experts
2. visually evaluating goodness of the underlying representation,
3. aiding case base maintenance, by revealing redundant features or noisy cases
4. presenting and explaining retrieved results to end users

The first three are concerned with building and maintaining textual case bases, and are "off-line" activities in that they do not directly concern retrieval. In contrast, the fourth is an "on-line" activity, and is outside the scope of the current paper. Also, it may be noted that throughout this paper, we will focus on visualizing the case base in its entirety, and not individual cases.

Our second goal is to evaluate case base complexity; this is important in facilitating the three off-line tasks mentioned above, particularly tasks 2 and 3. In case of task 2, a complexity measure would provide a quantitative basis for assessing the suitability of a representation, while visualizations aid qualitative judgements by humans. While visualization and complexity evaluation have often been treated in isolation, our current understanding is that they often share similar goals, and may exploit similar mechanisms to realize these goals as well.

Visualization is a well studied sub-field of text mining (TM) [5], and it is not surprising that most approaches investigated till date can be extrapolated to TCBR tasks. However, some differences are worth noting. Firstly, most visualization

approaches in TM focus either on visualizing clusters of documents, or of words, but not both. In TCBR maintenance tasks, we often want to highlight the nature of interrelationships between words (alternately higher level TCBR features) and documents (cases) that give rise to the clustering patterns, and serve as an explanation for the underlying complexity. This helps in case base maintenance, as we can identify noisy cases or redundant features [7]. A second distinction, and one that has a strong bearing on complexity evaluation, is the TCBR emphasis on the split between problem and solution components of a textual case. We choose a representation that maximizes the "alignment" [4] between problem and solution components of texts. This issue has not been explored by researchers in TM visualization. Thirdly, TCBR representations are often more knowledge rich in comparison to those used in TM or Information Retrieval (IR). In contrast to shallow Bag Of Words (BOW) representations used in TM/IR, TCBR often uses "knowledge entities" ranging from domain specific terms, phrases or syntactic patterns from Information Extraction, as features [14]. However, this distinction is not critical here since our approaches are agnostic to the kind of features, though both visualization and complexity measures can take into account sophisticated domain-specific similarity measures associated with knowledge rich features.

Our first contribution in this paper is the idea of visualizing a textual case base as an image displaying a matrix of cases and features such that interesting associations and clusters within the case base are revealed. We present a simple algorithm that generates this image by exploiting regularities across cases and features. The resulting image has more than just a visual appeal; the compressibility of the image is used to arrive at a novel measure of complexity called GAME (for Global Alignment MEasure) that estimates alignment between problem and solution components of cases. We present experimental studies to show that GAME correlates well with classifier accuracies in classification problems of varying complexity.

2 The "Case Base as Image" Metaphor

Let us consider a set of textual cases, each case consisting of a set of features. For simplicity, we treat words in the text as features; the ideas presented can easily be extended to deal with more complex features. Also, we will restrict our attention to the problem side of cases, for the moment. To illustrate our ideas, we model the documents in the toy Deerwester collection [6] as cases. This is shown in Fig. 1(a). An alternate representation is in the form of case-feature matrix shown in Fig. 1(b); elements are 1 when a feature is present in a case, 0 otherwise. It is straightforward to map this matrix onto an equivalent image, shown in Fig. 2(a), where the values 0 and 1 are mapped to distinct colours, a lighter shade denoting 1. We obtained this image, and for that matter all other images in this paper, using Matlab. Very simply put, this is the "case base as image" metaphor. However the image as it stands, is not very useful. Firstly, it conveys very little information about underlying patterns in terms of word or document clusters. Secondly, the image is highly sensitive to how the words and documents are arranged in the matrix; this is clearly undesirable. Thirdly, and we shall explore this in more detail in Section 3, the image tells us very little about the complexity of the underlying case base.

To address these limitations, we propose an algorithm that does a twofold transformation on the case-feature matrix to yield a matrix where similar cases (and similar features) are stacked close to each other. The output is a matrix, which when visualized as an image, captures the underlying regularities in the case base. Fig 3 shows a sketch of the algorithm. The broad idea is as follows. The first case row in the original matrix is retained as it is. Next, we compute the similarity of all other cases to the first case, and the case most similar to the first case is stacked next to it, by swapping positions with the existing second row. If more than one case is found to be equally similar, one of them is chosen randomly. In the next step, all cases excepting the two stacked ones are assessed with respect to their similarity to the second case. The case that maximizes a weighted combination of similarities to the first and second case (with higher weight assigned to the second case) is chosen as the third case, and stacked next to the second row. The process is repeated till all rows are stacked. In Step 2 of the algorithm, the same process is repeated, this time over the columns of the matrix generated by Step 1.

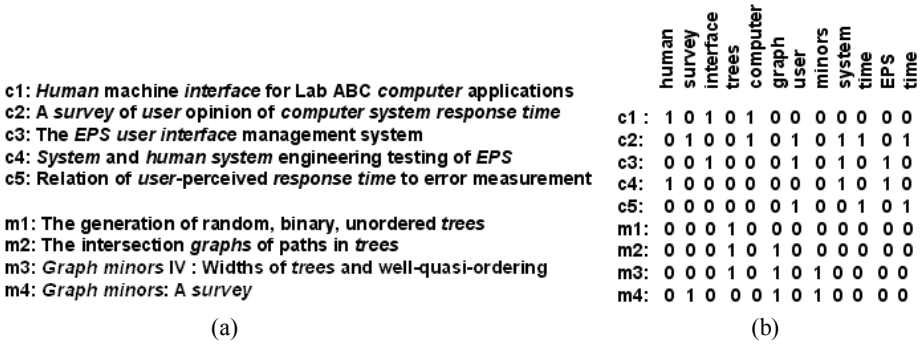


Fig.1. Documents in the Deerwester Collection

The weighted similarity evaluation is critical to the working of this algorithm and merits a closer look. The general rule for selecting the (k+1) row (case) is to choose the one that maximizes

$$\sum_{i=1}^{k} w_i \, sim(c_i, c) \text{ such that for all } 1 \le i < k, \; w_{i+1} > w_i \qquad (1)$$

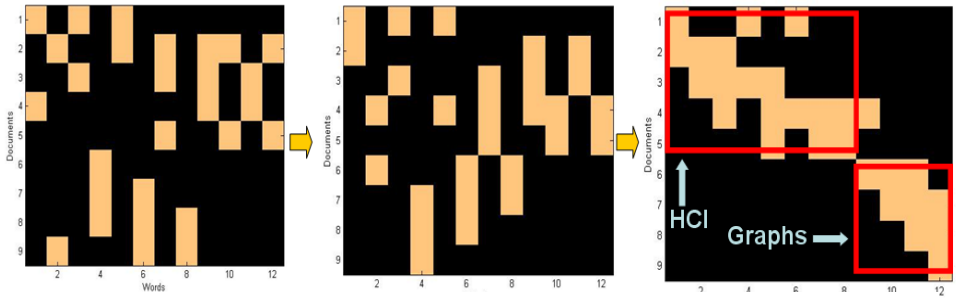
where k is the number of already stacked rows, c_i is the ith stacked case, c is a case whose eligibility for (k+1)th position is being evaluated, $sim(c_i, c)$ is the cosine similarity between cases c_i and c, and w_i is the weight attached to the similarity of c with the ith stacked case. In our implementations, we used

$$w_i = 1/(k - i + 1) \qquad (2)$$

The basic intuition behind this approach is that we want to ensure a gradual change in the way cases are grouped. This has implications for facilitating a meaningful display of clusters, and also for the complexity evaluation discussed in Section 3. If only $sim(c_k, c)$ were considered for the stacking process (which is equivalent to assigning 0 to all w_i, i = 1 to k-1) we may have abrupt changes resulting in an image that fails to reveal natural clusters. We note that for efficiency reasons, our implementation uses an approximation of (2), where we take into account only the previous 10 stacked

cases and no more, since the weights associated with very distant cases are negligible and have no significant effect on the ordering. Choosing the starting case for ordering cases is an important issue, that we examine in the next section.

Fig. 2(a) shows the image corresponding to an arbitrary arrangement of the documents in the Deerwester matrix. Fig. 2(b) shows the image after the rows are stacked. Fig. 2(c) is the final image after column stacking. It is interesting to see that the two broad topics within the collection, namely *Human Computer Interaction (HCI)* and *graphs* are clearly visible in Fig. 2(c) as two "chunks" of contiguous light shades. Also, there is a gradual transition in shades from *HCI* to *graphs*. This is useful in identifying "bridge words" that can serve to connect two topics; an example is word 9 ("survey") in Fig. 2(c) which is common to *HCI* and *graphs*. We can also visually identify cases that are in the topic boundaries and deal strongly with more than one topic. This has strong implications in case base maintenance tasks in terms of identification of noisy cases, and redundant features [7]. We have designed a simple interface that allows users to "navigate" the image, and visualize the "topic chunks", and words that describe those chunks. An example is shown in Section 5.



(a) (b) (c)

Fig.2. Images from Deerwester collection (a) arbitrarily stacked (b) after row stacking (c) after column stacking

Step 1 (Stack Rows)

Input : Case-Feature Matrix M
Output : Case-Feature Matrix M_R which is M stacked by rows
Method:
Instantiate first row of M_S to first row M
for k = 1 to (noOfRows-1) /*the index of the last case (row) stacked*/
  for j = (k+1) to noOfRows /* check through all candidate cases*/
    $wsim_j = 0$; /* $wsim_i$ weighted similarity of ith case */
    for i = 1 to k /* already stacked rows*/
      $wsim_j = wsim_j + wsim_j*(1/(k-i+1))*sim(c_i,c_j)$;
    end
  end
  choose j that maximizes $wsim_j$ and interchange rows (k+1) and j
end

Step 2 (Stack Columns)

Input : Case-Feature Matrix M_R generated by step 1
Output : Case-Feature Matrix M_C which is M_R stacked by columns
Method: same as in Step 1 except that columns are interchanged (based on feature similarity computed as cosine similarity between columns) instead of rows.

Fig.3. The Stacking Algorithm

3 Complexity Evaluation Using Compression

In this section, we explore how the image metaphor can be exploited to obtain a measure of the case base complexity. There are two reasons why complexity evaluation is useful. Firstly, we can predict difficulty of domains (datasets) for a given choice of representation (feature selection/extraction and similarity measures). Secondly, we can compare across different choices of representation over a fixed domain and choose the representation that minimizes complexity. We observe that complexity over a case base can be defined in two ways, namely Alignment Complexity (AC) and Collection Complexity (CC). The former, which is our main concern in this paper, measures the degree of "alignment" [4] between problem and solution components of textual cases. Measuring this helps us in answering the question "Do similar problems have similar solutions?" and thereby assessing the suitability of CBR (or alternatively the choice of representation) to that task. A special case of this problem is seen in classification domains, where the solution is replaced by class label. In measuring CC, the distinction between the problem and solution components of cases is ignored, and the complexity measure provides a measure of clustering tendencies exhibited by the case base. Thus a case base with cases uniformly distributed over the feature space has a high complexity; whereas, one with more well-defined clusters has a lower complexity [12]. Intuitively, since the stacked image captures regularities arising from topic chunks in the case base, we would expect that, all else being equal, stacked images from simpler domains will be more compressible, and thus have higher compression ratios, compared to ones from complex domains. This is because image compression algorithms typically exploit regularities to minimize redundancy in storage. Alternatively, a simple domain is one where case clustering serves as an explanation for feature clustering, and vice versa. We carry forth this intuition into our discussions of AC, since AC can be thought of as an extension of CC.

Alignment can be interpreted in two different ways. The first interpretation is a local one; an example is the case cohesion metric formulated by Lamontagne[4]. Here we look at a case, say C, in isolation, and determine two sets: set S_1, which comprises cases whose problem components are closest to the problem component of C (based on a threshold), and a set S_2, comprising cases whose solution components are closest to the solution of C. The overlap between S_1 and S_2 is used as a measure of alignment of C. This is a local metric, in that each case is evaluated on its own, and assigned a measure. The second interpretation is a global one based on how well the clusters derived from problem components of cases correspond to clusters derived from solution components. In this paper we adopt this second interpretation of alignment.

For measuring alignment, we construct two case-feature matrices: one based on problem components of cases, the other based on solution components. These two matrices are stacked as described in Section 2, to yield two images I_P and I_S respectively. I_P and I_S are now independently compressed to obtain compression ratios CR_P and CR_S respectively. For measuring alignment, it is interesting to compare the ordering of cases in I_P and I_S. One way of doing this is to create a fresh solution side image I_{SP} by stacking solution components of cases using the problem side ordering of cases as read out from I_P. We would intuitively expect I_{SP} to be less compressible than I_S, unless the case base is perfectly aligned. Compressing I_{SP} yields

a new compression ratio CR_{SP}. Let CR_{SMIN} denote the minimum compression ratio that can be obtained by reordering the solution components independent of the problem components. The Global Alignment MEasure (GAME) is given by $(CR_S - CR_{SMIN})/(CR_S - CR_{SP})$. A higher value of GAME indicates a better alignment.

GAME can be extended to classification domains where the class label is treated as a solution. In this case, our interest is in determining whether near-neighbours in the problem side ordering (as obtained from I_P) belong to the same class. We obtain a string of class labels corresponding to the problems as they appear in the problem side ordering. This allows us to do away with the image compression and resort to a simpler string based compression instead. As an illustration, let us consider a two class problem of 10 cases in the email domain, where cases C_1 through C_5 belong to class S (for SPAM) and C_6 through C_{10} belong to L (for LEGITIMATE). Let us assume that the problem side ordering of the cases after stacking is $C_1C_2C_6C_4C_5C_7C_3C_9C_{10}C_8$. Replacing each case identifier with its class label, we obtain the class string SSLSSLSLLL. The most easily classifiable case base would have a string SSSSSLLLLL, and the most complex would have SLSLSLSLSL. A compression algorithm that exploits contiguous blocks (but not compound repeating patterns like SL) would thus be ideal; Run Length Encoding is one such scheme. Using this, the complexity is a direct function of the number of the flips (changes from one class label to another, N to S or S to N in the above example). We define GAME complexity measure for classification as

$$\mathrm{GAME}_{\mathrm{class}} = \log\left(\frac{flips_{\max} - flips_{\min}}{flips - flips_{\min}}\right) = \log\left(\frac{(n-1) - (k-1)}{flips - (k-1)}\right) \qquad (3)$$

where k is the number of classes, n is the number of cases ($n > k$), $flips$ is the number of transitions from one class to another in the class string, $flips_{\min}$ is the value of $flips$ for the simplest possible case base having n cases and k classes, and $flips_{\max}$ is the value of $flips$ for the most complex case base. We note the most complex case-base presupposes a uniform class distribution; we then have $flips_{\max} = (n-1)$. A higher value of $\mathrm{GAME}_{\mathrm{class}}$ corresponds to a simpler domain; the most complex domain has $\mathrm{GAME}_{\mathrm{class}} = 0$. Thus we expect positive correlation of $\mathrm{GAME}_{\mathrm{class}}$ to accuracy results derived from classifiers. The logarithm has a dampening effect on the large values that could result when $n >> k$, $flips$. As a further detail, a small constant (say 0.01) should be added to the denominator to avoid division by zero when $flips = flips_{\min}$. Considering the inverse relation that exists between flips and compression ratio ($flips_{\min}$ corresponds to CR_S, and $flips_{\max}$ to CR_{SMIN}), and ignoring scaling due to logarithms, it is clear that $\mathrm{GAME}_{\mathrm{class}}$ can be viewed as an extension of GAME.

An important issue that merits closer attention is the choice of the starting case in the stacking process, and its influence on the visualization and complexity measure. Our experiments have shown that visualizations are not widely affected by the choice of starting cases, except for the shuffling in the order in which clusters are displayed. Even though the variance of GAME was found to be small over the choice of starting cases, we should, theoretically choose the maximum value that can be obtained. The arrangement that yields this value can be found by performing stacking using each case as the starting case at the time and picking the one that produces the maximum GAME score. More research needs to be done into finding efficient ways of pruning this search space to make the process less computationally expensive.

4 Experimental Results

Evaluating the general formulation of GAME involves a study of its correlation with an effectiveness measure (like precision/recall/F-measure) derived from subjective relevance judgments from experts over diverse casebases. Because of the difficulty in obtaining such TCBR datasets with relevance rankings, we evaluated the adapted version of GAME ($GAME_{class}$) over six different classification tasks.

For evaluating classification effectiveness in routing, we created datasets from the 20 Newsgroups [1] corpus. One thousand messages from each of the 20 newsgroups were chosen at random and partitioned by the newsgroup name [1]. We form the following four sub corpuses: SCIENCE which has 4 science related groups, REC which has 4 recreation related groups, HARDWARE which has 2 problem discussion groups, RELPOL which has 2 groups on religion and politics. Two datasets used for evaluation on spam filtering are: USREMAIL [11] which contains 1000 personal emails of which 50% are spam and LINGSPAM [8] which contains 2893 email messages, of which 83% are non-spam messages related to linguistics, the rest are spam. Equal sized stratified disjoint training and test sets were created, where each set contains 20 % of the dataset of documents randomly selected from the original corpus. For repeated trials, 15 such train test splits were formed. Documents were pre-processed by removing stop words and some special characters. We use an Information Gain based feature selection. Fig. 4 shows the $GAME_{class}$ values obtained over the 15 trials in each of the six datasets. Of the two class problems, LINGSPAM and USREMAIL have high $GAME_{class}$ values indicating that they are simpler compared to HARDWARE which has a low $GAME_{class}$ value. Table 1 suggests that $GAME_{class}$ predictions are supported by accuracy figures recorded by five classifiers. Support Vector Machines (SVMs) [2] have been shown to be very successful with textual data [5], Latent Semantic Indexing (LSI) and its class-aware version sprinkled LSI (LSISPR in the table) are interesting in the TCBR context, since they lend themselves to instance based retrieval, and incremental learning [3]. LogitBoost is a boosting approach grounded on weak learners in the form of decision stumps [5]. The current formulation of $GAME_{class}$ allows for more meaningful comparisons between problems when they have the same number of classes. So we compared the binary and four-class problems separately. The correlation coefficient of the $GAME_{class}$ score against classification accuracies over the four binary problems are shown in Table 2. We note a strong positive linear correlation of $GAME_{class}$ to all four classifiers. It is also interesting to note a stronger correlation of $GAME_{class}$ to LSISPR as compared to LSI, hinting at the importance of class knowledge. It is pointless to do correlation over the four-class datasets since we have just two of them; however we observe that $GAME_{class}$ declares SCIENCE to be more complex than REC, and this is confirmed by all classifiers. SVM being inherently a binary classifier was not tried on the 4-class datasets, though we plan to experiment with multi-class SVM in future. Figs. 5(a) and 5(b) shows stacked images from one of the trials in RELPOL and USREMAIL respectively. Of the two, RELPOL is sparser with less conspicuous chunks, thus partially explaining its lower GAME value. Fig. 5(c) shows the result of stacking on a representation generated by LSI; it is interesting to observe that the LSI image is relatively blurred; also the compressed LSI image is approximately 73% the size of

the original compressed image. We note that both LSI and LSISPR results were at a dimensionality setting where they yielded best performances [3].

5 Related Work

Visualization techniques in Text Mining have typically attempted to display one of word associations or document clusters, but seldom both. Techniques to display word associations include word association graphs and circle graphs [5]. For visualizing document clusters, a common approach is multidimensional scaling which projects documents in a high dimensional space to a two dimensional one, under the constraint of preserving the similarity relationships between documents, as closely as possible. An approach that comes close to our idea of stacking in terms of the generated layout is the Hierarchical Clustering Explorer [10] which dynamically generates clusters based on user-defined thresholds, and displays the mined document clusters. In addition to the fact that word clusters are not displayed, one other limitation of this

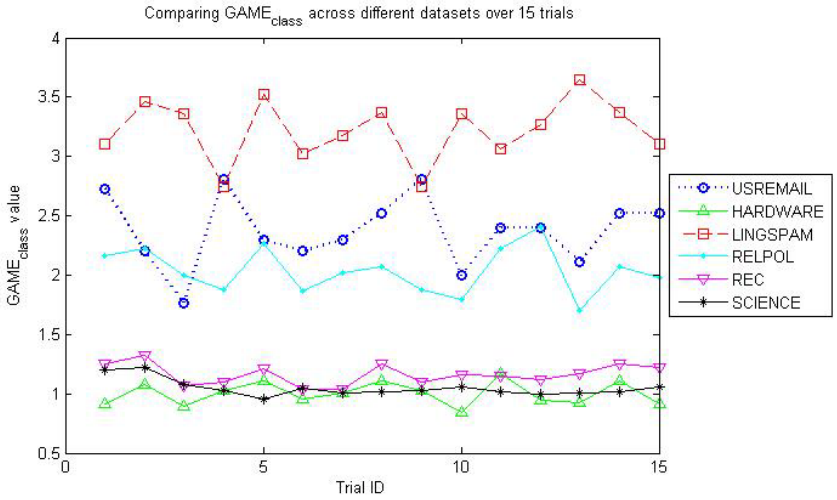


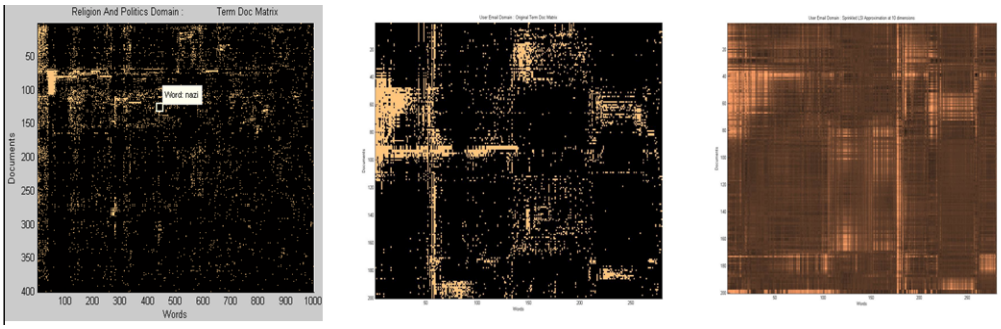
Fig. 4. GAME$_{class}$ values across different datasets

Table 1. GAME$_{class}$ and Accuracies obtained by different classifiers

	HARDWARE	RELPOL	USREMAIL	LINGSPAM		REC	SCIENCE
GAME measure	1.0028	2.0358	2.3728	3.2222		1.1629	1.0492
LSI + kNN-3	66.30	91.17	94.67	97.37		79.32	72.55
LSISPR + kNN-3	80.42	93.89	96.13	98.34		86.99	80.60
SVM	78.82	91.86	95.83	95.63		--	--
LogitBoost	77.99	79.67	92.67	95.80		87.15	73.77

Table 2. Correlation of GAME$_{class}$ with classifier accuracies over 4 binary classification problems

	LSI + kNN-3	LSISPR + kNN-3	SVM	LogitBoost
ρ	0.9176	0.9365	0.9023	0.8820



(a) (b) (c)

Fig. 5. Stacked Images

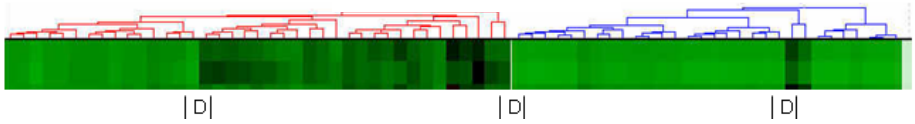


Fig. 6. A snapshot of hierarchical visualization (courtesy HCI Maryland website [10])

approach is that there is no clear way of choosing the right ordering between several sub-trees under a given node. This may lead to discontinuities in the image (some of which are marked by D in Fig. 6) and sudden change in concepts. Thus it would fail to reveal patterns revealed by the weighted stacking approach. An approach that comes close to showing both words and documents in the same space is WEBSOM [5]. WEBSOM fails to preserve the structure of cases as a set of feature values, and is unwieldy for case base maintenance. Furthermore, our approach has the relative advantage of being free from convergence problems faced by WEBSOM.

While compression models have been used in [11] for feature-free retrieval in CBR, it has not been used before for complexity evaluation. It will be interesting to examine parallels between conditional complexity measures [11] and GAME.

It would be interesting to explore parallels between "topic chunks" revealed by the stacked image, and concepts as mined by Formal Concept Analysis (FCA) [13]. While FCA has been applied to TCBR tasks, the inherent sparseness of textual data leads to generation of a large number of concepts that are brittle and unintuitive. Relaxing the strict closure requirements of FCA could possibly lead to "approximate concepts". Our intuition is that a topic chunk, when interpreted as a blurred rectangular version of the actual light shades in close proximity, may be a close analog to such an approximate concept. It is worth noting that this blurring operation can be viewed as smoothing of cases based on neighbourhood of each cell, thus achieving feature generalization. Blurring makes sense only on the stacked image since we are assured that neighbouring cells are likely to correspond to similar cases and features; it is meaningless on the original image where the arrangement is arbitrary. In our earlier work on LSI-based classification [3], we presented examples to show that lower rank approximations to case feature matrices generated by LSI can be regarded as blurred versions of the original. This parallel opens up avenues for exploring alternatives to LSI that tailor the blurring to cater to specific TCBR goals.

6 Conclusion and Future Work

We presented a simple approach to visualize textual case bases. The stacked image display can help knowledge engineers to get a bird's eye view of the domain, thus facilitating knowledge acquisition. The visualization has three main advantages over other approaches. Firstly, it shows case and feature clusters in relation to each other, thus allowing case clusters to be explained in terms of feature clusters, and vice versa. Secondly, since stacking does not rely on any abstraction, it preserves the structure of cases and displays case and feature vectors as they are. This helps case base maintenance since noisy cases, redundant features or "bridge" features are revealed. Finally, stacking is fast and simple to implement, has no convergence problems, and is parameter-free for all practical purposes. We have also introduced a complexity measure founded on the idea of stacking. We showed that in classification tasks, an adapted version of this measure corresponds closely to accuracies reported by standard classifiers. As part of future work, we would like to carry out an evaluation of the original GAME measure on unsupervised case bases over which relevance judgements are available. On the visualization front, an interesting extension to our current interface would be a facility to show feature associations in each topic chunk in the style of association graphs [5] rather than displaying just a list of features. This may enhance its usability for the knowledge engineer.

References

1. Mitchell, T.: Machine Learning. Mc Graw Hill International (1997)
2. Joachims,T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proc. of ECML, ACM Press (1998) 137–142
3. Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S., Harper, D.: Supervised Latent Semantic Indexing using Adaptive Sprinkling, in Proc. IJCAI (2007) 1582-7
4. Lamontagne, L.:Textual CBR Authoring using Case Cohesion, in TCBR'06 - Reasoning with Text, Proceedings of the ECCBR'06 Workshops (2006) 33-43
5. Feldman R., Sanger, J.: The Text Mining Handbook, Cambridge University Press (2007)
6. Deerwester S.,C., Dumais, S.,T., Landauer T.,K., Furnas,G.,W, Harshman, R.,A.: Indexing by Latent Semantic Analysis, JASIST, 41(6) (1990) 391-407
7. Massie, S.: Complexity Modelling for Case Knowledge Maintenance in Case Based Reasoning, PhD Thesis, The Robert Gordon University, 2006
8. Sakkis, G., Androutsopoulos, I.,Paliouras,G., Karkaletsis,V., Spyropoulos, C.D., Stamatopoulos,P.: A Memory-based Approach to Anti-Spam Filtering for Mailing Lists. Information Retrieval, 6: (2003) 49–73
9. Delany S.J., Cunningham P.: An Analysis of Case-base Editing in a Spam Filtering System in Proc of ECCBR (2004) 128–141
10. HCE visualization, HCI Lab, University of Maryland: http://www.cs.umd.edu/hcil/hce/
11. Delany, S.J., and Bridge,D.: Feature-Based and Feature-Free Textual CBR: A Comparison in Spam Filtering, Procs. of Irish Conference on AI and Cognitive Science (2006) 244-253
12. Vinay, V., Cox, I.J., Milic-Fralyling, N., Wood,K.: Measuring the Complexity of a Collection of Documents, Procs. of 28th ECIR (2006) 107 - 118
13. Díaz-Agudo, B., González-Calero, P.A.: Formal concept analysis as a support technique for CBR, Knowledge Based Syst. 14(3-4): (2001) 163-171
14. Brüninghaus, S., Ashley, K.D.: The Role of Information Extraction for Textual CBR. In Proceedings of 4th ICCBR, Springer, (2001) 74-89