[23] Shakir, Hussain Sabri, "Context-Sensitive Processing of Semantic Queries in an Image Database System", Information Processing & Management, Vol. 32, No. 5, 1996. P. 573-600.

[24] Zloof, M., M.; "Query-By-Example: A Database Language", IBM Systems Journal, Vol 16(4), 1977, P. 324-343.

[25] Zweigenbaum, "MENELAS: An Access System for Medical Records Using Natural Language", Computer Methods and Programs in Biomedicine, 1994, P. 117-120.

Keyword Association Network: A Statistical Multi-term Indexing Approach for Document Categorization

Kang H. Lee[*] *Judy Kay*[*] *Byeong H. Kang*[+]

[*]Basser Dept of Computer Science
University of Sydney
N.S.W., 2006, Australia
{kangl,judy@ cs.usyd.edu.au}

[+]School of Computing
University of Tasmania
Hobart, Tasmania, 7001, Australia
bhkang @ utas.edu.au

Abstract

A Keyword Association Network (KAN) is the network of keywords extracted from a collection of documents. In this network, the relationship between keywords is represented by a confidence value. It is argued in this paper that the semantics and importance of a word can be more clearly and accurately measured by making use of other words that are co-occurring in a given document. The term frequency used for measuring the importance of terms in most document categorization methods ignores this important aspect. A KAN is constructed on the basis of co-occurring terms in documents. If two tenns appear more than a certain number of times in the same documents, they are considered as having close relationship. This paper proposes using KAN as a basis for finding informative keywords and using a confidence value in the process of document categorization. The process of constructing and application of KAN for document categorization is presented and the performance comparison with a typical statistical single-term document categorization algorithm - TFIDF classifier - will be shown. The experimental results show that KAN gives significant benefits.

Keywords Document Categorization, Machine Learning, Statistical Multi-term indexing, Semantic-Meaning.

1 Introduction

Term indexing, sometimes known as feature selection or feature extraction, is concerned with extracting informative terms - keywords - from documents based on a weighting scheme. Term indexing has been studied using a growing number of statistical and machine learning techniques and is a crucial part of both document categorization and recommendation systems for textual information. For example, the adjusting parameter (5) extracted terms and their weights are used when calculating the distance or similarity between two documents.

Based on the weighting scheme, term indexing can be broadly grouped into the following three categories: statistical, probabilistic, and information theoretic [7]. In statistical weighting schemes, frequently occurring terms in documents are regarded as informative terms and the term frequency *(TF)* is assigned to term weight. It is also widely recognized that terms which occur in only a few other documents are more informative than ones that appear in many. This consideration results in introducing the inverse document frequency *(IDF)*. These two are combined into a single value, called *TFIDF* [19] and this is widely used for document categorization [8, 11, 12, 13, 22]. Probabilistic weighting schemes are also commonly used for term indexing. When applying such a weighting scheme to document categorization [3, 8, 10], the joint probability of term and category is computed from the training document set and used for the term weights to estimate the probability of a category given a document. Information theoretic methods are based on information gain [15]. In this method, the terms that are concentrated in particular documents are considered as informative terms by measuring signal-noise ratio. In this sense, it operates in a similar way to the *IDF* of statistical weighting schemes.

Term indexing is also categorized into single and multi-term indexing. In single-term indexing on which most previous research has been focused, term weighting schemes are applied to one word without considering the relationship between words. On the other hand, a different point of view on extracting informative terms is that a more accurate and precise meaning of a term can often be identified when looking at other terms in a document. For example, let

Proceedings of the 5th Australasian Document Computing Symposium,
Sunshine Coast, Australia,
December 1, 2000.

us consider the similarity between two documents where 'apple' appears in both documents. If one document is in the computer category and the other is about fruit, the term 'apple' has a different meaning in each document. If we adopt a single-term indexing scheme, the similarity value may be high and. as a result, two documents may be incorrectly considered as being similar. A multi-term indexing scheme is intended to solve the above problem by designing term weights to include information relationships between terms. Latent Semantic Indexing (LSI) [2] is an example of the multi-term indexing methods. It uses word relationships by coalescing terms which have similar meanings and has been successful in reducing the dimensionality in information retrieval area.

This paper introduces a statistical multi-term indexing scheme, the Keyword Association Network (KAN). In KAN, each word is connected with other words if they are assessed as being related and a confidence value [1] is used for measuring relationship between two words. The effectiveness of KAN for term indexing has been evaluated in document categorization. From the training and test documents in each category, our approach is to build a KAN using training data in each category and use the confidence value on the term weight when calculating similarities of test documents with categories. This approach is compared with a TFIDF classifier [16] that is one of the most widely used statistical single-term indexing approaches for document categorization.

The rest of this paper is organized as follows. Section 2 describes TFIDF classifiers and their problems. In section 3. we explain the process for building a KAN and applying it to document categorization. Experimental results are presented in section 4 to support our claim. Section 5 concludes and presents future work.

2 TFIDF classifier: A statistical single-term indexing approach

Static document categorization is the problem of assigning predefined categories to the test documents [21]. By contrast, dynamic document categorization, also called clustering, is the problem of classifying and categorizing a set of documents by grouping them [5, 12]. In this paper, we focus on the area of static document categorization.

A number of statistical learning methods have been applied to this problem in recent years. There are, in common, three main issues in applying algorithms to the document categorization problem: (1) What representation method should be used (2) How the large number of words dealt with in a representation and (3) Which algorithm is used. In the following sub-section, we look at TFIDF classifiers in terms of these issues and the problems these have for each aspect.

2.1 Document representation and the high **dimentionality** problem

To apply a learning algorithm, the first step is to transform text documents into a representation that is suitable for the algorithm to perform the categorization task. Information retrieval research suggests that words work well as representation units and that their position in a document is not very important for performance [9]. This leads to the bag-of-words representation that is widely used for document representation. However, when dealing with semi-structured documents (web documents and news articles), there is some work that uses additional information such as position, tag, or hyperlinks [4, 6]. In the bag-of-words representation, each distinct word has its frequency in a given document as a weight.

The problem in this representation is that it leads to high-dimensional word spaces. Frequently used approaches to reduce the number of different words are to use a 'stop-list' containing common words and apply a language specific stemming algorithm. Through pruning the infrequent and/or very frequent words, the size of the word space can be further reduced. However, it is noted in [9] that many irrelevant words that exist even after applying above approaches cause overfitting in measuring similarities between test documents and predefined categories. So, there is a need for a new representation method to further reduce the number of unimportant words or their influence in calculating similarities and. as a result, to improve the accuracy of document categorization.

2.2 TFIDF classifier

This classifier is based on the Rocchio relevance feedback algorithm [16]. Its major heuristic is the TFIDF word weighting scheme. Due to its various heuristic components, there are a number of similar algorithms corresponding to the particular choice of these heuristics. This sub-section describes two heuristics, the word weighting scheme and the similarity measuring method.

A TFIDF classifier builds on the following representation of documents, called the vector space model [18]. Each document D is represented as a vector $d = (v_1, v_2, \ldots, v_n)$. Here, each $v_i(d)$ is the weighting value of /th word in document D and it is calculated as a combination of two weighting schemes, TF(i, D) and IDF(r'). The term frequency TF(i, D) is the number of times the ith word occurs in document D and the inverse document frequency, IDF(z'), can be calculated as follows:

$$IDF(7) = \log[N/DF(/)]$$

- DF(i) is the number of documents in which /th word occurs at least once.
- N is the total number of documents.

$$pv_c = [\alpha \times (n_c)^{-1} \times \sum_{d \in c} d] - [\beta \times (N-n_c)^{-1} \times \sum_{d \notin c} d]$$

- α and β are adjustive parameters for positive and negative examples.
- n_c is the number of documents in the category C and N is the total number of documents in the training set of documents.
- d is the vector for document D.

Set of distinct words: W = {apple, windows, computer. web, www. file}

Set of documents (D)
d_1 apple, windows, computer
d_2 windows, computer
d_3 apple, computer, web, www
d_4 file, web, www

$\{F_1\}$

Word set	Support
apple	2
windows	2
computer	3
web	2
www	2

$\{CF_2\}$

Word set	Support
apple, windows	1
apple, computer	2
apple, web	1
apple, www	1
windows,computer	2
windows, web	0
windows, www	0
computer, web	1
computer, www	1
web. www	2

$\{F_2\}$

Word set	Support
apple, computer	2
windows, computer	2
web, www	2

Figure1: Example.

Then, the weight of /th word, $v_i(d)$, in document D is TF(/, D) x IDF(/). This weighting scheme says that a word is an important indexing term for document D if it occurs frequently. On the other hand, a word which occurs in many documents is rated as a less important indexing term due to its low inverse document frequency. Because document lengths may vary widely, a length normalization factor is applied to the term weighting function. A weighting equation that is used in this experiment is given in [23].

$$v_i(d) = \frac{[\log(TF(i, D)+1.0) \times IDF(i)]}{\sqrt{\sum_{i=1,n} [\log(TF(/, D)+1.0) \times IDF(/)]^2}}$$

A prototype vector pv_c for a category C is prepared by summing the vectors of the positive documents as well as those of the negative documents and, then calculating a weighted difference for each.

The similarity value between a category and a new document is obtained as the summation of the inner products between corresponding term vectors.

The problem in this classifier is that it is very sensitive to the number of irrelevant words, because all the words participate equally in the calculation of similarity. When the discriminating words are only a small subset of the whole word set, even two documents with identical values on these discriminative words may not be considered as near neighbors. Other statistical single-term based classifiers, such as k-NN classifier [13], have the same problem.

3 Keyword association network for document categorization

The reason for proposing a new indexing scheme is to give solutions for the following two important

F_1: Frequent 1-word sets; CF_2: Candidate 2-word sets:
F_2: Frequent 2-word sets

for all $d \in D$ do
  for all $w \in W$ do
    if w_i exists in d
      vr,.count++;
F_1 = { vr. e W | w_i.count>= minimum support}

Compute uncommon words which satisfy user specified frequency (minimum support).

when F_1 = {vr,, w_2, ... w_k}
CF_2 = 0;
for all w e F_1 do
  for (/= 1; / <k; /++) do
    CF_2 = CF_2 + {(w_i, w_{i+1}),(w_i, w_{i+2}),.... (vr,, w_k)}

Generate word pairs for uncommon words.

for all $d \in D$ do
  for all $cf \in CF_2$
    if two words in cf exists in d
      cf_i.count++;
F_2 = { cfe CF_2 | cf_i.count >= minimum support J

Create list of word pairs which satisfy required frequency.

Figure 2: Algorithm for generating frequent 2-word sets.

objectives: (1) to give a word an appropriate weight according to its semantic meaning in a given document (2) to remove the influence of unimportant words which are misleading or irrelevant to categorization. For example, a word can have several meanings, and its meaning might be identified by considering other words in a document. This is why we use a confidence value when calculating term weight. Also, large numbers of irrelevant words are the underlying cause of imprecise document categorization. In the following sub-sections, we represent KAN - a new multi-term indexing scheme, and explain the process of applying it to document categorization by giving an example.

3.1 Multi-term indexing scheme: KAN

Previous work showed that it is possible to automatically find words that are semantically similar to a given word based on the collocation of words [17, 20]. KAN is constructed based on this statistical method. The degree of relationship between two words is represented by a confidence value. This measure was used in finding association rules [1] that have recently been identified as an important tool for knowledge discovery in huge transactional databases. In KAN, the confidence value is used for measuring how the presence of one word in a given document may influence the presence of another.

Let us assume that there is a set of n unique terms $\{W = (w_1, w_2, \ldots, w_n)\}$ and a set of m documents $\{D = (d_1, d_2, \ldots, d_m)\}$Here, each $d_i = (w_1, w_2, \ldots, w_k)$ is a non-empty subset of W. The construction of KAN is based on support and confidence values, defined as follows.

Definition 1

The support of the term vr, - $SUP(w_i)$ - is the number of documents that contain w_i.

Definition 2

The confidence value of vr, to w_j - $CONF(w_i w_j)$ - is the percentage of documents which contain vr, and also have w_j, i.e., $SUP(w_i, vr_j) / SUP(w_i)$.

The problem of building the network is to find two terms that satisfy a user-specified minimum support and confidence. High confidence of w_i to w_j is interpreted as indicating that the semantic meaning of vr, can be identified with the existence of term w_j.

In Figure 1, we have the set of documents from a certain category, information technology. Through the usual preprocessing steps of stemming and use of a stoplist as mentioned in Section 2.1, we can get the set of distinct words that are considered informative words. Note that the term 'file' occurs in just one document and so it is not considered at this point. With the given set of documents (D), set of unique words (W), and user specified minimum support (in this example, it is 2), the algorithm in Figure 2 finds the frequent 2-word sets, F_2, which are groups of two words occurring frequently together in the set of

documents and satisfying the given minimum support. CF_2 is the candidate 2-word sets generated from the frequent 1-word sets, F_1. Figure 3 shows the network for above example. In the calculation of similarity between this category and a new document, if the document has both the words apple and computer they

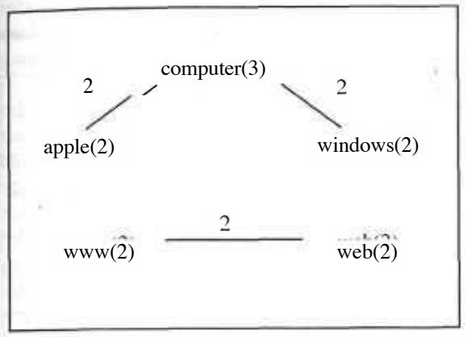

Figure 3: KAN for the example.

are considered as informative words. And, their confidence values are used for increasing their term weights in the similarity computation.

3.2 Applying KAN to document categorization

The most important factor in increasing the accuracy of categorization is to remove the influence of the large number of irrelevant terms that occur evenly across the categories. Those terms should be connected with many other tenns in the network. This means their confidence values are quite low compared to the confidence values of the important terms; they will significantly degrade the overall accuracy of categorization. So, we need to restrict the number of the participating tenns in the similarity measure through filtering out these irrelevant terms. To solve this problem, we use the inverse category frequency (ICF).

Definition 3

The inverse category frequency of term vr,, $ICF(w_i)$ is:

$$ICF(w_i) = [\log (C / c_i) / \log C] + 5$$

where c_i is the number of categories in which w_i appears in the KAN, C is the total number of categories, and 5 is a parameter that adjusts the impact of ICF values to term frequencies.

In a network, support of each term is replaced with the weighted support (WS), and it is used in computing the modified confinece value ($MCONF$).

Definition 4

The weighted support of term w_i, $WS(w_i)$, is:

$$WS(w_i) = SUP(w_i) / ICF(w_i)$$

Definition 5

The modified confidence value of w_i to w_j, $MCONF(w_i w_j)$, is:

$$MCONF(w_i w_j) = SUP(w_i, vr_j) / WS(w_j).$$

If a term appears only in one network, its ICF value will be 1 plus the adjusting parameter value (8). So. its weighted support (WS) in a network will be smaller than its original term frequency (TF) and. as a result, its confidence values to other terms in a network become greater than the original confidence values. In the other extreme situation, consider a term that appears in all given networks. Its ICF value will be same as the adjusting value since its numerator, log (C / c_i), in the definition 3 will be zero. So. its WS becomes much greater than its term frequency and its modified confidence values to other terms become much smaller. In this way, we can greatly decrease the impact of irrelevant terms to the similarity calculation.

In the similarity computation between a category and a new document, the weight (TFIDF) of each term in the category is increased by the value of multiplying its original term weight by its total confidence value. The total confidence value for each term is obtained by finding all the links from other terms in the KAN. summing up all the modified confidence values which are greater than the threshold. Thus, the similarity value between a document D and a category C. which are represented by the vectors of the form $(d_1, d_2, \ldots, d_k)$ and $(c_1, c_2, \ldots, c_k)$respectively, is computed as follows:

$$Sim(D, C) = \sum_{i=1,k} \{[c_i + (c_i \times \sum_{i=1,k,k\neq i} MCONF(w_i w_j))1 \times d_i\}$$

d_i: the weight of /th word in the vector of document D.

c_i: the weight of /th word in the vector of category C.

vr,.. vr: the ith and jth words in document D.

4 Experiments

Experiments were conducted to find out how well KAN performs in document categorization. The performance of KAN is compared with a TFIDF classifier in the following data set.

Category	Number of Training data	Number of Test data
earn	2709	1066
acq	1488	722
money-fx	460	222
grain	394	179
crude	349	215
trade	337	177
interest	289	133
wheat	198	89
ship	191	103
corn	159	63

Table 1: Number of training and test data in each category.

4.1 Charactreristics of data set

The data set is the Reuters-21578 text categorization test collection Distribution 1.0. In this collection, each article does not contain any meta-information (hyperlinks or tags) that could be used for extracting more important terms. Instead of analyzing all 135 categories, we select the ten most frequent categories and split the articles into training and test set according to the Modified Lewis Split as shown in Table I. For the preprocessing steps, we applied a stop-list and Porter's stemming algorithm [14] to the articles. Then we take terms that appear more than twice in a document as informative tenns.

4.2 Experimental results

Table 2 shows the accuracy of the two classifiers in each category. To build KAN for each category, the larger number between the integer number of 2% of training data and 5 was set as minimum support and minimum confidence = 50 was used for all categories. Also, when computing the ICF value, the adjusting parameter (5) was set to 3 for terms which appear in only one category and 0.05 for other terms. Comparing KAN and TFIDF classifiers, KAN tends to work better in all categories. The accuracy values of the two classifiers are much lower than expected in grain and this is due to the fact that grain has similar characteristics with corn and wheat.

The accuracy in each category seems to be affected by the existence of similar categories that have many keywords in common. So, the most important task in document categorization is to find out the informative and unique terms in each category and give high weights to them. Our multi-term indexing method, KAN and its new weighting scheme achieve comparable performance over TFIDF in this unfavorable experimental situation.

Category	TFIDF(%)	KAN(%)
earn	91.5	96.2
acq	67.7	84.6
money-fx	73.8	81.1
grain	42.5	50.3
crude	76.9	79.1
trade	90.8	98.3
interest	68.2	85.7
wheat	68.6	96.0
ship	60.2	74.8
corn	90.0	95.0

Table 2: Accuracy of classifiers in each category.

5 Conclusion and future work

In most single-term indexing methods, terms are indexed syntactically and the frequency is the only data showing their importance in the documents. It has been noted that their lack of ability to capture the semantic meanings of terms has reduced their performance in textual information systems.

In a given document, the exact meanings of some words can only be identified in relation to other words. Also, their meanings tend to change in accordance with other co-occuning words. KAN is a multi-term indexing method and designed to capture a certain level of the semantic meanings of words.

In this paper, this new multi-term indexing scheme called KAN was applied to document categorization. The expected advantages of applying KAN in the task of document categorization are: (1) unlike single-term indexing methods, it becomes possible to give weights to terms according to their semantic meanings in a given document. This is achieved by using the confidence value - $CONF(w_i w_j)$ - between two terms, and (2) the influence of a huge number of unimportant terms can be further removed by increasing their supports and is handled by dividing supports with their inverse category frequency *(ICF)* values, i.e. the use of $MCONF(w_i w_j)$. Our experiments indicate that our approach is more successful than a typical statistical single-term classifier (TFIDF).

In future work, KAN will be evaluated to establish how quickly its accuracy increases. It is also planned to build KAN at the sentence or paragraph level. When the document is quite long, it seems to be desirable to consider the sub-section of a document as a basis unit for building KAN. Through the experiments, it was identified that the perfomance of KAN is very sensitive to the parameters, such as 8, the minimum support, and the minimum confidence. The more research will be done to determine the optimum parameter values automatically in each category.

We have also noted that the application of KAN to other areas could be promising. For example, in textual recommendation systems, each user's profile built by KAN will represent her/his interests more accurately. Also, unlike single-term indexing methods KAN could be used to expand user's query in information retrieval systems. The KAN's usefulness in the selection of important words and its relationship information among keywords make it possible for the retrieval systems to suggest other keywords that are highly related to the user's input query.

References

[1] A. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast Discovery of Association Rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith. R. Uthurusamy, editors. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, pages 307-328, 1996.

[2] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, Vol. 37, No. 4, pages 573-595, 1995.

[3] L.D. Baker and A.K. McCallum. Distributed Clustering of Words for Text Classification. ACM SIGIR98, 1998.

[4] W.W. Cohen, Learning to Classify English Text with ILP Methods. Workshop on Inductive Logic Programming, Leuven. September, 1995.

[5] P. Cheeseman and J. Stutz. Baysian Classification (Autoclass): Theory and Results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, R. Uthurusamy, editors. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, pages 153-180, 1996.

[6] J. Furnkranz. Exploiting Structural Information for Text Classification on the WWW. In D. J. Hand, J. N. Kok, M. R. Berthold (eds.), Advances in Intelligent Data Analysis: Proceedings of the 3rd Symposium(IDA-99), Amsterdam, Netherlands. Lecture Notes in Computer Science 1642, Springer-Verlag, pages 487-497, 1999.

[7] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasnagottu. Information Retrieval on the World Wide Web. IEEE Internet Computing, pages 58-68, September-October 1997.

[8] T. Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In Proceedings of the 14th International Conference on Machine Learning ICML'97, pages 143-151, 1997.

[9] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Machine Learning: ECML-98, Tenth European Conference on Machine Learning, pages 137-142, 1998.

[10] D. Lewis and M. Ringuette. A Comparison of two learning algorithms for text categorization. In Third Annual Symposium on Document Analysis and Information Retrieval, pages 81-93, 1994.

[11] D.D. Lewis, R.E. Schapire, J.P. Callan. and R. Papka. Training Algorithms for Linear Text Classifiers. In SIGIR 96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 298-306, 1996.

[12] J. Moore. E. Han. D. Boley, M. Gini, R. Gross, K. Hastings. G. Karypis, V. Kumar, and B. Mobasher. Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering. In 7th Workshop on Information Technologies and Systems. Dec. 1997.

[13] S. Okamoto and K. Satoh. An Average-Case Analysis of k-Nearest Neighbor Classifier. In Proceedings of the First International Conference on Case-Based Reasoning, pages 243-264, 1995.

[14] M.F. Porter. An Algorithm for Suffix Stripping. Program, Vol. 14. No. 3, pages 130-137, July 1980.

[15] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[16] J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall Inc., 1971.

[17] G. Ruge. Experiments on Linguistically Based Term Associations. Information Processing & Management, 28(3), pages 317-332, 1992.

[18] G. Salton. Developments in Automatic Text Retrieval. Science, Vol. 253, pages 974-979, 1991.

[19] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, No. 5, pages 513-523, 1998.

[20] S. Sekine, J. Carroll, A. Ananiadou, and J. Tsujii. Automatic Learning for Semantic Collocation. Proceedings of the Third Conference on Applied Natural Language Processing, ACL. pages 104-110, 1992.

[21] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval, Vol. 1, No. 1/2, pages 67-68, 1999.

[22] T. Yavuz and A. Guvenir. Application of k-Nearest Neighbor on Feature Projections Classifier to Text Categorization. In Proceedings of the 13th International Symposium on Computer and Information Sciences - ISCIS'98, U. Gudukbay, T. Dayar, A. Gursoy, E. Gelenbe (eds.), Antalya, Turkey, pages 135-142, Oct. 26-28, 1998.

[23] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic Query Expansion Using SMART: TREC 3. The Third Text Retrieval Conference (TREC-3). National Institute of Standards and Technology Special Publication 500-207, Gaithersburg, MD, 1995.

The TREATS Approach to Reuse of Tables in Plain Text Documents.

L.E.Hodge[1], N.J. Fiddian and W.A.Gray

Department of Computer Science,
Cardiff University,
Cardiff, UK.

{scmleh|njf|wag}@cs.cf.ac.uk

Abstract.

In this paper we present the table processing approach employed by the TREATS (Table Recognition, Extraction, Analysis and Tranformation System) software toolkit.

In order to support the large variety of table layouts that appear in plain text documents, our system aims to identify the layout of cells that exist within a table and to tailor processing accordingly. This results in more effective processing than preexisting approaches that apply one general technique to all types of table.

The classification process is the key to processing and exploits a cellular automaton (CA) based approach to the identification of cell structure within a table. The input to the CA is a simple representation of the content of the source table. This is evolved via the application of intelligent transformation rules, resulting in a representation of the cell structure that exists within the table. Based on the combination of cell types that appear in this representation, the layout of the table can be determined and appropriate processing can be performed. During this processing, the content of the source table is transformed into a relational form suitable for reuse in other applications.

Keywords: Information retrieval, resource discovery.

1. Introduction.

The TREATS (Table Recognition, Extraction, Analysis and Transfonnation System) software toolkit [1,2] offers support for reuse of tabular content from a wide variety of source documents[2]. Of the supported types, tables in plain text form are by far the most difficult to reuse. Whilst other document types such as HTML and Latex contain table definitions that enable their content to be extracted via a parsing approach, with plain text the only guide to the structure of the table (the key to correct reuse) is the layout of its content. Combined with the variety of table layouts that are possible, this makes tables in plain text documents difficult to effectively and conectly reuse.

Although methods for the processing of such tables exist, we feel that in general a specific approach can only support reuse of a limited subset of the possible table layouts. In this paper, we describe the approach used by the TREATS toolkit that provides more wide ranging support by identifying and classifying table layout such that processing can be tailored accordingly.

During this discussion, we concentrate on processing the content of a table i.e. everything below the level of the column labels[3]. Whilst these labels are important to the understanding of table content and will need to be extracted and in some cases transformed[4] to enable reuse, they do not offer any indication of the cell structure of the table.

The remainder of a table is concerned with the presentation of data and consists of two components, the stub and the body [3]. In general, the stub is the leftmost column of a table and may contain either a simple column of entries or in some cases where grouping exists, nested entries that indicate the hierarchy within the data. This hierarchy may also be indicated through the use of spanning cells and where this occurs, we extend the definition of the stub to cover all columns that contain spanning cells involved in defining the hierarchy. Spanning and nested cells are discussed in more detail in section 3.1. The body

[1] With support from EPSRC and BT (Case Studentship).
[2] Currently, plain text, HTML and Latex are supported.
[3] The area where labels are displayed is often known as the boxhead.
[4] Processing of label structures is addressed elsewhere [1].