FUZZY ADAPTIVE CLUSTERING AND CLASSIFICATION FOR BROWSABLE DOCUMENT DIRECTORIES

NEAL RICHTER
RightNow Technologies,
Bozeman, Montana
nealr@rightnow.com

ZUZANA GEDEON
RightNow Technologies,
Bozeman, Montana
zgedeon@rightnow.com

STEPHEN DURBIN
RightNow Technologies,
Bozeman, Montana
sdurbin@rightnow.com

DOUG WARNER
RightNow Technologies,
Bozeman, Montana
doug@rightnow.com

BIKRAMJIT BANERJEE
Tulane University
New Orleans, Louisiana
banerjee@eecs.tulane.edu

ABSTRACT
We describe an algorithm used to automatically create a hierarchy of documents. This hierarchy is quickly regenerated and allows for dynamic changes in a document database. We use natural language processing, text clustering with fuzzy adaptive parameter techniques, classification rule extraction and summarization to create and maintain this hierarchy.

INTRODUCTION

Built around the core document knowledge base, RightNow™ eService Center (RNTeSC) uses a variety of AI techniques to facilitate the construction, maintenance, and navigation of the knowledge base (Warner et al., 2001, Durbin et al., 2002). These include collaborative filtering, swarm intelligence, natural language processing, text clustering and classification rule learning. RNTeSC is an integrated application that combines email management, web self service center, live collaborative chat, and knowledge management. From an AI perspective, the core of the application is the document knowledge base and the tools by which it is created, maintained and accessed. In this paper we will focus on a specific feature of the enduser portion of the application.

For many of our customers the Answer knowledge base is highly dynamic and may encompass 100's to many 1000's of documents. We term the knowledge base organic, because of the natural way it is seeded and evolves over time (Warner et al., 2001). Our global aim was to construct a simple and robust framework requiring minimal human resources. Our goal is to use appropriate AI methods to automate the development and maintainance of the knowledge base, lifting the organizational burden and workload from the expert knowledge providers.

Our 1100+ cusomers provide customer service knowledgebases to 10+ million visitors per month. These visitors are usually coming to a company's customer service site with a directed goal of getting help with a product or service. The end-user interface to the knowledge base contains a searchable list of documents ranked by a time adaptive visitation metric.

We recognized that a significant percentage of visitors may prefer an alternate inteface to the document database. They may not be able to express their question in the same terms or even a appropriate level of detail to allow them to enter an effective search query. To address this problem we wanted to design a system to autmatically render the documents into a browsable directory in the style of the Yahoo WWW directory.

In this paper we introduce the basic system in more technical detail than in previous publications. The system incorporates natural language feature selection, heriachical text clustering and classification techniques in order to achieve complete automation of this process. Note that this system is a completely unsupervised method of machine learning.

We create our hierarchy using a simplfied version of the fast hierarchical clustering algorithm BIRCH (Zhang, 1996). We add to BIRCH a fuzzy adaptive layer that changes the variety of clustering parameters to achieve optimal document trees. Later, we use the cluster vectors to create classification rules similar to the rules produced by RIPPER (Cohen, 1995). We also use the cluster vectors to form human readable linguistic summaries for the branches of the document tree. See Figure 1 for a block diagram of the system.

Recognizing the inherent multiplicity and subjectivity of document similarity relationships, we classify documents into multiple clusters. This makes use of the directory much more convenient, as the user can locate a document along various paths, and does not have to guess what rigid classification might control the listing The extracted classification rules also allow for dynamic updating of the directory by classifing new documents into appropriate clusters.

Although we allow for modification of the directory tree with the inclusion of new documents and the removal of old documents, after many changes to the knowledge base the original document tree may become obsolete. To alleviate this problem we detect high levels of document churn and automatically rerun the system. High document churn can also signal the addition of new knowledge into the system, so a recluster becomes more necessary the higher the churn factor.

The final tree is presented to the user in a format similar to tree-based directory navigation tools available in most modern operating system GUIs, with an expandable tree on the left, and the current directory on the right. Figure 2 is a screen shot of the document tree containing information from the United States Social Security Administration knowledgebase.

FEATURE SELECTION

The natural languge feature selection algorithm is a crucial component of the system. It allows us to vastly reduce the total space necessary to store the documents in memory and increase the speed of the clustering algorithm.

The document is first tagged with the BRILL POS tagger (Brill, 1995). We then identify noun phrases and other interesting liguistic features of the document. The words and phrases are then stored in a stemmed format via the Porter stemming algorithm (Porter, 1980). Each feature has a weight according to the type of feature and the occurance frequency in the document. Documents are represented by sparse vectors, common in information retrieval, using weighted keyword phrases as features.

Our software allows for explicit creation of the various limited hierarchies using ‘product/subproduct’ and ‘category/subcategory’ labels for each document. Knowledge base administrators can also enter keywords that are associated with the document for searching purposes. When these additional fields are present, we use them as highly weighted features for each document.

The RNTeSC keeps track of recent seach terms used by end-users. We use these term lists in the final step of feature selection to reinforce particular features. This allows the biasing of the feature weights according to what terms visitors are explicitly searching for.

ADAPTIVE CLUSTERING

We developed a simplified version of the BIRCH hierarchical clustering algorithm. At present the algorithm is quite different from the original. It uses different distance metrics, a slightly different in-memory tree representation and does not include any of the memory-disk swapping features of BIRCH.

The intent of the system is to produce an easily browsable document tree. This means that a balance is desired between the depth and width of the tree. During development we quickly realized that there was no way to pick a static set of parameters for the BIRCH algorithm (or most other clustering algorithms) that will perform well for a braod variety of document sets. At a given threshold and maximum branching factor, some document databases looked good, while others were either narrow and deep, or too wide and shallow. A narrow-deep tree requires the user to follow too many branches, an excessively wide tree presents too much information at each branch resulting in a cluttered appearance.

Here are the equations describing document and vector representations, vector norms and the distance metric and standard L_2 norms. Equation (1) refers to a document's sparse vector and assocated norm. Similarly, Eq. (2) refers to a cluster's average (sparse) vector and norm. Equation (3) is the inverted standard cosine distance.

$$D = \left\langle w_{D_i}(p_1), w_{D_i}(p_2), \ldots, w_{D_i}(p_N) \right\rangle \qquad ||D|| = \sqrt{\sum_{p_i} w_{D_i}(p_i)^2} \quad (1)$$

$$C = \left\langle w_{C_i}(p_1), w_{C_i}(p_2), \ldots, w_{C_i}(p_N) \right\rangle \qquad ||C|| = \sqrt{\sum_{p_i} \left(\frac{w_C(p_i)}{N_C(p_i)} \right)^2} \quad (2)$$

$$d(C,D) = \frac{1}{dist_{\cos}(C,D)} \quad (3)$$

FUZZY CONTROLLER

We designed an adaptive fuzzy parameter controller to adjust the distance threshold for clustering. First a heuristic function was used to estimate the branching factor and maximum tree depth given a number of documents. We bias this estimate prefering a slightly wider tree over a true balance of width and depth. When using highly varied document sources, a good initial estimate of the threshold value is difficult to get. Thus we start with a relatively high threshold value.

Using the initial parameters, the first tree is created and stored in main memory. The remander of the clustering iterations are handled by the fuzzy controller. Here is an outline of the process:

```
While (stopping criterion is not met) {
    Collect document tree metrics
    Adjust threshold with fuzzy rule system
    Recluster documents with new parameters
}
```

We use the following metrics as input to our fuzzy parameter controller:

andc = average number of documents per cluster
ateff = average threshold efficiency
aabf = average actual branching factor
avcav = average variance of the cluster vectors

The **andc**, **avcav**, & **ateff** are designed to judge the suitability of the current threshold, while **aabf** is intended to judge the maximum branching factor's correctness. Each metric is fuzzified with a five way membership function, VLOW to VHIGH. Each fuzzy rule casts a vote for an action, and at the end of each rule processing step, the action is taken according to the voting outcome. This is in contrast to standard defuzzification techniques, and was simpler to implement. Here is a sample fuzzy rule that uses **ateff** to vote for an increase or decrease in the next iteration's threshold.

$$\text{threshold votes} = \left\{ \begin{bmatrix} \mathrel{+}= -2 \\ \mathrel{+}= -1 \\ \mathrel{+}= 0 \\ \mathrel{+}= 1 \\ \mathrel{+}= 2 \end{bmatrix} \right\} \text{ if ateff} = \left\{ \begin{bmatrix} \text{VLOW} \\ \text{LOW} \\ \text{MED} \\ \text{HIGH} \\ \text{VHIGH} \end{bmatrix} \right\}$$

Writing a functionally complete rule set that will appropriately move the threshold in all cases turned out to be a hard task. Thus, we augmented the fuzzy controller's parameter recomendations with a limited threshold scanning loop. In the first phase, we start the threshold high and move it low in exponential steps searching for a high degree of votes. In the second phase the threshold is moved in smaller steps using the controller's votes. If the controller returns no votes for threshold change, we stop. Otherwise we stop after a maximum number of iterations.

CLASSIFICATION

After the final clustering iteration completes, we extract classification rules from the tree. This will allow us to add new documents to the tree without needing to recluster. The extracted rules are very similar to the types of rules that the RIPPER algorithm learns. It is important to note that this rule extraction step does not do any learning of its own, it merely forms conjunctive rules based on the cluster vectors in the tree.

We use only the top N phrases for each cluster. To ensure a proper hierarchy we need to employ shrinkage: adding parent rules to the rules of all the children with weight proportionally smaller than that of the parent. This will ensure that documents contained in a cluster sharing common parent still share parent's features and enforce a more correct hierarchy in subsequent classification. In addition we do some elimination of common, non-distriguishing features shared by peer clusters.

For the last step of the clustering and classification phase, we recreate the hierarchy using the newly created classification rules. This step slightly redefines the shape of original tree and ensures that the cluster descriptions are relevant and don't contain incorrectly clustered documents. It also allows documents to be inserted into multiple areas of the tree, giving the user more opportunities to find the document.

Finally, we define our branch and cluster summaries. The summaries consist of a set of unstemmed words and phrases with (hopefully) some descriptive meaning to the visitors. Using the cluster vector and the phrases produced by the feature selection step, we generate cluster descriptions, avoiding word repetitions. The summariztion system is biased towards prefering two-word phrases, it will use highly weighted single words if that does not create redundancies.

Reclustering is occasionaly neccesary due to document changes, removes and additions. The knowledge base administrator sets a document churn parameter at 10%, 20% or 30%, telling the algorithm when to reprocess the documents into a directory.

ALTERNATIVE USE OF THE HIERARCHY

RNTeSC includes options for automatic integration of email management and the web self service center. To possibly eliminate the need for customer service representatives to answer all questions sent through an e-mail gateway we employ an e-mail filter we call SmartAssistant. This automatic response manager attempts to find knowledge base documents relevant to the current e-mail and returns highly scored documents to the user as suggested reading.

Replies are generated through this module to match incoming e-mails with answers in the knowledge base. To improve the performance we incorporated a context test. The test filters the answers using the directory classification rules, making sure that question and answers both fall into the same area of the tree. This enforces that the overall topic of the question is the same as that of the answer.

CONCLUSIONS AND FUTURE WORK

During our work we identified several issues arising from using a combination of natural language feature selection, a BIRCH-like clustering algorithm and classification rules. First, we need to experiment and find a larger set of fuzzy rules to handle more situations. Ideally we would like to replace the threshold scanning method with a more sophisticated threshold searching algorithm to complement an expanded fuzzy controller.

In additon, we need to do a rigourous analysis of the algorithm against standard knowledge bases and algorithms. Note that the idea of using a fuzzy controller to adapt parameters for the clustering algorithm could be used more generally towards adapting the many parameters associated with common clustering algorithms.

We also have several ideas for improving the summarization step to produce more readable cluster sumamries.

Please see (Warner et al., 2001) and (Durbin et al., 2002) for more complete descriptions of the RightNow eService Center's capabilities and operation as well as other AI techniques used.

REFERENCES

(Brill, 1994) Brill, E., "Some advances in transformation-based part of speech tagging." *In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-1994),* pp. 722-727, 1994.

(Cohen, 1995) Cohen, W. H. "Fast effective rule induction." *Proceedings of the Twelfth International Conference in Machine Learning,* pp. 115-123. 1995.

(Warner et al., 2001) Warner, D., Richter, J. N., Durbin, S. D., and Banerjee, B., 2001, "Mining User Session Data to Facilitate User Interaction With a Customer Service Knowledge Base in RightNow Web." *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001),* pp. 467-472, 2001.

(Durbin et al., 2002) Durbin, S. D., Warner, D., Richter, J. N., and Gedeon, Z., "RightNow eService Center: Internet customer service using a self-learning knowledge base." *Proceedings of the Thirteenth Annual Conference of Innovative Applications of Artificial Intelligence (IAAI-2002),* pp. 815-821, 2002

(Porter, 1980) Porter M., 1980, An algorithm for suffix stripping. Program.

(Zhang, 1996) Zhang, T., Ramakrishnan, R., and Livny, M., "BIRCH: An Efficient Data Clustering Method for Very Large Databases." *In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD-1996),* pp. 103--114, Montreal, Canada, 1996.

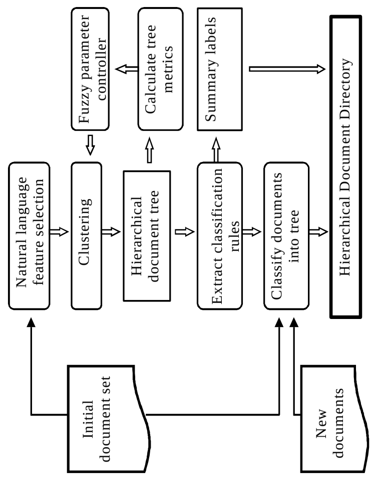
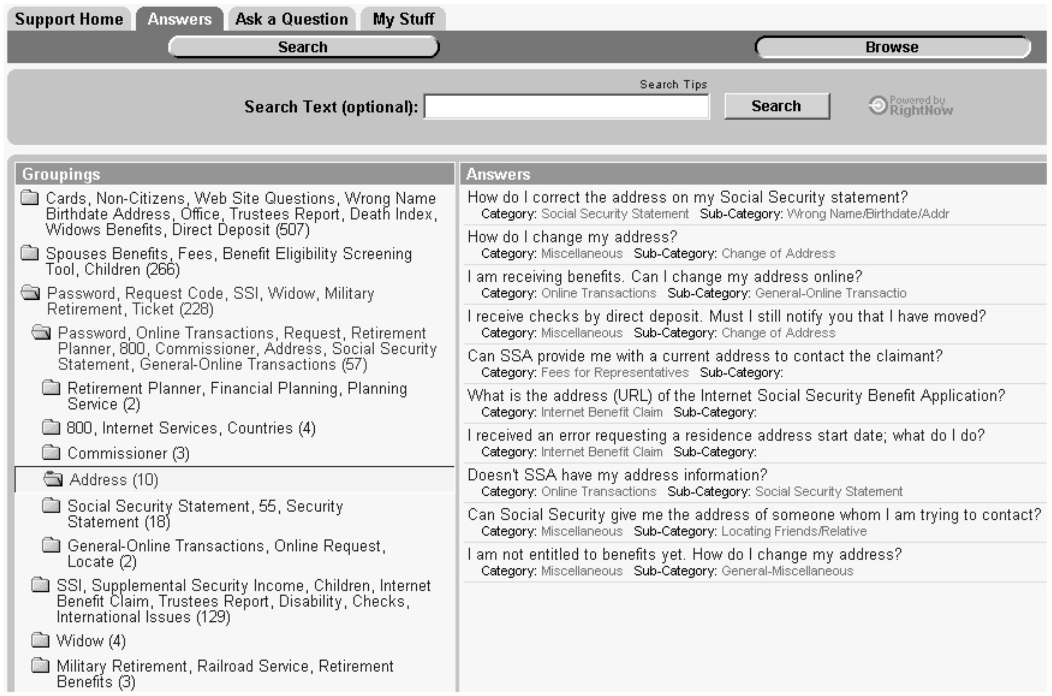

Natural language feature selection
Clustering
Fuzzy parameter controller
Hierarchical document tree
Calculate tree metrics
Extract classification rules
Summary labels
Classify documents into tree
Hierarchical Document Directory
Initial document set
New documents

Figure 1 System Diagram


Support Home | Answers | Ask a Question | My Stuff

Search | Browse

Search Text (optional): Search Tips Search

Groupings

- Cards, Non-Citizens, Web Site Questions, Wrong Name Birthdate Address, Office, Trustees Report, Death Index, Widows Benefits, Direct Deposit (507)
- Spouses Benefits, Fees, Benefit Eligibility Screening Tool, Children (266)
- Password, Request Code, SSI, Widow, Military Retirement, Ticket (228)
  - Password, Online Transactions, Request, Retirement Planner, 800, Commissioner, Address, Social Security Statement, General-Online Transactions (57)
    - Retirement Planner, Financial Planning, Planning Service (2)
    - 800, Internet Services, Countries (4)
    - Commissioner (3)
    - Address (10)
    - Social Security Statement, 55, Security Statement (18)
    - General-Online Transactions, Online Request, Locate (2)
  - SSI, Supplemental Security Income, Children, Internet Benefit Claim, Trustees Report, Disability, Checks, International Issues (129)
  - Widow (4)
  - Military Retirement, Railroad Service, Retirement Benefits (3)

Answers

How do I correct the address on my Social Security statement?
Category: Social Security Statement Sub-Category: Wrong Name/Birthdate/Addr

How do I change my address?
Category: Miscellaneous Sub-Category: Change of Address

I am receiving benefits. Can I change my address online?
Category: Online Transactions Sub-Category: General-Online Transactio

I receive checks by direct deposit. Must I still notify you that I have moved?
Category: Miscellaneous Sub-Category: Change of Address

Can SSA provide me with a current address to contact the claimant?
Category: Fees for Representatives Sub-Category:

What is the address (URL) of the Internet Social Security Benefit Application?
Category: Internet Benefit Claim Sub-Category:

I received an error requesting a residence address start date; what do I do?
Category: Internet Benefit Claim Sub-Category:

Doesn't SSA have my address information?
Category: Online Transactions Sub-Category: Social Security Statement

Can Social Security give me the address of someone whom I am trying to contact?
Category: Miscellaneous Sub-Category: Locating Friends/Relative

I am not entitled to benefits yet. How do I change my address?
Category: Miscellaneous Sub-Category: General-Miscellaneous

Figure 2 Document Directory Screenshot