

# An Interactive Automatic Document Classification Prototype<sup>†</sup>

Kirk Baker  
Collexis  
Bethesda, MD 20817  
baker@collexis.com

Archna Bhandari  
Office of Knowledge Management and  
Portfolio Analysis  
National Institutes of Health  
Bethesda, MD 20892

Rao Thotakura  
Division of Information Services  
National Institutes of Health  
Bethesda, MD 20817

<sup>†</sup>Disclaimer: This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any of its employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trade mark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government and shall not be used for advertising or product endorsement purposes.

## ABSTRACT

In this paper we report on a series of completed and ongoing experiments that involve the integration of fully automatic document classification techniques into an existing manually-oriented document retrieval system. We take our primary findings as positions on the design of an interactive document classifier and retrieval tool.

## General Terms

Design, Experimentation, Human Factors.

## Keywords

Information retrieval, machine learning, user interaction.

## 1. INTRODUCTION

In this paper we report on a series of completed and ongoing experiments that involve the integration of fully automatic document classification techniques into an existing manually-oriented document retrieval system. We take our primary findings as positions on the design of an interactive document classifier and retrieval tool.

The rest of this paper is split into three sections. Section 2 contains background information and a brief technical overview of a document classification system used to report funding allocations across a range of research categories. Section 3 outlines some of the formal system and user requirements that guided our integration of automatic classification techniques into an existing document classification system. In Section 4 we describe our positions on the design of an interactive document classification tool in terms of a functioning prototype that meets these requirements.

## 2. The Research, Condition and Disease Categorization (RCDC) Initiative

In 2006, the United States Congress mandated that the National Institutes of Health (NIH) establish a standardized, automated system for reporting its financial allocations to supported research

areas, conditions, and disease categories. The RCDC system is the implementation of this mandate.

It is implemented atop a vector space document retrieval model. Documents are preprocessed by a natural language processing module that extracts only those variations of term strings that correspond to concepts in the RCDC thesaurus. The RCDC thesaurus is a controlled medical ontology that draws from the Medical Subject Headings (MeSH)<sup>1</sup> and CRISP<sup>2</sup> databases, the National Cancer Institute thesaurus<sup>3</sup>, the UMLS Metathesaurus<sup>4</sup>, and Jablonski's Dictionary of Medical Acronyms and Abbreviations [1]. Documents are represented as weighted concept vectors where a frequency-based weighting scheme is applied to concept counts and then normalized such that all concept weights fall between 0 and 1.

Definitions of funding areas are also represented as weighted concept vectors along with a similarity threshold, where the set of concepts and weights are determined by subject matter experts and refined in conjunction with ongoing reviews of the set of documents that are retrieved. Concept weights in definitions of funding categories range from -1 to 1, or may be designated as mandatory or excluded. From an information retrieval standpoint, funding area definitions are treated as query vectors and a measure of similarity between each query and each document vector is calculated. When the computed similarity is above threshold for a given query and document, that document is classified as belonging to the given funding category.

The impetus for incorporating automatic document classification techniques into the RCDC system originated in response to the following specifications:

1. a need to alleviate the manual effort required in developing and maintaining category definitions;

---

<sup>1</sup> <http://www.nlm.nih.gov/mesh>

<sup>2</sup> [http://crisp.cit.nih.gov/crisp/CRISP\\_Help.help](http://crisp.cit.nih.gov/crisp/CRISP_Help.help)

<sup>3</sup> <http://ncit.nci.nih.gov>

<sup>4</sup> <http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html>

2. a desire to improve classification accuracy of selected existing categories;
3. the ability to support ad hoc category definition and retrieval in real-time.

In meeting these specifications, we were obligated to adhere to the conditions described in the following section.

### 3. Development of Automated Classification Techniques

Maintaining continuity with the existing RCDC system was a required condition for the integration of automated techniques into the grant categorization process. In terms of system requirements, this meant maintaining compatibility with the implemented vector space information retrieval model. Therefore, we restricted ourselves to linear classifiers, primarily linear support vector machines ([2], [3]), although experiments with linear perceptron show similar results ([4]).

Within the current RCDC application, the definitions of funding categories (i.e., the set of query terms and their ranking) are as important to subject matter experts as the set of retrieved documents. In practical terms, this means that not only should the output of a trained classifier be interpretable, it must be editable as well. Specifically, users requested the ability to:

1. limit the number of dimensions in the trained classification function;
2. delete dimensions that are intuitively irrelevant;
3. change the value of a dimension to match their intuitions about its relative importance;
4. add dimensions that were not part of the automatically generated classification function.

With regard to the documents retrieved or intended to be retrieved by a given query, users required the ability to indicate retrieval errors (false positives) and to augment the training data with externally labeled documents (false negatives) and incorporate these labels iteratively into the classifier's training procedure. In light of this requirement, we found that users should be given feedback about the inherent separability of the documents they are trying to classify. We found such feedback useful in tempering expectations of the performance of the automatic classifier and helping users understand why classification accuracy varies for different funding categories.

In the next section, we take the requirements above as our positions on the design of a document classification and retrieval tool that incorporates user interaction into the training procedure for an automatic classifier.

### 4. Positions on Designing an Interactive Automatic Document Classification System

In this section, we outline our positions on the design of an interactive document classification and retrieval tool that we prototyped for the RCDC project. Our positions are grounded in a series of experiments that measured classification accuracy for four categories – Lung Cancer, Breast Cancer, Prevention, and Orphan Drug. The data consisted of about 25,000 labeled grants

that were funded by NIH in 2008 (a subset of about 80,000 total applications funded for the year).

#### 4.1 Users must be able to edit the trained classification function.

In an interactive document classification system, the trained classification function must be interpretable by users. By default, linear classifiers like perceptron or SVM produce as output a weighted list of all (or nearly all, in the case of SVM) of the input dimensions. In our case, this vector typically contained around 20,000 dimensions, which is too many for a user to make sense of. Therefore, the first step we took in producing a human-interpretable classification function was to limit the number of dimensions over which the classifier operated. We evaluated several techniques for restricting the dimensionality of the training data and found that a simple, two-pass strategy worked as good as anything:

1. Train the classifier on the full-dimensional data set.
2. Remove all but the top- $n$  ranked dimensions in the output function from each item in the data set.
3. Retrain the classifier on the  $n$ -dimensional data set.

Happily, the optimal number of dimensions in our classification tasks turned out to be quite small (around 25-100 dimensions) relative to the dimensionality of the original data set. Another simple and effective way to restrict the dimensionality of the trained classification function is to remove low-ranked features from the data set before initially training the classifier (i.e., remove terms from a document vector if their weight is below some threshold).

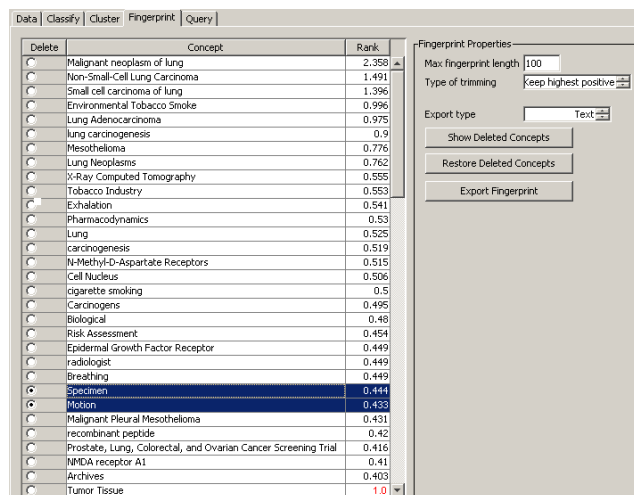


Figure 1. Screenshot illustrating functionality to edit the trained classification function by deleting features or changing their weights.

Even after limiting the number of features in the classification function, some items remain which are unacceptable to users' intuitions about whether they belong. For example, if a number of Lung Cancer grants originated from the N.C. Cancer Hospital, it is possible that a term like "North Carolina" would be heavily weighted in the resulting classification function. However, a user might feel that this term is inappropriate for a query intended to

define Lung Cancer grants in general. Therefore, our prototype allows users to indicate that particular features should be excluded from the training data.

Figure 1 shows a screenshot of a prototype application that illustrates this functionality. The primary view in Figure 1 contains a set of weighted features which represent the decision boundary for a sample document category. In this illustration, the two blue highlighted terms have been selected by the user for removal from the classification function. The highlighted features will be removed from the training data prior to retraining the classifier, and in essence become inactive for any future data points that may contain them. We found that allowing users to selectively remove unwanted features from the training data generally has very little impact on classification accuracy and occasionally improves it.

Sometimes users want to override automatically computed weights for a given feature, usually because they feel that it merits a higher weight than the classifier assigned it. The problem with attempting to manually adjust feature weights is that they are likely to change with the next training cycle. Our solution to this requirement has been to store user-modified concepts separately, train the classifier, and overwrite those user-modified concepts before the classification stage. In Figure 1, the weight of the last term shown (“Tumor Tissue”) has been overridden by the user to 1.0 and is depicted in red font.

Another way to allow users to interactively modify classifier output is to allow them to apply the classifier to an unlabeled or partially labeled document set. The documents that are returned can be designated by the user as positive or negative examples and incorporated into retraining the classifier (Figure 2).

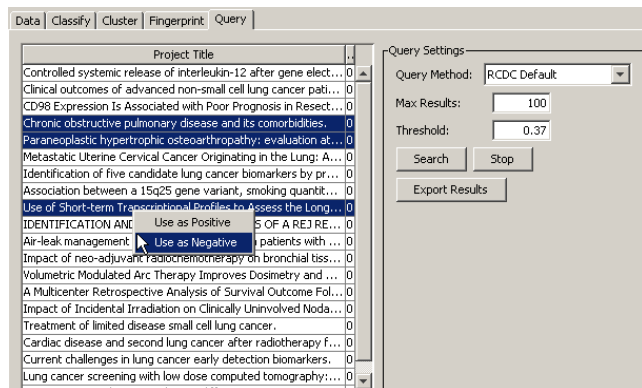


Figure 2. Using document relevance feedback to relearn the decision boundary.

## 4.2 Users must be given feedback on the inherent separability of their data set.

We sometimes observed users trying to distinguish document classes that are at best poorly separable over the given feature representation. For example, a category like Orphan Drug is defined axiomatically as research pertaining to any pharmaceutical agent used to treat a disease or condition that affects less than 200,000 people in the United States [5]. It is difficult to train a linear classifier to learn this distinction over document term vectors and produce an easily interpretable

classification function. In cases like this, users tend to get disgusted at the classifier’s poor performance and blame the computer.

We found it useful to guide users’ expectations of system performance by visualizing document similarities with a two-dimensional interactive cluster plot (Figure 3). When users see high overlap between positive and negative training samples, they understand that they’re asking the classifier to do something hard. For example, in the top plot in Figure 3, there is relatively poor separation of the sample document class (green squares labeled “Positives”) from the remainder of the document set (red squares labeled “Negatives”).

After the same data set has been restricted to 500 dimensions (from an original feature set of 7814 in this case), we observe less overlap of the two classes (bottom plot in Figure 3). The reduced feature set also corresponds to higher measures of classifier performance on a labeled test set. However, we found the type of visual feedback depicted in Figure 3 more effective than providing traditional evaluation measures like precision, recall, and F1 to users not accustomed to thinking in these terms.

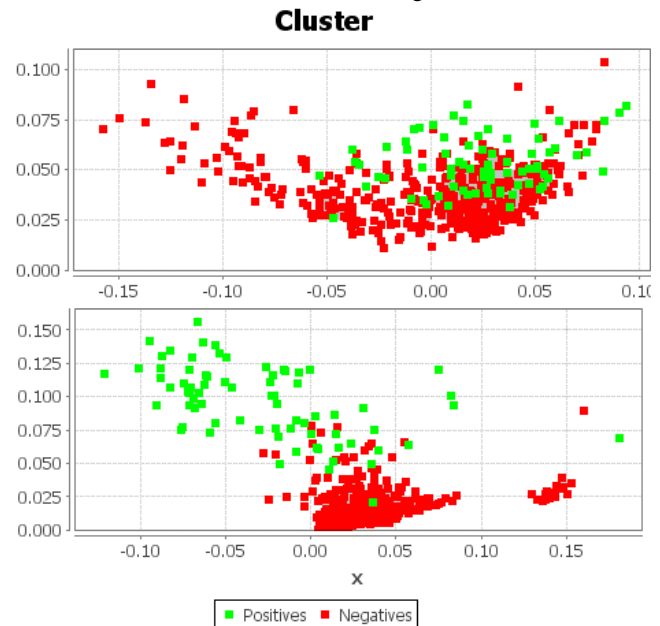
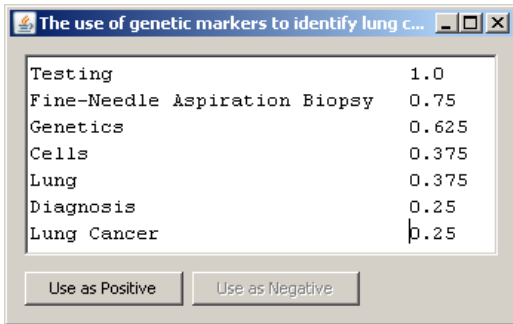


Figure 3. Document separation and number of features. In the top panel distances between documents are computed over the full set of 7814 features before applying dimensionality reduction necessary for two-dimensional visualization. In the bottom panel only 500 features were retained.

We also found it useful to allow users to interact with the classifier via the cluster plot (e.g., using the mouse to zoom or select individual data points). For example, after removing selected features from the training data, a user can examine the impact of this action by re-clustering the data and looking for relatively more or less class separation.

Alternatively, a user can select a document from the cluster plot to indicate whether it should be used as a positive or negative training point to retrain the classifier. Figure 4 shows a sample window that pops up when a data point is selected in the cluster

plot. From here the user can see information such as the document title and weighted features that are active in the document, and provide relevance feedback to the classifier prior to any additional training iterations.



**Figure 4. Sample document popup window when a data point is selected in the cluster plot.**

## 5. Acknowledgments

We are indebted to Reinder Verlinde who provided an extremely useful wrapper to libsvm and essentially kick-started this whole prototype.

## REFERENCES

- [1] Jablonski, S. 2008. Jablonski's Dictionary of Medical Acronyms and Abbreviations. 6th ed. Elsevier Health Sciences.
- [2] C.-W Hsu, C.-C. Chang, and C.-J. Lin. (2003). A practical guide to support vector classification. Technical report, Taipei.  
<http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- [3] Thorsten Joachims (2002). Learning to Classify Text using Support Vector Machines. Vol. 668 of Kluwer International Series in Engineering and Computer Science. Kluwer.
- [4] Calvin Johnson, William Lau, Archana Bhandari, and Timothy Hays. (2008). A Best-Fit Model for Concept Vectors in Biomedical Research Grants. AMIA 2008 Symposium Proceedings: 93.
- [5] RARE DISEASES ACT OF 2002. 107<sup>th</sup> Congress.  
[http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107\\_cong\\_public\\_laws&docid=f:publ280.107](http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ280.107)