2006

Improving text clustering for functional analysis of genes

Jing Ding
Iowa State University

Improving text clustering for functional analysis of genes

by

Jing Ding

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Computer Engineering; Bioinformatics and Computational Biology

Program of Study Committee:
Daniel Berleant, Co-major Professor
Eve Wurtele, Co-major Professor
Srinivas Aluru
Julie Dickerson
Sigurdur Olafsson

Iowa State University

Ames, Iowa

2006

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Jing Ding

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

For the Co-major Program

Signature was redacted for privacy.

For the Co-major Program

TABLE OF CONTENTS

Abstract

Continued rapid advancements in genomic, proteomic and metabolomic technologies demand computer-aided methods and tools to efficiently and timely process large amount of data, extract meaningful information, and interpret data into knowledge. While numerous algorithms and systems have been developed for information extraction (i.e. profiling analysis), biological interpretation still largely relies on biologists' domain knowledge, as well as collecting and analyzing functional information from various public databases. The goal of this project was to build a text clustering-based software system, called GeneNarrator, for functional analysis of genes (microarray experiments).

GeneNarrator automatically collected MEDLINE citations for a list of genes as the source of functional information. A two-step clustering approach was designed to process the citations. The first-step (text) clustering grouped the citations into hierarchical topics. The second-step (gene) clustering grouped the genes based on the similarities of their occurrences across the clusters resulting from step one. Hence, we planned to demonstrate how, instead of manually collecting and tediously sifting through potentially thousands of citations, biologists can be presented with dozens of topics as a summarization of the citations, and gene (groups) mapped to the topics.

In order to improve the first-step text clustering part of the system, several strategies were explored, including different vector space models (BOW-based or concept-based) for text representation, vector space dimensionality reduction (document frequency filtering), and multi-clustering. The most improvement came from multi-clustering. The clusterings were evaluated in terms of self-consistency and agreement with a manually constructed gold standard dataset using a newly proposed metric, normalized mutual information.

Chapter 1 Literature Review and Requirements Analysis

1.1 Functional analysis of microarray data

The rapid development of microarray technology has enabled biologists to simultaneously monitor the expression of hundreds or even thousands of genes in a single experiment. Automated data analysis methods and software tools are invaluable in aiding biologists to efficiently and timely process these large amounts of data. Numerous algorithms and systems have been developed for the "profiling" analysis, i.e. finding patterns in gene expression in response to environmental changes [57]. Interpreting the biological meaning of the patterns, however, still mainly relies on human experts' domain knowledge, as well as on finding previously reported information from literature and/or various public databases. While an expert's domain knowledge or manual collection from existing data may be sufficient for small data sets, it is unrealistic to expect someone to memorize functional properties of thousands of genes. Manually collecting, reading and summarizing them from the literature and/or public databases is tedious and time-consuming. Therefore, computer-aided functional analysis methods and tools are highly desirable.

There are a number of such systems that have been reported in the literature. In terms of the sources of functional information they rely on, they can be grouped into three categories.

1. Some rely on human curated functional annotations with the Gene Ontology [3] in public genomic databases. Systems in this category include [2,4,32,35,55,63,71].
2. Some collect functional information directly from online literature databases, e.g. MEDLINE [52]. Systems in this category include [5,8,25,33,36,45,54,59,61,62,70].
3. Hybrids take advantage of both human curated data and literature, as well as the structure of GO. Systems in this category include [21,37,60].

1.2 Gene Ontology-based functional analysis

The Gene Ontology (GO) is a controlled vocabulary for describing genes and their products from three perspectives: biological process, molecular function and cellular component. It was developed starting in 1998 by a group of member organizations, and is coordi-

nated by the Gene Ontology Consortium. In its 9/2005 release, there were 18,455 concepts organized hierarchically in three major branches corresponding to these three aspects: biological_process, molecular_function and cellular_component. The GO Consortium is also a repository of gene/product annotations contributed from many member organizations, e.g. FlyBase, the *Saccharomyces* Genome Database (SGD) and the Mouse Genome Database (MGD), to name a few. This makes it an invaluable source of functional information for annotating microarray experiments.

Robinson *et al.* [63] provided functional descriptions for a gene cluster by counting and tabulating their associated GO concepts. The functional properties were represented by the most frequently annotated GO concepts of the cluster. THEA [55] went one step further. It checked against a user-definable background whether the frequency of a GO concept associated with the cluster was statistically significantly different from chance. Another tool, GO-Mapper [71], provided a similar description using explicitly annotated GO concepts, except that it scored the GO concepts with actual gene expression levels. A drawback of these systems is the lack of summarization, which is more problematic when the gene cluster is big and the list of GO concepts is long.

A quite different approach was exploited by the GO-Cluster [2], which switched the order of expression profiling and functional analysis. In other words, it performed expression profiling on a subset of the genes in a microarray experiment that were associated with a node of the GO hierarchy tree. Although the strategy seemed new and interesting, it was incapable of providing biologists with a functional summarization given a list of genes of interest.

The Ontologizer [63] used not only the explicitly annotated GO concepts, but also their parent concepts as implicit annotations. A similar strategy was also used in THEA [55]. Obviously, the intention of using the implicit annotations was to provide some sort of summarization of the explicit annotations. However, the higher a GO concept in the hierarchy, the more likely it to be used as an implicit annotation. This made the summarization less meaningful, because it consisted of many high-level concepts and the more general concepts tended to have higher counts than the more specific ones.

The Gene Ontology Categorizer [32] took the summarization one step further. Instead of simply counting how many and how often GO concepts were annotated explicitly or implicitly to a gene cluster, it tried to score them taking into account both generality and specificity. An implicit concept scored higher in generality if it covered more explicitly annotated concepts, e.g., a common ancestor of those annotated concepts. On the other hand, a concept scored higher in specificity if it was closer to the explicitly annotated concepts, e.g., direct parents had higher speciality scores than grandparents. The final output was the concepts with the highest combined scores balanced in generality and specificity. They were general enough to summarize most of the explicit annotations, yet not too general to be meaningless.

The above-mentioned systems focused on summarizing the functional properties of a gene cluster, i.e., finding what properties were common among genes in the cluster. Some other methods were developed to find what properties were different within or about a gene cluster. Kennedy *et al.* [35] developed a method to analyze a large gene cluster by finding sub-clusters based on similarities in GO annotations. Implicit annotations (parent concepts of the explicitly annotated concepts) were also taken into account, albeit with lower weights. The more GO concepts shared in two genes' annotations, the more similar the two genes were rated. Sub-clusters of similar genes were then obtained using a distance-based clustering algorithm (Modified Basic Sequential Algorithmic Scheme). The sub-clusters were highlighted on a GO hierarchy graph to illustrate their relationships.

Badea [4] compared a list of GO-annotated genes (positive examples) against another "background" list (negative examples), and extracted those concepts associated with the positive examples, but not the negative examples. In more detail, first, they crossed out the concepts explicitly or implicitly occupied by the negative examples in the GO hierarchy. Next, they marked the concepts explicitly annotated to the positive examples. (But if a concept had already been crossed out, do not mark it.) Finally, the marked concepts were moved up towards the root concept until a crossed-out concept was encountered. These marked and upgraded concepts were the final output, which was interpreted by the author as a functional hypothesis of the positive gene cluster. Obviously, the output was very much influenced by the selection of the negative examples. A small list would result in very general and less in-

formative concepts, while a large list might cross out all the concepts and generate empty output. The dependency on the negative examples made the interpretation of this analysis highly questionable, unless there were standardized rules for picking the negative examples.

While the summarization-focused systems may be more attractive to biologists, differentiation-oriented methods may also be useful in some special cases. However, both types of the systems share some common limitations, which result from the insufficiency of GO.

- The GO itself is still under active development, and far from mature. Its development is not balanced. While some branches go down to very deep levels (maximum 18 levels) and contain very detailed concepts (e.g. GO:0000201 – nuclear translocation of MAPK during cell wall biogenesis), the GO coverage in some areas of biology is incomplete, for example, pathways [44] and immunology [32].
- GO is updated monthly, so keeping annotations always compatible with the latest GO release is not a trivial task.
- The GO annotations are mainly manually curated by human experts, so inconsistencies are more or less inevitable. Badea [4] complained that numerous mistakes in the Proteome HumanPSD database[1] had to be corrected by hand before performing the analysis. Joslyn *et al.* [32] pointed out that some annotations from the Gene Ontology Consortium were at levels too high to be informative to biologists.
- GO annotations are only available for well-studied genes in a few model organisms. Badea [4] could find annotations for only 26% (39 out of 149) of the genes of interest to perform the analysis. Analysis based on such a small sample is risky to extrapolate to the whole dataset.
- There is a time lag between the time when new data is available in the literature and the time when it is annotated into the databases.

1.3 Literature-based functional analysis

To overcome the limitations of the GO-based systems, some researchers turned directly to the biomedical literature for functional information. MEDLINE is one of the most

[1] http://proteome.incyte.com/

comprehensive and up-to-date online literature databases for life and medical sciences. It has collected about 16,000,000 citations dating back to the 1960's, and more than 10,000 new citations are added weekly. An important feature of MEDLINE citations is that most citations are assigned, by human curators, a list of Medical Subject Heading (MeSH) terms, which is a rich source of computer-friendly functional information. The systems that extract functional information directly from MEDLINE can be further divided into two subcategories based on whether or not they make the assumption (explicitly or implicitly) that genes with similar expression patterns are involved in the same functional pathways.

1.3.1 Assuming similar expressions imply same functional pathway

Under this assumption, the results of expression analysis, the gene expression clusters, are part of the input to the functional analysis. Then, the task of functional analysis is to answer the following questions.

1. Is a gene cluster functionally coherent?
2. If it is coherent, what are the functions?

Raychaudhuri *et al.* [59,61] developed a method called "neighbor divergence" to answer the first question. They assigned to a gene cluster a score that measured functional coherence. The score was based on mutual relevance among the articles associated with the cluster. The relevance was measured on the number of an article's "neighbors." Two articles were neighbors if they shared a certain amount of words. If an article had many neighbors in the cluster, it would have a high relevance score. If a cluster had many high-scored articles, it would have a high functional coherence score. However, even if the method could assign perfect scores to the gene clusters, biologists would not be satisfied. They would like to know WHAT those functions are.

If biologists had *a priori* knowledge (hypothesis) about the function of a gene cluster, PubMatrix [5] or LACK [36] could be a useful tool. Given two lists of query terms, PubMatrix automatically sent combined queries to PubMed, resulting in a frequency matrix of term co-occurrences in MEDLINE. One of the term lists could include the gene names in a cluster, and the other, the keywords of the hypothesis. Thus, a high-scoring matrix would tend to

confirm the hypothesis, while a low-scoring one would tend to deny it. LACK took three user-provided files as input: a list of differentially regulated genes with functional annotations, a list of all genes on the microarray with functional annotations as background knowledge, and a list of candidate keywords as the hypothesis. Then it detected the statistical significance of each keyword for the differentially regulated genes against the background. The usefulness of PubMatrix or LACK depends on the *a priori* hypothesis about the gene cluster, which is not always available. Tools without the need for *a priori* knowledge are therefore highly desirable.

It is expected that the articles related to a gene cluster share many common keywords. The task of a functional analysis tool is to detect, extract and display those keywords. Biologists then could sift through the extracted keywords and grasp the functional properties of the cluster.

Masys *et al.* [45] extracted MeSH terms and Enzyme Commission (EC) numbers as keywords from the `<MeshHeadingList>` and the `<ChemicalList>` fields assigned to MEDLINE citations which mentioned one or more genes of interest. Keyword frequencies were summed across a gene cluster, and aggregated to their parent terms in the MeSH or EC hierarchy. The statistical significance of the keywords (including the parent terms) was detected by comparing their frequencies to the corresponding baseline frequencies, which were obtained from 500 groups of 100 genes randomly sampled from a set of 37,000 genes. The significant keywords were highlighted on the MeSH hierarchy tree or the EC hierarchy tree to visualize their relationships.

MedMeSH Summarizer [33] also took advantage of the MeSH terms. However, instead of comparing the keyword frequencies against a background, it compared and ranked keywords by their statistics (mean, variance, entropy...) within the cluster across individual genes. For example, a keyword with high mean and low variance of frequency was probably a common major topic of the cluster, because the word was associated with the most genes at similar high frequencies. On the other hand, a keyword with moderate mean and high variance might be a minor topic restricted to a subgroup of the cluster. The entropy and the GINI index [10] could be good matrices for particularity of a keyword to the cluster, i.e. whether the keyword was evenly distributed among the genes.

Blaschke *et al.* [54,70] analyzed citations by their words directly. Single words were extracted from the citations (stemmed to root and stopwords removed). Two-word terms were detected statistically if their observed frequencies were well above the expected random co-occurrences. The significance of a keyword (single-word or two-word term) in regard to a gene cluster was determined by a z-test against all clusters. Biologists were given a list of significant keywords as functional descriptions of a gene cluster.

These systems share some common drawbacks arising from the underlying assumption that similar expressions imply same functional pathway. First, the fact is that genes involved in different pathways can have similar expression patterns in a particular microarray experiment. Second, single gene may participate in several pathways. In either case, the expected common keywords would be a mixture from different pathways, making them harder to interpret. These drawbacks could be avoided if the functional similarities were extracted without the assumption.

1.3.2 Not assuming similar expressions imply same functional pathway

Functional analysis without this assumption does not use any gene clustering data as input. Thus, biologists can compare results from the functional analysis against those from expression profiling. This is like looking at a complex object from two different directions in order to get its full picture.

Shatkey *et al.* [69] developed a theme retrieval system for large-scale gene analysis. A theme was essentially a set of MEDLINE citations with similar keyword frequencies. It was recursively built in an Expectation Maximization manner from one or more kernel documents provided by the user. First, keyword frequencies were counted for the kernel documents. The most frequent keywords (excluding stopwords) were then used to retrieve more citations from MEDLINE. The new citations were added to the kernel, and keyword frequencies recalculated. The process was repeated until some criterion was satisfied, e.g. a certain number of citations had been retrieved. After themes were retrieved for all of the genes, a pairwise similarity (or distance) metric was defined based on the portion of shared citations between two themes (genes). The final results were, for each gene, a theme with its

characteristic keywords and a list of the most similar genes. The main limitation of the system is its dependence on the kernel documents, which must be provided by the users. For microarray experiments involving a large number of genes, the effort and time needed to prepare kernel documents is prohibitive. Another limitation is the lack of summarization in the results. Given a list of hundreds or even thousands genes (each with a set of kernel documents) as input, the system outputs a theme and a list of similar genes for every gene. Browsing through hundreds or thousands of themes and lists, it is easy to get lost.

The semantic gene organizer (SGO) [25] was a proof-of-concept system to find "phylogenetic" relationships among genes using functional descriptions instead of sequences. A gene was represented as a vector of words. The words were collected from all MEDLINE citations cross-referenced to the same gene in three organisms (mouse, rat, human) in LocusLink[2]. To reduce the size of the vectors, SGO utilized the latent semantic indexing (LSI), which extracted the most important factors in a matrix. The LSI vectors were then built into a hierarchical tree using the popular phylogenetic tree building software PHYLIP [17]. SGO is a useful tool to find out whether some genes are functionally closely or remotely related. However, it cannot give the actual functional descriptions, because LSI has factored out the biological meaning in the original word vectors.

Chaussabel and Sher [8] reported a "literature profiling" system for mining microarray experiment data. It was based on text clustering. The system collected MEDLINE citations related to the genes in the analysis. The citations were broken into single words, filtering out ones that were too frequent or too rare. For every unique word, its relative frequency (number of citations containing the term / total number of citations) was calculated for each gene. The results were tabulated in a two-dimensional matrix, where each row represented a gene and each column a word. The matrix was analyzed using a clustering software package originally developed for gene-expression profiling. The resulting clustergram showed groupings of genes according to patterns of word frequencies. The text clustering-based approach overcame the limitations of the theme-based approach. It required only a list of genes as input, eliminating the dependency on kernel documents. The genes were clustered into groups

[2] http://www.ncbi.nlm.nih.gov/projects/LocusLink/

so that it was easier to get an overall picture for a large dataset. However, the choice of the clustering algorithm posed some other limitations:

- The algorithm was originally developed for gene-expression clustering. Gene-expression clustering is quite different from text clustering in terms of feature space dimensionality and data sparsity. For example, a microarray experiment of 1,000 genes at 10 time points generates a 1,000 × 10 matrix. Most of its elements have non-zero values. The total number of unique words in MEDLINE citations related to the 1,000 genes can easily exceed 10,000. For a particular gene, however, the number may be only 500. Hence, we are dealing with a 1,000 × 10,000 matrix with most of the elements being zero. The algorithm would perform poorly on this type of matrixes. The authors did aggressive filtering to drastically reduce the matrix width (keeping 101 out of 25,000 words). Some useful information could be lost in this filtering process.
- The number of MEDLINE citations related to each of the genes varied dramatically, from 0 to thousands. Well-studied genes with many citations had counts in a lot of terms and tended to dominate the clusters, while newly discovered or less-studied genes with only a few counts were easily neglected. Hence, the authors set a lower threshold of 5 citations for a gene to be included. The threshold cut out about 40% of the genes in their sample dataset, which made the system incapable of drawing a complete picture for all of the genes.
- The system counted only single-word terms. A lot of functional information is captured in multiple-word terms and so could not be extracted. For example, the meaning of "red blood cell" is lost when the term is broken down into three separate words "red", "cell" and "blood", and mixed with hundreds of other terms. The system had little chance of discovering a cluster of citations discussing red blood cells.
- The output was a huge 2D clustergram (number of genes × number of terms). It was not very user-friendly in terms of determining cluster boundaries, or checking gene-term relations (i.e. matching rows and columns in a hundreds × hundreds table).

1.4 Hybrid systems

Since there is functional information available in public genomic databases and literature, some researchers have developed functional analysis systems to take advantage of both.

Kiritchenko *et al.* [37] trained hierarchical text classifiers for GO concepts using the human-annotated data in genomic databases as a training set. The classifiers were then used to assign GO concepts to un-annotated genes based on their descriptions in the literature. Raychaudhuri *et al.* [60] tried another approach, assigning GO concepts to genes using maximum entropy. While these tools are useful, especially for genomic databases, to automatically or semi-automatically annotate new genes, they seem less suitable for providing a big picture to help biologists interpret the outcome of a microarray experiment.

Glenisson *et al.* [21] provided a big picture by clustering genes based on their text representations. The basic idea was the same as that of "literature profiling" [8], with the following differences.

- Each gene's functional description was collected from the Saccharomyces Genome Database [14] and the SWISS-PROT database [76], supplemented with 20 MEDLINE citations.
- The functional descriptions were represented in a predefined vector space of GO concepts. Each vector described the content of the functional descriptions of a gene in terms of GO concept occurrences.
- A document's vector elements were weighted using $tf \bullet idf$ (term frequency × inverse document frequency), instead of tf only, as in "literature profiling" [8]. This weighting method takes into account both the importance of a term to the document (using term frequency) and its importance to the entire document set (using inverse document frequency).
- The clustering algorithm was K-medoids [34], which generated flat clusters, in contrast to the agglomerative neighbor joining used in "literature profiling" which generated hierarchical clusters.

The authors tested the system with an artificial data set (three groups of a total of 116 genes hand-picked from three biologically distinct functional pathways in the MIPS database[3]), I have identified some hidden problems not revealed by the test.

- Each gene needed input from both genomic databases' annotations and twenty MEDLINE citations. This requirement is too high even for some genes in the well-studied model organisms. One of the genes in the test set did not have enough annotations or citations, and it was clustered into a wrong group. The applicability of the tool is thus limited.
- Text descriptions were mapped into GO vector space. This limits the analysis to the areas where GO has been developed. In addition, many GO concepts are extremely difficult to map, for example, GO:0000201 (nuclear translocation of MAPK during cell wall biogenesis).
- The clustering algorithm requires a parameter K, which is the number of clusters. In their test, the authors used 3, because they already knew the answer. Given the *a priori* knowledge and the small number of clusters (the chance of a correct random guess was high, 33%), their good test run was no surprise. In the real world, however, biologists usually don't know the number of clusters beforehand; and the actual number could be much higher.

1.5 Requirements analysis

Based on the review of the state of the art of functional analysis of microarray experiments, we designed and built a new system, GeneNarrator, which overcomes the limitations of the above-mentioned tools and extends their strengths. Below is the requirements analysis for GeneNarrator from five perspectives.

1. Intended application

GeneNarrator is intended for functional interpretation of *any* microarray experiments as long as there is some information in the literature about the genes involved. Annotations

[3] http://mips.gsf.de/genre/proj/yeast/

in public genomic databases are not required, since that would limit the application to well-studied genes in model organisms. *A priori* knowledge about the genes is not required either. With little or no modification, it should be able to analyze proteomic or metabolomic data too.

2. User input

User input should be as simple as possible, for example, a list of gene names (or protein or metabolite names). *A priori* knowledge based input is not required, since such knowledge is not always available. Even if users know something about the genes, preparing that knowledge in a usable format (such as kernel documents [69] or keyword lists [5,36]) is a big burden for users that we wish to avoid. Expression profiling results are not required either, so that the functional analysis is independent of expression profiling. These two independent analyses may complement each other and together provide a better picture of the subject.

3. Source of functional information

Functional information from MEDLINE citations should be permitted, since they are the most complete and up-to-date. Functional annotations in public genomic databases tend to cover well-studied genes in model organisms. Dependency on them would make GeneNarrator inapplicable to less-studied genes or non-model organisms. Avoiding such independency also avoids the concern of annotation errors and delayed updates [4].

4. Type of analysis

The analysis should provide a summarized picture of the thousands or even tens of thousands of MEDLINE citations collected for a given list of genes. Text clustering seems a natural choice, because it can divide a large number of documents into groups based on topic differences (similarity in word usage). The clustering solution should avoid some pitfalls illustrated in some of the above-reviewed systems.

- Hierarchical clustering algorithms are more suitable than flat ones. Users usually don't know how many clusters should be in the final result beforehand. While inspecting the clusters, they might decide to merge some clusters. Merging clusters is much easier for

hierarchical clusters than flat clusters. With hierarchical clusters, a merge decision can be made locally by considering sibling clusters only. On the other hand, with flat clusters, the decision has to be made after comparing all clusters.

- The text-clustering algorithm of choice should perform well in high-dimensional vector space. It is best to avoid forced transformation from high-dimensional space to low-dimensional space as done in "literature profiling" [8]. Such transformations tend to lose information.
- Well-studied genes with a lot of citations should not dominate the clustering process, with newly discovered or less-studied genes being neglected.
- Multiple-word terms should be incorporated in text clustering. Many biomedical concepts are multi-word terms. Breaking them down to unrelated single words may adversely affect clustering results, because useful information is lost.

5. Output of analysis

An intuitive GUI is necessary to display the analysis results, i.e. hierarchical structure of the topics, biological meaning of the topics, how many and what genes are in what topics, etc. This is a given in modern software design.

Chapter 2 Strategic Design

2.1 Design overview

Based on the above requirements analysis, a two-step clustering approach was designed for GeneNarrator to provide biologists with a functional summarization of a microarray experiment utilizing the information from MEDLINE (Fig. 1). The system takes as input a user-provided gene list, and automatically queries PubMed for citations mentioning one or more of the genes. Gene symbols, official names, synonyms and gene product names could all be included to retrieve more relevant citations. The pool of retrieved citations is grouped into functional topics using a text-clustering algorithm in step 1. Each gene is then represented as a topic distribution, a vector of occurrence counts (how many of its citations in which topics). Then the 2nd clustering step groups together the genes with similar distributions.

Text clustering-based functional genomic analysis is not new in the literature (see Chapter 1). However, the two-step clustering design distinguishes GeneNarrator from other systems. The rationale of this aspect of the design will be explained in the next section. The

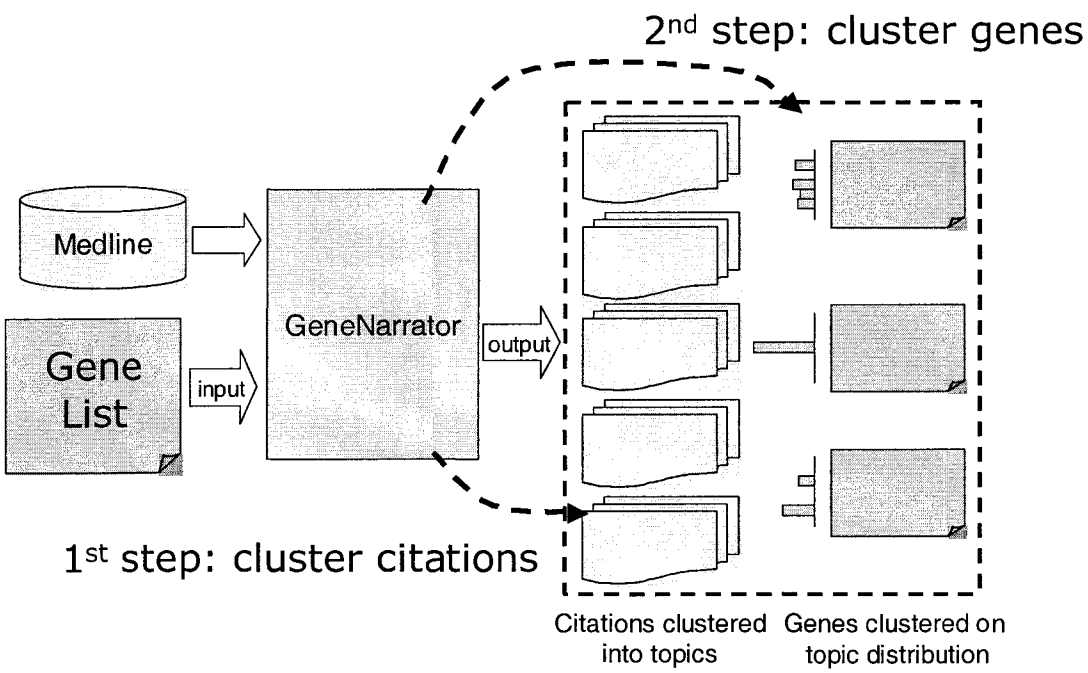


Figure 1. Overview of GeneNarrator

other two strategic design decisions, the choice of document representation and the choice of text clustering algorithm, will be discussed in the rest of the chapter.

2.2 Two-step vs. one-step clustering design

The two-step clustering design is one of the distinguishing features of GeneNarrator. The two-step design overcomes three main drawbacks of single-step designs:

- Dominance of well-studied genes over less-studied genes.
- Dilemma to assign less-studied genes.
- Difficulty in grasping clusters' biological meanings.

Some genes are well studied with hundreds or even thousands of hits in MEDLINE, while newly discovered or less popular genes may have only a few hits. In a one-step design, each gene is represented by all of its citations combined. Thus genes with a lot of citations are analogous to "thick books," while genes with only a few citations are more like "short memos." When the thick books are compared directly against the short memos, the thick books inevitably dominate. This was vividly illustrated in Fig. 2 of [8]. In the two-step design, however, such dominance can be avoided because citations for a gene are not bundled together. Instead, all citations from all genes are pooled together, thus each citation has the same weight in determining topics, regardless of its origin from a thick book or a short memo.

Many genes, especially well-studied ones, participate in several biological processes (pathways). Their text representations are therefore mixed with keywords from different pathways, just like thick books usually have different chapters covering different topics. In a one-step design, therefore, an individual topic cluster dominated by well-studied genes may discuss several pathways; and different topic clusters may overlap partially in some pathways. The overlap can make assigning a less-studied gene a dilemma, if it is discussed in the context of a single pathway. On one hand, it may be assigned to any of the overlapping clusters, because the pathway is discussed in all of them. On the other hand, it is wrong to assign it to any of the clusters, because the membership in a multi-pathway cluster requires coverage in more than one pathways. In the two-step approach, this is less likely a problem. Text clustering is expected to group the citations into "pure" topics discussing individual path-

ways, if an appropriate clustering algorithm is used. Assigning less-studied genes to individual pathways should not be a problem at all. Well-studied genes can be assigned to many topics simultaneously.

Finally, both approaches must provide a list of representative keywords and/or sentences for each topic cluster to make it biologically meaningful. From the users' point of view, it would be much easier to grasp the biological meaning if the keywords were from a single pathway rather than a mix of several pathways. Thus, the two-step design is inherently superior to the one-step design in this aspect.

2.3 Document representation

Before any clustering algorithm can be applied to a document set, the documents have to be transformed into some type of abstracted representation. The most widely used document representation in text clustering, as well as in other text mining tasks such as text classification and document retrieval, is the vector space model [70]. Given a set of m documents, the model encodes a document d_i as a vector in an n-dimensional vocabulary space

$$d_i = (w_{i1}, w_{i2}, ...w_{ij}, ...w_{in}), \quad i = 1,2,...,m$$

where w_{ij} is the weight of the j^{th} entry in the vocabulary in document d_i. The actual order of the vocabulary entries is not important. Variations exist for different definitions of the vocabulary and different weighting methods. The weighting method is closely tied to the clustering algorithm, so its discussion will be deferred to the next section. The rest of this section will be devoted to the vocabulary.

The most popular vocabulary approach is called "bag of words" (BOW), which, as the name suggests, includes the unique words in the document set [6,8,9,11,12,16,22,24,27-30,41,53,65,66,72,75,77,78]. Stopwords are usually excluded; and optionally the words are stemmed to their roots (e.g. suffixes removed). Some filtering or feature selection methods have been investigated in order to reduce the dimensionality of the vocabulary space [6,8,11]. Another choice of the vector space is a predefined controlled vocabulary (concept-based representation), such as the Medical Subject Heading (MeSH) [74], Gene Ontology (GO) [21], or other ad hoc ontologies [26].

It is obvious that the vector space model doesn't capture information in a document present in word *sequences* and grammar structures. Could the omission of this information adversely affect the results of text clustering? Observe that text information is captured in several levels.

- The single term level, where the information is intrinsic to the word, *e.g.* "insulin", "cell", *etc.*
- The ontology level, dealing with implicit relationships between concepts such as synonymous terms or related concepts. For example, a paper discussing "baseball" and a paper discussing "soccer" might be grouped together under "sports."
- The phrase/sentence level, dealing with explicit interactions between terms where grammar structures and word orders are important. For example, "insulin decreases blood sugar," "insulin-induced decrease of blood sugar," or "red blood cell."
- The logic flow level, including paragraphs, sections, chapters, *etc.*

Not all information is needed for text clustering, where the interest is more in grouping documents based on concepts, rather than on the interactions among the concepts. Hence, the first two levels of text information seem sufficient for text clustering.

A BOW-based representation can capture the information in level 1, while a concept-based representation can also handle level 2. It is therefore natural to hypothesize that a concept-based approach could outperform a BOW-based one. There are three pre-requisites to taking full advantage of a concept-based representation:

1. an ontology that encodes background knowledge about the domain of interest (genomics, in this case);
2. a natural language processing algorithm that efficiently and effectively maps free text to ontology concepts; and
3. a text clustering algorithm that can effectively utilize the ontology.

Given the state of the art in biomedical ontology design, natural language processing and text clustering, there is no guarantee that one representation will be better than the other. It is therefore intriguing to experiment with both approaches and compare them.

2.4 Choice of text clustering algorithm

The importance of an appropriate text clustering algorithm for the success of GeneNarrator cannot be overemphasized. There are many types of clustering algorithms with many variations for diversified applications (for a review, see [31]). Some algorithms were originally developed for other purposes, and later tried for text clustering. They fall into three categories: (1) *k*-means and its variations [21,22,26]; (2) variations of agglomerative hierarchical clustering [8,22]; and (3) self-organizing maps (SOM) and variations [27-30,38-40]. There is also an algorithm specially developed for text clustering: the cluster-abstraction model (CAM)[24].

Despite their successful application in many other areas, distance- or similarity-based algorithms (*k*-means, agglomerative hierarchical clustering and SOM) suffer from problems due to high dimensionality when applied to text clustering [26]. All of these algorithms consist of steps that iteratively find nearest neighbors. In a high dimensional space, however, numerous data points can appear to be at the same distance from a given point [7,23]. The "nearest neighbor" idea is thus less meaningful in the high dimensional context.

The Cluster-Abstraction Model (CAM) [24] avoids the calculation of distance or similarity. A CAM consists of a vocabulary and many topics organized in a hierarchical tree. Each topic is defined by a set of probabilities (*Pt*). Each probability is the likelihood of the topic containing a certain word in the vocabulary. Each leaf topic defines a unique route from the leaf to the root of the tree. The topics along the route are considered as different abstraction levels. The closer to the root, the higher the abstraction level. There is a document bin for each route. The bin is also defined with a set of probabilities (*Pb*). Each probability is the likelihood of the bin containing documents from a certain abstraction level. Finally, the entire model has a set of probabilities (*Pm*), each of which is the likelihood of the model containing documents from a certain bin. Given a model, a set of documents can be generated by iteratively picking a bin according to *Pm*, picking an abstraction level (topic) according to *Pb* given the bin, and producing words according to *Pt* given the topic. Clustering a set of documents is equivalent to finding a hierarchical topic structure and all of the probabilities, which can generate the documents with the highest likelihood. Compared to

other distance-based algorithms, especially agglomerative clustering methods, CAM has the following advantages:

- insensitivity to term weighting methods and distance (similarity) definitions;
- a statistically sound foundation;
- multiple levels of text clustering;
- representative keywords for topics (words with the highest probabilities); and
- efficient model fitting by annealed expectation maximization [64].

It was thus not difficult to decide on CAM as the choice of a text clustering algorithm for GeneNarrator.

Chapter 3 BOW-Based System: GeneNarrator I

3.1 Architectural overview of GeneNarrator I

GeneNarrator I consists of six modules: DocBuilder, LongBOW, CrossBOW, GeneSmith, ArrowSmith, and BOWviewer (Fig. 2). The core text clustering module (CrossBOW) was modified from the "bow" toolkit, an open-source library for statistical language modeling, text retrieval, classification, and clustering [46]. The other modules were developed around CrossBOW to prepare its input and process its output. The DocBuilder module retrieves MEDLINE citations that are related to at least one of the user-provided genes. The LongBOW module performs several pre-processing tasks on the citations including discarding stopwords, stemming, and detecting multiple-word terms. The CrossBOW module applies the CAM algorithm to the citations, and groups them into hierarchical functional topics. The ArrowSmith module extracts representative keywords from CrossBOW's output, and finds high-scoring sentences and citations for the topics in order to help users interpret their biological meanings. The more keywords a sentence or citation contains, the higher it scores. The GeneSmith module maps each gene into a distribution across the topics, and groups the genes with similar distributions. The BOWviewer module is a GUI for navigating the hierarchical topics, browsing the representative keywords, sentences and citations, and comparing

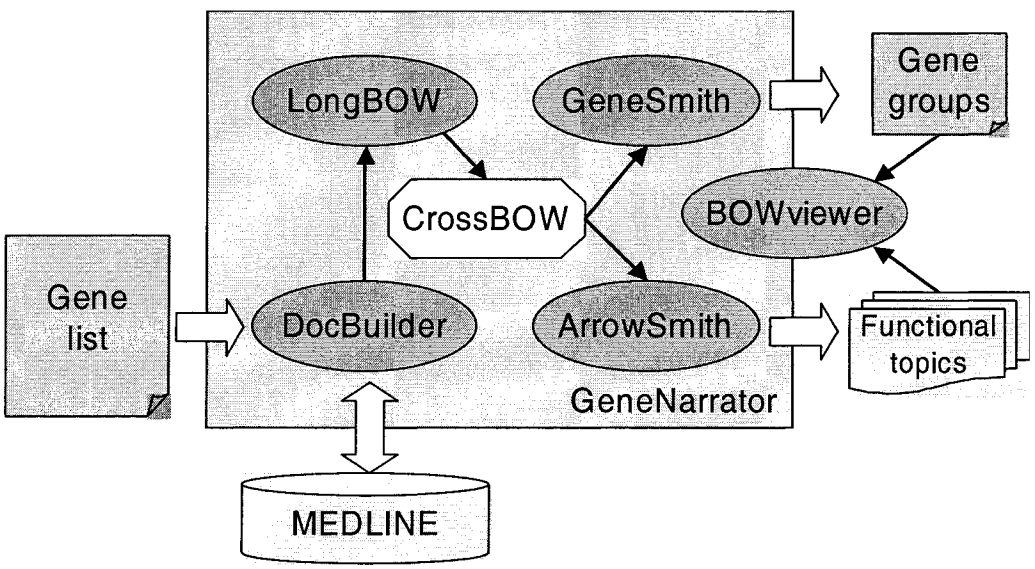


Figure 2. Architecture overview of GeneNarrator I

the genes' or gene groups' topic distributions. All modules were implemented in Java, except CrossBOW (which is in C).

3.2 Detailed description of individual modules

3.2.1 DocBuilder

Table 1. Required and optional input/output of DocBuilder

Input	A file containing a list of gene names. Synonyms and gene product names can also be included in the same line.
Output	• A directory of MEDLINE citations in plain text format. Each file contains one citation. The filename is the citation's PMID, and the file content is the citation's title and abstract. • A gene-PMID mapping file to keep track of which citation belongs to which gene.
Options	• A user may specify an upper limit of the number of citations retrieved for each gene. The option is designed to prevent well-studied genes from dominating the functional topics. • A user may specify a species name to filter out citations not related to the species.

Given a file containing a list of genes (1 gene/line), DocBuilder retrieves MEDLINE citations related to each of the genes via Entrez Programming Utilities (eUtilities) [51]. It consists of four sub-modules: a *querier*, a *sampler*, a *fetcher* and a *parser*. The querier embeds a gene name (together with its synonyms and product names if also provided) into a query, and sends the query to PubMed using eUtilities' ESearch function. PubMed returns a list of PMIDs. A user may set an upper limit for the number of PMIDs to be used in the following steps. If the number of returned PMIDs exceeds the upper limit, the sampler draws a random sample from the list. The returned or sampled PMIDs are recorded in the gene-PMID mapping file for later use. The fetcher then retrieves the PMIDs' full citations from PubMed using eUtilities' EFetch function. Finally, the parser extracts the titles and the abstracts from the retrieved citations, and writes them to plain text files. More details about the four sub-modules can be found in [15].

3.2.2 LongBOW

Table 2. Required and optional input/output of LongBOW

Input	The MEDLINE citations in plain text format (from DocBuilder).
Output	Modified citations: • Stopwords removed. • Word stemmed (suffixes removed). • Multiple-word terms detected.
Options	• A user may provide his/her own stopword list. • Sensitivity of multiple-word term detection is adjustable.

The LongBOW module preprocesses the MEDLINE citations in order to get better clustering results. The modifications include the following.

- Removal of stopwords, such as "that," "is," "you," "of," etc. A user may provide his/her own stopword list to replace the default stopword list.
- Stemming (suffix stripping). For example, "regulation," "regulating," "regulator," and "regulates" are all stemmed to "regulat." The stemming method is a Java implementation of the Porter stemming algorithm [56].
- Detection and labeling of multiple-word terms (MWTs).

Detection and labeling of MWTs is done in three steps (all of the citations are scanned three times). In the first pass, along with performing stopword removal and stemming, document frequencies (*df*: number of documents containing a specific term) and term frequencies (*tf*: number of total appearances of a specific term in the whole document set) of all unique single-word terms (SWTs) are counted. The total number of words (including stopwords and punctuations) is also recorded. After the first pass, a cut-off value of *df* is used to determine the "significant" SWTs, i.e. those with a *df* above the threshold.

In the second pass, unique double-word terms (DWTs) are counted. A DWT is defined as two consecutive SWTs without any stopwords or punctuation in between; both SWTs must be significant (above the *df* threshold). A DWT's observed count is tested against a null hypothesis, which assumes that two SWTs are next to each other by chance. The test is similar to the "t-test" of collocations described in [43]. Briefly, suppose the occurrences of single-word term w_1 in the entire document set is n_1, that of w_2 is n_2, the total number of words (including stopwords and punctuations) in the document set is N, and the observed double-word term $w_1\_w_2$ is n_{obs}. Then the probability of an occurrence of w_1 being

followed by w_2 under the null hypothesis is given by $p = n_2 / N$, and the expected occurrences of $w_1\_w_2$ is $n_{exp} = n_1 p = n_1 n_2 / N$. We can construct an approximate binomial test by

$$z = \frac{n_{obs} - n_{exp}}{\sqrt{n_1 p(1-p)}}$$

and reject the null hypothesis for large enough values of z. The DWTs that reject the null hypothesis are considered as significant DWTs.

In the third pass, significant DWTs are evaluated in the context of individual citations. A significant DWT is qualified only if its two contained SWTs are mentioned a similar number of times in a particular citation. For example, an occurrence of "*cell cycle*" would not be qualified as a DWT if the word "*cell*" was mentioned 10 times in a citation, while the word "*cycle*" only appeared once. Even though "cell cycle" might appear more frequently than expected by chance in the entire document set, it did not appear to be a major subject in the particular citation. Formally, given a significant DWT "w_1 w_2" with term frequencies tf_1 and tf_2 in the citation, and a predefined threshold α, $\alpha > 1$, the DWT is qualified if and only if $1/\alpha \le tf_1 / tf_2 \le \alpha$.

MWTs are detected if DWTs are chained together. Upon detecting a qualified DWT, LongBOW replaces the space with an underscore character (e.g. *cell_cycle*). The trick is to enable the CrossBOW module to treat the DWT as a single word.

3.2.3 CrossBOW

Table 3. Required and optional input/output of CrossBOW

Input	Modified MEDLINE citations (from LongBOW).
Output	• Hierarchical clusters of the citations. • Representative terms for each of the clusters.
Options	• A user may specify the branching factor and the maximum level of branching to control the number of clusters desired. • The number of representative terms can be specified.

The CrossBOW module was modified from the "bow" toolkit, a library for statistical language modeling, text retrieval, classification, and clustering [46]. CrossBOW is its text clustering component, which implements the Cluster-Abstraction Model algorithm described

in Chapter 2. It takes a set of documents (plain text files) as input, and clusters them into a hierarchical topic tree. Each document is assigned to one and only one of the leaf nodes (topics). Each topic is represented as a list of the most probable keywords. The topology of the tree (i.e. branching factors and maximum depth) can be specified as command line options. Many other command line options are available to meet various user needs. More details about the command line options can be found in its online manual. The modifications introduced for GeneNarrator included the following.

- Recognition of multi-word terms labeled by LongBOW. The default setting strips and discards the underscore characters, so we overrode the default.
- Addition of a command line option to change the number of topic keywords in the output.

3.2.4 ArrowSmith

Table 4. Required and optional input/output of ArrowSmith

Input	• Hierarchical topics (from CrossBOW). • Representative keywords for the topics (from CrossBOW). • The original citations (from DocBuilder).
Output	• Representative sentences and citations for each of the topics.
Options	• Scoring methods for representative terms.

Given the hierarchical topics and the representative topic keywords generated by CrossBOW, the ArrowSmith module scores the sentences and the citations within a topic, and picks those with the highest scores. According the scoring method used, each representative keyword is assigned a score. For example, the binary scoring method gives all representative keywords a score of one. Other scoring methods may assign different scores to different keywords based on their probabilities or ranks. A sentence's score is the sum of its contained keywords' scores, and a citation's score is the sum of its sentences' scores. The representative keywords, and the highest-scored sentences and citations, define the inferred biological meaning of each topic.

3.2.5 GeneSmith

Table 5. Required and optional input/output of GeneSmith

Input	• Hierarchical topics (from CrossBOW). • Gene-PMID mapping file (from DocBuilder).
Output	• Distribution of a gene's citations among the topics. • Gene groups with similar topic distributions.
Options	• A user may pick the clustering method to group the genes. • A user may specify the number of gene groups desired.

The GeneSmith module converts gene-to-citation mappings to a gene-to-topic distribution by straightforwardly counting. It further clusters genes based on their topic distribution. The choice of clustering algorithms is *k*-means or expectation maximization (EM) from the Weka machine-learning workbench [19].

3.2.6 BOWviewer

Table 6. Required and optional input/output of BOWviewer

Input	• Representative topic keywords (from CrossBOW). • Representative sentences and citations for the topics (from ArrowSmith). • Distribution of a gene's citations among the topics (from GeneSmith). • Gene groups with similar topic distributions (from GeneSmith).
Output	Screen displays.
Options	None

The BOWviewer module (Fig. 3) is a graphical user interface (GUI) for browsing the final analysis results: the topic hierarchy, the representative topic keywords, high-scoring sentences and citations for each topic, the genes' topic distributions, and the gene groups. Users can easily navigate through the hierarchical topic tree; browse topic keywords, high-scoring sentences and citations; and annotate the topics with biologically meaningful comments. It is also intuitive to check how many genes contribute to a topic of interest, or how a particular gene distributes among the topics.

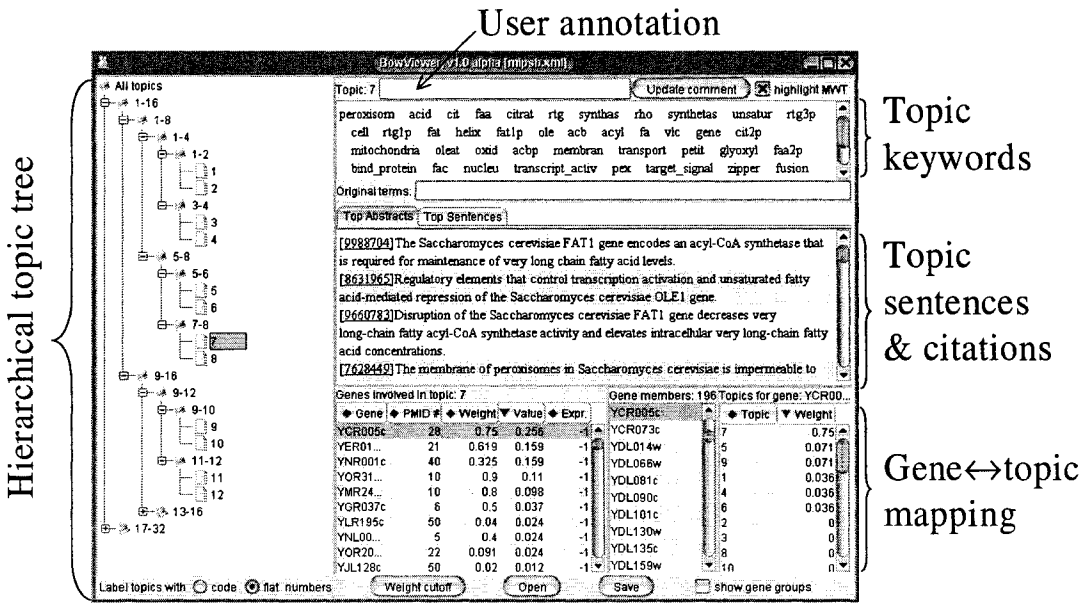


Figure 3. BOWviewer user interface

Chapter 4 Evaluation of GeneNarrator I

Since GeneNarrator takes a two-step clustering approach, it would be useful to see how effective GeneNarrator performed on each step. To achieve this purpose, real-world microarray experiment data would be difficult to use. Even though GeneNarrator could cluster document topics and search for functional groups for the genes that are significantly regulated in the experiment, there is no "gold standard" to compare the results with. Thus we don't know how many "true" groups or topics are present. Therefore, a handpicked list of genes from several pathways of a model organism was used for evaluation. The first section in this chapter is a review of the dataset. A brief review of the methods and indices for evaluating clustering algorithms will be given in the second section. In section 3, a new metric, normalized mutual information, is introduced. Finally, in section 4, the metric will be applied to GeneNarrator's analysis results.

4.1 The gold standard gene list and document set

The evaluation gene list contains 155 yeast genes manually selected from ten pathways in the comprehensive yeast genome database [48]. The gene symbols and their synonyms are listed in appendix A. The gene list was fed into DocBuilder; and 2819 unique PMIDs were returned (Table 7). The upper limit of PMIDs for a single gene was set at 50. Please note that there were some overlaps among the pathways both at the gene level and at the citation level. Care was taken in picking the pathways so that the overlap was kept at an acceptable low level, though some overlap was unavoidable. The higher degree of overlap at the citation level was expected, since it is common for papers to discuss several pathways. To build a more clearly separated document set is technically difficult, and its evaluation suitability is also questionable because it tends to over-estimate the effectiveness of a text-clustering algorithm.

Table 7. Handpicked genes from the Comprehensive Yeast Genome Database and the set of citations (PMIDs) retrieved from PubMed

ID	Pathway	# of genes	# of PMIDs
1	Sulfur amino acid biosynthesis	14	263
2	Biosynthesis of sphingolipids	15	230
3	Respiratory chain	40	643
4	Pyrimidine metabolic pathway	8	257
5	Krebs tricarboxylic acid cycle	15	209
6	Cell cycle control of DNA replication	24	1102
7	Pre-rRNA processing pathway	24	390
8	The ubiquitin-mediated proteolytic pathway	7	138
9	Early steps of protein translocation into the endoplasmatic reticulum	5	190
10	Vesicular protein transport in exo- and endocytosis	7	198
	Total	159	3620
	Unique total	155	2819

4.2 Evaluating clustering: literature review

Evaluation of clustering, an unsupervised machine-learning task, is more difficult than classification, a supervised machine-learning task. While the accuracy (or error rate) of classification is widely used for evaluating classifiers, there are no commonly accepted approaches for evaluating clustering. This is illustrated in Table 8, where the clustering evaluation methods used in some papers are listed. (The list is by no means exhaustive.) Different methods were used not only in different papers but also in the same papers, which indicated that no one method alone is trusted to evaluate a clustering method. The methods can be categorized into three strategies: (i) indices measuring agreement between two clustering results or a clustering result against a "gold standard;" (ii) metrics measuring cluster quality; and (iii) domain experts' or readers' subjective judgment. These are discussed further in the following paragraphs.

4.2.1 Subjective judgment

Subjective judgment may be explicit domain experts' opinions. Alternatively, the authors may simply present their experimental results, and let the readers determine whether the results make sense or not. This was the most popular method in our survey (Table 8); and sometimes it was the only approach used. Subjective judgment may be an important compo-

nent of the entire evaluation, since it is the end users who finally determine whether a tool is truly useful or not. However, it is not a good choice as a sole and formal evaluation method simply because it is inconsistent and influenced by many uncontrollable factors. Different experts may give completely different opinions. Even the same expert may change his/her mind over time. It is also impossible to compare algorithms from different studies solely based on subjective judgment.

Table 8. An incomplete list of evaluation methods for clustering in the literature

Ref.	Agreement Measures												Quality Measures				Subj.
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Steinbach 2000	x	x															
Goldszmidt 1998			x														
Hung 2004a, b			x										x				
Struble 2004														x			x
Beil 2002	x																
Hotho 2001															x	x	x
Schutze 1997				x													
Iliopoulos 2001																	x
Hofmann 1999																	x
Glenisson 2001						x	x								x		x
Chaussabel 2002																	x
Meila 2002					x												
Saporta 2002							x	x	x								
Denoeud 2004							x		x	x	x	x					

1: Weighted average of entropy
2: F-measure
3: Error rate/accuracy
4: Precision
5: Variation of information
6: *k*-NN learnability
7: (Adjusted) Rand index
8: Mac Nemar's test
9: Jaccard index
10: Transfer distance
11: Wallace index
12: Leman index
13: Average quantization error
14: Agglomerative coefficient
15: Silhouette coefficient
16: Mean squared error
17: Subjective judgment

4.2.2 Cluster quality measures

It seems natural to evaluate a clustering algorithm based on metrics that measure cluster quality. Clusters are of high quality if elements within a cluster are close to each other, while the distance between elements from different clusters is far apart. There are several metrics measuring cluster quality. Some of them are dependent on the underlying clustering algorithms, such as the average quantization error (AQE) for self-organizing maps (SOMs) [27-30], and the agglomerative coefficient for hierarchical algorithm [74]; while others are independent, such as the *Silhouette* coefficient (SC) [21,26] and the mean squared error [26]. SC can serve as a good example for this type of metrics (Fig. 4).

Let D be a set of documents and $\{D_1, D_2, \ldots D_k\}$ a partition of D. The *distance* of a document $d \in D$ to a cluster D_i is the mean distance from d to the documents p in D_i:

$$dist(d, D_i) = \frac{\sum_{p \in D_i} dist(d, p)}{|D_i|}$$

Let $a(d)$ be the *distance* of d to its assigned cluster, and $b(d)$ the distance of d to its nearest neighboring cluster:

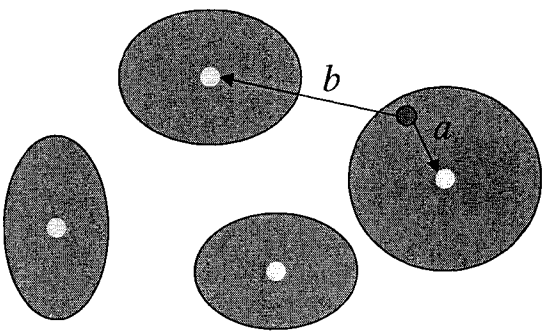
$$a(d) = dist(d, D_i), \; d \in D_i$$
$$b(d) = \min_{D_j, d \notin D_j} \{dist(d, D_j)\}$$

The *silhouette* $s(d)$ of document d and the *silhouette coefficient* SC of D are then defined as

$$s(d) = \frac{b(d) - a(d)}{\max\{b(d), a(d)\}}$$
$$SC = \frac{\sum_{d \in D} s(d)}{|D|}$$

Intuitively speaking, *SC* measures on average whether a document is closer to its own cluster's center, or to its nearest neighboring cluster's center. It is bounded between 0 and 1 for entire D, although negative $s(d)$ might be taken by some outliers. The interpretation of the value is shown in Fig. 4.

Since it is defined in terms of document distance, *SC* suffers from high dimensionality like distance/similarity-based clustering algorithms. In high dimensional space, data points appear next to each other at the same distances [7,23]. Hotho *et al.* developed an ontology-based text clustering system, observing that the *SC* dropped below 0.25 (not sepa-



SC	Separation
0.7~1.0	Excellent separation
0.5~0.7	Clearly separated
0.25~0.5	Separated with noise
< 0.25	Not separated

Figure 4. *Silhouette* coefficient

rated) whenever dimensionality was above 30. This issue is not special to *SC*, but may happen to any distance- or similarity-based metrics.

In high dimensional space, it is even possible for a "gold standard" document set to fall into the "not separated" range, which makes the cluster quality measures completely useless. Compared to a gold standard, the "absolute" quality of a clustering solution does not matter. What's important is how well the clustering result agrees with the original partition.

4.2.3 Agreement measures

Although a plethora of agreement metrics and indices have been proposed and/or tried in the literature, none of them seem to have gained popularity among researchers (Table 8). They can be further categorized into three subgroups: (i) metrics adopted from classification evaluation; (ii) member relation-based indices; and (iii) information-based metrics.

4.2.3.1 Metrics adopted from classification evaluation

Evaluating text classification is fairly straightforward. Given a set of documents, a classifier's task is to assign to each document one or more predefined class labels. Evaluating the classifier is to see whether the assigned class labels agree with the original known labels. Classification accuracy and error rate, as well as their derivations (e.g. precision, recall and F-measure), are widely accepted metrics. It is not surprising that some of them were adopted, with or without modification, into the closely related unsupervised machine-learning task, of which text clustering is an example [21,22,27-30,68,72].

A text-clustering algorithm tries to partition a document set, i.e. divide it into mutually exclusive subgroups. The number of clusters (partitions or subgroups) and their membership contents are up to the algorithm. In some cases, such as the family of *k*-means algorithms, the number of clusters is "guessed" by the user as an input parameter. Comparing the partition to the original class labels (which may or may not be a partition) is significantly different from the situation of evaluating classification. Hence, adopting the classification metrics to text clustering is questionable for at least two reasons.

First, if the number of clusters is more than the number of classes, some clusters have to be labeled as wrong and get penalized, no matter how much sense the extra clusters make.

This is unfair to the algorithms in the *k*-means family, since they are instructed to generate a particular number of clusters and get penalized for doing that because of users' wrong guesses. Hence, an ideal metric should be able to tolerate cluster numbers different from the number of classes.

Second, a clustering algorithm is likely to strip a small proportion out of a class, and put it in a separate cluster or mix it into another class. From clustering's point of view, this is not too bad, much better than scattering the proportion into many other classes. The algorithm should get some penalty for separating the proportion from the main class, but still get some partial credit for keeping the proportion together. However, none of the classification metrics gives this type of partial credit. Any documents clustered apart from their main classes are penalized to the same extent.

4.2.3.2 Member relation-based indices

Rand [58] proposed an objective criteria (the Rand index) to evaluate clustering methods. Let D be a set of n documents, and P and Q be two partitions on D. A pair of documents x and y ($x, y \in D$) can be joined or separated in P and Q. Let further r, s, u and v be the number of pairs related in P and Q as depicted in the table below

	Joined in P	Separated in P
Joined in Q	r	v
Separated in Q	u	s

The Rand index is then defined as the percentage of document pairs for which P and Q agree:

$$R(P,Q) = \frac{r+s}{r+s+u+v} = \frac{r+s}{n(n-1)/2}$$

While making sense, Rand index has drawbacks [47]. First, its baseline is not zero, since random guess may still make some pairs simultaneously joined or separated. This reduces the index's useful range to approximately (0.6, 1] [18]. Second, the baseline is a function of P and Q, and has to be recomputed for every pair of P and Q [47]. Hence, many variations of the index (Table 9) have been proposed in the literature for various considerations (for review, see [13,67]).

However, it seems that none of these variations was a silver bullet. Saporta and Youness [67] experimented with Rand, Mac Nemar and Jaccard's indices on 1,000 clustering simulations, in order to find out the distributional characteristics of the indices. Although they expected a deviation from the normal distribution since the simulations were not independent, they were still surprised at the observed bimodal distributions. They commented that "agreement measures are only one of the many facets of comparing partitions." This was another way to say, please do not put your faith on these indices.

Table 9. Variations of Rand index

Index	Definition	Rationale
Corrected Rand index	$HA(P,Q) = \frac{r - Exp(r)}{Max(r) - Exp(r)}$ *Exp*(*r*): expectation of *r* *Max*(*r*): maximum of *r*	Random guessing may make some pairs joined simultaneously in *P* and *Q*. They should be subtracted from the index.
Jaccard's index	$J(P,Q) = \frac{r}{r+u+v}$	It is the joined documents that define a partition (i.e., clustering). Separation is just a by-product.
Wallace index	$W(P,Q) = \frac{r}{\sqrt{(r+u)(r+v)}}$	Similar to Jaccard's index, except normalized by the geometric mean of joined pairs in *P* and *Q*.
Lerman index	$ICL(P,Q) = \frac{r - Exp(r)}{\sqrt{Var(r)}}$ *Var*(*r*): variance of *r*	Simultaneously joined pairs corrected for its expectation, and normalized to the standard deviation.
Mac Nemar's test	$Mc(P,Q) = \frac{u - v}{\sqrt{u+v}}$	A non-parametric statistical test to check equality of proportions of disagreement.

Denoeud *et al.* also criticized these indices, promoting the *transform distance* originally proposed by Régnier in 1964 [13]. The transform distance between two partitions is defined as the minimum number of steps to transform one partition to the other. The same transform distance may result in different index values and vice versa. For example, suppose one moves a single document from one cluster to another. The transform distance is always 1, regardless of the sizes of the source/target clusters. However, those indices are sensitive to the cluster sizes, since moving a document from a larger to a smaller cluster breaks more simultaneously joined pairs, resulting in smaller index values. There is no compelling reason why one approach should be favored over the other. In addition, transform distance has its own drawback, i.e. it is computationally expensive [13].

Although the critiques and the controversies regarding the indices and transform distance may explain (at least in part) the diversity of evaluation approaches for text clustering, they do not rule out using several metrics with well-understood semantics regarding partitions and their comparison provided by information theory.

4.2.3.3 Information-based metrics

The (Shannon) entropy of a discrete variable X measures its average information content in terms of bits. It is defined as

$$H(X) \equiv -\sum_x P(x) \log_2 P(x)$$

where $P(x)$ is the probability of X being in state x. Applying it to a set of documents D composed from m classes such that $D = \{D_1, D_2, \ldots D_m\}$, we have the entropy of D as

$$H(D) \equiv -\sum_{i=1}^{k} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}.$$

$H(D)$ equals zero when all documents are from the same class, and it reaches a maximum value when the set is evenly mixed. It is fairly straightforward to apply this measure to a clustering result. One simply averages the clusters' entropies weighted by the clusters' sizes. This was the approach used in [6,73].

Although it measures the "cleanness" of a clustering result, the average entropy is not a good indicator of the effectiveness of its underlying algorithm. To evaluate the effectiveness, we need to measure the "agreement" between the clustering result and the original classes, not the "cleanness" of the result. The "cleanness" is just a by-product of the "agreement;" it is a required condition, but not a sufficient one. An example may illustrate the idea more clearly. Suppose we have a set of documents from 3 classes (4, 400, and 4 from each class respectively). A random-guess clusterer is used to divide the set into 4 clusters. Each cluster might have one document from the first class, 100 from the second, and one from the third. The clusters are "clean" (100 out of 102 documents from the same class), so their average is also clean. However, there is no "agreement" at all!

The "cleanness" measure fails in the above example because the original document set is already "clean." The failure is the result of looking at the clustering from only one direction: whether the clusters are clean or not. To see agreement, we also have to look from

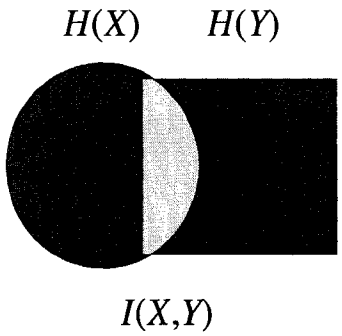


Figure 5. Entropy and mutual information

the other direction: whether the classes are clean or not. If we have both clean clusters and clean classes, then we have an agreement. Although we could calculate an average entropy across the clusters and another across the classes, this is not necessary because *mutual information* can capture exactly what we are looking for.

4.3 Normalized mutual information

The mutual information I between two discrete variables X and Y is defined as

$$I(X,Y) \equiv -\sum_{x\in X}\sum_{y\in Y} P(x,y)\log_2\left(\frac{P(x,y)}{P(x)P(y)}\right).$$

Intuitively speaking (Fig. 5), mutual information measures the overlap between the two variables, the information in X about Y and vice versa. Put in the context of clustering evaluation, the mutual information of a clustering $D = \{D_i,\ 1 \le i \le k\}$ and its original class labeling $C = \{C_j, 1 \le j \le m\}$ is

$$I(C,D) \equiv -\sum_{j=1}^{m}\sum_{i=1}^{k}\frac{n_{ij}}{\sum_{j=1}^{m}|C_j|}\log_2\left(\frac{n_{ij}\sum_{j=1}^{m}|C_j|}{|D_i|\bullet|C_j|}\right),$$

where n_{ij} is the number of elements in cluster i with class label j. $I(C, D)$ can be interpreted as the information about an element's class label if its cluster membership is known. This is very close to the agreement measure we are looking for, except that one piece is still missing. Since it measures the absolute size of the overlap, two large shapes may have an overlap smaller in proportion yet bigger in size than two small shapes. Hence, we further define the normalized mutual information (NMI) of C and D

$$NMI(C,D) \equiv \frac{I(C,D)}{\min\{H(C),H(D)\}},$$

as a measure of agreement between the clustering D and its original labeling C. The NMI has the following desirable properties.

- It is bounded by [0, 1].
- Its baseline is stable, and not sensitive to the size of document set, the number of classes, the number of clusters, *etc.*
- Random guesses get zero credit.
- Clustering with more clusters than there are classes can still get a perfect score (1.0) as long as each cluster contains only elements of one class.

4.4 Evaluating GeneNarrator I

Since NMI is defined on two partitions, it cannot be used directly on the gold standard datasets constructed above, the yeast gene list and the citation set. Neither is a partition because some elements are in more than one class. A work-around trick is to make them pseudo-partitions by duplicating those elements and assigning a copy to each of their containing classes. Of course, no clustering algorithm could be expected to differentiate those duplicated elements; and they will be assigned to the same cluster, resulting in mixing of the classes to some extent. Therefore, for a non-partitionable dataset, perfect clustering is not achievable, i.e., NMI < 1. The best achievable NMI for such a dataset can be estimated using simulated clustering, in which the non-duplicated elements keep their class labels and duplicate elements are assigned to the same class (randomly picked from the classes they actually belong to).

We estimated the best possible NMIs with five simulated clusterings for the gold standard gene list and the gold standard citation set (Table 10).

Table 10. Best NMIs achievable for the gold standard datasets

	Agreement with the standard (NMI: mean ± SD)
Gene list	0.882 ± 0.005
Citation set	0.843 ± 0.001

The effectiveness of GeneNarrator I was evaluated with five trials on the gold standard datasets. The command line options and the results are listed in Table 11. The agreements with the standard gene list and the citation set were not impressive, only 0.62 for the 1st–step clustering and 0.52 for the 2nd-step clustering, respectively (only 74% and 59% out of the best achievable). The result was not surprising when we looked at the consistency of the 1st-step clustering among the five trials (only 0.67 ± 0.02). If the very first step of the analysis cannot even produce consistent results, there is little to expect for the subsequent steps.

Since BOW-based GeneNarrator I did not produce especially high results in the evaluation, the question of whether the concept-based GeneNarrator II could do a better job became the basis for the next experiment.

Table 11. Evaluation of GeneNarrator I on the gold standard datasets (5 trials)

	1st-Step (Text) clustering	**2nd-Step (Gene) Clustering**
Command line parameters	LongBOW: • SWT *df* cut-off = 0.05 • DWT *p* value = 0.025 • DWT *tf* ratio = 3.0 • minimum word length = 2 • default stopword lists CrossBOW: • branch factor = 2 • maximum branch depth = 4	GeneSmith: • algorithm = *k*-means • *k* = 15
Consistency	0.67 ± 0.02	0.52 ± 0.04
Agreement	0.62 ± 0.01	0.52 ± 0.08
% Achieved	74%	59%

Chapter 5 Concept-based text clustering

Everyday experience tells us that background knowledge plays a role in manual grouping of documents. For example, documents about *red blood cells* should be grouped together with those discussing *erythrocytes*, since they are exact synonyms. It is also very common for a human reader to group closely related subjects under a bigger topic, e.g. to put documents about subjects *apple*, *orange* or *peach* under the topic *fruit*. Such knowledge is usually not explicitly stated in the documents, but widely accepted among the domain experts, and in some domains captured to some extent in the format of ontologies, e.g. the unified medical language system (UMLS) [50], the medical subject headings (MeSH) [49], the gene ontology (GO) [3,20,42] and the open biological ontologies [1] in the biomedical domain.

As discussed in the chapter 2, BOW-based text representations are incapable of capturing ontological information. Hence there is no way for BOW-based text clustering tools to take advantage of this background knowledge. For example, the words *apple*, *orange* and *peach* do not give a BOW-based algorithm any hint that they belong together under *fruit*. Although these documents might still end up together in the same cluster, that would probably be because they share many other common words, such as *vitamin*, *nutrition*, *healthy*, *tree*, *diet*, *etc*. The main concept words (*apple*, *orange* and *peach*) would play no role or even negative roles in the clustering. This was possibly one of the causes that GeneNarrator I performed inconsistently in the first-step clustering, since synonyms are widely used in MEDLINE citations and biomedical concepts are highly related. Concept-based methods might be expected to be able to help solve this problem.

Another potential strength of a concept-based text representation is that it might filter out irrelevant words. With the choice of an appropriate ontology, only the information relevant to the purpose need to be kept in the representation. Therefore, the clustering algorithms could also focus on what is important, and not be distracted by irrelevant keywords. Although similar effects may be partially achieved in BOW representation with an extended stopword list for filtering out "unwanted" keywords, it seems likely to be infeasible in a practical sense. Except for a relatively few very common stopwords (e.g. *be*, *of*, *not*, *that*, *etc.*),

there is no consensus about what is "unwanted." It is different from user to user, or even occasion to occasion for the same user. In addition, there is no community effort toward developing comprehensive stopword lists. On the other hand, it is a hot area right now in developing biological ontologies, which can be utilized for our purpose.

Our next goal was then to implement a concept-based clustering approach, and to see whether it can improve GeneNarrator's performance. As previously pointed out in chapter 2, there are three challenging technical requirements for a concept-based text clustering system (repeated here for convenience).

1. An ontology that encodes sufficient background knowledge about the domain of interest (genomics in our case).
2. A natural language processing algorithm that efficiently and effectively maps free text to ontology concepts.
3. A text-clustering algorithm that can effectively utilize the ontology.

These issues are discussed in the next three subsections.

5.1 Choice of ontology

Since we intended to develop a functional genomic analysis system, the Gene Ontology (GO) seemed natural to consider as a choice of ontology. The GO is a hierarchical controlled vocabulary for describing genes and their products from three aspects (three branches): biological process, molecular function and cellular component [3,20,42]. However, further evaluation revealed that GO's development was immature and had intrinsic design flaws.

5.1.1 Imbalance in the development of GO

GO was first proposed in 1998 by a joint force associated with several model organism genomic databases, Drosophila Genome Database (FlyBase)[4], Mouse Genome Informatics (MGI)[5], Saccharomyces Genome Database (SGD)[6] and the Institute of Genomic Research

[4] http://flybase.bio.indiana.edu/

[5] http://www.informatics.jax.org/

(TIGR), in response to the explosion of biological data resulting from technological advancement in sequencing, microarrays, proteomics and metabolomics. Seven years later, GO has grown tremendously. In its 9/2005 release, there were 19,465 concepts. Excluding unusable ones (such as "obsolete" and "relationship"), 18,455 concepts can be used for annotating genes and their products.

The first evidence of imbalance is the huge variation in leaf depth. A leaf concept is one without any child concepts. There are 11,369 leaves out of 18,455 total concepts. The depth of a leaf concept is defined as the maximum number of arcs to the root concept ("all"). The deepest leaf concepts have 18 steps (GO:0045856 = "positive regulation of pole plasm oskar mRNA localization" and GO:0045855 = "negative regulation of pole plasm oskar mRNA localization"), while the shallowest have only 2 steps (GO:0030533 = "triplet codon-amino acid adaptor activity", GO:0031386 = "protein tag", and GO:0045735 = "nutrient reservoir activity"). The average leaf depth is 7.67 ± 2.27 (mean ± SD). The variation reflected the fact that in some areas GO has gone to unnecessarily detailed levels (the necessity will be discussed shortly), while in other areas GO just scratched the surface. For example, one group [32] stated that GO had insufficient coverage in immunology. While building an automatic system for genome annotation and pathway identification, another group [44] decided to use the KEGG Orthology, instead of GO, as the controlled vocabulary because GO concepts did not match known pathways directly. It is arguable that GO cannot and should not cover everything. Immunology is a huge domain deserving an ontology of its own. However, there is no excuse for not covering pathways. Isn't it a major category of biological process?

The structural imbalance of GO is also shown from its use in annotation of gene/products. Ideally, concepts should be used at the same level of frequency. If some concepts are used extremely frequently, it is a strong indication that these concepts should branch into more detailed ones. On the other hand, if some concepts are rarely used, they should be eliminated from the ontology, and their parent concepts be used for annotations instead. It is also to be hoped for that the GO concepts annotated to a gene/product should be

---

[6] http://www.yeastgenome.org/

as specific as possible, i.e. using leaf concepts whenever possible. However, the current status of GO annotation is far from supporting such a scenario.

There are 25 public databases, including many model organism genomic databases such as FlyBase, SGD, MGI, and others contributing 7,577,336 annotations at the GO consortium website (in the 9/2005 release). The biggest contributor is the Universal Protein Resource (UniProt)[7] with 5,940,839 annotations, and the smallest contributor is AgBase[8] with 109 annotations. The total number of annotated genes/products is 1,926,085 (on average, 3.9 concepts per gene/product). Out of 18,455 usable concepts, 10,988 (59.5%) were actually used. Table 12 summarizes the usage of GO concepts in the annotations, breaking down to the three major branches. The most surprising fact was that about 40% of the concepts, including leaf concepts, were never used in any annotation. If they were never used, why would they be introduced into the GO in the first place? A closer look at some unused concepts might give us some clues.

Table 12. Usage of GO concepts in annotation

Branch	Usable	In use	%
Biological process	9,805	5,200	53.0%
Molecular function	7,076	4,685	66.2%
Cellular component	1,574	1,103	70.1%
Total	18,455	10,988	59.5%
Leaf	11,369	6,487	57.1%

One of the unused concepts was *nuclear translocation of MAPK during cell wall biogenesis* (GO:0000201). It was a subconcept of biological process, and its depth was 12. Using the concept as a query to PubMed returned 0 hits (on 03/02/2006). It is no wonder that the concept was not used in any annotations, because no one ever talked about it in the entire MEDLINE. The concept was also a good example for examining the structure of GO concepts, illustrating an intrinsic design flaw of GO.

---

[7] http://www.pir.uniprot.org/

[8] http://www.agbase.msstate.edu/

5.1.2 A critical defect of GO

Before we analyze the concept concept, let's pull back a step and think about what GO, as a controlled vocabulary, should be like. A controlled vocabulary is basically a simplified (restricted) language such as might be applicable to a specific domain. A language needs nouns to represent various entities (concrete or conceptual) in the world (domain), and probably fewer verbs to describe relationships among entities. The number of verbs needed depends on the types of relationships in the domain. For example, in the *Linnaean* taxonomy (a special type of ontology), there is only one type of relationship, subsumption, called "isa" in artificial intelligence. Is species *A* in genus *B*, which in turn is in family *C*? In other words, is it the case that *A* isa *B* isa *C*? Hence, the edges in a taxonomy tree can be interpreted as meaning isa, and this is sufficient for the *Linnaean* taxonomy. GO provides two verbs, "is_a" for the subsumption relationship, and "part_of" for the part-whole relationship. Are they enough?

Now let's have a closer look at the GO concept *nuclear translocation of MAPK during cell wall biogenesis* (GO:0000201). It actually doesn't name an abstract conceptual "entity", but tells a "story" involving spatial and temporal interactions among three entities (nucleus, MAPK, and cell wall). This looks funny from a language point of view. A speaker cannot tell a story by organizing nouns and verbs into sentences, because there are no verbs. All s/he can do is to create a new, giant, funny-looking noun. Such nouns contain so much information; they may be good only for one-time use. Therefore, the language is inflexible and lacks expressiveness.

Another consequence of the lack of relationships is overuse (hence abuse) of the existing ones, especially "part_of." The following examples show that "part_of" is used with different meanings.

- *Extracellular organelle* (GO:0043230) is "part_of" *Extracellular region* (GO:0005576)
- *mRNA editing complex* (GO:0045293) is "part_of" *cytoplasm* (GO:0005737)
- *Establishment of cellular localization* (GO:0051649) "part_of" *cellular localization* (GO:0051641)

The first example is a misuse. How can an organelle have a part-whole relation with a space? It is as ridiculous as saying "I am part_of my car, because I sit in it." It is not diffi-

cult to guess that the actual intention was to say, "the organelle IS LOCATED IN or OCCUPIES the region." However, since there are no other verbs available for describing spatial relationships, abusing "part_of" is inevitable. The second and the third examples are not wrong, but their meaning is quite different (spatial vs. temporal). Misuse and ambiguity may pose challenges for computer programs that automatically process annotations.

The unbalanced coverage of GO in many areas is just a matter of growing pains. Given time and resources, there is no doubt that GO can eventually cover the domain evenly. The lack of verbs, however, is a more serious defect, one which has severely hampered GO's development.

There are two possible explanations for the design of GO with only two types of relations (*is_a* and *part_of*). The first explanation is that the original designers considered the subsumption relation and part-whole relation sufficient in the world of GO. Such a simplified view of the domain was probably borrowed and extended from the *Linnaean* taxonomy, in which only the subsumption relation is needed. The later development of GO outgrew the domain boundaries drawn by the original designers. Hence, other types of relationships (i.e. spatial and temporal relations) sneaked in, resulting in ill-formed concepts.

An alternative explanation is that the original GO design was "naïve" [42]. It borrowed and extended the structure of the *Linnaean* taxonomy without considering whether or not that structure could meet the requirements of GO. Plausibly, there was no real requirements analysis at all. The genomic research community was in such need for a controlled vocabulary that GO was embraced without sufficient critical analysis, even though it had such a severe defect.

5.1.3 MeSH is the choice

The ill-formed GO concepts were difficult to work with. Its unbalanced and incomplete coverage of the biomedical domain made itself questionable as a source of background knowledge to guide text clustering. Hence, we turned to other alternatives, such as the open biological ontologies (OBO) [1], unified medical language system (UMLS) [50] and medical subject headings (MeSH) [49].

OBO is currently (as of Mar. 2006) hosting 55 ontologies (including the three major branches, *cellular component*, *molecular function* and *biological process* from GO) in seven formats (GO, OBO, protégé, XML, OWL, html and plain text). Some of them are targeted to very specific domains for specific species, such as *Arabidopsis gross anatomy*, *Mosquito gross anatomy*, and *Plasmodium life cycle*. There is obvious overlap among some ontologies. For example, there are four plant anatomy ontologies (*Arabidopsis gross anatomy*, *Cereal plant gross anatomy*, *Maize gross anatomy* and *Plant structure*) and two mouse anatomy ontologies (*Mouse adult gross anatomy*, *Mouse gross anatomy and development*). There seems little coordination among the development teams. More seriously, there is a lack of principles, guidelines or well-established best practices in biological ontology design. It is no surprise that we could not find a suitable ontology for GeneNarrator in OBO.

The unified medical language system (UMLS) was developed and is distributed by the national library of medicine (NLM). One of its components, Metathesaurus, is a very large vocabulary database containing biomedical and health related concepts (over 1 million concepts), their synonyms (5 million, some in multiple languages), and their relationships. It is built from more than 100 controlled vocabularies used in patient care, health services billing, public health statistics, indexing and cataloging biomedical literature, and /or basic, clinical, and health services research. It is too much for GeneNarrator.

The last candidate is the Medical Subject Headings (MeSH) thesaurus, also developed at NLM and one of the source vocabularies of UMLS. It seemed likely to serve GeneNarrator well due to its following characteristics.

- MeSH is designed for indexing, cataloging and searching biomedical literature, partially overlapping with GeneNarrator's purposes.
- With over 50 years of development and evolution since 1954, MeSH is much more mature than the newly developed ontologies, such as GO.
- The collection of synonyms for MeSH concepts is extensive.
- Most MEDLINE citations have a list of MeSH concepts assigned by human experts, which can be of great help in dealing with the next technical challenge, mapping free text to ontology concepts.

5.2 Concept mapping

5.2.1 MMTx

MetaMap Transfer (MMTx) is a supplemental project to the UMLS developed at NLM. It maps free text to the UMLS Metathesaurus concepts. In other words, it discovers what Metathesaurus concepts are discussed in the text. The input text is processed in MMTx through a pipeline of modules. First it is parsed into smaller and smaller components such as paragraphs, sentences, phrases, and tokens. Then UMLS concepts containing one or more of the tokens (or their variants) are retrieved as candidates, and the candidates are scored against the phrases. The final mapping is put together with the best candidates, like fitting together a puzzle, so that as many of the tokens in the original text are covered as possible, yet without overlapping. The process is illustrated with a sample sentence in Fig. 6.

The sentence was first broken down into four phrases (red bars). Then all UMLS concepts that contain at least one token in the phrases were retrieved as candidates (colored in blue). A token might retrieve multiple candidates (RNase); and variants of a token were also retrieved (*mediate*, *mediator* and *mediation* for mediated). The candidates were scored

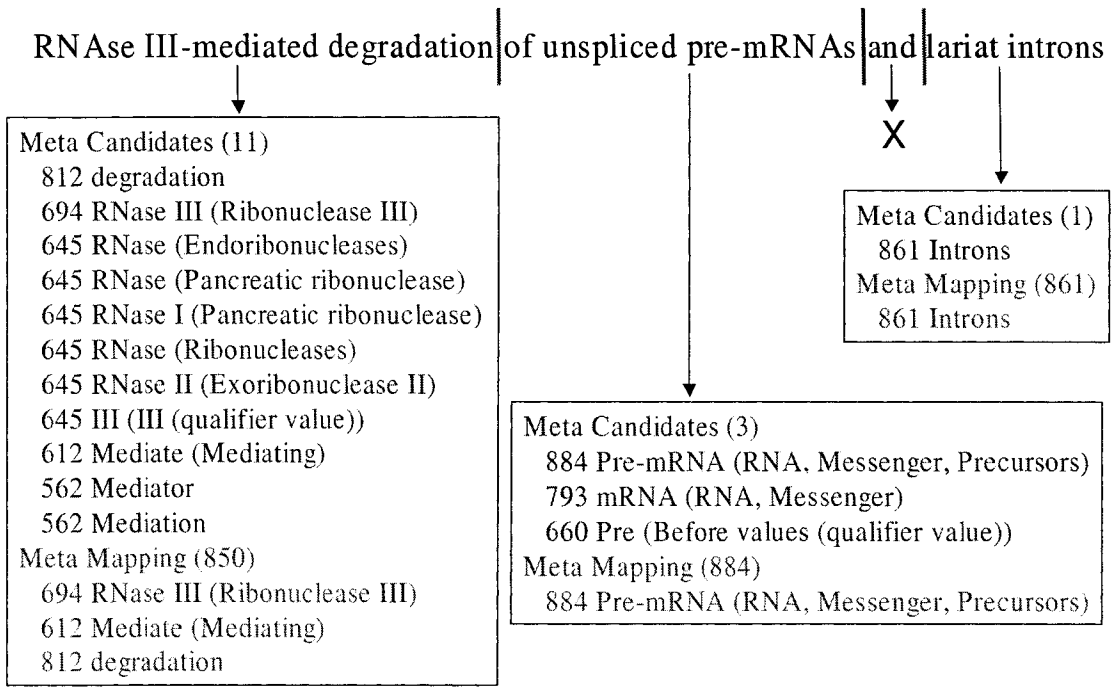


Figure 6. MMTx maps free text to UMLS concepts

based on the number of matching tokens and whether they were original tokens or variants. Finally, the first phrase was mapped to the three best candidates (*RNase III*, *mediate* and *degradation*), which covered all the tokens in the phrase without any overlap. The other phrases were processed similarly except that some tokens were not covered. Although MMTx can be configured to use other thesauri (e.g. MeSH) for GeneNarrator, its mapping strategy has some limitations.

- The mapping is limited within individual phrases; no concepts can cross phrase boundaries. In the above example, the MeSH concept *mRNA degradation* (D020871) was not discovered, because its constituent terms were in two phrases. Instead, two separate concepts (*mRNA* - D012333 and *degradation* - Q000378) were mapped. Yet there is no doubt that the single concept *mRNA degradation* contains richer information than the two separate ones.
- There is no compelling reason to avoid overlapping concepts (i.e. concepts sharing some tokens). In fact, token sharing is very common in MEDLINE citations. For example, in *nicotinic and muscarinic acetylcholine receptors*, the two underlined tokens are shared by two concepts. MMTx would find *muscarinic acetylcholine receptors* (D011976), but *nicotinism* (*nicotine poisoning*) instead of *nicotinic acetylcholine receptors*.

The limitations forced us to design our own concept-mapping module, the MeSH Miner.

5.2.2 MeSH Miner

Designed to overcome the limitations of MMTx discussed above, the MeSH Miner maps MEDLINE citations to MeSH concepts. It consists of two components, a normalized MeSH lexicon and a mapping module.

The lexicon normalizes MeSH concepts' names and synonyms, converting each name (or synonym) into a bag of tokens in their base format using the SPECIALIST Lexicon tool[9] (a component of UMLS). For example, the concept *Adverse Drug Reaction Reporting Sys-*

[9] http://www.nlm.nih.gov/research/umls/meta4.html

tems (D016907) is normalized to {adverse, drug, reaction, report, system}. All tokens in the lexicon are assigned frequency-based scores: $S(t) = \max\{(10 - \log_2 F), 0\}$, where F is the number of concepts containing the token t.

The mapping module's processing workflow is illustrated in Fig. 7. It takes a citation's title, abstract and manually curated MeSH list as input. The title and abstract are parsed into individual sentences. The mapping is done sentence-by-sentence; no mapped concepts can cross two or more sentences. A sentence is first tokenized; and the tokens are normalized to their base forms. Then all concepts containing one or more of the tokens are retrieved and scored as candidate concepts. A candidate's score is the sum of the scores of the matched tokens scores minus the scores of the unmatched tokens. There is an added bonus if all tokens are matched, and a penalty if other tokens are mixed in between.

$$S(C) = \sum_{t \in \text{matched}} S(t) - \sum_{t \in \text{not\_matched}} S(t) + \text{bonus(all\_match)} - \text{penalty(gap)}$$

The bonus and the penalty can be fixed scores, or can be based on the number of tokens matched or mixed. Candidates with scores lower than a predefined threshold are discarded. Also discarded, optionally, are candidates whose direct or indirect children are also candidates but with better scores. Any candidates not discarded are the mapped concepts of the sentences. Finally, the concepts in the human-curated MeSH list are optionally added to the mapped ones if they are not already included. An optional branch filter can be applied to the mapping module so that only the concepts from the selected MeSH branches (e.g. *Anatomy*, *Diseases*, *Chemicals and Drugs*, and *Biological Sciences*, *etc.*) are mapped.

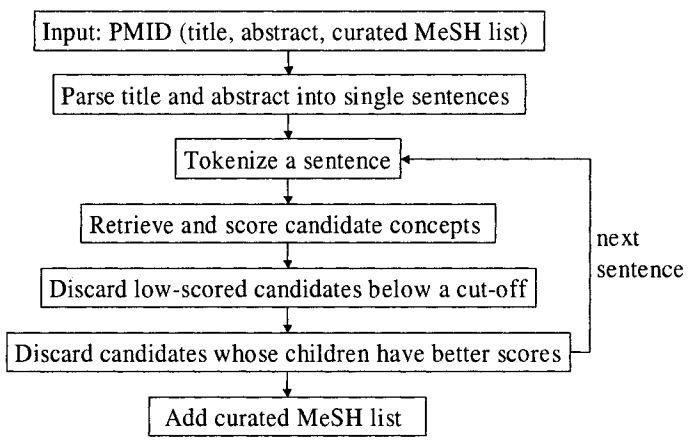


Figure 7. Processing workflow of MeSH Miner

MeSH Miner was tested on a sample sentence "RNAse III-mediated degradation of unspliced pre-mRNAs and lariat introns", and compared against MMTx (Table 13). The all-match bonus and the gap penalty were fixed at 20 and 5, respectively, and the cut-off score was set at 10. Most concepts mapped by MMTx were also identically discovered by MeSH Miner, except that the concept *degradation* was replaced by the more specific *mRNA degradation*. MMTx could only map to *degradation* because of the limitation of phrase boundaries. MeSH Miner was designed to ignore the phrase boundaries so that more complex and/or specific concepts could be discovered.

Table 13. Comparison between MMTx and MeSH miner

MMTx		MeSH Miner	
Score	**Concept**	**Concept**	**Score**
812	degradation	mRNA degradation	37
884	Pre-mRNA	Pre-mRNA	30
861	Introns	Introns	29
694	RNase III	RNase III	28
612	Mediate (Mediating)	Mediating	27
		mRNA	25

5.3 Concept-based representation and clustering

With titles and abstracts mapped to MeSH concepts, MEDLINE citations can be represented using a vector space model in which each citation is viewed as a vector of concepts (a VOC):

$$d_i = (C_{i1}, C_{i2}, ...C_{ij}, ...C_{in}), \quad i = 1,2,...,m$$

where d_i is the i^{th} document in a set of m documents, and the vector space consists of n concepts. Although the VOC representation looks quite similar to the BOW representation, there is a major difference. While the elements in a BOW representation are orthogonal, those in a VOC representation may be related by an ontology. For example, BOW views *apple*, *orange* and *baseball* as three unrelated things. The difference between *apple* and *orange* is the same as that between *apple* and *baseball*. This is not the case for VOC, because the ontology has the knowledge that *apple* is closer to *orange* than to *baseball*.

It is, however, not immediately clear how the knowledge provided by the ontology can be utilized to cluster the vectors. Most clustering algorithms, such as *k*-means and agglomerative hierarchical algorithms, are distance-based, relying on the orthogonal assumption for distance calculations. Although the CAM algorithm does not explicitly depend on the orthogonal assumption, it has no place to utilize an ontology, hence implicitly assumes that all concepts are unrelated. To take advantage of the ontology, a method called "concept upgrading" was developed, which attempts to mimic how a human expert utilizes background knowledge in clustering documents.

Consider how an expert might cluster a set of documents. If there were enough documents discussing *apple*, *orange* or *peach* to form clusters by themselves, he/she would simply treat them as unrelated and put them into separate clusters. If there were too few documents for that, s/he would put documents about *apple*, *orange* or *peach* together under a more general topic like *fruit*. The same idea is implemented in "concept upgrading." If a concept's document frequency (the number of documents containing the concept) is below a pre-defined threshold, the concept is aggregated (upgraded) to its parent concept. If a parent has aggregated all its children and is still below the threshold, the parent itself will be upgraded. The upgraded vectors can then be clustered using the CAM algorithm.

5.4 Evaluation of concept-based text clustering

The above strategies for concept mapping and clustering were implemented in GeneNarrator, and evaluated on the gold standard citation set with four slightly different tests. The optional branch filter (*Anatomy*, *Diseases*, *Chemicals and Drugs*, and *Biological Sciences*) was applied to all four tests. A preliminary trial indicated that CrossBOW could not split the citation set at the root level under the default parameters (branch-factor = 2, max-branch-depth = 4) if the filter was not applied. The manually curated MeSH list was not added to the mapping in test 1. Test 3 and 4 tried concept upgrading with two different *df* threshold (5 and 10, respectively). Each test was run for three or five trials. The test results in terms of self-consistency among trials and agreement with the standard are summarized in table 14, which also provides a comparison of these with the BOW-based clustering results.

Table 14. Evaluation of concept-based text clustering in comparison with BOW-based clustering

	Test 1	Test 2	Test 3	Test 4	BOW
Branch filter	yes	yes	yes	yes	
MeSH list	no	yes	yes	yes	
Upgrade threshold	N/A	N/A	5	10	
Self consistency	0.67 ± 0.02	0.68 ±0.02	0.69 ± 0.02	0.68 ± 0.04	0.67 ± 0.02
Agreement	0.51 ± 0.01**	0.54 ± 0.02**	0.52 ± 0.01**	0.52 ± 0.02**	0.62 ± 0.01

** $p < 0.001$ compared to BOW-based result (two sample t-Test assuming equal variances)

Several observations are evident from the above results.

- Self-consistency of the clustering was not affected.
- Adding the human-curated MeSH list or concept upgrading had little effect on either consistency or agreement.
- Concept-based clustering significantly decreased agreement with the standard.

A possible cause of the decreased agreement is that MeSH concepts could not cover all the details in the text; some detailed information, which might be of value to the clustering, was lost in the process of concept mapping. MeSH is designed for indexing and categorizing the entire MEDLINE database (~17,000,000 citations). Thus comprehensive coverage of all details is impractical and unnecessary. There are a total of 23,885 concepts in MeSH (2006 edition), out of which ~15,000 are under the branches we are interested in (*Anatomy*, *Diseases*, *Chemicals and Drugs*, and *Biological Sciences*). The total number of concepts mapped to the gold standard citation set was about 1,500, while the BOW representation had about 15,000 dimensions. In particular, many protein or gene names were lost. A concept-based text representation without loss of relevant and important details is desirable.

5.5 Hybrid representation and clustering

A hybrid text representation, called VOWC (vector of words and concepts), was then designed to address the above requirement. VOWC is a straightforward extension of VOC, involving only the following small modification to the MeSH Miner. After a citation is

mapped to a VOC, unmapped tokens are not thrown away but instead are appended to the vector, converting it to a VOWC.

Table 15. Comparison of clustering performance among three methods of text representations.

Representation		No filter	Filter		
			$df > 5$	$df < 500$	$5 < df < 500$
Consistency	BOW	0.67 ± 0.02			$0.69 \pm 0.01^{+}$
	VOC	0.68 ± 0.02			
	VOWC	0.67 ± 0.01	$0.72 \pm 0.02^{++}$	$0.69 \pm 0.01^{+}$	$0.73 \pm 0.02^{++}$
Agreement	BOW	0.62 ± 0.01			0.61 ± 0.01
	VOC	$0.54 \pm 0.02^{**}$			
	VOWC	0.61 ± 0.01	0.63 ± 0.01	$0.64 \pm 0.00^{*}$	0.62 ± 0.02

$^{+}$ * $p < 0.05$ compared to BOW-based result (two sample t-Test assuming equal variances).

$^{++}$ ** $p < 0.001$ compared to BOW-based result (two sample t-Test assuming equal variances).

VOWC was also evaluated on the gold standard citation set. In addition, a document frequency (*df*) filter was also tested. The filter discards concepts (and words) in the documents if they appear too frequently or too rarely in the entire citation set. The evaluation results were tabulated in Table 15, and the following observations may be made.

- VOWC permitted clustering agreement, lowered in VOC, to the same level as in BOW, indicating that MeSH concepts could not cover all the details necessary for the clustering.
- In terms of clustering consistency, the three representations had no differences.
- Filtering too frequent and/or too rare concepts (words) could improve self-consistency of the clustering. Although statistically significant, the improvement was too marginal to have much impact.
- The effect of filtering on clustering agreement was hardly noticeable.

It was perhaps disappointing to see that neither the VOC nor the VOWC representations improved text clustering much in terms of consistency and agreement. The only improvement in consistency came from *df* filtering, which has nothing to do with the concept-based representations. Yet this improvement was too marginal to have impact on the agreement, let alone on further enhancing the gene analysis. A new strategy was needed.

6.2 Implementation

Given a set D of n documents and a hierarchical clustering algorithm A, the multi-clustering algorithm identifies a certain percentage (p) of the documents as the "core" set in the following steps.

1. Construct a fully connected, undirected, weighted graph $G = \{V, E\}$, where each vertex v in V is mapped one-to-one to a document d in D. Thus there is an edge e_{ij} in E connecting the pair of vertexes v_i and v_j for each i and j. All edge weights are initialized to 0.
2. Run A on D once and get a solution. Update the edge weights based on relationship of document pairs in the solution. $w_{ij} := w_{ij} + r_{ij}$, where r_{ij} is the level of the lowest common parent node of documents d_i and d_j in the cluster hierarchy (Fig. 8).
3. Repeat step #2 k (user defined) times.
4. Remove E from G. Sort all edges in E by weight descendingly. Define an empty set C of core documents.
5. Restore edges from E into G one-by-one in the sorted order, connecting vertexes to small islands and small islands to big islands. Once the size of a growing island exceeds a pre-defined threshold s, it is removed from G and the corresponding documents added into C. All edges involving the vertexes in the island, whether they have been put back in G or remain in E, are discarded.
6. Repeat step #5 until the size of C exceeds the target ($p \times n$).

The resulting islands in the core set C can be treated as a flat clustering solution, in which the documents remaining in the graph is regarded as forming a "miscellaneous" clus-

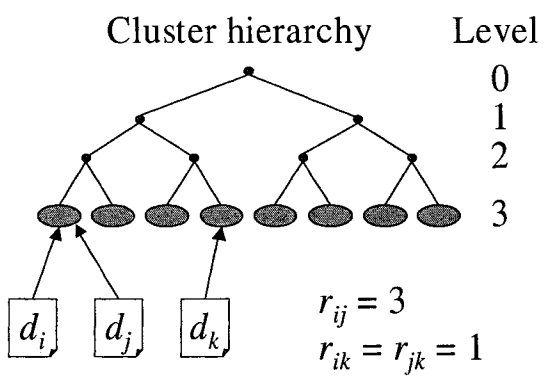


Figure 8. Relationship of document pairs in a hierarchical clustering solution

ter. If a hierarchical clustering solution is still needed, the original algorithm can now be applied to C, which is D with the outliers removed.

6.3 GeneNarrator II

The *df* filtering module and the multi-clustering module, as well as the MeSH miner (concept mapping module), were integrated into GeneNarrator II (Fig. 9). Some modules were modified from GeneNarrator I (DocBuilder II, ArrowSmith II and VOWCViewer), while CrossBOW and GeneSmith were imported without change. The modules are summarized in Table 16.

Table 16. Description of GeneNarrator II modules

Module	Description	Modification/Addition
DocBuilder II	Retrieve MEDLINE citations relevant to a list of genes.	Modified to extract manually curated MeSH list.
MeSH Miner	Identify MeSH concepts in free text. Combine them with manually curated MeSH concepts and unmapped terms to convert a citation into a VOWC.	New module (replacing LongBOW).
df Filter	Discard too frequent or too rare concepts/terms from the VOWC representations.	New module.
Multi-clusterer	Identify and discard outlier citations through multiple runs of CrossBOW.	New module.
CrossBOW	Text clustering module implementing CAM algorithm.	Not modified.
GeneSmith	Calculate genes' topic distributions. Cluster genes based on their topic distributions.	Not modified.
ArrowSmith II	Extract representative keywords for topics. Score and rank representative sentences and citations.	Modified to deal with mapped MeSH concepts.
VOWCViewer	GUI interface for exploring the analysis results.	Modified for MeSH concepts.

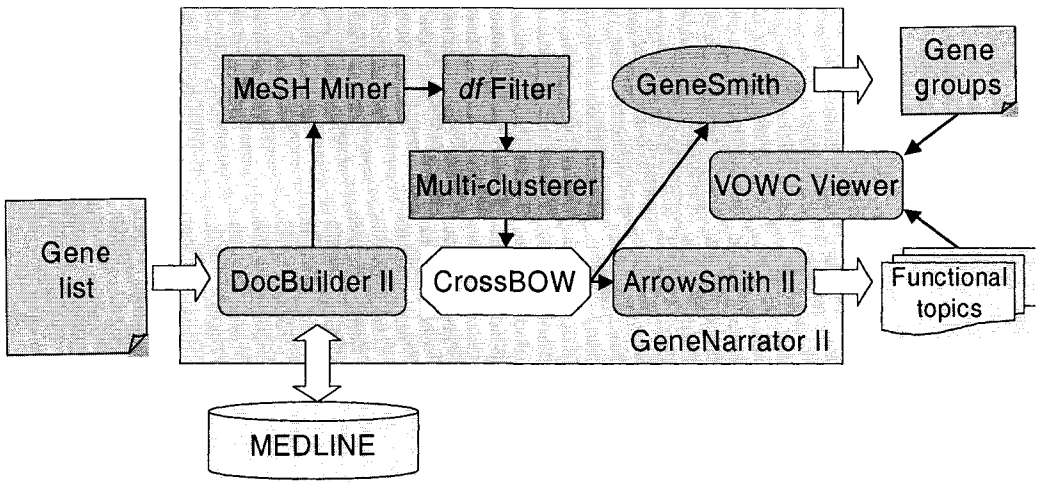


Figure 9. Architecture of GeneNarrator II

6.4 Evaluation of GeneNarrator II

6.4.1 Text (1st-step) clustering

GeneNarrator II was again evaluated on the gold standard citation set. Various parameters of *df* filtering and multi-clustering were tested in order to find the best combination for the most improvement. The evaluation workflow was illustrated in Fig. 10. The original citation set was first converted to VOWC representation using the MeSH miner. Then *df* filtering was applied to the converted set to remove too frequent and/or too rare concepts and terms (high-pass: $df > 5$, low-pass: $df < 500$, and mid-pass: $5 < df < 500$), resulting in three copies of the representation. Next, each copy underwent multi-clustering using the CAM (in CrossBOW) algorithm with $k = 10$; and four core document sets of various sizes (60%, 70%, 80% and 90% of the original size) were extracted. As a control condition, four subsets of various sizes (60%, 70%, 80% and 90% of the original size) were randomly sampled from the mid-pass set. Finally, the core sets and the random sets, as well as the three filtered sets, were clustered using CrossBOW 3 to 5 times for each set. Self-consistency and agreement with the gold standard were calculated and compared for each condition.

The results are summarized in Fig. 11. The following observations are evident from

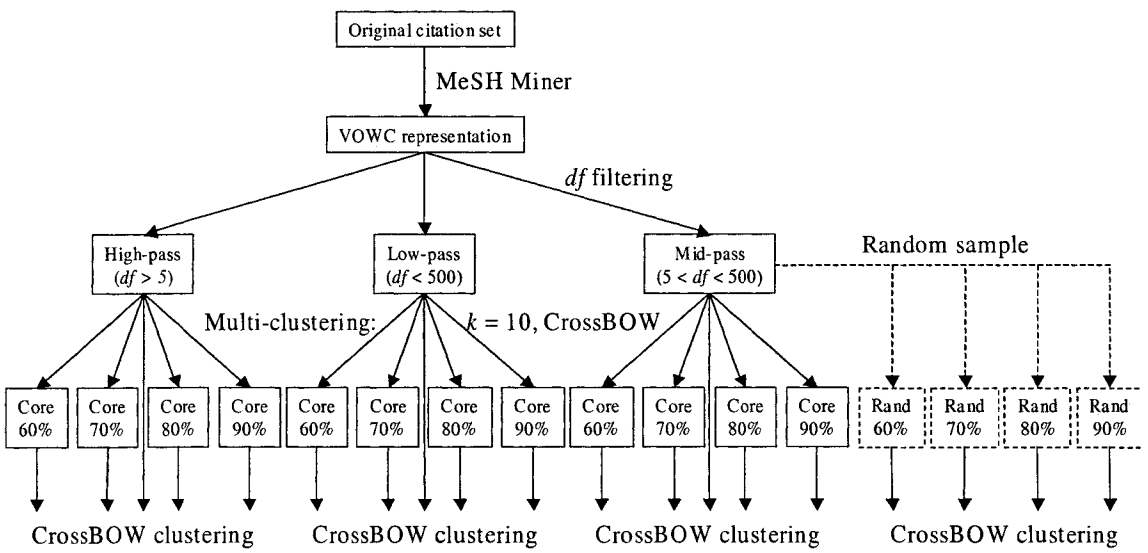


Figure 10. Evaluation workflow for the multi-clustering algorithm

the results.

1. Multi-clustering works! With the outliers removed, the core document sets (especially the 60% and the 70% ones) have much better clustering results in terms of self-consistency and agreement compared to the original document set. The improvement is significant both statistically ($p < 0.001$ for all non-random points at 60% and 70% compared to the corresponding points at 100% on the same curve) and practically (improvement greater than 15%).
2. Although not as significantly as multi-clustering, *df* filtering can further improve clustering, especially in terms of consistency.

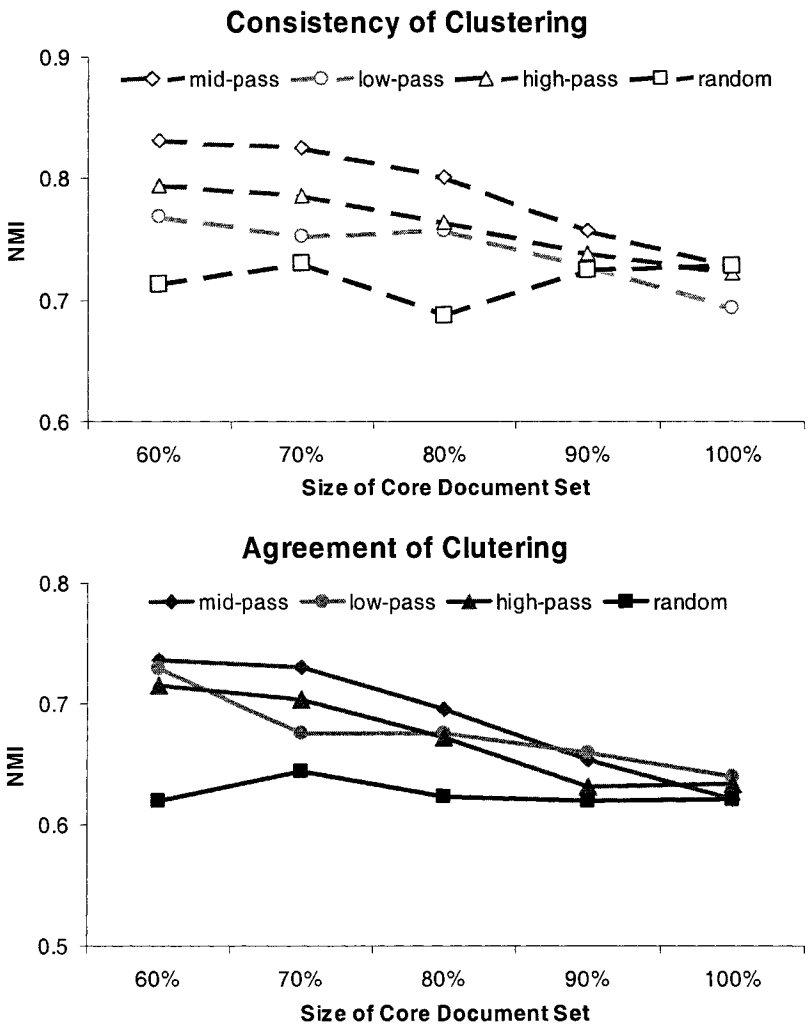


Figure 11. Effect of multi-clustering and *df* filtering on consistency and agreement of text clustering

3. The most improvement was achieved at the 60% core size of multi-clustering with mid-pass filtering of *df* with 0.83 ± 0.02 in consistency and 0.74 ± 0.01 in agreement, in comparison to 0.67 ± 0.02 (consistency) and 0.62 ± 0.01 (agreement) for BOW-based clustering in GeneNarrator I. This is about 84% (0.74 out of 0.88) of the best achievable agreement for the citation set of 0.88 (see Section 4.4). The improvement at 70% core size was about as good as 60% (0.83 ± 0.02 and 0.73 ± 0.01 in consistency and agreement, respectively).
4. The randomly sampled subsets did not show any improvement. It is thus safe to rule out the argument that improvement from multi-clustering is due to decreasing the size of the document set.

6.4.2 Gene (2nd-step) clustering

The performance of GeneNarrator I on the results of 2nd-step (gene) clustering was disappointing with 0.52 ± 0.04 and 0.52 ± 0.08 consistency and agreement, respectively (see section 4.4 for more details). The low performance was due largely to inconsistency and disagreement in the results of 1st-step (text) clustering. With the significant improvements that were obtained from multi-clustering and *df* filtering in the 1st-step clustering, it seemed interesting to question how much the improvement would propagate to the results of 2nd-step clustering. This was done as described next.

The GeneSmith module of GeneNarrator I was responsible for the 2nd-step clustering process. This is where the genes are clustered based on what document clusters they appear in. This process was applied to the multi-clustering results derived from the mid-pass VOWC representation set. The results are summarized in Fig. 12. The consistency of gene clustering was significantly improved compared to GeneNarrator I, e.g. from 0.52 ± 0.04 to 0.66 ± 0.04 at 60% core size ($p < 0.001$). However, no significant improvement was observed for the agreement of gene clustering, e.g. 0.52 ± 0.08 (BOW) vs. 0.52 ± 0.08 (60% core size).

The above results might sound disappointing, but they actually were much better if viewed from the perspective of topic hierarchy. This is illustrated in Table 17 with the seven

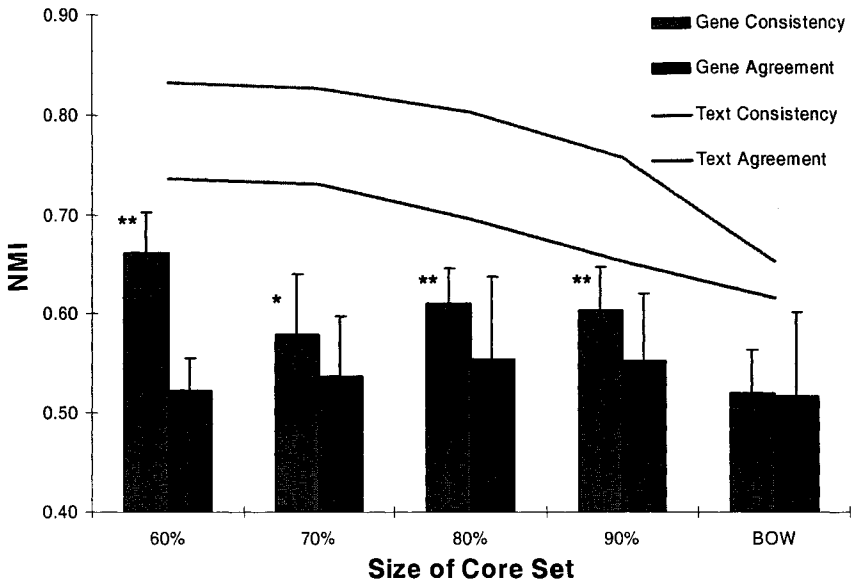


Figure 12. Evaluation of gene clustering based on *df*-filtered multi-text clustering.

* $p < 0.05$, ** $p < 0.001$ compared to the gene consistency of BOW.
(two-sample t-Test assuming equal variance, error bar = standard deviation).

genes from the ubiquitin-mediated proteolytic pathway, which have a relatively simple topic distribution in a single topic (/1/0/0/0/1). The genes contributed 87.2% (102 of 117) of the citations to the topic. Several factors lowered the agreement NMI resulting from not clustering them in the same group.

Table 17. Topic distributions of the genes in the ubiquitin-mediated proteolytic pathway

Gene	Gene group	Major Topic (%)
YDR394w	7	/1/0/0/0/1 (5/6 = 83%)
YGL048c	7	/1/0/0/0/1 (45/45 = 100%)
YKL145w	7	/1/0/0/0/1 (17/19 = 89%)
YOR117w	7	/1/0/0/0/1 (3/3 = 100%)
YOR259c	7	/1/0/0/0/1 (19/22 = 86%)
YGR270w	10	/1/0/0/1/1 (1/2 = 50%)
YDL007w	11	/1/0/0/0/1 (12/15 = 80%)

1. The 2nd-step clustering algorithm (*k*-means) is too affected by details, and not sensitive enough to the big picture. Thus it seems natural for a human expert to put gene YDL007w in the same group as the other group #7 genes. Yet the algorithm put it in another group, because either the main distribution in the topic (80%) was just below a threshold (between 80% and 83%), or the distribution of 3 other citations (≤ 20%) qualified it for group #11.

2. One or two citations' disagreement does not inordinately impact the overall agreement of text clustering. However, for genes with only a few citations, one or two citations can make big difference.
3. Although not obvious in the above example, the *k*-means algorithm fails to take advantage of the topics' hierarchical structure. In effect, topics are considered orthogonal to one another. Therefore if a gene made a major contribution to topic /1/0/0/0/0, it would not be recognized as closely related to topic /1/0/0/0/1, and hence would have little chance to be clustered into the relevant group.
4. The NMI metric itself does not take into consideration the hierarchy either. Therefore, if two neighboring nodes exchanged some elements (genes or citations), the actual disagreement would be much less than perceived from the lowered NMI.

Nevertheless, the majority of the membership was still captured in the above example. In addition, gene grouping based on functional topics is not the only important thing about GeneNarrator; the result of text clustering can be viewed for individual genes. In other words, users can browse a single gene's topic distribution, with or without considering its grouping within the set of genes.

6.4.3 Biological meaning of text clusters

The CrossBOW (text clustering) module can output its assessment of the most probable concepts and terms for each cluster (topic), and these can be used to describe the biological significance of the topic. GeneNarrator also scores sentences and citations for containing the concepts and terms. High-scoring sentences and citations may further help capture biological meanings. However, biological significance is difficult to evaluate in the same way as clustering consistency and agreement. Whether or not a list of concepts and terms makes biological sense is subjective, and might even vary over time for the same evaluator. Hence, we only give some example concepts/terms here and point out some interesting observations.

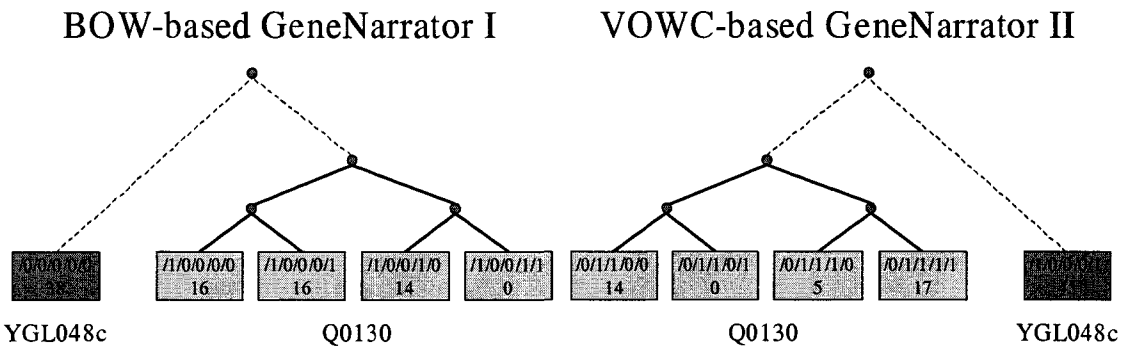


Figure 13. Comparison of two genes' topic distributions in BOW- and VOWC-based clustering

The example shown in Fig. 13 compared two genes' topic distributions in BOW-based GeneNarrator I and VOWC-based GeneNarrator II analyses. One gene (YGL048c) was from the ubiquitin-mediated proteolytic pathway. Another (Q0130) was from the respiratory chain pathway. YGL048c had 50 citations in the BOW representation and 45 in the VOWC representation; and Q0130 had 50 in BOW and 38 in VOWC. The reduction in the total number of citations was because the multi-clustering in GeneNarrator II discarded some outlier citations. The patterns of the distributions in both analyses were similar, with YGL048c concentrating in one topic and Q0130 spreading into several neighboring topics. The topologies of the branches in the two hierarchies were equivalent, even though they were labeled differently.

Table 18. Comparison of the keywords for the ubiquitin-mediated proteolytic pathway

BOW /0/0/0/0/0	VOWC /1/0/0/0/1
proteasom, *sug*, *rpt*, *atpas*, **ubiquitin**, rpn, transcript, protein, *gal*, mts, tbp, **proteas**, *cim*, **proteolysi**, ufd, receptor, **proteolyt**, yta, *tfiia*, pa, channel, ms, *regulatori_complex*, cad, msug, conjug, famili, activ_domain, *phosphoryl*, transcript_activ, toa, cap, **protein_degrad**, nucleu, **ubiquitin_protein**, manduca, pre, nob1p, ubr, ism, fza, *transcript_factor*, *bind_protein*, muscl, gankyrin, die, *nuclear*, put_atpas, *rna_polymeras_ii*, hormone	**proteasome**, *Adenosinetriphosphatase*, 26s, *sug1*, Endopeptidases, *Transcription Factors*, 20s, **Peptide Hydrolases**, **Ubiquitin**, *DNA-Binding Proteins*, **Ubiquitins**, Trans-Activation (Genetics), *Multienzyme Complexes*, Repressor Proteins, *sug2*, *gal4*, Cysteine Endopeptidases, *Transcription Factor TFIIA*, Protein Subunits, **proteasomal**, Tissues, RNA, *RNA Polymerase II*, cDNA, DNA Repair, *rpt1*, DNA-Directed RNA Polymerases, tfiih, *TATA-Box Binding Protein*, Gene Expression, *rpt4*, *rpt2*, cdc68, Protein S, Chromatin, lid, *rpt6*, mts2, Adenosine Triphosphate, Transcription Factor TFIID, *cim5*, *cim3*, *Nuclear Proteins*, *Phosphorylation*, **Hydrolysis**, Methylation, Protein Isoforms, aaa, **ubiquitinate**, Cell Line

The top 50 representative (most probable) concepts/terms of the topics may be compared by viewing Tables 18 and 19. Overlapping concepts/terms are highlighted in colors, with the red ones were directly related to the pathways. The respiratory chain pathway included three topics: mitochondrial genome, mitochondrial protein synthesis and mitochon-

drial ATP biogenesis. GeneNarrator I and II agreed quite well with each other on the topics, indicated by many highlighted (overlapping) concepts/terms.

Table 19. Comparison of keywords for the pathway of respiratory chain

BOW /1/0/0/0/0	VOWC /0/1/1/1/1 (mitochondrial genome)
cystein, trna, cystathionin, petit, cys, gene, sulfur, oah, methionin, enzym, **mitochondri_genom**, *genom*, lyas, plant, **mitochondri_dna**, clone, serin, shlase, str, **atp**, sulfhydrylas, ori, homocystein, *exon*, *schizosaccharomyc_pomb*, acetylserin, rho, acetylhomoserin, cyp83b, **mitochondri**, nidulan, cyp83a, cytoplasm, transposit, acetyltransferas, fungi, biosynthesi, pomb, met, aspergillu, ibs, beta_synthas, homolog, male, glucosinol, glu, sulphur, satas, synthas, sulphat	Introns, **Mitochondrial DNA**, cob, *Genome*, **Cytochromes b**, RNA, *Exons*, Cytochromes, **Electron Transport Complex IV**, RNA Splicing, petite, oxi3, Transfer RNA, Recombinant DNA, maturase, Oxidoreductases, Reading Frames, *Schizosaccharomyces pombe Proteins*, Cluster Analysis, DNA Restriction Enzymes, Genetic Recombination, Ribosomal RNA, Gene Order, rRNA Genes, Nucleic Acid Conformation, Open Reading Frames, cox1, oxi1, Cytochrome b Group, Apoproteins, Base Pairing, Nucleotides, Species Specificity, Protein Splicing, Nucleic Acid Repetitive Sequences, RNA Splice Sites, oxi2, Structural Genes, Antibodies, Endoribonucleases, Peptides, Fungal RNA, hybridization, 10b, Fungal Genome, d273, oli2, Nucleotidyltransferases, Cytochromes a, Nucleic Acid Hybridization
BOW /1/0/0/0/1	VOWC /0/1/1/1/0 (mitochondrial protein synthesis)
mrna, pet, cox, *translat*, transcript, gene, mutat, cob, **mitochondri**, cbp, fumaras, *cox2p*, synthesi, rna, oxi, nuclear_gene, nuclear, mss, ai, cox1p, aep, box, **respiratori**, *translat_activ*, *codon*, ts, nam, cbs, fum, **cytochrom**, suppressor, bi, arg8m, utl, excis, **mitochondri_gene**, **mitochondri_mrna**, mss51p, protein, phenotyp, synthet, **mitochondri_translat**, *translat_product*, *cox3p*, *pet111p*, oxa, **mitochondri_transcript**, coxiii, pre_mrna, suv	*mRNA*, *Protein Biosynthesis*, cox3, Gene Expression, pet122, pet54, *cox2*, cbp1, Membrane Tissue, pet494, 5' Untranslated Regions, Membrane Proteins, *cox2p*, *Codon*, *pet111*, Untranslated Regions, **Mitochondrial Proteins**, Initiator Codon, arg8m, 3' Untranslated Regions, **Mitochondria**, **Electron Transport Complex IV**, Ribosomal Proteins, Nuclear Proteins, Alleles, Fungal Gene Expression Regulation, RNA Stability, mitochondrially, Polyribosomes, Reporter Genes, Elements, Temperature, **Cytochromes**, chimeric, nuclearly, Suppressor Genes, **Ferricytochrome c'**, **Cytochromes c**, Oxidoreductases, Gene Deletion, Ribosomes, Isoenzymes, Trans-Activators, Prostaglandin-Endoperoxide Synthases, aug, Biogenesis, Prokaryotic Initiation Factor-2, cox1p, Cold, Protein Precursors
BOW /1/0/0/1/0	VOWC /0/1/1/0/0 (mitochondrial ATP biogenesis)
atp, oscp, **atpas**, oligomycin, beta_subunit, **atp_synthas**, cox5b, cox5a, *oxygen*, sector, residu, *phosphoryl*, oxid, coq, heme, **adp**, hap, cyc, amino_acid, cyt, membran, oli, mtatpas, imp, aerob, atpas_subunit, f1f, *proton*, **mitochondri_atpas**, enzym, **atp_synthas_complex**, amino_acid_substitut, **atpas_activ**, vb, **yeast_atp_synthas**, som, hydrophob, acid, uas, chromatographi, aminolevulin, lethal, bovin, viia, alpha_subunit, mgi, hem, qh, *oxidas_subunit*, **mitochondri_atp_synthas**	**Adenosinetriphosphatase**, **Adenosine Triphosphate**, *Proton-Translocating ATPases*, oscp, **Mitochondrial Proton-Translocating ATPases**, Oligomycins, **Mitochondria**, Membranes, *atp2*, Protein Subunits, Amino Acids, *Oxidative Phosphorylation*, *Phosphorylation*, Membrane Proteins, **Mitochondrial**, diseases, terminus, *Protons*, *atp1*, Adenosine diphosphate, GAP-43 Protein, Antibodies, lethality, Amino Acid Substitution, nonfermentable, Glycerol, *atp6*, Proteolipids, **Bacterial Proton-Translocating ATPases**, *atp4*, petite, Protein Precursors, Structure-Activity Relationship, Organelles, Hydrolysis, Protein Conformation, Glycogen Synthase, fermentable, Cell Division, Oligonucleotide Probes, Family Characteristics, Heart, Enzymes, Membrane Potentials, Cross-Linking Reagents, Infertility, aap1, Mutagenesis, hexahistidine, **Mitochondrial Proteins**, Biogenesis

Even though the VOWC representation did not improve clustering quality in terms of consistency or agreement, it did provide better keywords than BOW representation. For example, the keyword *bind_protein* (stemmed from *binding protein*) provided by BOW for the ubiquitin-mediated proteolytic pathway was too general or vague, while *TATA-Box binding protein* or *DNA-binding protein* given by VOWC was more meaningful.

Chapter 7 Discussion & Future work

7.1 GeneNarrator and software engineering

The development of GeneNarrator followed the principles of software engineering at several levels. First, the overall research design followed the general lifecycle of incremental software development, which consists of major phrases (requirements/specification analysis, design and implementation) and incremental cycles among phases. The first three chapters roughly map to the major phrases, and the chapters about improving text clustering resemble the cycles between design and implementation.

The principles were also followed at strategic design level: modularization and decoupling. Modularization requires that complex system be divided into subsystems (modules). Decoupling demands interaction (dependency) between modules to be minimized, so individual modules can be replaced, modified and improved independently without breaking other modules. The two-step clustering approach modularizes and decouples text clustering and gene clustering from the complex task of genomic functional analysis. In addition to the biological advantages of the design (discussed in Chapter 2), the decoupling makes it possible to improve the two steps independently. Thus, GeneNarrator becomes an experimental platform for testing various text clustering algorithms and strategies for application in bioinformatics.

Finally, software design patterns, proven best solutions for particular types of problems, were applied throughout the implementation. For example, the overall architecture of GeneNarrator I and II used the pipeline pattern, and the BOWViewer used model-view-control (MVC) pattern.

This project is a demonstration that good software engineering practices can be applied to bioinformatics and lead to flexible, extensible and maintainable software applications.

7.2 Agreement measure

It came to us as a surprise that the text clustering research community could not agree on a measure for agreement. The newly proposed measure, normalized mutual information (NMI), has the following desirable properties, making it suitable for wide acceptance.

- NMI has well defined semantics from information theory. It measures the percentage of overlapping between two random variables' information. Under the context of clustering, it measures how certain the membership in one partition is if the membership in another partition is known. This is *the* semantics of an agreement measure.
- It is bounded by [0, 1].
- Its baseline is stable, and not sensitive to the size of document set, the number of classes, the number of clusters, *etc.* These two properties make it possible to compare different algorithms in different studies.
- Random guesses get zero credit.
- Perfect scores are still possible even if an algorithm (e.g. *k*-means) is given a wrong guess about the number of clusters in the dataset. The last two properties make NMI a "fair" measure.

7.3 Use of background knowledge (ontologies) in text clustering

This study has tested ontology-based text clustering. We hypothesized that the clustering might be improved due to the following factors.

- Focus can be on relevant information captured in the ontology, without the distraction of irrelevant terms.
- Clustering can be guided by the background knowledge encoded in the ontology. For example, mapping synonymous terms to the same concept can avoid counting them as different things (as the case in BOW-based clustering). Concept upgrading combines less-frequent, closely related concepts into more general concepts, so that they may have more impact on the clustering results.

The results however showed that the VOC representation alone actually decreased the quality of clustering. The hybrid VOWC representation restored the quality to the same level as the BOW representation. On the other hand, improvement was found in presentation of the

biological meaning of the text clustering (see Chapter 6). Concept mapping in hybrid concept-based text representation could be considered as equivalent to the multiple-word term detection in BOW-based representation, except that the detection is knowledge-guided instead of relying on statistics. This is how human experts recognize multiple-word terms/concepts. However, the improvement was difficult to measure in terms of a bounded and baseline-stable number, like NMI.

Ontology-based clustering might still provide more dramatic improvements in clustering in the future. The three prerequisites for a successful implementation are

1. an ontology capturing the domain knowledge,
2. a mapping algorithm identifying ontology concepts in free text, and
3. a clustering algorithm that can utilize the ontology.

These are not available or are in their infant stages at the present time. With the advance of research in these areas, ontology-based text clustering may still eventually show additional improvements.

7.4 Dimension reduction

Several text clustering systems, reviewed in Chapter 2, were forced to reduce the dimensionality of the vector space (e.g. keywords filtering) dramatically. For example, "literature profiling" [8] kept only 101 keywords out of 25,000 unique words. Hotho *et al.* [26] reduced the dimensionality to below 25. As these were not evaluated in terms of consistency or agreement using NMI, their results were not directly comparable to GeneNarrator's results. However, it is hard to put faith in the reliability of such results with so much information thrown away.

The problem of high dimensionality results from the dependence on finding the "nearest neighbor", which in turn depends on distance calculation. In high dimensional space, data points seem to often be at about the same distance, so the "nearest neighbor" is less meaningful [7,23]. Without the need for calculating distance, the CAM algorithm handles document vectors of high dimensionality (~20,000) with ease. In fact, aggressive reduction in dimensionality made CAM perform poorly. In preliminary experiments (data not shown) when *df* filtering reduced the dimensionality to ~1,000, CAM failed to split the root

topic into two child topics. In the reported experiments, the dimensionality was reduced to ~2,000; this was translated to a marginal improvement in clustering consistency.

7.5 Multi-clustering

The most dramatic improvement came from the multi-clustering idea, which was inspired by observing how human experts deal with inconsistency caused by outliers. The final result reached 84% of the best achievable agreement with the gold standard. Observe that even human experts cannot agree with each other 100%, and one expert may change his or her mind over time. Hence, the actual improvement might be even better than suggested by the 84% achievement. In addition, as illustrated in Chapter 6 and discussed further in the next section, viewing the result under the hierarchical context may look even better.

7.6 Clustering and comparing hierarchical structures

The input to both of the two clustering steps contained hierarchical information. The VOC or VOWC representation of MEDLINE records (input to the 1st clustering) was backed by the hierarchical MeSH ontology, and the genes' topic distributions (the input to the 2nd clustering step) were built on the topic hierarchy resulted from the 1st-step clustering. Yet neither clustering algorithm (CAM or *k*-means) benefited from the hierarchies. The input was treated as flat vectors; and the hierarchies were simply ignored. This inevitably degraded the quality of the clusterings, as discussed in Section 6.4.2. The attempt to utilize the hierarchy, the concept upgrading method, tried a heuristic approach. Significant improvement was not obtained, however, as discussed in the previous section. This was likely due to the quality of the ontology and/or the performance of the mapping algorithm.

The newly proposed metric, normalized mutual information (NMI), compares agreement between two flat clustering solutions, as do various other indices and metrics (reviewed in sections 4.2.3.2 and 4.2.3.3). This "flat" behavior of NMI made hierarchical clustering results look worse than they were from a biological perspective, because MEDLINE records misclassified into clusters near their ideal clusters were treated the same as records misclassified into distant, unrelated clusters. Hence, the development of hierarchy-aware clustering algorithms and comparing metrics should be able to improve GeneNarrator further.

7.7 Conclusion

This dissertation project contributed to the field of bioinformatics in the following ways.

1. A new metric (normalized mutual information) was described and used for evaluating a clustering algorithm in terms of its self-consistency and agreement with a gold standard.
2. Several strategies for text clustering of biological texts were implemented and tested, such as different text representations (BOW- vs. concept-based vector space models), dimension reduction (*df* filtering) and multi-clustering. The most improvement came from multi-clustering, which identified and discarded outliers from a document set.
3. A two-step clustering approach was designed for clustering functionally related genes based on information from texts. This might be applied, for example, to functional analysis of microarray experiments.
4. System variants GeneNarrator I and GeneNarrator II were implemented and compared. The GeneNarrator design facilitates comparisons that can test hypotheses about using concepts and ontologies in clustering of biological texts. It also illustrates a two-step clustering approach for clustering topics into related areas from raw clusters of input data such as MEDLINE records.

Appendix: Gold standard gene list

Format: gene_symbol|synonym1|synonym2|...

YBR294w|SUL1|SEL3|SFP
YLR092w|SUL2|
YJR010w|MET3|
YKL001c|MET14|
YPR167c|MET16|
YJR137c|ECM17|MET5
YFR030w|MET10|
YNL277w|MET2|
YLR303w|MET17|MET25|MET15
YGR155w|CYS4|STR4|NHS5|VMA41
YER091c|MET6|MET6
YAL012w|CYS3|CYI1|STR1|FUN35
YLR180w|SAM1|ETH10
YDR502c|SAM2|ETH2
YFL025c|BST1|
YDR072c|IPT1|SYR4
YMR272c|SCS7|FAH1
YLR372w|SUR4|ELO3|SRE1|VBM1
YKL004w|AUR1|ABR1
YPL057c|SUR1|BCL21|CSG1
YBR036c|CSG2|CLS2
YGL225w|GOG5|VRG4|VAN2|MCD3
YEL042w|GDA1|SYGP-ORF16
YCR034w|FEN1|ELO2|GNS1|VBM2
YDR297w|SUR2|SYR2
YDR062w|LCB2|YD9609.16
YKR053c|YSR3|LBP2
YJL134w|LCB3|YSR2|LBP1
YMR296c|LCB1|
YML120c|NDI1|
YKL148c|SDH1|SDHA
YKL141w|SDH3|CYB3
YLL041c|SDH2|SDHB|SDH
YDR178w|SDH4|
Q0105|COB|CYTB
YBL045c|COR1|QCR1
YOR065w|CYT1|CTC1
YJL166w|QCR8|COR5
YDR529c|QCR7|UCR7|COR4|CRO1
YFR033c|QCR6|CR17|UCR6|COR3
YPR191w|QCR2|UCR2|COR2|COXCH2
YEL024w|RIP1|
YHR001w-a|QCR10|
YGR183c|QCR9|UCR9
YGL191w|COX13|
YLR038c|COX12|
YDL067c|COX9|
YLR395c|COX8|
YMR256c|COX7|
YHR051w|COX6|
YNL052w|COX5A|
YGL187c|COX4|
YDR322c-a|TIM11|ATP21|ATPJ
YDR377w|ATP17|
YDL004w|ATP16|ATPDELTA
YPL271w|ATP15|
YLR295c|ATP14|
YKL016c|ATP7|
YDR298c|ATP5|OSCP
YPL078c|ATP4|LPF7
YBR039w|ATP3|
YJR121w|ATP2|

YBL099w|ATP1|
Q0085|ATP6|OLI2|OLI4|PHO1
Q0080|AAP1|ATP8
Q0130|OLI1|ATP9
Q0045|COX1|OXI3
Q0250|COX2|OXI1
Q0275|COX3|OXI2
YEL021w|URA3|MLF2
YMR271c|URA10|
YBL042c|FUI1|
YBR021w|FUR4|MLF1
YML106w|URA5|PYR5
YKL216w|URA1|
YLR420w|URA4|
YJL130c|URA2|
YLR304c|ACO1|GLU1
YOR136w|IDH2|
YDR148c|KGD2|
YFL018c|LPD1|DHLP1|HPD1
YOR142w|LSC1|PSC4
YGR244c|LSC2|
YPL262w|FUM1|
YNL037c|IDH1|
YNR001c|CIT1|LYS6|GLU3
YKL085w|MDH1|
YIL125w|KGD1|OGD1
YJL194w|CDC6|
YMR001c|CDC5|PKX2|MSD2
YAL040c|CLN3|WHI1|DAF1|FUN10|CST7
YBR160w|CDC28|SRM5|CDK1|HSL5
YFR028c|CDC14|OAF3
YLR103c|CDC45|
YMR199w|CLN1|PSC1
YPL256c|CLN2|PSC2
YLR079w|SIC1|SDB25
YFL009w|CDC4|
YDR328c|SKP1|CBF3D
YDR054c|CDC34|UBC3|DNA6
YDL132w|CDC53|
YGR108w|CLB1|SCB1
YPR119w|CLB2|
YDL155w|CLB3|
YLR210w|CLB4|
YPR120c|CLB5|
YGR109c|CLB6|
YKL022c|CDC16|
YHR166c|CDC23|
YBL084c|CDC27|SNB1
YDR052c|DBF4|DNA52
YDL017w|CDC7|SAS1
YMR239c|RNT1|
YLR175w|CBF5|
YHR089c|GAR1|
YDL014w|NOP1|
YLL011w|SOF1|
YNL282w|POP3|
YHR065c|RRP3|
YGL171w|ROK1|
YDR021w|FAL1|
YMR229c|RRP5|FMI1
YCL031c|RRP7|
YPL266w|DIM1|
YOR048c|RAT1|HKE1|XRN2
YNL221c|POP1|
YBR257w|POP4|
YDR478w|SNM1|
YGL078c|DBP3|
YGL173c|KEM1|SEP1|XRN1|DST2|RAR5|SKI1|ST
PBETA
YHR069c|RRP4|
YGR195w|SKI6|RRP41|ECM20
YDL111c|RRP42|

YCR035c|RRP43|
YOL021c|DIS3|RRP44
YOR001w|RRP6|
YDR394w|RPT3|YTA2|YNT1
YOR117w|RPT5|YTA1
YKL145w|RPT1|YTA3|CIM5
YGR270w|YTA7|
YDL007w|RPT2|YTA5|YHS4
YOR259c|RPT4|CRL13|SUG2|PCS1
YGL048c|RPT6|SUG1|TBY1|TBPY|SCB68|CIM3
YLR378c|SEC61|
YPL094c|SEC62|
YOR254c|SEC63|NPL1|PTL1
YLR292c|SEC72|SEC67|SIM2
YJL034w|KAR2|GRP78|BIP
YFL005w|SEC4|SRO6
YOR089c|VPS21|YPT51|YPT21|VPT21
YLR262c|YPT6|
YER031c|YPT31|YPT8
YGL210w|YPT32|YPT11
YFL038c|YPT1|YP2
YML001w|YPT7|AST4|VAM4

References

1. Open Biological Ontologies. http://obo.sourceforge.net/

2. Adryan, B. and R. Schuh (2004) Gene-Ontology-based clustering of gene expression data. *Bioinformatics* 20(16): 2851-2852.

3. Ashburner, M. , C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Epping, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, and G. Sherlock (2000) Gene Ontology: tool for the unificaiton of biology. *Nature Genetics* 25: 25-29.

4. Badea, L. (2003) Functional Discrimination of Gene Expression Patterns in Terms of the Gene Ontology. *Pacific Symposium on Biocomputing* 8: 565-576.

5. Becker, K., D. Hosack, G. Dennis, R. Lempicki, T. Bright, C. Cheadle, and J. Engel (2003) PubMatrix: a tool for multiplex literature mining. *BMC Bioinformatics* 4(1): 61.

6. Beil, F., M. Ester, and X. Xu (2002) Frequent Term-Based Text Clustering. *SIGKDD 02.*

7. Beyer, K., J. Goldstein, R. Ramakrishnan, and U. Shaft (1999) When Is "Nearest Neighbor" Meaningful? *Proc. of ICDT-99 217-235.*

8. Chaussabel, D. and A. Sher (2002) Mining microarray expression data by literature profiling. *Genome Biology* 3: research0055.1-research0055.16.

9. Cutting, D., D. Karger, J. Pedersen, and J. Tukey (1992) Scatter-gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of SIGIR'92.*

10. Damgaard, C. and J. Weiner (2000) Describing Inequality in Plant Size or Frcundity. *Ecology* 81: 1139-1142.

11. Dash, M., K. Choi, P. Scheuermann, and H. Liu (2002) Feature Selection for Clustering

- A Filter Solution. *IEEE International Conference on Data Mining (ICDM'02) 115.*

12. Debole, F. and F. Sebastiani (2003) Supervised Term Weighting for Automated Text Categorization. *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing* 784-788.

13. Denoeud, L., H. Garreta, and A. Guenoche (2004) Comparison of Distance Indices Between Partitions. *Proceedings of Applied Stochastic Models in Data Analysis on CD ROM.*

14. Department of Genetics, School of Medicine, Stanford University (2004) The Saccharomyces Genome Database. http://www.yeastgenome.org/

15. Ding, J. and D. Berleant (2005) MedKit: a helper toolkit for automatic mining of MEDLINE/PubMed citations. *Bioinformatics* 21(5): 694-695.

16. Fang, Y.C., S. Parthasarathy, and F. Schwartz (2001) Using Clustering to Boost Text Classification. *Workshop on Text Mining (TextDM'2001).*

17. Felsenstein, J. PHYLIP Home Page. http://evolution.genetics.washington.edu/phylip.html

18. Fowlkes, E.B. and C.L. Mallows (1983) A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association* 78: 553-569.

19. Frank, E., M. Hall, L. Trigg, G. Holmes, and I.H. Witten (2004) Data mining in bioinformatics using Weka. *Bioinformatics* 20(15): 2479-2481.

20. Gene Ontology Consortium Gene Ontology. http://www.geneontology.org/

21. Glenisson, P., P. Antal, J. Mathys, Y. Moreau, and B. De Moor (2003) Evaluation of the Vector Space Representation in Text-Based Gene Clustering. *Pacific Symposium on Biocomputing* 8: 391-402.

22. Goldszmidt, M. and M. Sahami (1998) A Probabilistic Approach to Full-Text Document

Clustering. ITAD-433-MS-98-044, SRI International.

23. Hinneburg, A., C.C. Aggarwal, and D.A. Keim (2000) What is the Nearest Neighbor in High Dimensional Spaces? *Proc. of VLDB-00* 506-515.

24. Hofmann, T. (1999) The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data. *Proceedings of the International Joint Conference on Artificial Intelligence.*

25. Homayouni, R., K. Heinrich, L. Wei, and M.W. Berry (2005) Gene clustering by Latent Semantic Indexing of MEDLINE abstracts. *Bioinformatics* 21(1): 104-115.

26. Hotho, A., A. Madche, and S. Staab (2001) Ontology-Based Text Clustering. *Workshop "Text Learning: Beyond Supervision", IJCAI 2001.*

27. Hung, C. and S. Wermter (2003) A Dynamic Adaptive Self-Organising Hybrid Model for Text Clustering. *Proceedings of The Third IEEE International Conference on Data Mining* 75-82.

28. Hung, C. and S. Wermter (2003) A Self-Organising Hybrid Model for Dynamic Text Clustering. *Proceedings of the The Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.*

29. Hung, C. and S. Wermter (2004) A Time-Based Self-Organising Model for Document Clustering. *Proceedings of the International Joint Conference on Neural Networks 17-23.*

30. Hung, C., S. Wermter, and P. Smith (2004) Hybrid Neural Document Clustering Using Guided Self-organisation and WordNet. *Issue of IEEE Intelligent Systems*: 68-77.

31. Jain, A.K., Murty M.N., and Flynn P.J. (1999) Data Clustering: A Review. *ACM Computing Surveys* 31: 264-323.

32. Joslyn, C.A., S.M. Mniszewski, A. Fulmer, and G. Heaton (2004) The Gene Ontology

Categorizer. *Bioinformatics* 20(suppl_1): i169-177.

33. Kankar, P., S. Adak, A. Sarkar, K. Murari, and G. Sharma (2002) MedMeSH Summarizer: Text Mining for Gene Clusters. *Proceedings of the Second SIAM International Conference on Data Mining.*

34. Kaufman, L. and Rousseeuw, P. (1990) Finding groups in data, Wiley-Interscience.

35. Kennedy, P.J., S.J. Simoff, D. Skillicorn, and D. Catchpoole (2004) Extracting and Explaining Biological Knowledge in Microarray Data. *The 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

36. Kim, C.C. and S. Falkow (2003) Significance analysis of lexical bias in microarray data. *BMC Bioinformatics* 4: 12.

37. Kiritchenko, S., S. Matwin, and A.F. Famili (2004) Hierarchical Text Categorization as a Tool of Associating Genes with Gene Ontology Codes. *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics.*

38. Kohonen, T., S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela (2000) Self Organization of a Massive Document Collection. *IEEE Transactionss on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery* 11: 574-585.

39. Lagus, K. (2000) Text Mining with the WEBSOM. Helsinki University of Technology, Finland.

40. Lagus, K. (2000) Text Retrieval Using Self-Organized Document Maps. Technical Report A61, Helsinki University of Technology, Laboratory of Computer and Information Science. ISBN 951-22-5145-0.

41. Larsen, B. and C. Aone (1999) Fast and Effective Text Mining Using Linear-Time Document Clustering. *Proc. of KDD '99.*

42. Lewis, S. (2004) Gene Ontology: looking backwards and forwards. *Genome Biology* 6(1): 103.

43. Manning, C.D. and H. Schutze (1999) *in* Foundations of Statistical Natural Language Processing (Manning, C.D. and Schutze, H., Eds.), pp. 1515-189 The MIT Press, Cambridge, Massachusetts, USA.

44. Mao, X., T. Cai, J.G. Olyarchuk, and L. Wei (2005) Automated genome annotation and pathway identification using the KEGG Orthology (KO) as a controlled vocabulary. *Bioinformatics* 21(19): 3787-3793.

45. Masys, D.R., J.B. Welsh, J. Lynn Fink, M. Gribskov, I. Klacansky, and J. Corbeil (2001) Use of keyword hierarchies to interpret gene expression patterns. *Bioinformatics* 17(4): 319-326.

46. McCallum, A.K. (1996) Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. http://www-2.cs.cmu.edu/~mccallum/bow/

47. Meila, M. (2002) Comparing Clusterings. University of Washington Statistics Technical Report 418.

48. Munich information center for protein sequences Comprehensive Yeast Genome Database. http://mips.gsf.de/proj/yeast/pathways/

49. National Library of Medicine Medical Subject Headings. http://www.nlm.nih.gov/mesh/meshhome.html

50. National Library of Medicine Unified Medical Language System. http://www.nlm.nih.gov/research/umls/

51. National Library of Medicine (2004) Entrez Programming Utilities. http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html

52. National Library of Medicine (2004) MEDLINE.

http://www.nlm.nih.gov/pubs/factsheets/medline.html

53. Nigam, K., A.K. McCallum, S. Thrun, and T.M. Mitchell (2000) Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39: 103-134.

54. Oliveros, J.C., C. Blaschke, J. Herrero, J. Dopazo, and A. Valencia (2000) Expression Profiles and Biological Function. *Genome Informatics* 11: 106-117.

55. Pasquier, C., F. Girardot, K. Jevardat de Fombelle, and R. Christen (2004) THEA: ontology-driven analysis of microarray data. *Bioinformatics* 20(16): 2636-2643.

56. Porter, M.F. (1980) An algorithm for suffix stripping. *Program* 14: 130-137.

57. Quackenbush, J. (2002) Microarray data normalization and transformation. *Nature Genetics* 32: 496-501.

58. Rand, W.M. (1971) Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association* 66: 846-850.

59. Raychaudhuri, S. and R.B. Altman (2003) A Literature-based Method for Assessing the Functional Coherence of a Gene Group. *Bioinformatics* 19: 396-401.

60. Raychaudhuri, S., J. Chang, P. Sutphin, and R.B. Altman (2002) Associating Genes with Gene Ontology Codes Using a Maximum Entropy Analysis of Biomedical Literature. *Geneome Research* 12: 203-214.

61. Raychaudhuri, S., H. Schutze, and R.B. Altman (2002) Using Text Analysis to Identify Functionally Coherent Gene Groups. *Genome Res.* 12(10): 1582-1590.

62. Renner, A. and A. Aszodi (2000) High-throughput functional annotation of novel gene products using document clustering. *Pac Symp Biocomput* : 54-68.

63. Robinson, P.N., A. Wollstein, U. Bohme, and B. Beattie (2004) Ontologizing gene-expression microarray data: characterizing clusters with Gene Ontology. *Bioinformatics* 20(6): 979-981.

64. Rose, K., E. Gurewitz, and G. Fox (1990) Statistical mechanics and phase transitions in clustering. *Physical Review Letters* 65: 945-948.

65. Ruiz, M.E. and P. Srinivasan (2002) Hierarchical Text Categorization Using Neural Networks. 5(1): 87-118.

66. Ruiz, M.E. and P. Srinivasan (2003) Hybrid Hierarchical Classifiers for Categorization of Medical Documents. *Proceedings of the 2003 Conference of ASIST.*

67. Saporta, G. and G. Youness (2002) Comparing Two Partitions: Some Proposals and Experiments. *Proceedings in Computational Statistics 2002 15th Symposium 243-248.*

68. Schutze, H. and C. Silverstein (1997) Projections for Efficient Document Clustering. *Proc. of SIGIR-97* 74-81.

69. Shatkay, H., S. Edwards, W. Wilbur, and M. Boguski (2000) Genes, Themes, and Microarrays: Using Information Retrieval for Large-Scale Gene Analysis. *8th International Conference on Intelligent Systems for Molecular Biology (ISMB) 19-23.*

70. Slaton, G. and McGill, M.J. (1983) Introduction to Modern Information Retrieval, McGraw-Hill, New York, NY.

71. Smid, M. and L.C.J. Dorssers (2004) GO-Mapper: functional analysis of gene expression data using the expression level as a score to evaluate Gene Ontology terms. *Bioinformatics* 20(16): 2618-2625.

72. Steinbach, M., G. Karypis, and V. Kumar (2000) A Comparison of Document Clustering Techniques. *Proc. TextMining Workshop, KDD 2000.*

73. Steinbach, M., G. Karypis, and V. Kumar (2000) A Comparison of Document Clustering Techniques. University of Minnesota, Technical Report #00-034.

74. Struble, C.A. and C. Dharmanolla (2004) Clustering MeSH Representations of Biomedical Literature. *HLT-NAACL 2004 Workshop: Biolink 2004, Linking Biological Lit-*

erature, Ontologies and Databases pp. 41-48.

75. Sun, A. and E.-P. Lim (2001) Hierarchical Text Classification and Evaluation. *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001) 521-528.*

76. Swiss Institute of Bioinformatics ExPASy - Swiss-Prot and TrEMBL. http://us.expasy.org/sprot/

77. Yang, Y. (1999) An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval* 1: 69-90.

78. Yang, Y. and X. Liu (1999) A Re-examination of Text Categorization Methods. *22nd Annual International SIGIR* 42-49.

VITA

NAME OF AUTHOR: Jing Ding

DATE AND PLACE OF BIRTH: January 5, 1967, Shanghai, China

DEGREE AWARDED:

B.S. in Biophysics, Fudan University, Shanghai, China, 1989

M.S. in Toxicology, Iowa State University, Ames, Iowa, 2000

M.S. in Computer Engineering, Iowa State University, Ames, Iowa, 2003

PROFESSIONAL EXPERIENCE:

Assistant Researcher, Shanghai Institute of Physiology, Chinese Academy of Sciences, Shanghai, China, 1989 – 1997

Teaching Assistant, Research Assistant, Iowa State University, Ames, Iowa, 1997 – 2005

Research Intern, Procter and Gamble Company, Cincinnati, Ohio, 2002, 2003

Senior Systems Consultant, The Ohio State University Medical Center, Columbus, Ohio, 2006 – present

PROFESSIONAL PUBLICATIONS

- JOURNAL ARTICLES

1. Ding J, Hughes LM, Berleant D, Fulmer AW, and Wurtele ES (2006) PubMed Assistant: a biologist-friendly interface for enhanced PubMed search. *Bioinformatics* 22(3): 378-80. Epub 2005 Dec 6.
2. Ding J, Viswanathan K, Berleant D, Wurtele E, Ashlock D, Dickerson J, Fulmer A, and Schnable PS (2005) Using the Biological Taxonomy to Access Biological Literature with PathBinderH. *Bioinformatics* 21(10): 2560-2562.
3. Ding J, Berleant D (2005) MedKit: A Helper Toolkit for Automatic Mining of MEDLINE/PubMed Citations. *Bioinformatics* 21(5): 694-695.

4. Ding J, Drewes CD, Hsu WH (2000) Behavioral effects of ivermectin in a freshwater oligochaete, Lumbriculus variegatus. *Environmental Toxicology and Chemistry* 20(7):1584-90.
5. Ding J, Xu XZ, Li CY (1998) Dye coupling between visual cortical (area 17) neurons of adult rats - a study on brain slices. *Acta Physologica Sinica* 50(3): 241-248.
6. Ding J, Xu XZ, Li CY (1998) Short-term temporal integration of the in vitro rat visual cortex (area 17). *Acta Biophysica Sinica* 14(3): 478-484.
7. Cheng H, Grodnitzky JA, Yibchok-anun S, Ding J, Hsu WH (2005) Somatostatin increases phospholipase D activity and PIP2 synthesis in clonal beta-cells HIT-T15. *Molecular Pharmacology* 67(6): 2162-72. (e-print: March 22, 2005; DOI: 10.1124/mol.104.010470)
8. Yibchok-anun S, Cheng H, Abu-Basha EA, Ding J, Ioudina M, Hsu WH (2002) Mechanisms of bradykinin-induced glucagon release in clonal alpha-cells In-R1-G9: involvement of Ca(2+)-dependent and -independent pathways. *Molecular and Cellular Endocrinology* 192(1-2): 27-36.

- BOOK CHAPTERS

1. Dickerson JA, Berleant D, Du P, Ding J, Foster CM, Li L, and Wurtele ES, "Creating, modeling, and visualizing metabolic networks," chapter 17 in H. Chen, S. S. Fuller, C. Fiedman, and W. Hersh, eds., *Medical Informatics: Knowledge Management and Data Mining in Biomedicine*, Springer, 2005, pp. 491-518.

- PROCEEDINGS - FULL PAPER REVIEWED

1. Ding J, and Berleant D (2005) Design of a Standoff Object-Oriented Markup Language (SOOML) for Annotating Biomedical Literature. *Proceedings of the Seventh International Conference on Enterprise Information Systems* (ICEIS 2005), May 25-28, Miami/FL, pp. 382 – 385.
2. Ding J, Berleant D, Xu J, and Fulmer AW (2003) Extract biochemical interactions from MEDLINE abstracts using a link-grammar parser. *Proceedings of the Fifteenth IEEE Conference on Tools with Artificial Intelligence* (ICTAI 2003), Nov. 3-5, Sacramento, pp. 467-471.

3. Ding J, Berleant D, Nettleton D, and Wurtele ES (2002) Mining MEDLINE: abstracts, sentences, or phrases? *Pacific Symposium on Biocomputing* 2002: 326 – 337.

- OTHER PUBLICATIONS

1. Ding J, Viswanathan K, Berleant D, Wurtele E, Ashlock D, Dickerson J, Fulmer A, and Schnable PS (2004) PathBinderH: a tool for sentence-focused, plant taxonomy-sensitive access to the biological literature. *Software Artifact Research and Development Laboratory Technical Report*: SARD11-19-04.
2. Ding J, and Berleant D (2004) Design a Standoff Object-Oriented Markup Language (SOOML) for Text Mining. *Iowa State University ECpE Technical Report*: TR-2004-04-2.
3. Ding J (2003) PathBinder: a repository of sentences containing chemical interactions extracted from MEDLINE. (MS thesis)
4. Ding J (2000) Lethal and sublethal effects of ivermectin in a freshwater oligochaete, Lumbriculus variegates. (MS thesis)

- PRESENTATIONS AND POSTERS

1. [2005] Ding J, and Berleant D, "Design of a Standoff Object-Oriented Markup Language (SOOML) for Annotating Biomedical Literature," The 7th International Conference on Enterprise Information Systems, Miami, FL, May 24 – 28.
2. [2005] Ding J, and Berleant D, "Design a Standoff Object-Oriented Markup Language (SOOML) for Text Mining," External Advisory Board Poster Session, Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, April 14.
3. [2005] Hughes L, Ding J, Viswanathan K, Fulmer AW, Berleant D, and Schnable PS "PathBinderH: a Tool for Linnaean Taxonomy-Aware Literature Searches," Plant and Animal Genome XIII Conference, San Diego, January 15-19.
4. [2003] Berleant D and Ding J, "Extracting Protein Interactions from Sentences: the PathBinder Project," 3rd Annual UI/ISU Bioinformatics Workshop, Iowa City, IA, April 25.

5. [2003] Ding J, Xu J, and Fulmer AW, "Gene Ontology Toolkit: A Comprehensive Software Package for Working with Gene Ontology," The 11th International conference on Intelligent Systems for Molecular Biology (ISMB), Brisbane, Australia, June 29 – July 3.
6. [2002] Ding J, Berleant D, Nettleton D, and Wurtele ES, "Mining MEDLINE: abstracts, sentences, or phrases?" The Pacific Symposium on Biocomputing, Kauai, Hawaii, January 3 – 7.
7. [2001] Ding J, Berleant D, Nettleton D, and Wurtele ES, "Mining MEDLINE: Abstracts, Sentences, or Phrases?" Plant Sciences Institute Scientific Symposium, Ames, Iowa, October. 19.
8. [2000] Ding J, Duwairi B, Miao J, and Berleant D, "Three Systems for Improved Access to Literature on the Web," Joint Bioinformatics Workshop, Iowa City, IA, November 3 – 4.
9. [1999] Ding J, Drewes CD, and Hsu WH, "Lethal and sublethal effects of ivermectin in a freshwater oligochaete, *Lumbriculus variegates*." The 39th Annual Meeting of the Society of Toxicology, Philadelphia, PA, April 4 – 7.

- SOFTWARE

1. Ding J (project architect/lead developer), GeneNarrator: A Text Clustering-Based Microarray Functional Analysis Tool. http://metnetdb.gdcb.iastate.edu/genenarrator
2. Ding J (project architect/lead developer), PubMed Assistant: A Biologist-Friendly Interface for Enhanced PubMed Search. http://metnetdb.gdcb.iastate.edu/browser
3. Ding J (project architect/lead developer), MedKit: A Helper Toolkit for Automatic Mining of MEDLINE/PubMed Citations. http://metnetdb.gdcb.iastate.edu/medkit
4. Ding J (sub-project architect/lead developer), PathBinder: A Text Mining-Based Biochemical Interaction Database. http://metnetdb.gdcb.iastate.edu/pathbinder
5. Ding J (principal developer), PathBinderH: Taxonomy-Aware Access of Biomedical Literature. http://pathbinderh.plantgenomics.iastate.edu/PathBinderH