Methods for accurate retrieval of MEDLINE citations in functional genomics

Mehmet Kayaalp,[a] Alan R. Aronson,[a] Susanne M. Humphrey,[a] Nicholas C. Ide,[a] Lorraine K. Tanabe,[a] Lawrence H. Smith,[a] Dina Demner,[a,b] Russell R. Loane,[a] James G. Mork,[a] Olivier Bodenreider[a]

[a]National Library of Medicine, Bethesda, Maryland
{Kayaalp, Alan, Humphrey, Ide, Demner, Loane, Mork, Olivier}@nlm.nih.gov;
{Tanabe, LSmith}@ncbi.nlm.nih.gov

[b]University of Maryland, College Park, Maryland

Abstract

The lack of discipline and consistency in gene naming poses a formidable challenge to researchers in locating relevant information sources in the genomics literature. The research presented here primarily focuses on how to find the MEDLINE® citations that describe functions of particular genes. We developed new methods and extended current techniques that may help researchers to retrieve such citations accurately. We further evaluated several machine learning and optimization algorithms to identify the sentences describing gene functions in given citations.

Keywords: Genomics; MEDLINE; MeSH; Information Retrieval; Propositional Logic; Decision Lists; Machine Learning; Bayesian Networks; Model Averaging; Probabilistic Inference.

1 Introduction

Genomics research has created a wealth of information in a relatively short period of time. A downside of this rapid growth has been the inability of the research community to establish a disciplined and consistent labeling system for labeling new information (such as naming new genes and proteins). In the absence of such a systematic information labeling discipline, accessing certain genomic information might be insurmountable for researchers who are not in the circle of that particular genomics research.

To better understand the problem and perhaps to devise some remedies, the National Library of Medicine® (NLM®) and University of Maryland (UMD) teamed up to participate in the genomics track of the 12th Text Retrieval Conference (TREC-12) in 2003.

After describing the primary task of the genomics track in the next section, we introduce three different approaches to solving the problem in separate sections. In the subsequent section, we explain how the outcomes of these methods were combined. In Section 7, the secondary task and several methods towards the possible solutions are discussed. Our conclusions can be found in Section 8.

2 Primary Task

The primary task of the genomics track was defined as ad hoc information retrieval of MEDLINE citations that contain descriptions of a function of a gene given in the query of interest.

The provided corpus consisted of over a half million MEDLINE citations indexed between April 1, 2002 and April 1, 2003. The training query set consisted of 50 queries. Each query corresponded to a gene and was composed of a set of gene identifiers such as *official gene name*, *official symbol*, *alias symbol*, *preferred product*. The test query set contained the same type of information for another 50 genes.

The source of the gene information was the curated genes represented as NLM's LocusLink (LL) database. An LL record contains links in the form of unique identifiers to MEDLINE citations found in NLM's bibliographic resource known as PubMed®. Such a link, along with a brief description of gene function from the MEDLINE article, comprises a GeneRIF (Gene References Into Function). In the example shown in Table 1, the unique PubMed identifier (PMID) 11859139 and the passage next to it are a GeneRIF in the LL record for Interleukin-5 of *Mice*.

The first GeneRIF citation of the example shown in Table 1 is

> Mishra A, Hogan SP, Brandt EB, Rothenberg ME.: *IL-5 promotes eosinophil trafficking to the esophagus*. J Immunol. 2002 Mar 1;168(5):2464-9. PMID: **11859139** [PubMed - indexed for MEDLINE]

In this case, the description in the GeneRIF corresponds to the title of the MEDLINE citation. We informally call such a citation a GeneRIF citation.

Table 1: A Small Portion of LocusLink Record[1]

Mus musculus	
Il5	interleukin 5
LocusID	16191
Locus Type	gene with protein product, function known or inferred
Product	interleukin 5
Alternate Symbols	Il-5
Gene References into Function	
PMID	*GeneRIF*
11859139	IL-5 promotes eosinophil trafficking to the esophagus
11960640	Role of IL-5 during primary and secondary immune response to acetylcholine receptor.

GeneRIF citations were considered the gold standard given that they were the most reliable information resource practically available at the time of the primary task design.

For training purposes, the GeneRIF citations associated with training queries were provided. For test queries, participants were asked not to retrieve and use associated GeneRIF citations. They were expected to develop methods/tools that would label all citations either as positive (i.e., relevant) or negative for every test query, and to submit all positive documents in rank order.

3 An Information Retrieval Approach

In this portion of our study, we used an in-house IR tool called Search Engine (SE) to identify GeneRIF citations. SE was developed at NLM to enable consumers of *ClinicalTrials.gov* to locate information relevant to their needs (McCray, Ide, Loane, & Tse, 2004).

To objectively evaluate SE's performance, Inquery (Callan, Croft, & Harding, 1992) was used to establish a baseline for the evaluation of SE. The best performance we could obtain using Inquery on the training dataset was 0.34 in mean average precision.[2]

[1] An actual LocusLink record of *Locus:16191* contains a larger set of GeneRIF records among other entries such as Gene Ontology terms.

[2] This result was obtained by using the *sum#* and *n#* operators of Inquery , where $n = |query\ term\ tokens| + 2$, and indexing MeSH fields separately (Callan et al., 1992).

3.1 Search Engine

The corpus was parsed by SE by (1) identifying XML fields, (2) tokenizing words, numbers, and non-alphanumeric characters, and (3) indexing all tokens associated with XML fields.

3.1.1 Tokenization

The retrieval was case insensitive. All letters were converted to lower case. All consecutive white spaces were collapsed to a single white space. Tokens containing both alphabetic and numeric characters such as *JAK2* were also searched for their hyphenated variants such as *JAK-2*.

Queries were preprocessed before scoring documents. Commas in queries were treated as separators of independent query terms. Parenthetical expressions in the gene names delimited by white space were considered as separate query terms; however, any other parenthetical expression such as 1(2)*gd*2 was considered as a single token.

3.1.2 Scoring

The final document score was a conjunction of three part scores:

1. on species of interest,
2. on query terms, and
3. on key terms that occurred frequently in GeneRIF citations

If the exact species name or a variant of the term denoting the organism of interest (i.e., org^+) was not found in the `<MeshHeading>` field, then the likelihood of the document was of interest (i.e., d^+) was lowered drastically. Namely,

$$\frac{P(d^+|org^+)}{P(d^+|org^-)} = 1000 \qquad (1)$$

Each query term t_i (e.g., *Slowpoke binding protein*) observed in an XML field (e.g., $xml(t_i) =$ `<ArticleTitle>`) was associated with a subjective probability $P(xml(t_i))$, which reflects our belief how likely a document is of interest given that the query term was observed in $xml(t_i)$.

$$\begin{aligned} P(d^+, t_i) &\equiv P(d^+, xml(t_i)) \\ &= P(d^+)P(xml(t_i)|d^+) \end{aligned} \qquad (2)$$

The values of $P(xml(t_i)|d^+)$ for fields `<ArticleTitle>`, `<MeshHeading>`, `<NameOfSubstance>`, and `<AbstractText>` are 0.9, 0.6, 0.6, and 0.4, respectively, and $P(d^+) = 0.8$.

If the observed term t_i was not an exact query term, but an inflectional variant $lex(t_i)$ or a synonym $syn(t_i)$ of the exact query term, then

$$P(d^+,t_i) \equiv P(d^+, xml(lex(t_i))) = P(d^+)P(xml(lex(t_i))|d^+) = P(d^+)P(xml(t_i)|d^+)P(lex(t_i)|d^+) \quad (3)$$

The values of $P(lex(t_i)|d^+)$ and $P(syn(t_i)|d^+)$ are 0.9 and 0.8, respectively.

There were a number of other factors considered in scoring query terms, such as:

1. The specificity of the query term $spc(t_i)$ to the gene of interest. The term $P(spc(t_i)|d^+)$ was a multiplicative factor similar to $P(lex(t_i)|d^+)$ and had different probability assignments for preferred gene names (0.9), official gene names (0.8), preferred symbols (0.7), official symbols (0.6), symbol aliases (0.5), preferred product names (0.3), product names (0.3), protein alias (0.1), and derived terms[3] (0.01).
2. Multiword terms with $P(spc(t_i)|d^+) > 0.5$ were subject to phrase relaxation: If all tokens of a multiword phrase were found in arbitrary locations of an XML field of a document, then
$$P(d^+,t_i) = P(d^+)P(xml(t_i)|d^+) P(rlx(t_i,n,m)|d^+) \quad (4)$$
where $P(rlx(t_i,n,m)|d^+)$, n, and m denote conditional probability of the phrase relaxation score, the number of subphrases and number of tokens in term t_i and have the following characteristics:
$$0.01 \le P(rlx(t_i,n,m)) = 0.01^{\frac{n-1}{m-1}} \le 1 \quad 1 \le n \le m \ne 1 \quad (5)$$
3. Names of species were mapped into corresponding Medical Subject Headings (MeSH®) term equivalent:
   Homo sapiens → *Human*
   Mus musculus → *Mice*
   Rattus norvegicus → *Rats*
   Drosophila melanogaster → *Drosophila*
4. The query term *Rats* would match both MeSH terms "*Rats, Mutant Strain*" and "*Rats*." The latter is considered as an exact match. Query terms that exactly matched to the MeSH terms of interest contributed to the overall score with higher conditional probabilities.

When multiple query terms were observed in a given document, the probability that the document was of interest was computed as the union of the probabilities of all occurrences. For example, let d be a document, in which the query terms t_i and t_j were observed once. The probability that d was of interest was computed as follows:

$$P(d^+,t_i \cup t_j) = P(d^+,t_i) + P(d^+,t_j) - P(d^+,t_i \cap t_j) \quad (6)$$

The probability term of the co-occurrence $P(d^+,t_i \cap t_j)$ was computed with the assumption that t_i and t_j were independent from each other.

$$P(d^+,t_i \cap t_j) = P(d^+,t_i)P(d^+,t_j) \quad (7)$$

If a term or its lexical variant $lex(t_i)$ was observed in a given XML field of a document *J* times, the probability of the document relevancy was computed as follows:

$$P(d^+,\bigcup lex(t_i)) = P(d^+)4\sum_{j=1}^{J} P(lex(t_i)|d^+)5^{-j} \quad (8)$$

Earlier research suggested that key context terms (i.e., with unusually high frequency counts only in that context) may be beneficial to filter in documents of interest from large corpora (Tanabe et al., 1999). A new set of key context terms $\{k_1,...,k_{46}\}$ frequently occurred in GeneRIF citations, such as *genetics*, *gene expression*, and *sequence*, was collected ad hoc. The third part of the document probability was computed by evaluating the document with respect to all lexical variants $\{lex(k_i)\}$ of these terms.

$$P(d^+, xml(k_i)) = P(d^+)P(xml(k_i)|d^+) \quad (9)$$

where $k_i \in \{\{lex(k_1)\},...,\{lex(k_{46})\}\}$.

The final document probability was obtained as

$$P(d^+,org,\bigcup t,\bigcup k) = P(d^+,org)P(d^+,\bigcup t)P(d^+,\bigcup k) = P(d^+,org)\bigcup_{t_i} P(d^+,t_i)\cdot\bigcup_{k_i} P(d^+,k_i) \quad (10)$$

3.1.3 Results and Analysis

The performance of SE was evaluated as 0.4168 in mean average precision using the trec_eval program provided by TREC organizers. In order to understand which heuristics played the major factor in obtaining this result, SE was rerun by excluding a different heuristic at each run. The largest performance drop (0.085) was observed when the organism names on the test queries were not mapped to the corresponding MeSH terms. A similar performance drop (0.081) was observed when the differentiations among XML fields were disregarded. When one-to-one (exact) mapping between MeSH terms and query terms was not taken into account, the retrieval performance dropped by 0.06. Performance drops

[3] A derived term is a portion of a query term that is delimited by commas or parentheses.

through the exclusions of any other heuristics (searching hyphen variants, frequently co-occurring terms, phrase relaxation, and synonymy) were insignificantly small (between 0.03 and 0.003).

The relatively low performance contributions of phrase relaxation and synonymy were surprising since these features provide empirically significant benefits in *ClinicalTrials.gov* environment. The difference might be due to the relatively well structured queries and the low counts of gene name synonymy found in the current version of the Unified Medical Language System® (UMLS®).

The possible performance differences between SE and other IR approaches might partially be due to SE's tokenization and scoring methods. For example, SE conserves parenthetical information in terms such as 1(2)*gd*2 and includes them in the search while others might search *gd*2 only. Unlike many other IR approaches, SE does not use an inverse document frequency statistics in evaluating the importance of a token, since some commonly occurring words (e.g., *rat*, *fat*, *human*) may play crucial roles under specific conditions, as in the current task and should not be devaluated due to their high frequencies in the corpus.

4 A Rule-Based Approach

This portion of our research focuses on how effectively we can utilize MeSH resource. MeSH is a controlled vocabulary consisting of medical terms organized in a set of broader-narrower hierarchies.[4] A decision list composed of 18 nodes each of which was a propositional logic clause (i.e., a rule-based search statement consisting of MeSH terms using Boolean logic connectors) was developed to identify GeneRIF citations in the MEDLINE test corpus. A decision list (Rivest, 1987) is a special case of decision trees where each internal decision node has only one direct descendent (see Figure 1).

The traversal on the decision list is terminated when the propositional clause (*PC*) represented on the currently visited decision node is satisfied. If none of the *PC*s were satisfied, the document would be labeled as *negative* and would not be retrieved. The approach of using a sequence of models (such as *PC*s) and switching between them in this manner is sometimes called model switching (Kayaalp, Pedersen, & Bruce, 1997).

[4] More information about MeSH can be found in http://www.nlm.nih.gov/mesh/meshhome.html.

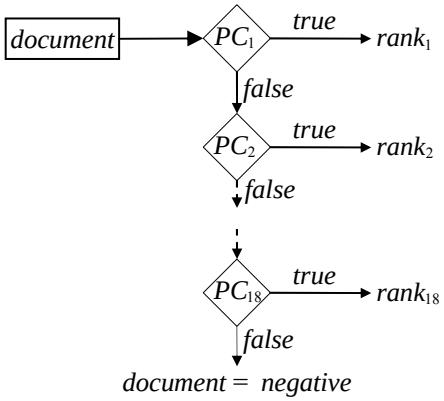

Figure 1: A Decision List with 18 *PC* Nodes

4.1 Controlled Vocabulary Search

In contrast to our other two approaches, this part of the study mainly relies on search on the structured data portion (MeSH) of the corpus. Since MeSH is a controlled vocabulary (as opposed to free text), the presented search method is called controlled vocabulary search (CVS).

CVS is composed of three subsequent phases:

1. Identification of MeSH terms in queries through pattern matching against MeSH records
2. Transformation of each query into a decision list of 18 *PC*s with MeSH terms
3. Search against the MeSH fields of each MEDLINE citation, and output the search result.

The first phase consisted of pattern matching between query terms and MeSH terms that are found in MeSH *regular descriptor* file and MeSH *supplementary concept records* file. Techniques were used for query expansion, tokenization, and eliminating results due solely to matching an acronym on the query side with an acronymic MeSH term. Names of species provided in query terms were converted to the corresponding MeSH terms as stated in Section 3.1.2.[5]

The second phase consisted of constructing a decision list of 18 *PC*s for each query. Each *PC* is a conjunction of three parts: (1) species name, (2) gene name, (3) MeSH qualifiers describing biomedical functions, with the exceptions of $PC_{16} - PC_{18}$, which were composed of only (1) and (2). These parts correspond to how indexers would likely index a document discussing function of a particular gene in a particular species. A simplified example of *PC* is

[5] This approach was considered a manual run since the algorithms identifying MeSH terms for gene names were subsequently modified for the test queries before the test run.

shown in Proposition (11), which corresponds to gene query 18, where the species name is *Mice*, gene name is *Interleukin-5*, and a set of MeSH qualifiers *genetics*, *physiology*, and *metabolism*:

$$PC_i\,(Query = 18):\\ Mice \wedge Interleukin\text{-}5 \\ \wedge\,(genetics \wedge (physiology \vee metabolism)) \quad (11)$$

Any MEDLINE citation indexed under *Mice* and one of the following terms would be retrieved (i.e., be satisfied by Proposition (11)):

- Interleukin-5/genetics/physiology
- Interleukin-5/genetics/metabolism
- Interleukin-5/genetics/physiology/metabolism

The order of the propositional clauses (*PC*s) was designed to maximize retrieval precision.

Ad hoc analysis of the training dataset yielded a set of nine MeSH qualifiers (e.g., *genetics*, *physiology*, *metabolism*) and 16 MeSH hierarchical nodes (e.g., *Cell Physiology*, *Gene Expression*, *Neoplasm Protein*) that occurred frequently in GeneRIF citations.

The last phase was the search for relevant citations against the test corpus and output of retrieval results. A citation was retrieved if at least one of the *PC*s was satisfied. As illustrated in Figure 1, the rank of a document was determined by the decision list order of the *PC* that retrieved the document first.

4.2 Results and Analysis

The retrieval performance of CVS was 0.34 (in mean average precision) on the training query set and 0.23 on the test query set.

We also tried an alternative ranking strategy based on the number of functional keywords (Tanabe et al., 1999) contained in the retrieved documents. The strategy improved the retrieval performance by 0.04, which however was not sustained when CVS was used along with SE and a collocation network to retrieve GeneRIF citations as explained in Section 6.

Retrospective analysis of results suggest that recall rate might have been improved if an additional set of *PC*s corresponding to a single *gene* conjunct was appended to the existing *PC*s.

As expected, many GeneRIF citations contained MeSH terms of proteins that were associated with the genes of interest. The current CVS algorithms for matching genes with MeSH protein terms found a reasonable match in 92% of cases, suggesting that a maintained crosswalk between genes and MeSH protein terms would be a valuable knowledge resource for the CVS method to answer gene queries posed to MEDLINE.

5 A Machine Learning Approach

Conventionally, supervised learning involves a classification task and a training dataset. A training corpus is usually evaluated as a bag of words associated with a specific class. After the model is learned on training corpus, it is used to classify a new corpus (test set). For example, given all MEDLINE citations published before 2003, a model may be learned to identify whether a citation is about a particular gene (e.g., *Interleukin-5*). Then that model may be used to classify any MEDLINE citation published in 2003 with the same classification criterion (e.g., whether a citation is about *Interleukin-5* or not).

In contrast to the usual classification task, the current problem does not yield a persistent class that remains the same in both training and test cycles. Each test of the primary task involves a new class that was not available during the training phase. Obviously, an *Interleukin-5* model (i.e., its *class* = *Interleukin-5*, its *feature set* = $\{Interleukin, IL, ...\}$, and parameters defined on them) would be useless in classifying whether a document is a citation on *Tropomyosin-1*(*alpha*). In other words, our higher-level question becomes:

> *How can we solve a classic information retrieval problem through machine learning methods?*

This part of the study was exploratory in nature; we looked for empirical evidence as to whether abstraction could be an answer to the above question. The class assignment was abstracted from a particular gene name to a Boolean decision of relevancy; i.e., $class = +$ denotes that a document is about a gene of interest.

Abstracting only class (from a particular instance of gene name such as *Interleukin-5* to a generic gene of interest) would yield a model that learns a set of genes. Such a model may be trained to discriminate whether a document is a GeneRIF citation. The features of such a model may be a set keywords that would be specific to GeneRIF domain as the ones used in SE or MedMiner (Tanabe et al., 1999) but not specific to a particular gene. Obviously, such a system cannot identify whether a citation is about a particular gene of interest, which however was required by the primary task. In this study, in addition to abstracting the class of interest, the feature set was also abstracted from a set of phrases to a set of phrase containers, *n*-grams, of different sizes. For example, a gene name *Indian hedgehog* would be abstracted to

2-gram, and if a document contains the term then the variable 2-gram = + .

The premise was that if the training and test corpora were randomly sampled from the same population of citations, then the characteristics of $(class, \{n\text{-gram}\})$ distribution could be informative for inferring the document class of interest. For example, it is expected that a document that has a three-word phrase as part of the original query term is more likely to be a document of interest compared to another document that only has a two-word phrase of the query term.

$$\begin{aligned} &P(class = + \,|\, 2\text{-gram} = +, 3\text{-gram} = -) \\ &< P(class = + \,|\, 3\text{-gram} = +, 4\text{-gram} = -) \end{aligned} \quad (12)$$

Thus, two documents each of which satisfies two different conditions in (12) would be assigned different probability scores and ranked accordingly. The training corpus would serve as a means of learning the values of these probability mass functions.

5.1 Collocation Networks

A collocation network is a Bayesian network whose structure reflects the dependency hierarchy of morphological (e.g., lexical) and/or conceptual (e.g., semantic) collocations observed in corpora of interest. In this work, the focus is on lexical collocations. The root of the (collocation) network (see Figure 2) is the class *C* representing whether a given document *d* is of interest. Two other morphological constructs, titles and abstracts, were also considered. Their representations in the network were based on the assumption that the presence of lexical structures in titles is independent of the presence of lexical structures in abstracts, given *C*.

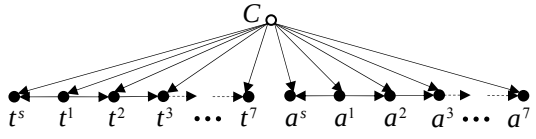

Figure 2: The Collocation Network Structure

Each descendent of the root is a member of an ordered set of *n*-grams, where $1 \le n \le 7$. Let w_i and w_j denote two different words, and (w_i, w_j) denote a two-word phrase in which w_i is followed by w_j. Since $P((w_i, w_j)|d)$, the probability that (w_i, w_j) is present in a given document *d*, depends on the presence of both w_i and w_j in *d*, a natural order for the *n*-gram dependency relationships may be

$$w^{n-1} \rightarrow w^n \quad (13)$$

where w^{n-1} is an $(n-1)$-gram and w^n is an *n*-gram containing w^{n-1}. The topological order in (13) was assumed in the network. In Figure 2, t^n and a^n where $1 \le n \le 7$ denote *n*-grams obtained from gene names and symbols that were observed in titles and abstracts, respectively. The nodes t^s and a^s represent gene symbols, which usually are single words, from titles and abstracts, respectively.

The *n*-gram variables of the network were generated by conserving the word order in gene names and gene symbols. For example, the gene name *Slowpoke binding protein* yields the following three *n*-grams in the first pass:

$$\begin{aligned} 1\text{-gram} &= \{(slowpoke), (binding), (protein)\} \\ 2\text{-gram} &= \{(slowpoke, binding), \\ &\quad (binding, protein)\} \\ 3\text{-gram} &= \{(slowpoke, binding, protein)\} \end{aligned} \quad (14)$$

In the second pass, each *n*-gram set is populated with the lexical variants of its elements using the SPECIALIST Lexicon (McCray, 1998). The lexical variants of *slowpoke*, *binding*, and *protein* are *slowpokes*, *bindings*, *bind*, *binds*, *bound*, *bounded*, *bounding*, *bounds*, and *proteins*, based on which all combinations are generated for each *n*-gram. If a given document abstract contains a three word phrase which is a member of the 3-gram set {(*slowpoke*, *binding*, *protein*),…,(*slowpokes*, *bounds*, *proteins*)} , then the variable a^3 would be labeled as positive for that document. All variables were labeled accordingly and this protocol was followed for each query (i.e., for each gene) separately.

5.2 Results and Analysis

Unlike SE, the collocation network presented in this study was in an early phase of its development. Even though it was not ready to be used as a stand-alone IR tool, our preliminary results on the training set (using leave-one-out cross-validation) indicated that the collocation network in its current state of development might improve overall retrieval performance if its results were combined with the results of SE and CVS using model averaging as explained in Section 6. For the training dataset, the retrieval performance of the collocation network was 0.24 in mean average precision using leave-one-out cross-validation. For the test queries, the retrieval performance of the collocation network was 0.11. The retrieval performance of the system combined with the other two systems is analyzed in Section 6.

Given the training results were obtained through leave-one-out cross-validation, which is known to be a very conservative measurement, the degradation of the performance indicates that the parameters learned on the training set were not as applicable to the test set.

The underlying assumption that was made in retrieving documents for the test queries was that training and test queries were selected randomly from the same population of queries. Any selection bias (such as elimination of certain queries through post-processing) may have caused degradation of the results. Had we had a larger set of training queries (a larger sample size), parameter learning and retrieval results might have been more robust.

This portion of the study was intended to evaluate the value of collocation information. The results suggest that a collocation network based IR system using MeSH indices may be useful in classification and retrieval of genomic information.

6 Model Averaging

In this study, we combined outputs of SE, CVS, and a collocation network through model averaging using uniform priors. We call the resulting system SCC. Model averaging is in line with the Bayesian approach, which suggests using all possible models in making inference. Since it is generally intractable, the approach is usually relaxed to model selection or selective model averaging (Heckerman, Meek, & Cooper, 1999).

Every document *d* in the corpus was labeled by each system as negative or as positive for a given query. Every positive label was associated with a rank order. Given a query and a document *d*, a composite relevancy score $crs(d)$ was obtained as follows:

$$crs(d) = c - \sum_{s \in \{SE, CVS, coNet\}} \alpha_s \cdot rank_s(d) \qquad (15)$$

where $rank_s(d)$ is the rank order of *d* according to system *s*; α_s is the prior and was set to $1/3$ for every *s*; $c \geq \sum_s \max(rank_s(d))$ is a constant and was set to 3000 in this study; and *coNet* denotes the collocation network. If *d* was evaluated as negative by *s*, then $rank_s(d) = c$. The rank order of the document was determined by a decreasing order of $crs(d)$, where

- the most relevant document has the highest $crs(\cdot)$, and
- documents with equal $crs(\cdot)$ share the same rank.

6.2 Results and Analysis

The retrieval performance of the combined system (SCC) was measured in mean average precision as 0.52 on the training set and 0.40 on the test set. Analysis of the test set results of SCC and SE reveals that SE was more precise in ranking documents but SCC was more robust in recall where the retrieval maximum was set at 1000 documents. SCC not only recalled all positive cases that SE identified but also recalled additional cases that SE missed. The difference in robustness is more obvious in Figure 3, where AUCs denoting the areas under the receiver operating characteristics (ROC) curves were plotted per query for both SE and SCC. As seen in this plot, SCC consistently performed with an $AUC > 0.7$.

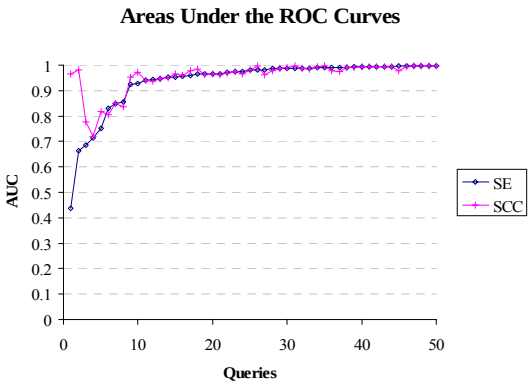

Figure 3: Retrieval performances of SE and SCC compared using ROC metric with queries sorted in increasing order of AUC of SE.

The difference between SE and SCC in terms of AUC is statistically significant: the mean AUC score of SE remains outside of the 95% confidence interval of AUC of SCC (see Figure 4). The values were obtained through bootstrapping with 10,000 samples.

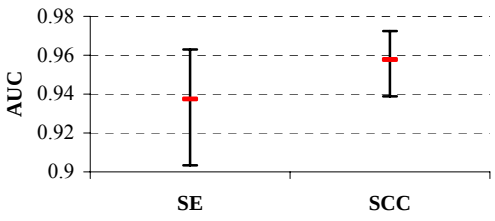

Figure 4: The mean AUC of SE remains outside of the 95% confidence intervals of AUC of SCC.

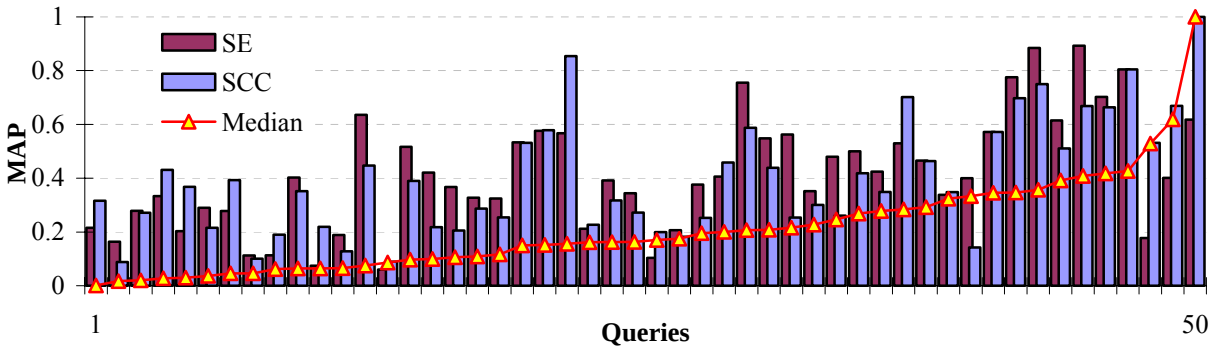

Figure 5: Retrieval performances of SE and SCC compared using mean average precision (MAP) against median performance values of other systems with queries sorted by Median in ascending order.

In Figure 5, the retrieval performances of SE and SCC were also compared against the median retrieval performance of other systems participated to the genomics track in 2003.

Except in one case, the retrieval performance of SCC did not drop below the median performance level.

7 Secondary Task

The secondary task of the genomics track was an information extraction (IE) task and its goal was defined as reproducing the GeneRIF annotations from MEDLINE citations. Our initial analysis revealed that 95% of GeneRIF annotations were taken form titles or abstracts of the corresponding MEDLINE citations, 42% of which were direct quotations. Thus, we decided that finding best sentences in the corresponding MEDLINE citations might serve the purpose of the secondary task.

We used a set of 9,403 recent MEDLINE documents associated with LocusLink GeneRIF records. After excluding 133 abstracts associated with the TREC test set, the documents were divided into a training set and a development test (*devtest*) set. The documents were segmented into sentences from the titles and abstracts so that each annotation A_i was associated with a set of candidate sentences $C_{i,1},...,C_{i,n_i}$. The Dice score $s(C_{i,j})$ was computed for all *i* and *j*.

Our objective was to find a selection function $sel(A_i)$, which returns a sentence $C_{i,m}$ that maximizes the Dice score; i.e., $s(C_{i,m}) \geq s(C_{i,j})$ for all *j*. If the best possible candidate was selected, the average Dice score on the training set was 78.6%. Restricting candidate sentences to either titles or abstracts only, the best possible average Dice scores were 55.0% and 62.2% respectively. Selecting the first sentence from each title yielded 54.7%, which we used as baseline.

7.1 Methods and Results

We assumed that the optimal selection function depends on textual features of the candidate sentences. A very broad range of features familiar to the field of information retrieval and text summarization was considered, as summarized in Table 2. Feature values $f_j(C)$ were computed for each feature f_j and candidate sentence *C*. Two forms for the selection function were considered: (A) a weighted linear form, and (B) a predicate calculus form. In both cases, machine learning algorithms were used to search for the optimal instantiation.

A. In the weighted linear formula, weights w_j were associated with each feature, and a selection score was computed for each candidate sentence by the formula $sel\_score = \sum_j w_j f_j(C)$. The selection function was then defined by selecting from the candidate sentences the one with the largest selection score. Two indirect methods were employed to quickly compute w_j and obtain an estimate for *sel_score*:

   1. Linear regression was used with $f_j(C)$ to predict $s(C)$, and this gave an average Dice score of 48.1%, with the highest weight assigned to the ABS feature (which indicates whether a sentence is in the abstract or title).
   2. The CMLS algorithm (Zhang & Oles, 2001) was used to predict the candidate from each document with the highest Dice score (the objective is 1 for the candidate with the highest Dice score and 0 for all other candi-

Table 2: Lexical Features Used in Secondary Task

Name	Value
ABS	1 if sentence is in abstract
ALAST	1 if sentence is last sentence of abstract
ANUM	the sentence number for sentences in the abstract
GENE	1 if the sentence contains a gene name, determined by Abgene (Tanabe & Wilbur, 2002)
GOOD-LEN	1 if the sentence is in the title and has 5 or more words, or it is in the abstract and has 40 or fewer words
GOOD-NUMS	1 if sentence is in title, or number 7, 8, or 9 in abstract
HD*w*	1 if word *w* occurs as the head word of a noun phrase in the sentence (for 92 most frequent stemmed words in the training set)
LEN	number of characters in sentence
MAX-SENT	maximum Dice coefficient of this sentence to some other sentence in the document
MM	number of MedMiner keywords in the sentence (Tanabe et al., 1999)
MM*k*	1 if MedMiner keyword *k* occurs in the sentence (for the 78 most frequent keywords in the training set)
NBW	number of words that also occurred in the Brown or Wall Street Journal corpus
NUM-CAPS	number of capital letters occurring in the sentence
NUMDIGITS	number of digits occurring in the sentence
NUM-WORDS	number of words occurring in the sentence
REL	the relevancy score based on most frequent words in a document (Ishikawa, Ando, & Okumara, 2001)
ST*n*	the score for semantic index *n* (for 129 different types (Humphrey, Rindflesch, & Aronson, 2000))
T1*t*	1 if POS tag *t* is the first tag of the sentence (for 34 most frequently occurring first POS tags in the training set). For example, T1*DT* = 1 if the first word of the sentence is a determiner.
TDICE	the maximum Dice coefficient of the sentence with a sentence in the title (for abstract sentences only)
TNUM	the sentence number for sentences in the title
TNM	the sentence number of sentences in the abstract or the sentence number of the sentence in the title plus the number of sentences in the abstract
WORDS	sum of Bayesian weights of words in sentence

dates). The best Dice score obtained with this approach was 53.3% using only the ABS indicator, and could not be significantly increased by adding other features.

Finally, we implemented an incremental search algorithm to maximize the Dice score directly, one feature at a time. With weights trained on the training set and feature combinations selected based on the average Dice score of the *devtest* set, the best selection score function obtained was

$$sel\_score(C) = -f_{\mathrm{ABS}}(C) + 0.19\, f_{\mathrm{REL}}(C) - 0.22\, f_{\mathrm{T1}DT}(C) \quad (16)$$

which gave an average Dice score of 54.42% on the *devtest* set. The features ABS, REL, and T1 are described in Table 2.

B. We also sought a predicate calculus formula to decide if a given title sentence was a best candidate GeneRIF. We used the Aleph inductive logic programming (ILP) system (Muggleton, 1995; Srinivasan, 2000) to induce a Prolog program (ILP theory) to find good titles using the features REL, NUMWORDS, NUMDIGITS, NUMBROWNWSJ, MM*n*, MM, and NUM-CAPS. Title sentences that had minimum and maximum Dice scores among all candidates were used as negative and positive training examples, respectively. Consistent with all other findings, the induced program almost always selected one of the title sentences. If the ILP theory rejected a title, an abstract sentence covered by the ILP theory was selected. This resulted in an average Dice score of 48.6%, comparable to the linear regression result. In the TREC test set, all but seven GeneRIF candidates were indicated by ILP to be selected from titles.

Based on the average Dice score on the *devtest* set, the best performing method was based on the linear selection score shown in Equation (16). We used this to obtain our TREC submission for the secondary task, which had a average classic Dice of 50.36% on the TREC testing set. We pointed out the large negative weight assigned to the ABS feature in Equation (16) caused all sentences to be selected from titles.

Only five of these sentences were the second sentence in the title and the remaining 134 were the first. For comparison, selecting the first sentence from each title gives an average classic Dice score of 49.83%.

8 Conclusions

The primary task of the track was ad hoc retrieval of GeneRIF citations from a subset of MEDLINE corpus. We applied three approaches: (1) a conventional information retrieval approach using SE, (2) a rule-based decision making approach using CVS, and (3) a machine learning approach using a collocation network. The SE's test results and the test result of SCC, which is a combination of these three systems, were submitted.

Empirical results suggest that SE performed with high precision in most of the cases while SCC consistently showed robust performance.

SE is a deployed system servicing to the on-line users of *ClinicalTrials.gov*. Both CVS and collocation networks are in early phase of their development and can be improved significantly based on the experience that we gained in this study.

The results on the IR portion of the problem were submitted in two sets: (1) the documents retrieved by SE alone and (2) the documents retrieved by the ensemble called SCC, whose retrieval performances were measured in mean average precision as 0.42 and 0.40, respectively. The mean average precision of SE was slightly better than that of SCC, which was statistically not significant. SCC was evaluated as a more robust retrieval system than SE, where the difference of mean AUCs of two systems was statistically significant.

In the information extraction portion of the problem, experiments with different models yielded similar results that are comparable to selecting titles as GeneRIFs with a Dice-coefficient performance measure of 50%. A method capable of selecting the best GeneRIF candidate sentence from a document can achieve a Dice score exceeding 70%. A wide range of lexical features and several machine learning algorithms were applied to select candidates, yet the best result selected all candidates from titles and performed only 0.53% better than selecting the first sentence from the title. A deeper linguistic analysis at semantic or discourse level might give better performance; however, the heterogeneity of GeneRIF choices suggests even more elusive pragmatics and/or cognitive modeling may be required to achieve optimal results.

References

Callan, J. P., Croft, W. B., & Harding, S. M. (1992). The INQUERY Retrieval System. Proceedings of the *Third International Conference on Database and Expert Systems Applications*, 78–83.

Heckerman, D., Meek, C., & Cooper, G. F. (1999). A Bayesian Approach to Causal Discovery. In C. Glymour, & G. F. Cooper (Eds.) *Computation, Causation, & Discovery*, 141–165. Menlo Park, CA: AAAI/MIT Press.

Humphrey, S. M., Rindflesch, T. C., & Aronson, A. R. (2000). Automatic Indexing by Discipline and High-Level Categories: Methodology and Potential Applications. Proceedings of the *11th ASIST SIG/CR Classification Research Workshop*, 103–116.

Ishikawa, K., Ando, S., & Okumara, A. (2001). Hybrid Text Summarization Method Based on the TF Method and the LEAD Method. Proceedings of the *Second NTCIR Workshop*.

Kayaalp, M., Pedersen, T., & Bruce, R. (1997). A Statistical Decision Making Method: A Case Study on Prepositional Phrase Attachment. Proceedings of the *1997 Meeting of the ACL SIG in Computational Natural Language Learning* (CoNLL97), 33–42.

McCray, A. T. (1998). The nature of lexical knowledge. *Methods of Information in Medicine*, 37(4-5), 353–360.

McCray, A. T., Ide, N. C., Loane, R. R., & Tse, T. (2004). Strategies for Supporting Consumer Health Information Seeking. Accepted for publication in the proceedings of *MedInfo 2004*.

Muggleton, S. (1995). Inverse Entailment and Progol. *New Generation Computing*. Special Issue on Inductive Logic Programming, 13, 245–286.

Rivest, R. L. (1987). Learning Decision Lists. *Machine Learning*, 2, 229–246.

Srinivasan, A. (1999). The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/research/areas /machlearn/Aleph/.

Tanabe, L., Scherf, U., Smith, L. H., Lee, J. K., Hunter, L., & Weinstein, J. N. (1999). MedMiner: An Internet Text-Mining Tool for Biomedical Information, with Application to Gene Expression Profiling. *BioTechniques*, 27(6), 1210–1217.

Tanabe, L., & Wilbur, W. J. (2002). Tagging Gene and Protein Names in Biomedical Text. *Bioinformatics*, 18, 1124–1132.

Zhang, T., & Oles, F. J. (2001). Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval*, 4, 5–31.