Automatic Construction of Knowledge Base from Biological Papers

Yoshihiro Ohta * **Yasunori Yamamoto**
Tomoko Okazaki **Ikuo Uchiyama** **Toshihisa Takagi**
Human Genome Center, Institute of Medical Science,
University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108 Japan
{yoh,yas,okap,uchiyama,takagi}@ims.u-tokyo.ac.jp

Abstract

We designed a system that acquires domain specific knowledge from human written biological papers, and we call this system IFBP (Information Finding from Biological Papers). IFBP is divided into three phases, *Information Retrieval* (IR), *Information Extraction* (IE) and *Dictionary Construction* (DC). We propose a query modification method using automatically constructed thesaurus for IR and a statistical keyword prediction method for IE. A dictionary of domain specific terms, which is one of the central knowledge sources for the task of knowledge acquisition, is also constructed automatically in the DC phase. IFBP is currently used for constructing the *Transcription Factor DataBase* (TFDB) and shows good performance. Since the model of knowledge base construction that is adopted into IFBP is carried out entirely automatically, this system can be easily ported across domains.

Introduction

Every day there is a large influx of newly published papers, each with the potential of being relevant to the interest of multiple researchers and an amount of information in genome area has been growing rapidly in recent years. As the volume of biological information increases, the demand for the domain specific knowledge base grows. Most data on biological functions, such as molecular interactions which are crucial for the next stage of genome science, are still only in literatures. The task of knowledge base construction has generally been done by human experts. However, as the flood of information increases, it becomes difficult for humans to construct and maintain the knowledge base consistently and efficiently.

Therefore, we present an automatic knowledge base construction system to support biologists who are in need of constructing a knowledge base. The task of this system IFBP is divided into three phases, IR, IE and DC. IR returns a set of papers that is relevant to a user's needs (Lewis & Jones 1996), and IE returns a structured representation of knowledge within the retrieved paper (Cowie & Lehnert 1996), which will constitute the entry of the knowledge base. With IFBP, papers of interest can be collected from a text database effectively and exhaustively in the IR phase, and keywords are automatically extracted from retrieved papers in the IE phase. In addition, the domain specific dictionary are also constructed automatically in the DC phase as the central sources for the task of IFBP (see Figure 1). The overall system architechture is outlined in the last section, and algorithms adopted to each phase will be described in detail.

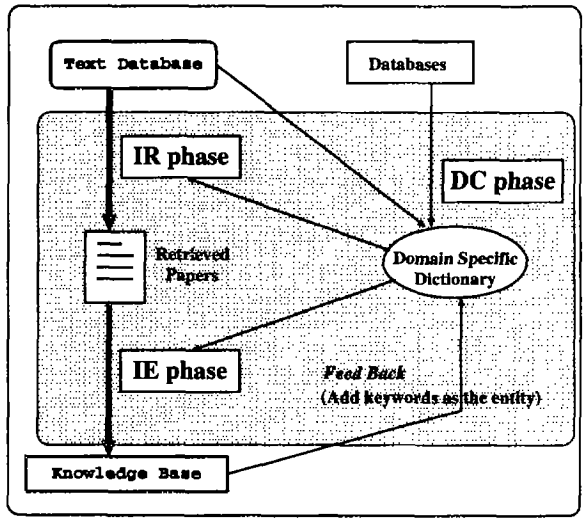


Figure 1: Overall architecture of IFBP

DC(Dictionary Construction) phase

Although a dictionary of domain specific terms is one of the central knowledge sources of the knowledge acquisition system, the on-line domain specific dictionary

*Present address: Tokyo Research Laboratory, IBM Japan. 1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242 Japan

hardly exists due to the difficulties in its construction, that is, the task of dictionary construction is time-consuming and labor-intensive. Therefore, we propose an automatic dictionary construction method adopted in the DC phase (Yamamoto 1996). The DC phase constructs domain specific dictionary from a corpus and existing databases with minimal human supervision (see Figure2).

At the first step of our strategy, terms categorized in a specific class such as "protein name" are collected from the public databases, for example SWISS-PROT, PRF and PIR etc., and transformed into appropriate forms. We call this initial collection a base dictionary.

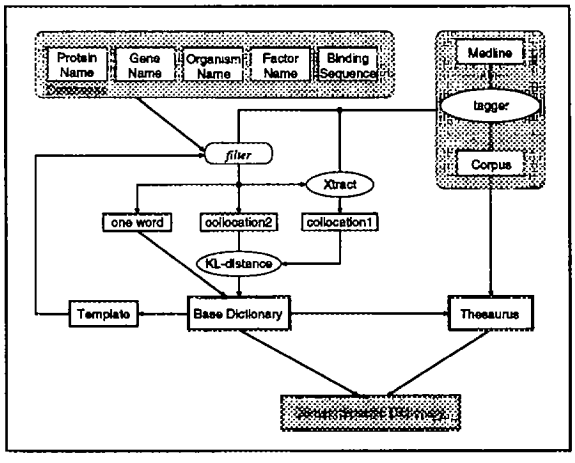


Figure 2: Automatic dictionary construction in DC phase

Next, context of each term in the actual literatures are examined and hierarchical term clustering is performed on the base dictionary. For further discussion regarding this automatic dictionary construction task, see (Yamamoto 1996). The resultant clusters can be used as a thesaurus for IR and IE.

While there are several clustering algorithms, such as single-link method and Ward's method, almost all the previous algorithms use the measure of distance between two objects and merge the close ones (Anderberg 1973, Willett 1988). However, a probabilistic clustering algorithm called Hierarchical Bayesian Clustering (HBC) constructs a set of clusters that has the maximum Bayesian posterior probability and this maximization is a general form of the well known *Maximum Likelihood* estimation. IFBP adopts HBC as clustering algorithm since better performance was obtained than the other clustering algorithm through preliminary experiments in text clustering area (Iwayama & Tokunaga 1995).

Automatic thesaurus construction

The outline of the algorithm HBC are briefly reviewed next and this system's method of constructing the thesaurus using HBC are proposed in the following.

HBC(Hierarchical Bayesian Clustering) HBC constructs a cluster hierarchy from bottom to top by merging two clusters at a time. At the beginning, each datum belongs to a cluster whose only member is the datum itself. For every pair of clusters, HBC calculates the probability of merging the pair and selects the best one with the highest probability for the next merge. The last merge step produces a single cluster containing the entire data set.

Formally, HBC selects the cluster pair whose merge results in the maximum value of the posterior probability $P(C|D)$, where D is a collection of data (i.e., $D = d_1, d_2, \cdots, d_N$) and C is a set of clusters (i.e., $C = c_1, c_2, \cdots$). Each cluster $c_j \in C$ is a set of data and the clusters are mutually exclusive. At the initial stage, each cluster is a singleton set; $c_i = d_i$ for all i. $P(C|D)$ defines the probability that a collection of data D is classified into a set of clusters C. Maximizing $P(C|D)$ is a generalization of *Maximum Likelihood* estimation.

To examine the details of merge process, consider a merge step $k+1 (0 \leq k \leq N-1)$. By the step $k+1$, a data collection D has been partitioned into a set of clusters C_k. That is each datum $d \in D$ belongs to a cluster $c \in C_k$. The posterior probability at this point becomes

$$P(C_k|D) = \prod_{c \in C_k} \prod_{d \in c} P(c|d)$$

$$= \prod_{c \in C_k} \prod_{d \in c} \frac{P(d|c)P(c)}{P(d)}$$

$$= \frac{\prod_{c \in C_k} P(c)^{|c|}}{P(D)} \prod_{c \in C_k} \prod_{d \in c} P(d|c)$$

$$= \frac{PC(C_k)}{P(D)} \prod_{c \in C_k} SC(c) \quad (1)$$

Here, $PC(C_k)$ corresponds to the prior probability that N random data are classified into a set of clusters C_k. This probability is defined as follows:

$$PC(C_k) = \prod_{c \in C_k} P(c)^{|c|} \quad (2)$$

$SC(c)$ defines the probability that all the data in a cluster c are produced from the cluster and is defined as

$$SC(c) = \prod_{d \in c} P(d|c) \quad (3)$$

When two clusters $c_x, c_y \in C_k$ are merged, the set of clusters C_k is updated as follows:

$$C_{k+1} = C_k - \{c_x, c_y\} + \{c_x \cup c_y\} \quad (4)$$

After the merge, the posterior probability is inductively updated as

$$P(C_{k+1}|D) = \frac{PC(C_{k+1})}{PC(C_k)} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)} P(C_k|D) \quad (5)$$

Note that this updating is local and can be done efficiently, because the only recalculation necessary since the previous step is the probability for the merged new cluster, that is, $SC(c_x \cup c_y)$. The factor $\frac{PC(C_{k+1})}{PC(C_k)}$ can be neglected for maximization of $P(C|D)$, since the factor would reduce to a constant regardless of the merged pair. See (Iwayama & Tokunaga 1995) for further discussion.

Thesaurus construction This section concerns clustering terms based on the relations they have with attributes. In order to apply HBC to clustering terms, we need to calculate the elemental probability $P(d|c)$, a cluster c actually contains its member term d. To calculate this probability, this paper follows SVMV (*Single random Variable with Multiple Values*) (Iwayama & Tokunaga 1994). With SVMV, the DC phase classifies terms, each one of them being represented as a set of attributes. Each term has its own attributes (the set of nouns co-occurring with this particular term in a sentence is currently used in IFBP) gathered from a corpus, see Figure3.

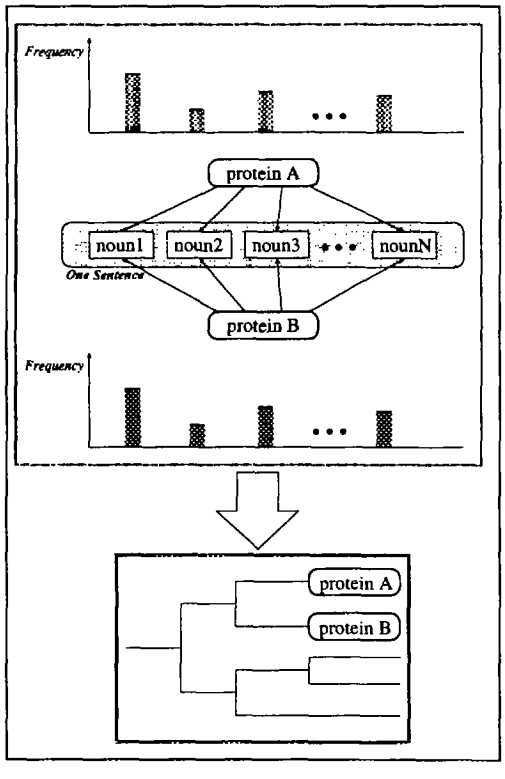


Figure 3: co-occurring nouns as the attribute

In SVMV, while cluster c is a set of terms, it is also represented as a set of attributes that all the members of c co-occur with. Consider an event $A = a$ where a randomly extracted attribute A from a set of attributes is equal to a. Conditioning $P(d|c)$ in each possible event gives

$$P(d|c) = \sum_a P(d|c, A = a)P(A = a|c) \quad (6)$$

if we assume conditional independence between c and d given $A = a$,

$$P(d|c) = \sum_a P(d|A = a)P(A = a|c) \quad (7)$$

Using Bayes' theorem, this becomes

$$P(d|c) = P(d) \sum_a \frac{P(A = a|d)P(A = a|c)}{P(A = a)} \quad (8)$$

Since each $P(d)$ appears in every estimation of $P(C|D)$ only once, this can be excluded for maximization purpose. Other probabilities, $P(A = a|d), P(A = a|c)$, and $P(A = a)$ are estimated from given data by using simple estimation as below. $P(A = a|d)$ is the relative frequency of an attribute a co-occurring with a term d. $P(A = a|c)$ is the relative frequency of an attribute a co-occurring with terms in cluster c. $P(A = a)$ is the relative frequency of an attribute a appearing in the whole training data.

Figure 4 shows an example of dendrogram automatically constructed in our experiments. Domain specific dictionary constructed like above can be used as the machine-readable domain specific dictionary, and also supports high performance retrieval and keyword prediction in IFPB, refer to (Riloff 1995).

IR(Information Retrieval) phase

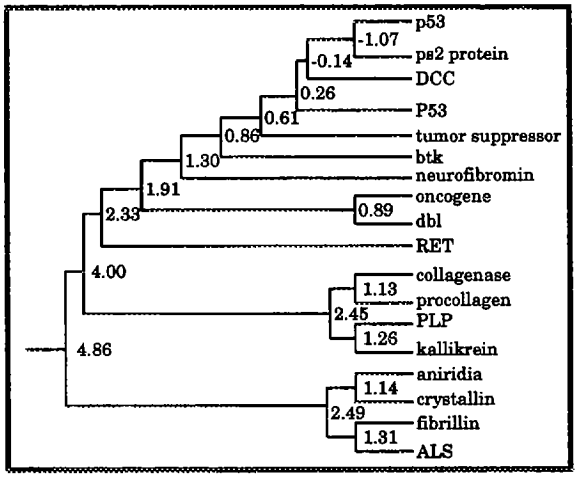
The IR module evaluates a large incoming stream of documents to determine which documents are sufficiently similar to a user's need at the broad subject level. There are several approaches to information retrieval, such as decision rule based (Apte, Damerau, & Weiss 1993), knowledge base based, text similarity based, and so on. This paper focuses on text retrieval based on text similarity and employs the basic vector space model (Buckley et al. 1995, Salton 1988).

Vector space model

The vector space model assumes that each document may be represented by a vector whose non-zero coordinates correspond to the terms contained in that document. Let Q_j stand for a query vector and W_i a document vector, the document representation in vector space is defined as follows.

$$W_i = (w_{i1}, w_{i2}, \ldots, w_{it}) \quad (9)$$

Here, w_{it} represents the weight of term t in document W_i. A weight of zero is used for terms that are absent from a particular document, and positive weights



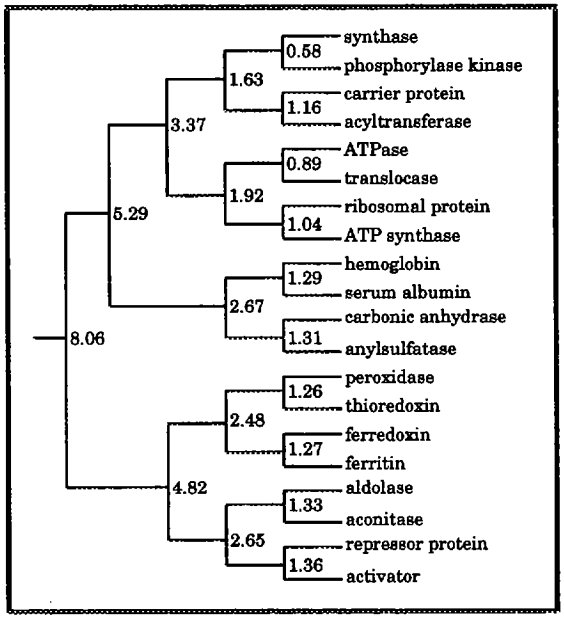


Figure 4: A part of the dendrogram created automatically

characterize terms actually assigned. The assumption is that all terms are available for the representation of the information. Term weighting is one of the important issues in text retrieval. Originally, term weighting has been widely investigated (Salton & McGill 1983, Salton 1988). A high-quality term weighting formula (Salton & Buckley 1988, Salton 1988) w_{ik}, the weight of term T_k in document W_i is

$$w_{ik} = \frac{(\frac{1}{2}(1 + \frac{f_{ik}}{f_i})) \times log\frac{N}{n_k}}{\sqrt{\sum_{j=1}^{t}\left[(\frac{1}{2}(1 + \frac{f_{ik}}{f_i})) \times log\frac{N}{n_k}\right]^2}} \qquad (10)$$

Here f_{ik} represents the occurrence frequency of T_k in W_i, and f_i is the maximum of f_{ik} over all terms T that occur in W_i. The augmented normalized term frequency ($\frac{1}{2}(1+\frac{f_{ik}}{f_i})$) is employed. N is the collection size, and n_k the number of documents with term T_k assigned. The factor $log\frac{N}{n_k}$ is an inverse collection frequency ("idf") factor which decreases as terms are used widely in a collection, and the denominator in Eq. (10) is used for weight normalization. The weights assigned to terms in *documents* are much the same. The terms T_k included in a given vector can in principle represent any entity assigned to a document for identification.

When the document W_i is represented by a vector of the form $(w_{i1}, w_{i2}, \cdots, w_{it})$ and the query Q_j by the vector $(q_{j1}, q_{j2}, \cdots, q_{jt})$, a similarity computation between the two items can conveniently be obtained as the inner product between corresponding weighted term vector as follows:

$$S(W_i, Q_j) = \sum_{k=1}^{t}(w_{ik} \times q_{jk}) \qquad (11)$$

Thus, the similarity between two texts (whether query or document) depend on the weights of coinciding terms in the two vectors.

Query Modification

In the task of IR, one of the most important and difficult operations is generating useful query statements that can retrieve papers needed by the user and reject the remainder. To obtain a better representation of the query, we propose a new query modification method. The method of modifying query uses automatically constructed thesaurus to perform high performance retrieval (Ohta 1997). In our modification method, the terms similar to the attribute in initial query are added in proportion to the distance between the term and the attribute using term-distance matrix Eq.(13). This term-distance matrix is obtained from the domain specific dictionary which is constructed automatically in the DC phase.

Query expansion and term-reweighting using term-distance matrix

The term-distance matrix is obtained from the domain specific dictionary, which has thesaurus information by using the probabilistic algorithm HBC. This paper defines the distance between two terms as the posterior probability calculated in the first step of HBC.

$$d_{xy} = \frac{PC(C_{k+1})}{PC(C_k)} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)} \tag{12}$$

The above d_{xy} indicates the distance between term t_x and term t_y. With this d_{xy}, the term-distance matrix is defined as below.

$$D = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_n \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix} \end{array} \tag{13}$$

$$where, d_{ii} = 1, \quad d_{ij} = d_{ji}, \quad 0 \leq d_{ij} \leq 1.$$

Our experience suggests that many users approach a search with already having some papers in mind that express their interest, and request for papers that are most similar to the paper already judged relevant by the searcher. In this way, initial query is usually constructed using some relevant papers that users have in their mind. With these papers, initial query is defined as:

$$Q = \frac{1}{R} \sum_{Rel} \frac{W_i}{|W_i|} \tag{14}$$

Here, R is the number of relevant papers. Because the number of terms in this initial query is not sufficient for retrieving enough papers, this paper present the modified query defined below to perform high performance retrieval.

$$Q_m = \sum_{k=1}^{n} Q^T \vec{e_k} \cdot D\vec{e_k} \tag{15}$$

The modified query defined by Eq.(15) has more attributes than the initial query, and as a result the number of terms that are in both vector of the query and the relevant paper increases. In addition, when the set of terms are added to the initial query, the distance between two terms are also multiplied by the matrix Eq.(13). Therefore, this query modification method can add the set of terms in proportion to the importance of the terms, and will contribute to the high performance retrieval.

Evaluation of modified query

To evaluate the retrieval performance of IFBP, the recall and precision are evaluated with the corpus, which consists of 5000 papers obtained from the MEDLINE abstracts. 149 papers were relevant to the domain, and in this experiment we chose the transcription factor as subject of our analysis and others as non-relevant. Recall and precision are calculated by the following equations:

$$Recall = \frac{T_P}{T_P + T_N} \tag{16}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{17}$$

Here, T_P is *the number of papers that are retrieved correctly*, $T_P + T_N$ is *the number of relevant papers in the collection of papers* and $T_P + F_P$ is *the number of total papers retrieved.*

In general, recall and precision are mutually exclusive factors, that is, high recall value is obtained at the cost of precision, and vice versa. Figure 5 shows the retrieval performance of IFBP in which the query is generated from fifty relevant papers. The graph shows superior ability of modified query obtained through this query modification method, and the point at which eighty papers are retrieved is the cross point of recall and precision.

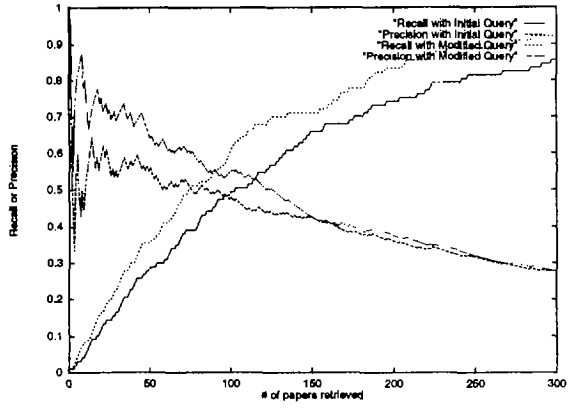


Figure 5: Retrieval performance of modified query

IE(Information Extraction) phase

The IE phase analyzes unrestricted texts in order to extract information concerning pre-specified types of events, entities or relationships (Riloff 1994, MUC5 1993). For example, information available in unstructured text can be translated into databases which can be probed by users through standard queries, see Figure 6.

This ability of IE phase can support a part of the task of knowledge base construction. To do so, IFBP predict keywords related to the knowledge base entry from the papers retrieved in IR phase. WWW interface of IFBP is also presented for the facility of monotonous work. This chapter first describes how IFBP predict

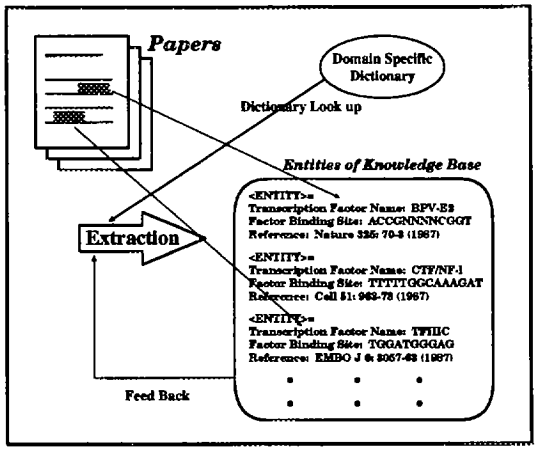


Figure 6: Transformation of text's structures in IE phase

keywords from papers, and then describes interaction between the domain specific dictionary (Liddy, Paik & Yu 1994).

Keyword prediction

One of the major problems in the accurate analysis of natural language is the presence of unknown words, especially names. While it seems convenient to use a list of all the names, names come and go. To resolve this problem, this paper adopts the scoring of words using cross-entropy as shown below, so the complex heuristics are not necessary.

$$M_{td} = \frac{f_{td}}{f_d} log \frac{f_{td}/f_d}{f_{tc}/f_c} \tag{18}$$

Here, f_{td} and f_{tc} is the frequency of term **t** in an input document **d** and corpus **c**. f_d and f_c is the total frequency in a input document **d** and corpus **c**. This information-theoretic measure are also known as KL-distance (Kullback & Leibler 1951). With this scoring method we can obtain the candidate for the keyword.

In addition, domain specific term often appears as collocation. To resolve the ambiguity of recognizing compound words, this paper also defines the score of collocation as bellow.

$$S(c_l|J_l J_{l+1} \cdots J_{l+m-1} N_{l+m} N_{l+m+1} \cdots N_{l+m+n})$$
$$= f(c_i) \cdot \left(\frac{1}{m} \sum_{k=0}^{m-1} M(J_{l+k}) + \frac{1}{n+1} \sum_{k=0}^{n} M(N_{l+m+k}) \right) \tag{19}$$

Here "$J_l J_{l+1} \cdots J_{l+m-1} N_{l+m} N_{l+m+1} \cdots N_{l+m+n}$" is the adjacent word strings of the adjective J and the noun N, and $M(A)$ is the score of a word A defined by Eq.(18).

This paper defines that any pair of adjacent nouns and adjectives is regarded as a potential phrase. The final list of phrases is composed of those pairs of words occurring in 10 or more documents of the document set. In addition, IFBP uses n-gram statistics to filter out non-sense word strings and can predict keywords in high accuracy, for further discussion see (Yamamoto 1996). Using this score, we can recognize and automatically extract domain specific terms in collocation level from natural language texts. The example of the ranking of predicted keywords from 5 papers in the domain of transcription factor (described in the following section) is shown in Figure 7.

```
15.13120  binding site
          [32x(0.25420+0.21865)]
 8.97350  DNA binding
          [25x(0.10474+0.25420)]
 7.47648  transcription factor
          [24x(0.17148+0.14004)]
 3.11744  binding affinity
          [8x(0.25420+0.13548)]
 3.00468  consensus sequence
          [6x(0.27883+0.22195)]
 2.75968  factor binding
          [7x(0.14004+0.25420)]
 2.65518  enhancer element
          [6x(0.18751+0.25502)]
 2.61352  DNA sequence
          [8x(0.10474+0.22195)]
 2.51214  binding specificity
          [6x(0.25420+0.16449)]
 2.26820  DNA binding domain
          [5x(0.10474+0.25420+0.09470)]
 2.07738  recognition sequence
          [6x(0.12428+0.22195)]
 2.03352  gene promoter
          [6x(0.07965+0.25927)]
 1.75791  gene transcription
          [7x(0.07965+0.17148)]
 1.74450  binding domain
          [5x(0.25420+0.09470)]
```

Figure 7: Predicted keywords with score

Dictionary look up and search for the knowledge base

The predicted keyword is scored as described above, and IFBP can distinguish important keywords from others with this score. Although keyword prediction with this method can filter out non-sense word strings to some extent, not all predicted keywords are appropriate terms which are suited for the knowledge base entity. This paper uses the domain specific dictionary constructed in the DC phase to resolve this problem. In addition, IFBP also looks up the knowledge base entities. By searching for the knowledge base entities,

we can sort out the new term that is more impressive for the researcher from the old one that is known well already. The predicted keywords that are classified in this way can contribute to the retrieval of more impressive papers and keywords, and will lead to the construction of excellent knowledge base.

Overall system architecture and WWW interface

Overall architecture of IFBP

We designed and implemented the knowledge acquisition system IFBP that is divided into three phases, IR, IE and DC (see Figure 1).

The IR phase can not only retrieve papers with high potential for being relevant, but also ranks the paper set according to each paper's predicted likelihood of relevance. That is, IFBP is also concerned with "computing the degree of relevance of papers". The IE phase can predict keywords from the relevant papers retrieved in IR phase, and support the construction and management of knowledge base.

In addition, predicted keywords in IE phase are also added as the entity of domain specific dictionary that is constructed in DC phase, and the automatically re-constructed dictionary will be used in future retrieval and extraction operation. Our experiments have shown that this re-constructed dictionary contributes to the improvement of retrieval and extraction performance (Ohta 1997).

Domain of IFBP and corpus

This paper chose the Transcription Factor DataBase (TFDB) as subject of analysis (Okazaki, Kaizawa & Mizushima 1996). As NCBI stopped the maintenance of Transcription Factor Database (TFD) since 1993, it is maintained as a new database TFDB on SYBASE System 11 at National Cancer Center Research Institute, which takes over some parts of the database focusing on the DNA binding sequence data.

The current system is conducted for the corpus described below. For building a tagged corpus, we have used the MEDLINE abstracts as text source All abstracts are tagged beforehand with a part-of-speech tag from Penn Treebank tagset (Marcus, Santorini & Marcinkiewicz 1993) using a slightly customized version of Brill's tagger (Brill 1994).

IFBP is now used as the assistant system constructing and maintaining the TFDB, and shows good retrieval and extraction performance.

WWW interface of IFBP

The WWW interface has been designed and implemented, and this interface can support the researchers in constructing knowledge base by simplifying the task. What can be done automatically with IFBP are listed below.

- Construct the on-line domain specific dictionary and refer to the entries in the separate window.
- Retrieve relevant papers from a textual database, and rank the paper set according to each paper's predicted likelihood of relevance.
- Check out the terms that appears in both relevant papers and knowledge base entity.
- Predict keywords in these relevant papers including unknown words by statistical method.

By the assistance described above, the user can construct a knowledge base by just clicking the keywords shown in the WWW interface. An example of this WWW interface implemented for the construction of TFDB is shown in Figure 8.

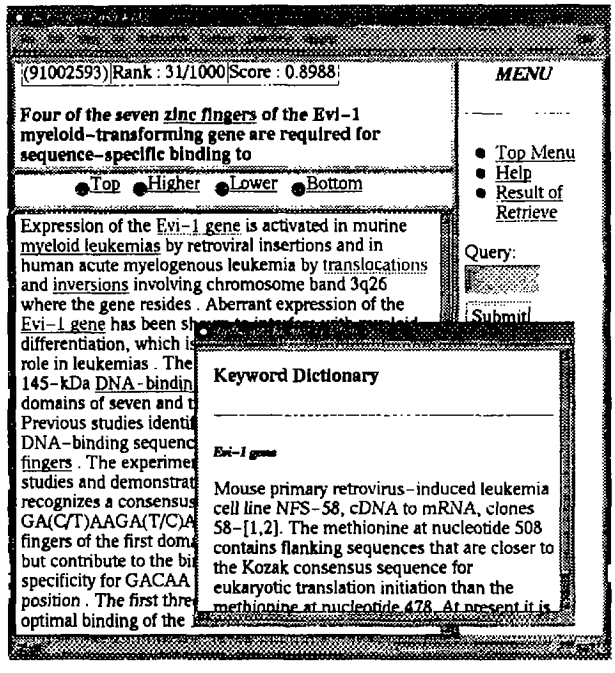


Figure 8: WWW interface for TFDB

Conclusion

We designed the system called IFBP that automatically acquires domain specific knowledge from biological papers. IFBP is divided into three phases, IR, IE and DC. In the IR phase, the query will be modified using domain specific dictionary, which was constructed in the DC phase, and the modified query can retrieve papers needed by the user from textual database, and accurately ranks the paper set according to the likelihood of relevance. In the IE phase, keywords are extracted automatically in a collocation level by a statistical approach from the set of papers retrieved in IR phase, and can filter out non-sense word strings

by looking up the dictionary and entities of knowledge base. The DC phase constructs an on-line domain specific dictionary as a central knowledge source, and the WWW interface of IFBP can simplify the task of knowledge base construction. The important aspect of the model proposed in IFBP is that because models of IR, IE and DC which are adopted into IFBP are carried out entirely automatically, this system can be easily ported across domains.

Acknowledgments

This work was supported in part by a Grant-in-Aid (08283103) for Scientific Research on Priority Areas from The Ministry of Education, Science, Sports and Culture in Japan.

References

Lewis, D. and Jones, K. 1996. Natural Language Processing for Information Retrieval. *Communications of the ACM*, 39(1).

Cowie, J. and Lehnert, W. 1996. Information Extraction. *Communications of the ACM*, 39(1).

Iwayama, M. and Tokunaga, T. 1995. Hierarchical bayesian clustering for automatic text classification. In Proceedings of the International Joint Conference on Artificial Intelligence.

Anderberg, M. 1973. *Cluster Analysis for Applications.* Academic Press.

Willett, P. 1988. Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management* 24(5): 577–597.

Iwayama, M. and Tokunaga, T. 1994. A probabilistic model for text categorization: Based on a single random variable with multiple values. In Proceedings of 4th Conference on Applied Natural Language Processing (ANLP '94), 162–167.

Apte, C.; Damerau, F.; and Weiss, S. 1993. Automatic learning of decision rules for text categorization. Research Report RC19979(82518). IBM.

Buckley, C.; Salton, G.; Allan, J.; and Singhal, A. 1995. Automatic Query Expansion Using SMART: TREC 3. In Proceedings of the Third Text REtrieval Conference (TREC-2). NIST Special Publication.

Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5).

Kullback, S. and Leibler, R.A. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86.

Riloff, E. 1995. Dictionary Requirements for Text Classification: A Comparison of Three Domains. In Working Notes of the AAAI Spring Symposium on Representaion and Acquisition of Lexical Knowledge: Polysemy, Ambiguity and Gnerativity, 123–128.

Riloff, E. 1994. Information Extraction as a Basis for Portable Text Classification Systems. Ph.D. diss. Department of Computer Science, University of Massachusetts Amherst.

Liddy, E. D.; Paik, W.; and Yu, E. S. 1994. Text Categorization for Multiple Users Based on Semantic Features from a Machine-Readable Dictionary. *ACM Transactions on Informaion Systems*, 12(3): 278–295.

Okazaki, T.; Kaizawa, M.; and Mizushima, H. 1996. Establishment and Management of Transcription Factor Database TFDB. In Proceedings of the Seventh Workshop on Genome Informatics, 218–219.

Marcus, M.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19(2): 313–330.

Brill, E. 1994. Some Advances in Transformation Based Part of Speech Tagging. In Proceedings of AAAI.

1993. *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, San Francisco, CA, Morgan Kaufmann.

Ohta, Y. 1997 Representaion of Retrieval Query using Corpus and Machine-Readable Dictionary. Master thesis, The Graduate School of The University of Tokyo.

Yamamoto, Y. 1996. Constructing a Dictionary of Biological Terms for Information Extraction. Master thesis, Graduate School of Information Science and Engineering, Department of Computer Science, Tokyo Institute of Technology.

Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill Publishing Company.

Salton, G. 1988. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley.