

PathwayFinder: Bridging the Way Towards Automatic Pathway Extraction

Daming Yao¹, Jingbo Wang², Yanmei Lu³, Nathan Noble⁴, Huandong Sun⁴,
Xiaoyan Zhu⁴, Nan Lin³, Donald G. Payan³, Ming Li⁵, Kunbin Qu⁶

1: School of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

2, 4, 5: Computer Science Department, University of California at Santa Barbara, Santa Barbara, California, CA 93106, USA

3, 6: Rigel Pharmaceuticals Inc, 240 East Grand Avenue, South San Francisco, CA 94080, USA

dyao@uwaterloo.ca, 519-888-4567 Ext 3287, FAX: 519-885-1208

jbwang, mli@cs.ucsb.edu

yly, nlin, dgpayan, kqu@rigel.com, 650-624-1100, FAX: 650-624-1101

Authors 1 and 2 contributed equally. Corresponding authors: 5 and 6.

Abstract

Automatically mining protein pathway information from the vast amount of published literature has been an increasing need from the pharmaceutical industry and biomedical research community. This task has been proved to be a formidable one. Many systems have been implemented, but few are practical. Some are too restricted and some are overly ambitious. This paper presents the PathwayFinder system with two key innovations that give the system simultaneously generalization power and practical capabilities: (a) PathwayFinder is designed with appropriate level of users' involvement on information extraction, based on the authors' belief that totally automatic pathway retrieval is beyond the current technology; (b) A novel multi-agent architecture is designed to support the need of user-computer interactions and domain extensions. As a result, PathwayFinder is flexible, easy to use, and extendable to be customized to other domains without losing accuracy. We have applied PathwayFinder system to study the ubiquitin cascade pathway. After processing over 8,000 abstracts with 65,000 sentences, over 1,800 relationships have been recovered.

Keywords: Pathway extraction, multi-agent, user-involved extraction, natural language processing, ubiquitin pathway

1. Introduction

There are currently over 12 million PubMed citations dating back to the mid-1960's from the PubMed service at NCBI. The number of published papers is growing at an exponential rate in the last several decades. For some intensively studied research areas, such as the ubiquitination pathway cascade, it becomes impossible for a single person to grasp all the concepts and ideas in a field from this massive information.

Normally, the most important information within raw texts is the relationship among the entities, such as "protein A is activated by protein B" and "small molecule C can inhibit such a process". This kind of information is critical to the process of drug development, from early target identification, to target validation, to lead profiling, and finally to lead development and pre-clinical studies. It serves as the basis for experiment design, assay development, project planning and decision-making. Effectively extracting such relationships from bulky raw texts to structure them into a standard format and turn them into accumulated knowledge is the key to supporting an efficient, rapid drug development process and the goal of our project.

Recently there have been a number of projects aimed at conducting Information Extraction (IE) automatically in the biomedical domain. For a detailed review, see the **Background** section of this paper. In general there are three categories of IE systems. The first one uses simple statistical methods,

such as co-occurrence, to identify protein and gene names. The second category applies natural language processing techniques, such as speech tagging and grammar parsers to handle complex sentences. The third category applies more sophisticated natural language technologies that can handle anaphora as well as extracting a broader range of information.

One major weakness of the current three categories of IE systems is the lack of interaction between systems and users. More accurately, although some systems also provide ways to interact with users, the objectives have always been over-ambitious to focus on fully automated IE system. The interactions extracted from these established systems are rather limited due to their confined domain knowledge integrated with the systems. The templates with which these systems are supplied allow only factual information about particular and *a priori* chosen entities (protein groups, cell types etc.) to be assembled from the analyzed documents. The weakness restricts the current IE systems to a limited application range.

PathwayFinder automatically performs the information extraction, recovers the relationships among the entities of interest, and then builds the knowledge database through manual curation. The system is not limited to any specific domain and is highly expandable in terms of relationships and pattern definitions through rich user-involvement features. Emphasis is on the collaboration between domain users and the system to complete the extraction tasks together for customized purpose. Current versions of all PathwayFinder agents are implemented in Java, which makes them platform-independent.

2. Background

Among the relationships of biomedical entities, protein-protein interactions are the most basic ones for many biological processes and the building blocks for cellular pathways. A large part of this information is embedded in a large amount of biological literature. For this reason, some systems have been developed to extract protein-protein interactions automatically. They can be classified into the following groups:

On-line protein-protein extraction systems [1, 12, 21]. Similar to information retrieval systems, these systems require users to input keywords—one or several protein names, and then retrieve related abstracts or papers from PubMed or other free text sources according to the keywords. After that, the systems will process the retrieved documents with the aid of preset internal patterns, extract the interactions and present them to the user. On-line extraction systems are quite useful for users searching for particular targets. However, the pattern set used is small and pre-defined which limits their performance.

Systems targeting a particular sub-domain [13, 7, 17]. These systems normally have pre-conditions, such as a well-defined protein name dictionary or pre-retrieved data set with the protein names. These systems are rather restricted for practical applications, even with claimed high precision and recall rates.

Systems migrated from general IE systems [7, 20]. The templates and rules for these systems are manually customized to fit biological sub-domains, which requires substantial efforts even for moderate performance.

General interaction extraction systems [16, 10, 4]. Some systems emerging in recent years target general interaction extraction in biological literature. These systems normally have well-defined patterns and strong language processing ability. The initial results are promising: Friedman *et al* [4] claim to have 96% of precision and 63% of recall for a paper they tested.

A lot of effort has also been put into individual components of the extraction procedure, such as document retrieving and clustering [8, 11], protein interaction database construction [18], protein name recognition [5, 19, 3, 6], tagging [2], parsing [15, 10], extraction rule development [9, 14] and pathway visualization [12, 21].

Most of these systems, except GeneWays [4], are using fixed patterns related with only a small number of verbs (“interact”, “activate”, “bind”, “inhibit”, “associate”, etc.). GeneWays manually collects many more patterns with a larger verb set and classifies them into 14 semantic classes. Since a few patterns cannot cover all scenarios, more effort needs to be made on pattern definition to increase the recall rate.

Statistical methods such as Hidden Markov Models and the Maximum Entropy algorithm, which yield good results in general information extraction, are seldom used in biological extraction systems except the protein name identification [3], because of the difficulty in obtaining large training sets.

None of the current pathway extraction systems is widely accepted by biologists. We believe the reason for this low acceptance is that the current pathway extraction systems are trying to replace the biology interpretation in the pathway extraction process, which is not practical for the following reasons:

- The credibility of the results are not properly reflected, which decreases further utilization;
- Some crucial information is not extracted, such as conditions of interactions;
- The varied and constantly changing demands from biologists.

To fill the gap, we introduce the user-involved extraction in the PathwayFinder system with the belief that the extraction process is a collaborative task involving both the biologists and the extraction system.

3. Multi-Agent Architecture

User-involved extraction demands great flexibility of the extraction procedure. To satisfy the demand, we introduce an innovative method in PathwayFinder system—a multi-agent architecture, which is composed of five types of components: manager agent (MA), extractor agent (EA), interface agent (IA), toolbox and databases, as shown in Figure 1.

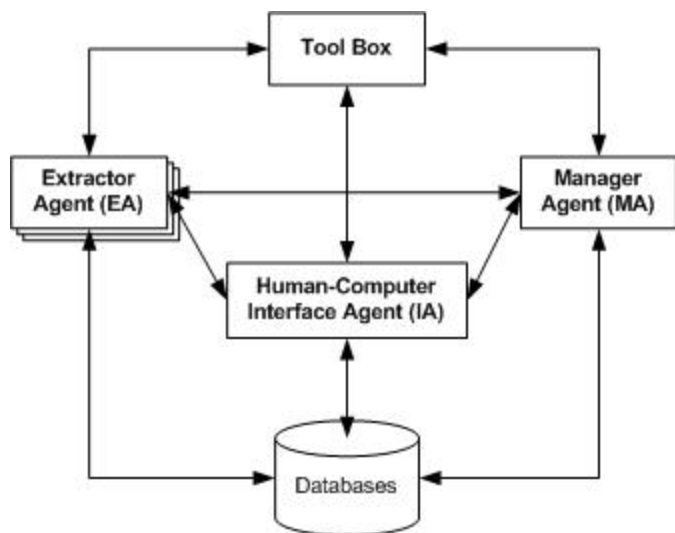


Figure 1. PathwayFinder system architecture

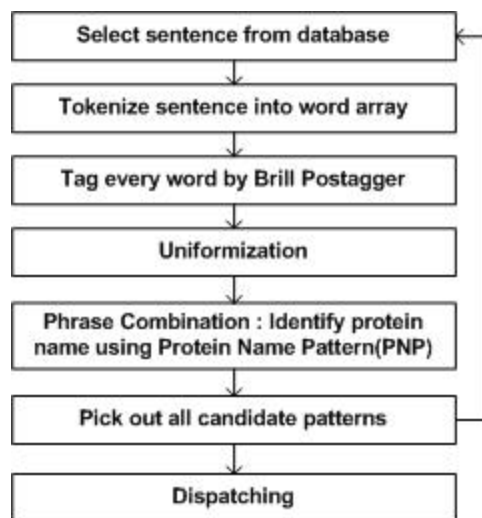


Figure 2. Preprocessing procedure

The extraction procedure is broken down into two parts: preprocessing and extraction, which are handled by MA and EA respectively. The current PathwayFinder has only one MA, since preprocessing is similar in pathway extraction tasks. More MAs could be deployed for more general extraction tasks. According to the sub-domain definition in MA, multiple EAs can be created to handle different targets. IA is introduced as a mediator between users and other components to enhance users’ role in the extraction procedure. Through IA, users are essentially integrated into the extraction procedure with rich user-involved features. For example, user can create interaction patterns through IA (see details in Section 4). Another example is that a confirmation of an extracted pathway from a

user will be directed to the proper EA, which increases the rank of the pattern used.

3.1 Manager Agent (MA)

MA provides two functions. One is to preprocess sentences, and the other is to organize EAs according to the sub-domain definitions. Figure 2 shows the procedure of preprocessing, which are divided into six steps. Firstly, every sentence is tokenized into a word array. The result is inputted into the Brill’s POS tagger [2], which determines the part-of-speech tag for each word. Next, the results from the first step are fed into the “Uniformization” step to generate the uniform word and uniform tag for each word. In the phrase combination step, Protein Name Pattern is used to identify proteins in the sentence. Then, all the candidate patterns of the sentence are picked out by looking up each word in the dictionary. If the sentence contains one or zero proteins or matches zero patterns, the preprocessor skips to the next sentence. Otherwise, with the identified protein names, the sentence is directed to the proper EA according to domain definition for further extraction. The MA also accepts requests from IAs, and directs them to the proper EAs.

3.2 Extractor Agent (EA)

An EA conducts the extraction task for sentences in its own domain, which is defined in MA, and maintains the related patterns and statistical data. The implemented pattern-matching algorithm is similar to the Maximum Entropy Approach [22]. We have defined three types of features. The first is adjacent feature fa , which reflects the relations between uniformed tags. For example, tag pair “DT” and “NN” is an adjacent feature “DT-NN”, which reflects the probability of those two tags being in the same phrase. The second is grammar feature fg , which reflects the relation between a lower level tag and its parent tag in the grammar tree of a sentence. For example, “NP-NP” is a grammar feature which reflects the probability of the parent “NP” tracing down to the lower level “NP”. These features are pre-calculated from sample data. The third is pattern feature fp , which describes meaningful relations between elements in a sentence, such as “PROTEIN-activate-PROTEIN”. Instead of the entropy, we calculate maximum feature scores of feature sets applicable to a sentence. The first calculated score is Adjacent-Pattern Feature score $A(p)$, which is the maximum combination of all adjacent features and pattern features fit for the sentence. $A(p)$ makes use of the grammar information contained in the user-defined patterns, and measures the closeness of the sub-sentences and the patterns. It is introduced to compensate the deficiency of parsing caused by the complexity of biology literatures. Another score is Grammar-Pattern Feature score $G(p)$, which is the maximum combination of all grammar features and pattern features fitting the sentence. It measures the matching between parsing results and the patterns. These scores are calculated as follows:

$$A(p) = \max(\prod_{f \in S_i} (fa, fp) | S_i \subset S)$$
$$G(p) = \max(\prod_{f \in S_j} (fg, fp) | S_j \subset S)$$

Where p is the pattern applied, f is the applicable feature, S is the set of all features in the sentence, and S_i and S_j are the feature subsets. For each action word, the qualified match with the highest combined score of $A(p)$ and $G(p)$ is selected as the extracted interaction.

Special Cases Processing

Besides the general grammar and pattern rules, there are some other rules existing in literatures, which will affect the accuracy of extraction. Figure 3 shows some of the special cases handled by PathwayFinder.

Protein-Interaction interaction. In some cases, such as “..the protein HHR6B inhibits the interaction of cdc34 and ICP0..”, the protein HHR6B does not interact with another protein, but works on an

interaction. This kind of interaction is extracted with a multi-scan technique. Sentences containing at least one protein-protein interaction are tagged, and new nodes with the “INTERACTION” tag substitute the extracted interactions. If this modified sentence matches any pathway patterns containing an “INTERACTION” tag, a nested interaction may be extracted. In this case, the “subject” or “object” of the extracted interaction is a link leading to the nested interaction. An example is given in Figure 3E.

A) slash in an interaction

Sentence ID: 75
Sentence: "...it is possible that the RPN-11/F55A11.3 interaction is involved in ..."
Action word: "interaction"
Associated pattern(s): "ROTEIN/PROTEIN interaction", "interaction between/of PROTEIN and/with PROTEIN"
Applied pattern: "ROTEIN/PROTEIN interaction"
Extracted Interaction: "RPN-11 -> interaction -> F55A11.3"

B) slash as "and"

Sentence ID: 127
Sentence: "To examine whether direct protein-protein interactions between CCTs and the COP/DET/FUS proteins are ..."
Action word: "interaction"
Associated pattern(s): "ROTEIN/PROTEIN interaction", "interaction between/of PROTEIN and/with PROTEIN"
Applied pattern: "interaction between PROTEIN and PROTEIN"
Extracted Interaction: "CCT -> interaction -> COP", "CCT -> interaction -> DET", "CCT -> interaction -> FUS"

C) "and" in parallel structure

Sentence ID: 1623
Sentence: "The fUBR11-1367 and UBR11-1140f, which, respectively, contained and lacked the RAD6-binding site (Fig. 2A), bound to GST-CUP9 with similar affinities (Fig. 5B, lanes 4-6 vs. lanes 1-3)."
Action word: "bind"
Associated pattern(s): "PROTEIN bind to/with/on PROTEIN", "PROTEIN bind PROTEIN"
Applied pattern: "PROTEIN bind to PROTEIN"
Extracted Interaction: "fUBR11-1357 -> bind -> GST-CUP9", "UBR11-1140f -> bind -> GST-CUP9"

D) "and" in an interaction, and "or" in a parallel structure

Sentence ID: 258
Sentence: "To test for the potential direct interactions between COP10 and the COP9 signalosome or COP1, a yeast two-hybrid assay was performed."
Action word: "interaction"
Associated pattern(s): "ROTEIN/PROTEIN interaction", "interaction between/of PROTEIN and/with PROTEIN"
Applied pattern: "interaction between PROTEIN and PROTEIN"
Extracted Interaction: "COP10 -> interaction -> COP9", "COP10 -> interaction -> COP1"

E) "PROTEIN-INTERACTION" interaction

Sentence ID: 2184
Sentence: "Covalent attachment of SUMO-1 to Mdm2 requires the activation of a heterodimeric Aos1-Uba2 enzyme (ubiquitin-activating enzyme (E1)) followed by ..."
Action word: "attachment", "require"
Associated pattern(s): "attachment of ROTEIN to PROTEIN", "INTERACTION require PROTEIN"
Applied pattern: "attachment of ROTEIN to PROTEIN", "INTERACTION require PROTEIN"
Extracted Interaction: "SUMO-1 -> attachment -> Mdm2", "{SUMO-1 -> attachment -> Mdm2} -> require -> E1"

Figure 3. Extraction examples. For each sentence, the patterns loaded for extraction are called associated patterns. Among them, the patterns used to extract the interactions are called applied patterns.

Special proteins. Protein state and activity modification, such as mutants and protein inhibitors, are also identifiable. Special identifying processes are added during preprocessing. For example, “MEK inhibitor” is identified as “[x inhibit MEK]” instead of “MEK”, “COP1 mutants” is identified as “COP1 mutant” protein, instead of “COP1” protein.

Negative words. PathwayFinder identifies negative words within the scope of the interaction by searching through an editable negative word list, and removes those results. For example, in the following sentence, no interaction is extracted because of the negative words “neither” and “nor”.

“Neither rce1-null nor yor291w-null mutations affected PIO or the phenotype of spf1- or ste24-null mutants.”

Slash (“/”). Slashes may have different meanings in different sentences. It can be a simplified form of an interaction, as shown in Figure 3A, or an “and/or” relation, as shown in Figure 3B. These two cases are successfully distinguished by patterns, since the former case has the subject and object in one word, which contains a “/”.

“And” and “Or”. “And” and “Or” may indicate an interaction or a parallel relation between proteins. To distinguish them, we create a relation set for the proteins connected by each “and” or “or” in a sentence. If both the subject and the object of the extracted interaction are in the same relation set, then the relation set is ignored. Otherwise, the “and” or “or” indicates a parallel relation, and expands the related subject or object. Examples are given in Figure 3C and Figure 3D.

3.3 Human-Computer Interface Agent (IA)

IA provides rich user-involved features to absorb users’ domain knowledge into the extraction process, and provide the required results to users. As an important function of PathwayFinder system, it is described in detail in Section 4.

3.4 Toolbox

Some utilities are relatively independent of the extraction process. We wrapped them into separate components, and put them into a toolbox. Currently, there are three tools integrated in the PathwayFinder system:

Paper Crawler, which grabs abstracts and papers from PubMed or other sources and stores them in the paper database.

Language Processing Server (PFSCTools Server), which provides tagging and parsing services; PFSCTools Server integrates Brill’s POS tagger [2] and Link Grammar parser [23] into one program. This server supports TCP/IP connections like a web server. Unlike a web server, which responds to the GET or POST services of HTTP, the PFSCTools server utilizes the RPC (Remote Procedure Call) service. The main application can connect to the PFSCTools server through a network for POS tagging or parsing services.

Benchmark Comparer, which compares the extracted results with the standard results, both of which are in XML format.

The components in the toolbox are not agents, since they do not have learning capability. To speed up the processing, we can have several instances for each tool. For example, we can have several Language Processing Servers running on different computers to deal with the requests from different agents. Furthermore, these tools are independent, and can be replaced later by substitutes with better performance.

3.5 Databases

PathwayFinder uses several databases. Collectively they store papers in text format, protein names, patterns, extracted pathways, and agent specific knowledge that includes special cases of words, proteins, and the usage data collected.

4. IA: User-Involved Design

IA acts as a mediator between users and agents. It provides two types of functions: customization, which customizes MA and EAs to absorb users' domain knowledge and improves the extraction performance; and interactive presentation, which provides user-friendly interfaces for users to access the extracted pathway information.

4.1 Customization

The variation of users' requirements is not limited to the query they initiate, but includes the domain they define, the patterns they select, and the results they are interested in. For example, when biologists query on "CCT" in PathwayFinder, they may not be looking for a company or project named as CCT, nor the definition of protein CCT, but the interactions between protein CCT and other proteins. The customization process not only helps users to tell the system what they want, but also integrates their domain knowledge into the system to improve the extraction capability.

Preprocessing Customization In the PathwayFinder system, we introduce four new tags in tagging process: "PNI" (Protein Name Identification), "PNK" (Protein Name Keyword), "PROTEIN" and "INTERACTION", to indicate simple or complex protein names. If a word has a "PNI" tag, it means that it is a candidate protein name; if a word has a "PNK" tag, then it is evidence that its neighboring words are protein names. After the phrase combination process, the recognized protein names are labeled as "PROTEIN", and some complex structures equivalent to proteins, such as "MEK inhibitor", are labeled as "INTERACTION". Users can browse through the "PNI" list to identify the true or false protein names, and edit the "PNK" list to indicate the new or special cases to improve the protein identification process. This knowledge is under constant development, and is not pre-fixed.

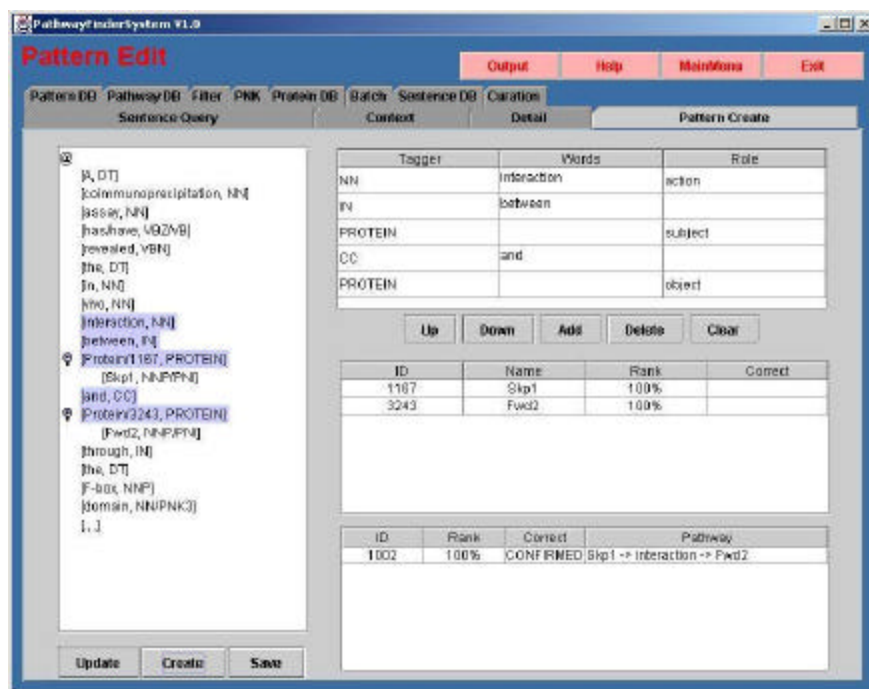


Figure 4. Demonstration of sample pattern creation. In the left window, key elements of a pattern are selected from the sentence by the user. The selected elements are transformed into a sample pattern in upper-right window. And the pathways extracted from the sentence are listed at the lower-right window.

Pattern Customization A pattern contains all the key elements that can identify a protein interaction. In most other pattern-based information extraction systems, patterns are fixed, and can only be added or modified by the system developer. This is suitable for some special study cases, but can hardly adapt to users' frequently changing requirements.

The PathwayFinder system provides an open and flexible mode to create patterns combining users' domain knowledge. Users do not need to know a lot about Natural Language Processing. They just indicate which words represent a protein-protein interaction in one sentence, and pick them out by a simple point-and-click method, as shown in Figure 4. The pattern created by a user is called a "sample pattern", which represents an exact protein-protein interaction. A sample pattern is not directly used in the subsequent pattern matching, since many sample patterns have the same grammar constituents and there is considerable redundancy. We use generalized patterns to eliminate this redundancy. One general pattern is the summation of many sample patterns that have the same grammar constituents. When a new sample pattern is added into the pattern database, it is merged into an old general pattern if they have the same grammar constituents, as shown in Figure 5; or a new general pattern is created if there is not any suitable general pattern to match the sample pattern.

A. The existing general pattern

Tagger	Words	Role
NN	interaction	action
IN	between	
PROTEIN		subject
CC	and	
PROTEIN		object

B. The user-created sample pattern

Tagger	Words	Role
NN	conjugation	action
IN	of	
PROTEIN		subject
CC	and	
PROTEIN		object

C. The updated general pattern

Tagger	Words	Role
NN	interaction conjugation	action
IN	between of	
PROTEIN		subject
CC	and	
PROTEIN		object

Figure 5. Demonstration of pattern generation. The user-created pattern (B) is merged into the existing general pattern (A) with the same grammar constituents, which results in a new general pattern (C).

To prevent the adverse effects of user-created erroneous patterns, the following usage data are collected to evaluate each pattern:

- **Loading factor**—total number of times that the pattern is loaded for matching;
- **Using factor**—total number of times that interactions are extracted by the pattern;
- **User confirmation factor**—total number of times that the extracted interactions by this pattern are confirmed by users; and
- **User denial factor**—total number of times that the extracted interactions by this pattern are denied by users.

A high loading factor with a low using factor means that the pattern is rarely used, which implies that the pattern may be too short or too general to be effective. And a high user denial factor reflects that the pattern is likely to be an erroneous one. The combination of these four factors gives an objective

The color in the diagram presents the credibility: the color of rectangles presents the protein credibility, and the color of lines presents the interaction credibility. Users can constrain the credibility interactively by providing the thresholds with sliding bars, which will customize the diagram dynamically.

5. Results

Comprehensive test of the system is difficult, since there is no standard sample data set for pathway extraction. For benchmark comparison purpose, we collect a small data set with 12 papers and 116 abstracts, which are manually extracted by a biologist for protein names and interactions. Among them, 10 abstracts are used for testing, and others are used as training materials. The data set can be downloaded at: http://monod.uwaterloo.ca/~dyao/PathwayFinder/pf_index.htm for any future comparison in this community.

The PathwayFinder system started from the state that no pattern or protein names were preloaded. With a short tutorial, a domain user began to use the PathwayFinder system. He started from pattern creation, either from some sentences or his own knowledge, and curation for the protein names identified by the system. In a matter of hours, there were 167 user-defined patterns created, which were automatically summarized into 60 general patterns. Those patterns were ranked according to the times they were used and the user's feedbacks. Some patterns, such as "transfer PROTEIN from PROTEIN" (rank: 0.011), had low ranking because of the user's rejection on the related pathways. It meant these patterns were not proper for this EA. The system was able to extract 84.3% of all manually extracted interactions in the training process, and 64.7% in the testing process. The overall recall rate should be further improved when more papers are fed into the system, because many interactions are stated in different papers repetitively. With the curation functions, the inaccurate results were corrected.

We have applied the PathwayFinder system to the ubiquitin cascade pathway. With various key words search from PubMed, over 8,000 abstracts that contains over 65,000 sentences have been processed. Over 1,800 relationships have been recovered after the curation. Ubiquitin and ubiquitin like molecules are small proteins that become conjugated to a substrate as a way to regulate a variety of cellular processes, such as protein degradation, localization, activity modification and signal transduction. Normally the cascade involves three enzymes: ubiquitin (like) activating enzyme (E1), ubiquitin (like) conjugating enzyme (E2) and ubiquitin (like) ligation enzyme (E3). We have cloned a novel ubiquitin conjugating enzyme E2.12. To find the implications of its function, we searched our PathwayFinder system. In this system we combined the literature knowledge and our internal protein-protein interaction data. A death-associated protein 6 (Daxx) has been found to regulate apoptosis. The PathwayFinder indicates that Daxx interacts with an ubiquitin-like molecule SUMO1 and SUMO1's E2, Ubc9. From our internal yeast-two-hybrid data, we found that Daxx associates with SUMO3, E2.12 and TRAF4. Therefore, it is likely that E2.12 is an E2 for SUMO1 or SUMO3, RingFinger domain containing protein TRAF4 acts as an E3 to transfer SUMO1 or SUMO3 to Daxx. Thus sumolation regulates Daxx's activity by modulating its degradation. The protein-protein interaction network implies that E2.12 is an E2 for SUMO ubiquitin-like molecules and it links to apoptosis pathway through Daxx. In agreement with the inferred function of E2, further in house experiments indicate that siRNA of E2.12 inhibits cell proliferation (data not shown).

6. Summary

In the PathwayFinder system, the extraction process is separated from the query interface, and is performed in the background. Since users access the extracted results directly, not only can queries be

processed much faster, but also further processing of the results such as curation is more easily accomplished.

The main contribution of the PathwayFinder is the significantly improved usability and extendibility provided by the user-involved extraction. We emphasize users' involvement in every aspect of our system. During preprocessing, users can specify the PNIs and PNKs to identify special forms of proteins; during the extraction, users can create, edit or delete patterns according to their domain knowledge. They can also confirm or deny the extracted results, which will not only increase the accuracy, but also affect pattern ranking in future extraction. Users can also track down the relationships between proteins on the diagram, which clearly states the credibility of the results, and jumps directly to the related text source for further reference. To support the user-involved extraction, we introduce an innovative multi-agent architecture. It provides not only the flexibility to support users' constantly changing requirements, but also the flexibility of the system itself, which makes it much more extendible.

Compared with fully automatic extraction systems with fixed targets, PathwayFinder can adapt to any targets specified by users, and the shifting of targets is handled automatically by generating new agents.

To handle the adverse effect of incorrect user-created patterns, the statistic data is applied to adjust the pattern ranking, which will diminish the influence of improper patterns effectively.

User-involved extraction also lowers the requirements of the language processing unit. Since the user-defined patterns often contain the substructure of the relevant sub-sentence, we introduce the adjacent pattern feature to combine this information with the language analysis result and compensate the deficiency of parsing.

The feedbacks of the system from domain users are positive. With rich user-involvement support, they can create their own patterns according to their domain knowledge without assistance from computer experts. The initially extracted results can act as an index to access the related context. And with proper curation, the results can be further used to find functional pathways, or other potential relations between proteins.

References

1. C. Blaschke, M. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: Protein-protein interactions. In Proceedings of the AAAI Conference on Intelligent Systems in Molecular Biology, pages 60–67, 1999.
2. Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.
3. N. Collier, C. No, and J. Tsujii. Extracting the names of genes and gene products with a hidden markov model. In Proc. COLING 2000, pages 201–207, 2000.
4. C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 suppl. 1:74–82, 2001.
5. K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In Pacific Symposium on Biocomputing 3, pages 705–716, 1998.
6. V. Hatzivassiloglou, P. A. Duboue, and A. Rzhetsky. Disambiguating proteins, genes, and rna in text: A machine learning approach. *Bioinformatics*, 17 no. 1:1–10, 2001.
7. K. Humphreys, G. Demetrios, and R. Gaizauskas. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In Proceedings of the Pacific Symposium on Biocomputing, pages 505–516, Hawaii, Jan 2000.

8. I. Iliopoulos, A. J. Enright, and C. A. Ouzounis. Textquest: Document clustering of medline abstracts for concept discovery in molecular biology, Oct 2001. <http://www.smi.stanford.edu/projects/helix/psb01/ili.pdf>.
9. Michael Krauthammer, Pauline Kra, Ivan Iossifov, Shawn M. Gomez, George Hripcsak, Vasileios Hatzivassiloglou, Carol Friedman, and Andrey Rzhetsky. Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics*, 18 suppl.1:249–257, 2002.
10. G. Leroy and H. Chen. Filling preposition-based templates to capture information from medical abstracts. In *Pacific Symposium on Biocomputing 7*, pages 350–361, 2002.
11. Edward M. Marcotte, Ioannis Xenarios, and David Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17:359–363, 2001.
12. S. Ng and M. Wong. Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Informatics*, 10:104–112, 1999.
13. Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. Automatic extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17 no.2:155–161, 2001.
14. T. Oyama, K. Kitano, K. Satou, and T. Ito. Extraction of knowledge on protein–protein interaction by association rule discovery. *Bioinformatics*, 18 no.5:705–714, 2002.
15. J.C. Park, H.S. Kim, and J.J. Kim. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar, Oct 2001. <http://citeseer.nj.nec.com/384291.html>.
16. D Proux, F Rechenmann, and L Julliard. A pragmatic information extraction strategy for gathering data on genetic interactions. In *Proc Int Conf Intell Syst Mol Biol 8*, pages 279–285, 2000.
17. T.C. Rindfleisch, L. Tanabe, J.N. Weinstein, and L. Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proceedings of the Pacific Symposium on Biocomputing, Hawaii, Jan 2000*.
18. C. Sanchez, C. Lachaize, F. Janody, B. Bellon, L. Roder, J. Euzenat, F. Rechenmann, and B. Jacq. Grasping at molecular interactions and genetic networks in drosophila melanogaster using flynets, an internet database. *Nucleic Acids Res*, 27 no.1:89–94, 1999.
19. Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18 no.8:1124–1132, 2002.
20. J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the Pacific Symposium on biocomputing*, pages 541–551, Hawaii, Jan 2000.
21. L. Wong. Pies, a protein interaction extraction system, Oct 2001. <http://www.smi.stanford.edu/projects/helix/psb01/wong.pdf>.
22. Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.
23. Daniel Sleator and Davy Temperley. Parsing English with a Link Grammar, Carnegie Mellon University technical report, CMU-CS-91-196, Oct. 1991