Bioinformatics Advance Access published January 22, 2004

Extracting human protein interactions from MEDLINE using a full-sentence parser.

Nikolai Daraselia[*]*, Anton Yuryev, Sergei Egorov, Svetalana Novichkova, Alexander Nikitin, and Ilya Mazo*

Ariadne Genomics, Inc., 9700 Great Seneca Hwy, Rockville, MD 20850

Abstract

Motivation. The living cell is a complex machine, that depends on the proper functioning of its numerous parts, including proteins. Understanding protein functions and how they modify and regulate each other is the next great challenge for life-sciences researchers. The collective knowledge about protein functions and pathways is scattered throughout numerous publications in scientific journals. Bringing the relevant information together becomes a bottleneck in a research and discovery process. The volume of such information grows exponentially, which renders manual curation impractical. As a viable alternative, automated literature processing tools could be employed to extract and organize biological data into a knowledge base, making it amenable to computational analysis and data mining.

Results. We present MedScan, a completely automated NLP-based information extraction system. We have used MedScan to extract 2976 interactions between human proteins from MEDLINE abstracts dated after 1988. The precision of the extracted information was found to be 91%. Comparison with the existing protein interaction databases BIND and DIP revealed that 96% of extracted information is novel. The recall rate of MedScan was found to be 21%. Additional experiments with MedScan suggest that MEDLINE is a unique source of diverse protein function information, which can be extracted in a completely automated way with a reasonably high precision. Further directions of the MedScan technology improvement are discussed.

Availability: MedScan is available for commercial licensing from Ariadne Genomics, Inc.

Contact: nikolai@ariadnegenomics.com

Running head: "Extraction of human protein-protein interactions from MEDLINE"

Introduction.

Information about protein function and cellular pathways is central to the system-level understanding of a living organism. The problems that could be tackled, should such data be easily available, include: quantitative simulation of complex cellular processes, identification of key elements in signal transduction, cross-talk between different pathways, mechanisms of gene co-regulation, and many others.

There is a number of available databases covering different aspects of protein function, such as protein–protein interaction (DIP(*URL1)*, BIND(*URL2*)), regulatory gene networks (GeNet(*URL3*), TRANSPATH(*URL4*)) or signaling pathways (CSNDB(*URL5*), SPAD(*URL6*)). However, being dependent on human experts, they rarely store more than few thousands of the best-known protein relationships and do not contain the most recently discovered facts and experimental details. There is an urgent need for an automatic system capable of accurate extracting protein function information from literature. Few systems aimed at solving this task have been recently proposed. They range in approaches from simple statistical methods to advanced natural language processing (NLP) techniques.

The simplest way to extract protein relations from the literature is to detect the co-occurrence of protein names in a text (Stephens *et al.*, 2001). However, by its nature, the name co-occurrence detection yields very little or no information about the type of a described relation and therefore the co-occurrence data may be misleading. More sophisticated information extraction approaches rely on the matching of pre-specified templates (patterns) or rules (such as precedence/following rules of specific words). The underlying assumption is that sentences conforming exactly to a pattern or a rule express the predefined relationship(s) between the sentence entities. In some cases, these rules and patterns are augmented with additional restrictions based on syntactic categories and word forms in order to achieve better matching precision. The pattern-based systems have been applied to extract protein-protein interaction (Sekimizu *et al.*, 1998; Blaschke *et al.*, 1999; Ono *et al.*, 2001) and pathway information (See-Kiong and Wong, 1999).

More advanced systems utilizing shallow parsing techniques have been described to extract protein interactions (Thomas *et al.*, 2000), enzyme reactions and protein structure information (Humphreys *et al.*, 2000), or functional relations between proteins (Park *et al.*, 2001). Unlike word-based pattern matchers, shallow parsers perform

[*] To whom correspondence should be addressed

partial decomposition of a sentence structure. They identify certain phrasal components and extract local dependencies between them without reconstructing the structure of an entire sentence. The precision and recall rates reported for shallow parsing approaches are 50-80% and 30-70%, respectively. Interestingly, most of the described systems are designed to extract only one specific aspect of protein function information.

The most promising candidates for a practical information extraction system are ones based on full-sentence parsing as they deal with the structure of an entire sentence and therefore are potentially more accurate. An example of such a system is GENIES (Friedman *et al*., 2001), which utilizes a parser and a semantic grammar consisting of a large set of nested semantic patterns (incorporating some syntactic knowledge), reflecting most frequently used sentence structures. Unlike other systems, GENIES is capable of extracting a wide variety of different relations between biological molecules as well as nested chains of relations. However, the downside of the semantic grammar-based systems like GENIES is that they may require complete redesign of the grammar in order to be tuned to a different domain

We believe that a key to efficient information extraction is in a modular architecture that separates natural language processing and information extraction into different modules. The NLP module deals with the domain-independent sentence structure decomposition, while the information extraction module can be reconfigured towards different tasks. Such approach was proposed by Yakushii (Yakushiji, *et al.*, 2001) who described a two-step information extraction system where the first step is construction of the sentence argument structure using general-purpose domain independent parsers (XHPSG and EngCG) and the second one is domain-specific frame extraction (not implemented at the time of publication.). The paper presented the results of the preliminary system testing, which justified the approach, however, to the best of our knowledge, no further results have been published since. We have previously reported a context-free biomedical domain-oriented NLP engine that parses sentences from MEDLINE abstracts into a set of alternative semantic trees (Novichkova *et al.*, 2003). In this paper, we present the MedScan - a complete information extraction system, which interprets these semantic structures using a pathway-oriented ontology and extracts protein function information. We have used MedScan to extract 2976 interactions between human proteins from 3.5 million sentences from MEDLINE abstracts dated after 1988. The processing took less than 10 hours and the precision of extracted information was 91%. Comparison with existing protein interaction databases BIND and DIP revealed that 96% of extracted information is. The recall rate of MedScan for extraction of protein interactions was found to be 21%.

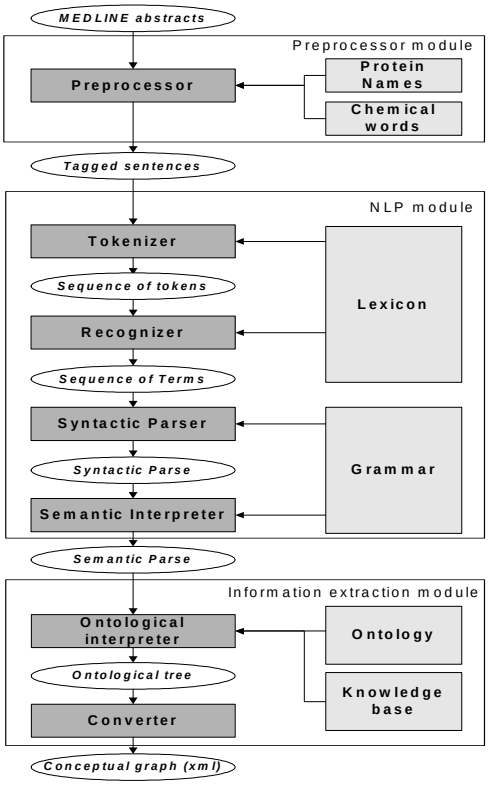

Figure 1. The components and processing steps of the MedScan system.

System Overview

The MedScan is a three-tier information extraction system based on a full sentence parsing approach. Conceptually, it contains three modules: i) preprocessor aimed to identify and tag various biomedical domain-specific concepts; ii) NLP engine constructing the set of alternative semantic sentence structures; and iii) information extraction module acting as a domain-specific filter for these structures and extracting information in a form of conceptual graph. An overview of MedScan architecture is presented in Figure 1. Below we provide a brief description of the preprocessing and NLP components, which are described in details elsewhere (Novichkova *et al.*, 2003), and concentrate on principles and algorithms invoked by the information extraction module.

The preprocessor reads the XML-based format of a MEDLINE record and splits the MEDLINE abstract into individual sentences. Next, it utilizes a provided protein name dictionary to identify and tag protein names and to select the sentences containing at least one protein name. It also identifies and tags notation of chemical substances using a pre-compiled list of chemical name "root" words. Our approach for a protein name identification is based on the application of a specialized tokenizer designed to ignore many variations of protein name spelling followed by a simple and efficient subsequence search algorithm applied to token sequences. Preprocessor does not attempt to decipher abbreviations and relies on alternative protein names being provided explicitly and is designed to ignore colliding gene names. In the final output, identified protein names are labeled with an identifier provided with the dictionary (we use HUGO ids). All tagged names are considered to be notations of proteins unless explicitly described as genes by specific terms ("gene" or "mRNA"). We use this

simplified approach because all potential protein relations are subsequently passed through the ontological filter. Because we are interested in protein-function information extraction, the preprocessor outputs only those sentences that contain at least one identified protein.

NLP component of a MedScan is a biomedical domain-oriented NLP engine that processes sentences from MEDLINE abstracts and produces a set of semantic structures representing the meaning of each sentence (Novichkova *et al.*, 2003). It is based on a context-free grammar and a lexicon developed specifically for MEDLINE. Processing is done in two steps. First, a syntactic parser constructs a set of alternative syntactic structures of an input sentence. Because syntactic knowledge is ambiguous in its nature, a single sentence usually yields many (sometimes up to more than 100,000) alternative parses. Next, semantic processor transforms each of them into a corresponding semantic tree. This conversion is based entirely on syntactic knowledge – namely, information about the predicate structure (number and order of arguments), and information about "prepositional patterns" associated with some of the lexemes (mainly verbs and verb nominalizations). The constructed semantic trees consist of nodes each representing a single lexeme (see examples on the figures 3A and 3B). The nodes are connected by labeled vertexes, which can be split into two major categories: *thematic role* vertexes and *attributive* vertexes. Thematic role vertexes reflect essential relations and are constructed from verb and noun compliments. Attributive vertexes correspond to more subtle qualifying or aggregative relations in a sentence usually expressed by prepositional phrases, coordinating and subordinating clauses, and conjunctional phrases. The difference in type is encoded by the vertex labels, which capture both the nature of the semantic relation between lexemes and information about its syntactic origin. The vertex labels play a crucial role in a subsequent process of ontological interpretation.

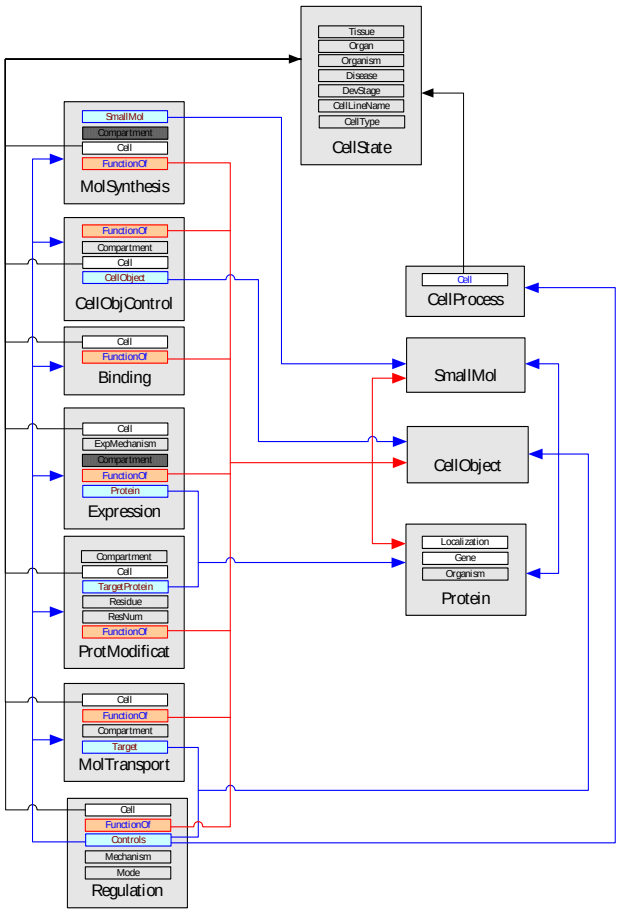

Figure 2. An overview of MedScan protein function ontology. Frames are shown as large rectangles. Slots are displayed as smaller rectangles inside the frames and are connected by arrows (each representing one ontological link) to target frames. Only the most important frames and links are shown. Frames from the Entity group are on the right, while frames from the Control group are on the left.

Ontology

The ontological interpreter utilizes an ontology and a knowledge base to perform evaluation of each semantic tree generated by the NLP component and to convert the valid ones into ontological representation. Ontology serves as a knowledge model for a particular domain of science.

Development of ontologies for molecular biology is a relatively new field yet a few well-designed ontologies exist. We have carefully considered them and concluded that no single one is readily suited for our purposes. Some ontologies focus on a very narrow aspects of protein functions such as biochemical reactions (*EcoCyc*, Karp *et al.*, 1999; *EMPathIE*, Humphreys *et al.*, 2000), 3D structure (*PASTA*, Humpreys *et al.*, 2000), ribosomal RNA function (*RiboWeb* Chen *et al.*, 1997). Others (*The Molecular Biology Ontology*, Schultze-Kremer, 1997) are, on contrary, too general for protein function representation. *Gene Ontology* (Ashburner *et al.*, 2000) is a well-established ontology developed specifically for annotating protein functions, but it completely lacks methods of representation of binary protein relations and focuses instead on the creation of lists of cellular processes, cellular components and "molecular functions". However, for the construction of our ontology it was very useful to include concepts from the Gene Ontology. The ontology most closely related to ours is the one proposed by Rzhetsky (Rzhetsky *et al.*, 2000). Our ontology is somewhat simpler and is specifically tailored towards the representation of protein relations as directed binary links with attributes.

Ontology is defined as the collection of concepts representing domain-specific entities, the set of relationships between the concepts, and the range of admissible values for each concept. We have adopted a traditional frame-based representation for ontology. Each concept is described by a *frame*, which has a unique name and contains a set of *slots*.

A frame can be instantiated with an admissible value. For example, admissible values for the frame *Protein* include all available protein identifiers; frame *RegulationType* has three admissible values: positive', 'negative' and 'unknown'.

Frame slots describe logical relations between frames. Generally speaking, each frame slot can be filled with several frames. We use triplets {*Frame1*, **Slot1**, *Frame2*}, called ontological links, to uniquely identify a single logical relationship between two frames. *Frame1* in such a triplet is called a parental frame; *Frame2* - child frame. Ontological links are directional, connecting a slot of the parental frame with a single child frame. During the process of ontological interpretation, ontological links constrain which ontological frames can be slot arguments of other frames.

The ontology we have developed is designed to reflect multiple aspects of protein functions, including protein enzymatic activity, cellular localization, protein-protein interaction and organization in complexes, gene expression, regulation of various cellular processes and protein organization into cellular pathways. It was constructed after analyzing about 2,000 MEDLINE abstracts describing various functional properties of different classes of proteins, including receptors, enzymes, transcription factors, channels, structural proteins, and transporters. The overall structure and most important frames and links between them are illustrated in Figure 2.

Almost all ontological frames are organized into two groups: **Entity** group that includes frames for *Protein*, *CellObject*, *CellProcess*, and *SmallMol*; **Control** group that includes frames describing the functions of proteins and compounds. The concept of Control is central to our ontology. The function of an Entity (a protein or a compound) is always represented as a set of Controls connecting it to another Entity or another Control. The *Regulation* frame is the most general frame in this group and represents an underspecified functional connection ("regulate", "activate", "inhibit") when exact details are not present. Other frames from this group represent specialized types of regulation events, such as various types of protein post-translational modifications (*ProtModification*), regulation of cellular molecular transport (*MolTransport*), direct interactions between molecules and between molecules and cellular components (*Binding*), regulation of protein expression (*Expression*), regulation of small molecule synthesis (*MolSynthesis*), and regulation of cellular component biogenesis (*CellObjControl*). A Control itself can be the subject of another Control. In this way we can describe functional interference between multiple proteins. For instance, the interaction of two proteins or a biochemical reaction can be influenced by a drug (i.e., "drug A inhibits phosphorylation of protein B by protein C").

Knowledge Base

MedScan utilizes the ontology as a filter to select correct semantic sentence structures and convert them into *frame trees*. A frame tree is a structure used to describe the ontological meaning of a sentence. It consists of ontological frame instances: each initialized with an admissible value and with some of the slots filled with permitted argument frames. Frame instances in the frame tree are connected by ontological links in such a manner that each slot has a single outcoming link (from a set defined in ontology), and each frame in a tree has only one incoming link.

The frame tree is build incrementally by combining the meaning (or *sense*) of individual lexemes stored in the knowledge base. Our knowledge base has been constructed by manually assigning senses to about 6000 words including biological terms and various relations between them. Only lexemes, which belong to the domain of interest are parts of the knowledge base and bear ontological meaning.

We differentiate between two types of lexemes: *template* lexemes and *connecting* lexemes and use different mechanisms to capture their meaning in the knowledge base. Template lexemes correspond to the concepts described by the frames in the ontology. Connecting lexemes usually express relations between entities and are best described by ontological links. In principle, new frames can be added to the ontology to cover these lexemes as well. We decided against that, in part, to limit the size of ontology and keep it manageable, but also to have a consistent data model.

The senses of *template* lexemes are represented by miniature frame trees. The "root" of a frame tree is called the *head frame*. For example, the sense for the lexeme "inhibit" (Figure 3A, top) has a head frame *Regulation.* The head frame slot **Mode** is filled with the frame *RegulationType* initialized by a value "negative". In a simplest case, a sense of a template lexeme is composed of a single frame initialized with one of its admissible values. Some slots of the head frame may be labeled with *thematic roles*. The set of these roles is identical to one utilized for the labeling of semantic vertexes. The labels serve for mapping of semantic vertexes to frame slots during the ontological interpretation. For example, in the sense "inhibit" the slot **FunctionOf** is labeled "agent", **Regulates** – "patient".

Senses of *connecting* lexemes are defined through a set of ontological links. For example, a lexeme "catalyze" is defined as a single ontological link {*MolSynthesis*, **FunctionOf** -> *Protein*}. Senses of less selective lexemes may contain many (or even all) ontological links; for example a lexeme "act" is defined as a set of 20 links connecting **FunctionOf** slots of different frames from the Control group to the appropriate frames from the Entity group (Figure 3B, top). When describing a connecting lexeme, thematic role labels are assigned to both frames connected by each ontological link, for example in sense "catalyze" frame *Protein* is labeled "agent", *MolSynthesis* – "patient'. These roles are used by the interpretation algorithm to match ontological links to semantic vertexes.

Ontological Interpretation

Ontological interpretation is a semantic tree-driven process performed recursively in a top-down manner starting from the top of a semantic tree. The algorithm traverses a semantic tree and incrementally builds a frame tree using senses of template lexemes as building blocks and connecting them by ontological links if they are permitted by

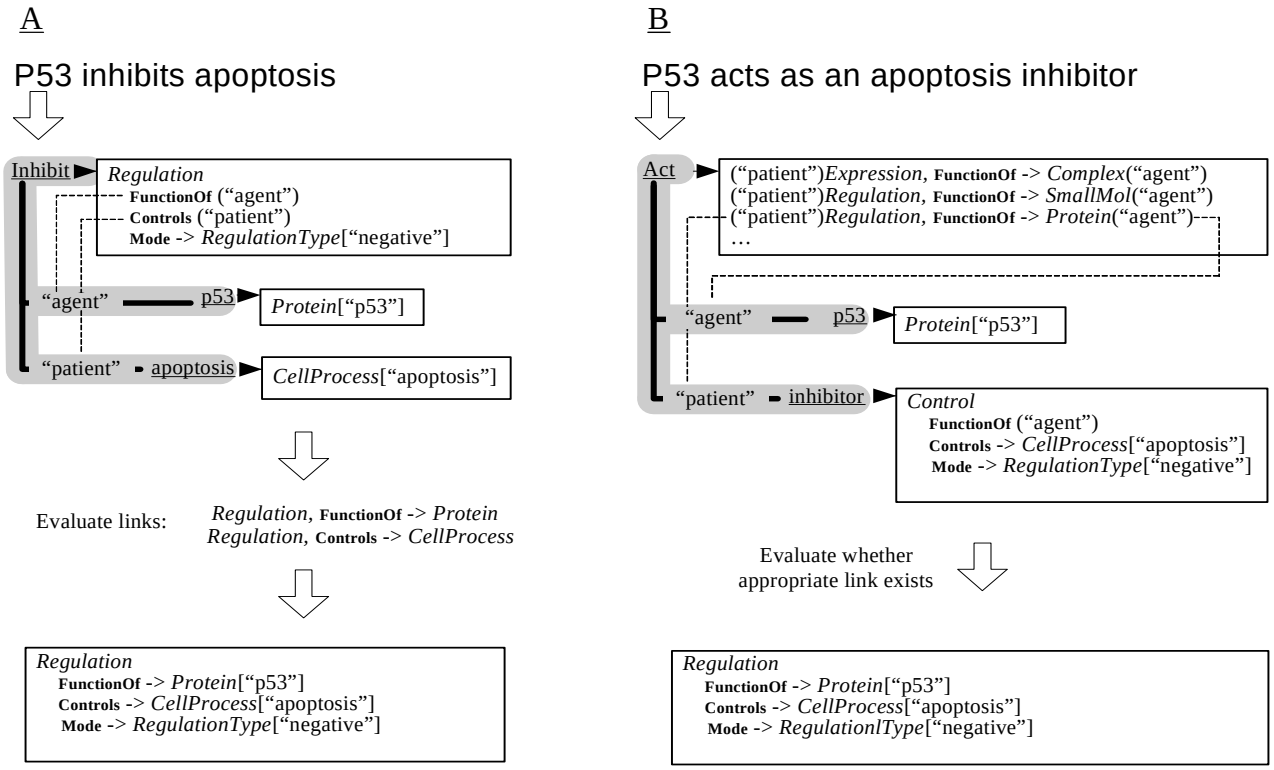

Figure 3. Verification and conversion of a semantic tree into a frame tree. A – interpretation of template lexeme links. B – interpretation of connecting lexeme. The interpreted semantic tree is shown on the left top part of each picture on a grey background with lexemes underlined. The semantic vertexes are shown as solid lines with thematic role labels in quotes. Immediately right to each lexeme is shown its sense, pointed to by a small solid arrow (►). Lexeme senses and frame trees (partially and completely built) are shown as blocks. Within each sense or frame tree, frame names are shown in italics, corresponding slot names in - reduced size bold typeface, and instantiating frame values (if present) – in square brackets. For filled frame slots, ontological links to the argument frame are indicated by arrows (->). For empty slots, corresponding role labels are indicated in parentheses. The correspondence between semantic vertex labels and slot labels in lexeme senses are shown by dotted lines. See detailed explanation of an algorithm in text.

ontology. At each step, a single semantic vertex is processed and a new sense is linked to the frame tree after a candidate link is constructed and its existence in ontology is verified.

When processing a semantic vertex between template lexemes, the algorithm first finds senses in the knowledge base that correspond to the parent and child lexemes. The link should be constructed between head frames of the found senses. The appropriate slot of the parental sense is identified by comparing the label of the processed semantic vertex to thematic slot labels of the head frame of parent sense. The candidate link constitutes the following triplet: i) head frame of parent sense, ii) one of its slots, and iii) the head frame of the sense for the child lexeme. The validation is performed by searching for an identical triplet in ontology. This process is illustrated in Figure 3A.

When interpreting connecting lexemes, the link is attempted to be formed between two child arguments of the lexeme. The interpretation is valid if the sense of the connecting lexeme contains the link between the head frames of child lexemes in an orientation defined by the link role labels. For example, in a sentence "*p53 acts as an apoptosis inhibitor*" when interpreting a semantic node "act", algorithm verifies that its sense in the knowledge base contains the link between frames *Regulation* (a head frame of "inhibitor") and *Protein* (a head frame of "p53"). If the validation succeeds, new links are incorporated into a constructed tree in a top-down orientation. In an example above, a parental frame of a found link *Regulation* becomes a root frame of a frame tree and the {*Regulation*, **ControlOf** -> *Protein* ["p53"]} branch is added to a frame tree (Figure 3B).

Any ontology is always a simplified and partial model of a knowledge domain and real sentences can be only partially mapped to the ontology. Unmapped information (for instance experimental details, references to similar work, etc.) may span parts or even most of the semantic tree. The goal, therefore, becomes to intelligently identify relevant parts of the sentences. Different syntactic constituents have different grammatical roles and different

implications on a projection of a sentence meaning to ontology. Some of them (prepositional phrases or lexemes in noun phrases) do not significantly affect it; others (for example verb and noun compliments) are critical for the meaning, and yet another ones (appositions, coordinated and relative clauses, or conjunctions) represent either independent or only loosely coupled pieces of information. To account for these differences, our algorithm takes advantage of the knowledge of the syntactic structure of an entire sentence and applies the link validation procedure with different stringency to different branches of a semantic tree. Stringency depends on the syntactic nature of a branch (captured in the vertex labels) and its hierarchical position. If a thematic role vertex fails to pass the validation, entire branch is rejected, while attributive vertexes (usually created from prepositional phrases, noun phrases or conjunctions), which fail validation, are simply ignored. In addition, the interpretation is performed only for branches stemming from the root of a semantic tree. This strategy allows for accurate search for the ontologically relevant information while concurrently discriminating against incorrect alternative syntactic structures. Notably, because the interpretation may leave out branches of low importance, it may generate identical frame trees from the same sentence. This occurs if a skipped branch was the source of syntactic ambiguity; identical trees are subsequently filtered out during the final output.

Converter, the last module in MedScan pipeline, converts a constructed frame tree into a form of generalized conceptual graph, and records it in an XML-based format. It transforms a frame tree into the set of functional links between proteins, cellular processes, cellular components, and small molecules based on the categorization of ontological frames into **Control** and **Entity** groups. The type of link ("expression control", "binding", "molecular transport", "protein modification", etc) reflects the nature of functional interaction between entities. The links can be directed ("protein modification") or undirected ("binding") depending on the nature of described functionality. The links are also attributed with additional extracted information such as effect ("positive" or "negative"), mechanism (for example "phosphorylation"), a reference to the MEDLINE abstract and the sentence it was extracted from. Converter is configured not to output the negatively labeled relations.

Results and Discussion

We have chosen to test MedScan by evaluating its ability to extract information about human direct physical protein-protein interactions from MEDLINE abstracts. Protein interactions describe explicit biological phenomena that can be unambiguously detected by a human reader and is commonly used to evaluate information retrieval systems (Blaschke *et al.*, 1999, Ono *et al.*, 2001). However, the syntactic nature of sentences describing interactions makes automatic extraction of information quite a challenging task and could best serve as a practical test for a MedScan performance. To illustrate the latter point the reader is invited to compare the following sentences: "*a and b interact*", "*a and b interact with c*", "*a and b interact with each other*" and "*a and b interact with high affinity*". Despite their syntactic similarity the interactions described by them are different and are unlikely to be correctly extracted by simpler pattern matching-based techniques.

The dictionary of human protein names necessary for the preprocessing-level protein name identification was compiled on a basis of HUGO consortium data and additionally enriched by incorporating protein names, aliases, descriptions and gene names from the linked SwissProt, TREMBL and Locus Link database entries. This dictionary was then curated in order to remove entries constituting single frequently used normal English words. Approximately 600 of colliding gene names have been either resolved or removed. The resulting non-redundant list contained approximately 68,000 protein aliases and descriptions for 15,000 human proteins.

From approximately 3.5 million MEDLINE abstracts dated after 1988 3.4 million sentences containing at least one notation of a human protein have been selected in a preprocessing step. Out of them only 1.2 million (35%) have been successfully parsed by an NLP component of MedScan and submitted for information extraction; 3601 total interactions corresponding to 2,976 distinct protein-protein interactions have finally been extracted. The processing speed on AMD XP 1800 PC computer with 256M RAM was 10 milliseconds per sentence.

To evaluate the quality of gathered data, we have manually reviewed 361 randomly extracted protein interactions and associated source sentences and concluded that 327 (91%) of them are correct. When we analyzed the remaining 34 erroneous facts we noticed that in 19 cases the misinterpreted sentences contained the words "interact" or "interaction" referring to the functional interference between the proteins, mostly, cytokines, rather than physical interaction. Unfortunately, correct interpretation of such artifacts is outside the scope of natural language processing.

To further validate the interactions extracted by MedScan we have compared them with a human subset of protein interactions available in public databases BIND (*URL2*) and DIP (*URL1*). Both DIP and BIND are curated databases and contain 781 and 86 interactions between human proteins, correspondingly. The correspondence between DIP, BIND and MedScan proteins was established based on GenBank GI identifiers and direct comparison of protein names. The overlap between different sources was rather low. Only 125 interactions retrieved by MedScan were found either in BIND (23 out of 86) or in DIP (92 out of 781).

The MedScan recovery rate was estimated by the manual analysis of 91 randomly selected sentences from 43 abstracts containing information about protein-protein interaction between at least two proteins, and was found to be 21%. This low recovery rate is primarily due to the low (34%) coverage of the MedScan's NLP component. We have also noticed that most of the extracted information is unique: only 478 facts out of 2,976 were extracted more than once. This in contrast to the reports concerning redundancy of MEDLINE information (Blaschke and Valencia, 2002). It is possible that high proportion of the unique facts extracted by MedScan may be attributed to the low recall rate of the system. On the other hand, if the MEDLINE is, indeed, a highly redundant source of information, actual recovery rate of MedScan (or any other information extraction system) when applied to the whole MEDLINE can be hard to evaluate.

We have further tested MedScan by conducting a large-scale information extraction experiment to retrieve different types of regulatory links between proteins, cell processes and small molecules. The information extracted included such phenomena as gene expression regulation, molecular transport, protein modification, regulation of cell processes, indirect causal relations between different entities and other types of data conforming to a general notion of molecular networks and represented in the described ontology. 100,000 facts were captured from the full release of 2002 MEDLINE. The precision of data extraction in this experiment was found to be similar to the precision rate of protein interactions extraction. The full report and analysis of the extracted data will be published in details elsewhere.

Manual analysis of a few hundred sentences revealed that significant proportion of the protein function information in MEDLINE is expressed in a form of raw experimental data presentation and is not covered by our ontology. Interpretation of experimental data is not a trivial task (even for human experts) and extracting information from the description of experimental results extends beyond the natural language processing field into the domain of logical inference and logic programming. For example, the sentence "Number of MDA-2 cells with apoptotic phenotype increased 50% after treatment with wortmannin" implies that the wortmannin somehow controls apoptosis, but this knowledge cannot be inferred from the sentence by solely NLP methods.

In view of that, we envision two major directions of further MedScan development. First, we intend to bring the efficiency of NLP module above the 90% mark, primarily by improving the grammar, size and quality of the lexicon, and preprocessing algorithm (Novichkova *et al.*, 2003). This will increase the recovery rate of the pipeline 3-4 fold. Second, we are currently testing applicability of available inference engines (i.e. Prolog) for extraction of logically implicit data.

In conclusion, MedScan is high precision information extraction system capable of extracting various types of protein function information encoded in a form of extendable ontology. Utilization of ontology provides an ability to change the scope of extracted information, making entire system more flexible, and along with high performance, favorably differentiates it from the other systems. Existing MedScan technology readily allows extraction of about 100,000 facts concerning human protein function with a precision high enough for practical application of the system. The high precision of the MedScan stems from its full-sentence parsing approach and presently comes at the price of a lower recall rate. However, the volume of data can be increased several times by implementing a reasonable set of improvements to the system, extending the ontology towards the description of experimental data and application of logical inference methods in order to convert the experimental result into the protein function information.

URL References

URL1: http://dip.doe-mbi.ucla.edu/
URL2: http://www.bind.ca/
URL3: http://www.csa.ru/Inst/gorb_dep/inbios/genet/genet.htm
URL 4 http://193.175.244.148/
URL 5 http://geo.nihs.go.jp/csndb/
URL 6: http://www.grt.kyushu-u.ac.jp/eny-doc/

References

Ashburner M., Ball C.A., Blake J.A., Botstein D., Butler H., Cherry J.M., Davis A.P., Dolinski K., Dwight S.S., Eppig J.T., et al (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet, 25:25-29

Blaschke, C., Andrade, M.A., Ouzounis, C., and Valencia, A. (1999). Automatic extraction of biological information from scientific text: protein – protein interactions. Ismb: 60-67.

Blaschke C., and Valencia, A. (2002) The frame-based module of the Suiseki information extraction system, IEEE Intelligent Systems 17: 14-20.

Chen R.O., Felciano, R, and Altman, R.B. (1997) RIBOWEB: linking structural computations to a knowledge base of published experimental data. Ismb 5:84-87.

Friedman, C. Kra, P., Yu, H., Krauthammer, M., and Rzhetsky, A. (2001) GENIES: a natural language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 17, Suppl 1: S74-S82.

Humphreys, K., Demetriou, G., and Gaizauskas, R. (2000). Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. Pac Symp. Biocomput.: 505-516.

Karp, P.D., Riley, M., Paley, S.M., Pelligrini-Toole, A., and Krummenacker, M. (1999). Eco Cyc: Encyclopedia of Escherichia coli genes and metabolism. Nucleic Acid Res. 27: 55-58.

Novichkova, S., Egorov, S., and Daraselia, N. (2003) Medscan, a natural language processing engine for MEDLINE abstracts. Bioinformatics, in press.

Ono, T., Hishikagi, H., Tanigami, A, and Takagi, T. (2001). Automated extraction of information on protein – protein interactions from the biological literature. Bioinformatics 17: 155-161.

Park, J.C., Kim, H.S., and Kim, J.J. (2001). Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. Pac. Symp. Biocomput. 6: 396-407.

Rzhetsky, A., Koike, T., Kalachikov, S., Gomez, S. M., Krauthammer, M., Kaplan, S.H., Kra, P., Russo, J.J., and Friedman, C. (2000). A knowledge model for analysis and simulation of regulatory networks. Bioinformatics 16: 1120 – 1128.

Schultze-Keremer, S. (1997) Adding semantics to genome databases: towards an ontology for molecular biology. Ismb 5, 272-275.

See-Kiong, N., and Wong, M. (1999). Toward routine automatic pathway discovery from on-line scientific text abstracts. Genome Informatics 10: 104 – 112.

Sekimizu, T., Park, H.S., and Tsujii, J. (1998). Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. Genome informatics 9: 62-71.

Stephens, M., Palakal, S., Mukhopadhyay, S., and Raje, R. (2001). Detecting gene relations from MEDLINE abstracts. Pac Symp Biocomput.: 483 – 495.

Thomas, J., Milward, D., Ouzounis, C.A., Pulman, S., and Caroll, M. (2000). Automatic extraction of protein interactions from scientific abstracts. Pac. Symp. Biocomput.: 541-552.

Yakushiji, A., Tateisi, Y., Miyao, Y., and Tsujii, J. (2001). Event extraction from biomedical papers using a full parser. Pac. Symp. Biocomput. 6: 408-419.