Utilizing Text Mining Results: The PastaWeb System

G. Demetriou and R. Gaizauskas
Department of Computer Science
University of Sheffield
Western Bank
Sheffield S10 2TU UK

{g.demetriou, r.gaizauskas}@dcs.shef.ac.uk

Abstract

Information Extraction (IE), defined as the activity to extract structured knowledge from unstructured text sources, offers new opportunities for the exploitation of biological information contained in the vast amounts of scientific literature. But while IE technology has received increasing attention in the area of molecular biology, there have not been many examples of IE systems successfully deployed in end-user applications. We describe the development of PASTAWeb, a WWW-based interface to the extraction output of PASTA, an IE system that extracts protein structure information from MEDLINE abstracts. Key characteristics of PASTAWeb are the seamless integration of the PASTA extraction results (templates) with WWW-based technology, the dynamic generation of WWW content from 'static' data and the fusion of information extracted from multiple documents.

1 Introduction

The rapidly growing volume of scientific literature, a by-product of intensive research in molecular biology and bioinformatics, necessitates efficient and effective information access to the published text sources. Information retrieval (IR) techniques employed in WWW interfaces such as PubMed and Entrez are very useful in browsing bibliographic databases and in facilitating the linking between protein or genome sequences and related references. But the jobs of automatically locating and extracting specific information within the texts require more specialised Natural Language Processing (NLP) techniques and have been the object of work in *information extraction* (IE) or *text mining* ((Cowie and Lehnert, 1996)).

While current work on IE in biology has concentrated by and large on the refinement of IE techniques and improving their accuracy, the incorporation of an IE system's output into effective interfaces that genuinely assist the biological researcher in his/her work is equally important, and has been neglected to date. Of course improving IE techniques, their accuracy and cross-domain portability are important research objectives for language technology researchers. But given that the techniques will remain imperfect for the foreseeable future, we must also ask how biologists can benefit today from the limited capabilities of existing IE technology.

In this paper we describe an approach to providing effective access to the results of the Protein Active Site Template Acquisition (PASTA) system (Humphreys et al., 2000; Demetriou et al., 2002), an IE system that extracts information about amino acid residues in protein structures and their roles in protein active sites directly from the published literature. To experiment with a mechanism for delivering PASTA results to biologist end-users, we have developed the PASTAWeb interface, a WWW-based interface that offers search and browsing facilities to the extracted protein structure information, as well as to the original text sources (MEDLINE

Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain, Philadelphia, July 2002, pp. 77-84.
Association for Computational Linguistics.

abstracts). PASTAWeb provides transparent text access and advanced navigation capabilities to enable users to track and display the relevant information from text to text. The PASTAWeb facilities enable users to find answers to implicit questions such as *What are the important residues for trypsin?* or *Is serine found in the active site of amylase?* and to track the flow of information for specific classes of biological entities (residues, proteins, species) from text to text.

Given the performance limitations of current IE technology, it is to be expected that some of the extracted information may only be partially correct, missing or spurious. The PASTAWeb interface compensates for the loss of information by supporting rapid, easy verification by scientists of the extracted information against the source texts.

2 IE and its Application to Biomedical Texts

Perhaps not surprisingly, the identification of biomedical terms in scientific texts has proved to be the easiest extraction task and has demonstrated acceptable levels of performance, not too far from the best results achieved in the NE task in the MUC competitions, despite differences between the domains (i.e. names of persons, organisations etc. in MUC vs. terms identifying proteins, genes, drugs etc. in biomedical domains). The techniques used for this task vary from rule-based methods (Fukuda et al., 1998; Humphreys et al., 2000), to statistical methods (Collier et al., 2000) and statistical-rule-based hybrids (Proux et al., 1998).

More complex IE tasks involving the extraction of relational information have also been addressed by the bioinformatics community. These include protein or gene interactions (Sekimizu et al., 1998; Thomas et al., 2000; Pustejovsky et al., 2002), relations between genes and drugs (Rindflesh et al., 2000) and identification of metabolic pathways (Humphreys et al., 2000). The range of techniques used in these systems varies considerably, but in most cases requires the application of more sophisticated NLP methods including part-of-speech tagging, phrasal or syntactic parsing and (for some systems) semantic analysis and discourse processing.

To date IE researchers working on biological texts have concentrated on building or porting systems to work in biological domains. This paper addresses the issue of utilising the IE results, after describing, in the next section, the underlying PASTA extraction system – what it is designed to extract, how it works, and how well it fares in blind evaluation using conventional evaluation metrics.

```
<RESIDUE-134> :=
   NAME: SERINE NO: 87
   SITE/FUNCTION: "catalytic"
                  "calcium-binding"
                  "active-site"
   SEC_STRUCT: "helical"
   QUAT_STRUCT: <not specified>
   REGION: "lid"
   INTERACTION: <not specified>

<IN_PROTEIN> :=
   RESIDUE: <RESIDUE-134>
   PROTEIN: <PROTEIN-2>

<IN_SPECIES> :=
   PROTEIN: <PROTEIN-2>
   SPECIES: <SPECIES-5>

<PROTEIN-2> :=
   NAME: "triacylglycerol lipase"

<SPECIES-5> :=
  NAME: "Pseudomonas cepacia"
```

Figure 1: PASTA template example

3 The PASTA System

The overall aim of the PASTA system is to extract information about the roles of residues in protein molecules, specifically to assist in identifying active sites and binding sites. We do not describe the system in great detail here, as this is described elsewhere (Demetriou et al., 2002).

3.1 PASTA Extraction Tasks

3.1.1 Terminological Tagging

A key component of PASTA, and of various other IE systems operating in the biomedical domain is the identification and classification of textual references (terms) to key entity types in the domain. We have identified 12 significant classes of technical terms in the PASTA domain: *protein, species, residue, site, region, secondary structure, supersecondary*

structure, quaternary structure, base, atom (element), non-protein compound, interaction. Guidelines defining the scope of the term classes were written, and an SGML-based markup scheme specified to allow instances of the term classes to be tagged in texts[1].

3.1.2 PASTA Template Filling Tasks

The PASTA template conforms to the MUC template specification and is *object oriented*. Slot fillers are of three types: (1) *string fill* – a string excised directly from the text (e.g. `Pseudomonas cepacia`); (2) *set fill* – a normalised form selected from a predefined set (e.g. the expressions `Ser` or `serine` are mapped to `SERINE`, one of a set of normalised forms that represent the 20 standard amino acids); (3) *pointer fill* – a pointer to another template object, used, e.g., for indicating relations between objects.

To meet the objectives of PASTA, three template elements and two template relations were identified. The elements are `RESIDUE`, `PROTEIN` and `SPECIES`; the two relations are `IN_PROTEIN`, holding between a residue and the protein in which it occurs, and `IN_SPECIES`, holding between a protein and the species in which it occurs.

An example of a template produced by PASTA for a Medline abstract is shown in Figure 1, which illustrates the three template element objects and two template relation objects. As can be seen from the figure, the `<RESIDUE>` template object contains slots for the residue name and the residue number in the sequence (`NO`). Secondary and quaternary structural arrangements of the part of the structure in which the residue is found are stored in the `SEC_STRUCT` and `QUAT_STRUCT` slots respectively. The `SITE/FUNCTION` slot is filled with widely recognizable descriptions that indicate that this residue is important for the structure's activation (e.g. `active-site`) or functional characteristics (e.g. `catalytic`). The `REGION` slot is about the more general geographical areas of the structure (e.g. `lid`) in which this particular residue is found[2]. The `INTERACTION` slot captures textual references to hydrogen bonds, disulphide bonds or other types of atomic contacts. At this point the only attributes extracted for protein and species objects are their names.

[1]The term class annotation guidelines are available at: `http://www.dcs.shef.ac.uk/nlp/pasta`.

[2]A residue may belong to more than one region

3.2 System Architecture

The PASTA system has been adapted from an IE system called LaSIE (Large Scale Information Extraction), originally developed for participation in the MUC competitions (Humphreys et al., 1998). The PASTA system is a pipeline of processing components that perform the following major tasks: text preprocessing, terminological processing, syntactic and semantic analysis, discourse interpretation, and template extraction.

Text Preprocessing The text preprocessing phase aims at low-level text processing tasks including the analysis of the structure of the MEDLINE abstracts in terms of separate sections (e.g. the title, author names, abstract etc.), tokenisation and sentence boundary identification. With respect to tokenisation, tokens are identified at the subword level resulting in the splitting of biochemical compound terms into their constituents which need to be matched separately during the lexical lookup phase. For example, the term `Cys128` is split to the three-letter residue abbreviation `Cys` and the numeral `128`.

Terminological Processing The aim of the 3-stage terminological processing phase is to identify and correctly classify instances of the term classes described above in section 3.1.1. During the *morphological analysis* stage individual tokens are analysed to see if they contain interesting biochemical affixes such as `-ase` or `-in` that indicate candidate protein names respectively.

During the *lexical lookup* stage the previously tokenised terms are matched against terminological lexicons which have been compiled from biological databases such as CATH[3] and SCOP[4] and have been augmented with terms produced by corpus processing techniques (Demetriou and Gaizauskas, 2000). Additional subcategorisation information is provided for multi-token terms by splitting the terms into their constituents and placing the constituents into subclasses whose combination is determined by grammar rules.

[3]http://www.biochem.ucl.ac.uk/bsm/cath/index.html

[4]http://scop.mrc-lmb.cam.ac.uk/scop/

	Development		Interannotator		Blind	
	Recall	Precision	Recall	Precision	Recall	Precision
Terminology recognition	88	94	92	86	82	84
Template extraction	69	79	78	80	69	64

Table 1: Summary evaluation results for term recognition/classification and template extraction.

Finally, in a *terminology parsing* stage, a rule-based parser is used to analyse the tokenisation information and the morphological and lexical properties of component terms and to combine them into a single multi-token unit.

Syntactic and Semantic Processing Terms classified during the previous stages (proteins, species, residues etc.) are passed to the syntactic processing modules as non-decomposable noun phrases and a part-of-speech tagger assigns syntactic labels to the remaining text tokens. With the application of phrasal grammar rules, the phrase structure of each sentence is derived and this is used to build a semantic representation via compositional semantic rules.

Discourse Processing During the discourse processing stage, the semantic representation of each sentence is added to a predefined *domain model* which provides a conceptualisation of the knowledge of the domain. The domain model consists of a concept hierarchy (ontology) together with inheritable properties and inference rules for the concepts. Instances of concepts are gradually added to the hierarchy in order to construct a complete *discourse model* of the input text.

Template Extraction A template writing module scans the final discourse model for any instances that are relevant to the template filling task, ensures that it has all necessary information to generate a template and fill its slots, and then formats and outputs the templates.

3.3 Development and Evaluation

Following standard IE system development methodology, a corpus of texts relevant to the study of protein structure was assembled. The corpus consists of 1513 Medline abstracts from 20 major scientific journals that publish new macromolecular structures. Of these abstracts, 113 were manually tagged for the 12 term classes mentioned above and 55 had associated templates filled manually. These annotated data were divided into distinct training and test sets.

The corpus and annotated data assisted in the refinement of the extraction task definitions, supported system development and permitted final blind evaluation of the system. Detailed results of the evaluation, for each term class, and for each slot in the templates, can be found in Demetriou et al. (2002). In Table 1 we present the summary totals for the development corpus, the unseen final evaluation corpus (Blind) and the human interannotator agreement where one annotator is taken to be the gold standard and the other scored against him/her. The evaluation metrics are the well known measures of *precision* and *recall*.

4 The PastaWeb Interface

The PASTAWeb interface[5] is aimed at providing quick access and navigation facilities through the database of the PASTA tagged texts and their associated templates. PASTAWeb has borrowed ideas from the interface component of the TRESTLE[6] system Gaizauskas et al. (2001) developed to support information workers in the pharmaceutical industry. Key characteristics of PASTAWeb are the seamless integration between the PASTA IE results and WWW-based browsing technology, the dynamic generation of WWW pages from "static" content and the fusion of information relating to proteins and amino acid residues when found in different sources.

4.1 PASTAWeb Architecture

The PASTAWeb architecture is illustrated in Fig 2.

[5] Accessible at `http://www.gate.ac.uk/cgi-bin/pasta.cgi?source=start` or via the PASTA project home page at `http://www.dcs.shef.ac.uk/nlp/pasta/`

[6] TRESTLE:Text Retrieval Extraction and Summarisation Technologies for large Enterprises

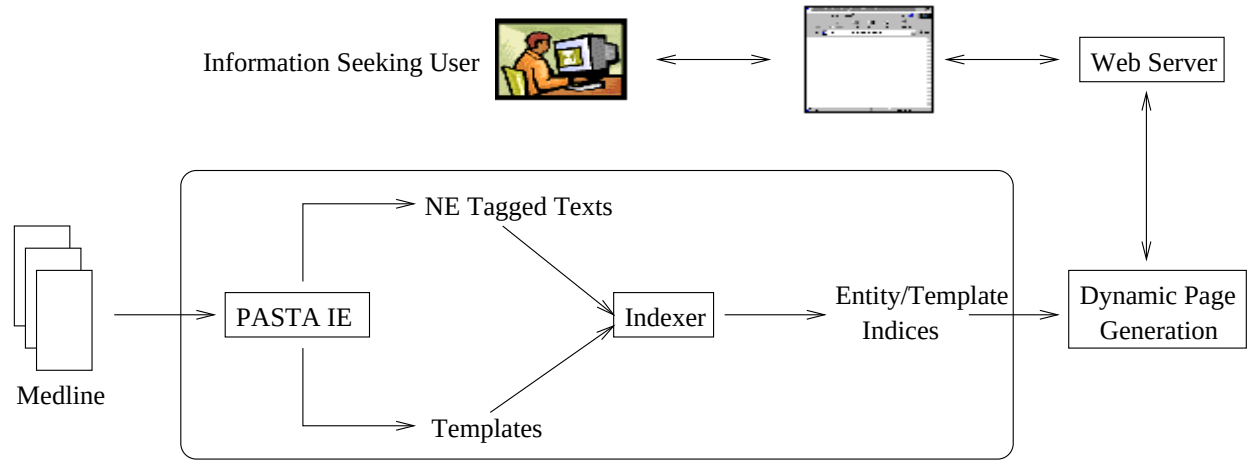

Figure 2: The PASTAWeb Architecture

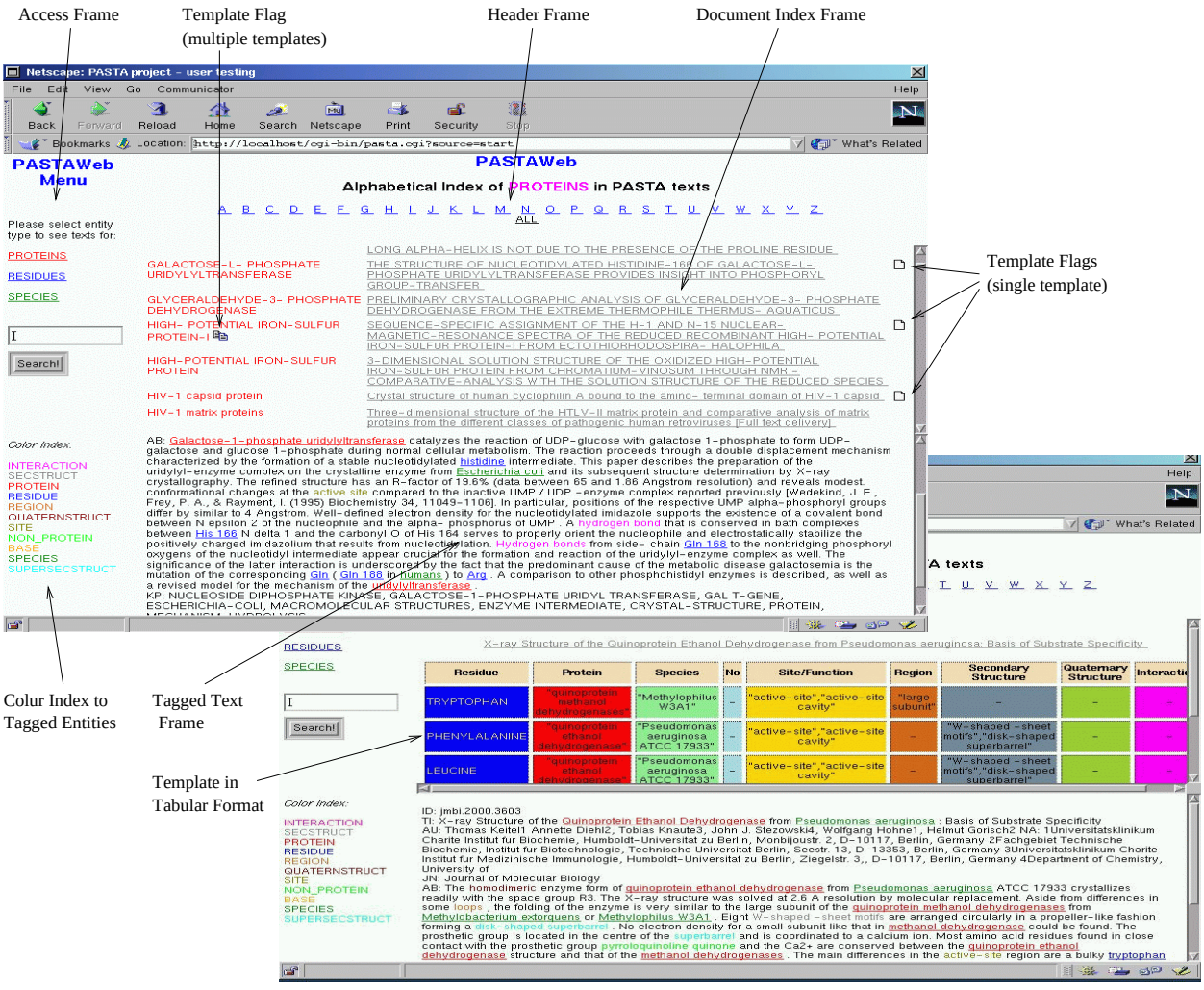

Figure 3: The PASTAWeb Interface

Initially, MEDLINE abstracts are fed through the PASTA IE system which produces two kinds of output: (i) texts annotated with SGML tags describing term class information for protein, residues, species, regions, and (ii) templates which are used as the main stores of information about residues including relational information between proteins and residues and between proteins and species.

Once PASTA has run, a separate indexing process creates three indices. The first associates with each processed document the terminology tagged version of the text and any templates extracted from the text. The second is a relational table between each document and each of the instances of the main term classes (i.e. proteins, residues or species) mentioned in the document. This index also points to the title of the document, because the title can provide vital clues about the content of the text.

The final index is used to assist the 'fusion' of the information in templates generated from multiple texts for the same protein. This index provides information about those proteins for which there are templates generated from multiple documents. Due to variations in the expression of the same protein name from text to text, the identification of suitable templates for fusion is not trivial. The problem of matching variant expressions of the same term in different databases is a well known problem in bioinformatics. The current implementation of the indexing addresses this problem using simple heuristic rules. Simply put, two templates are considered suitable for fusion if the protein names either match exactly (ignoring case sensitivity) or they include the same "head term". The applicability of the heuristic for finding a "head term" is limited to constituent terms ending in `-ase` or `-in` (to exclude common words, such as "protein", "domain" etc.). For example, the protein terms "scorpion toxin", "diphtheria toxin" and "toxins" would match with each other because they all include the head term "toxin". Consequently, the corresponding template information about the residues occurring in these proteins would be merged into a single table, though information about which slot fillers belong to which term variant is retained.

The decision to do the matching of variant names at the index level and not at the interface level is simply due to operational issues. Matching the protein names from multiple texts involves the pairwise string comparisons between all proteins in the PASTA templates. The number of these comparisons increases very rapidly as new texts and templates are added to the database and it was found that is causes considerable delay to the operation of the PASTAWeb interface.

Since information seeking tasks of molecular biologists may require complex navigation capabilities, the storing of the results in "static" HTML pages would have been unsuitable both practically (more difficult to implement pointers between different pieces of information and to alter and maintain pages) and economically (requires more disk space). We therefore opted for a dynamic page creator that is triggered by the users' requests expressed as choices over hypertext links. The dynamic page creator compiles the information from the indices and the associated databases (texts and templates) and sends the results to the WWW browser via a Web server. In the dynamically created pages, each hypertext link encodes the current frame, the information to be displayed when the link is selected, and the frame in which this information is to be displayed. For example, the hypertext link for a title of a document encodes information about the document id of the document as well as about the target frame in which the text will be displayed. Clicking on this link expresses a request to PASTAWeb for displaying that particular text in the target frame. The whole operation of PASTAWeb loosely resembles the operation of a finite-state automaton.

4.2 Interface Overview

PASTAWeb offers a number of ways of accessing the protein structure information extracted by PASTA. As shown in Fig 3 the interface layout can be split into four main areas (frames). On the left side of the page we find the "Access Frame" which allows the user to select amongst text access options. These options include browsing the contents of the text databases via either the protein, the residue or the species indices or via a text search option over these indexed terms.

The right hand side of the screen is split into three frames. The top frame, so called "Header Frame"(see Fig 3), is used to generate an alphabetical index for protein or species names whenever the

user has chosen the protein or species access modes for navigation. For residues, rather than an alphabetical index, a list of residue names is displayed in the "Header Frame". This is because while the number of protein names and their variants is probably indeterminable, the number of residues remains constant (i.e the 20 standard amino acids).

Just below the "Header Frame" is the "Document Index Frame" which initially serves to display the automatically generated indices together with document information. The "Index Frame" is split into two columns, the left of which is used to present an alphabetically sorted list of the chosen type of index (i.e. protein, residue, species). The right column occupies more space because it displays the list of corresponding document titles (as extracted by the PASTA IE system). These titles are presented as clickable hyperlinks to the full texts each of which can be displayed in the "Tagged Text Frame" below.

A second use of the "Index Frame" is for displaying template results, explained in more detail below.

4.3 Term Access to Texts

A typical interaction session with PASTAWeb requires the user to select one of the three term categories in the "Access Frame", i.e. proteins, residues or species. The "Header Frame" then displays a list of alphabetical indices (for proteins and species) or a list of residue names. Selecting any of these indices, e.g. "M" for proteins, activates the dynamic generation of a list of protein terms that are indexed by "M" (on the left) of the "Index Frame" and their corresponding document titles (on the right). Different font colours are used to distinguish between the two different kinds of information.

The selection of any of the title links causes the system to dynamically transform the PASTA-tagged text from SGML to HTML and display it in the bottom "Tagged Text Frame" with the recognised term types highlighted in different colours. The colour index for the term categories can be viewed in a frame just below the "Access Frame" (the "Colour Index Frame"). Each tagged protein, species or residue term is itself a hyperlink which can be used to dynamically fetch the indices of the texts in which this term occurs and display them in the "Index Frame".

Using this functionality, the user can navigate through a succession of texts following a single term or at any point branching off this chain by selecting another term and following its occurrences in the text collection.

4.4 Web-based Access to Templates

Unfortunately, although the type of object-oriented template produced by PASTA (Fig 1) is an efficient data structure for storing complex information, it is not suitable for displaying to end-users. For this reason, the templates are dynamically converted to a format that can be readily accommodated to the screen's layout while being at the same time easily accessible. The format chosen for displaying the PASTA templates is tabular and is implemented as an HTML table (see background picture in Fig 3).

Access to the templates produced by PASTA is facilitated by special template "icons" or "flags" which are displayed next to text titles or protein terms in the "Index Frame".

When a "single" template icon is displayed to the right of a title, this serves to flag that a template for this text is available and can be accessed by clicking on the icon. On the other hand, when a "double" template icon is displayed next to a protein name in the left column of the "index frame", this indicates that there are multiple templates (i.e. templates extracted from different texts) for this protein. Clicking on either of these icons will trigger PASTAWeb to scan the corresponding object-oriented templates, analyse their structures and convert them into tabular format. In the case of fused templates the information is assimilated into a single template. The template information is then displayed in the "Index Frame" together a hyperlink to the title of the original text which, when selected, displays the (tagged) text in the "Tagged Text Frame". This enables the user to retrieve more detailed information from the text if needed, or to inspect and verify the correctness of the extracted information.

PASTAWeb offers a simple and easy to use mechanism for the tracking of information for a specific entity from text to text, but can also assist in the linking of information between different entities in multiple documents. Starting with a specific protein in mind for example, a molecular biologist may want to investigate structural similarities between that and other proteins with respect to what has been described in the literature.

5 Conclusions and Future Work

We have described PASTAWeb, a prototype Web-based interface to the IE results of the PASTA system. Scientists in the area of molecular biology can benefit from the novel navigation and information tracking capabilities of PASTAWeb and use it as a tool for fast access to specialised information.

At the time of writing, the database of processed texts accessible through PASTAWeb is rather small (975 texts in total). The rate at which new articles appear on MEDLINE and the limited resources devoted to PASTA make it prohibitive to develop and maintain for PASTAWeb a text database of size comparable to MEDLINE. Nevertheless, PASTAWeb offers the core technology for the development of a fully automated IE system whose input can be based on automated updates ("autoalerts") from MEDLINE without human intervention. Current work concentrates on the development of such an automated software component and on the feasibility of expanding the system's navigation capabilities to allow users to link together information provided by PASTAWeb and by related servers such as the Protein Data Bank or SWISSPROT.

Finally, the utility of an interface such as PASTAWeb can only be truly assessed by user evaluation. Usability evaluation should be carried out using both qualitative and quantitative methods. Qualitative evaluation should be used to review the users' perceptions about the design, their preferred strategies for accessing information and whether they find the system easy to use and useful for performing their tasks. Quantitative evaluation should focus on measures of activity time, efficiency in tracking relevant information and on analysing the effect "noise" in the IE results has on user satisfaction. However, while this evaluation remains to be done, we believe that the work presented here provides concrete, constructive ideas about how to effectively utilise the output of IE systems in the biology domain.

References

N. Collier, C. Nobata, and J. Tsujii. 2000. Extracting the names of genes and gene products with a hidden markov model. In *Proc. of the 18th Int. Conf. on Computational Linguistics (COLING-2000)*, pp. 201–207.

J. Cowie and W. Lehnert. 1996. Information extraction. *Communications of the ACM*, 39(1):80–91.

G. Demetriou and R. Gaizauskas. 2000. Automatically augmenting terminological lexicons from untagged text. In *Proc. of the 2nd Int. Conf. on Language Resources and Evaluation (LREC-2000)*, pp. 861–867, Athens, May-June.

G. Demetriou, R. Gaizauskas, P. Artymiuk, and P. Willett. 2002. Protein structures and information extraction from biological texts: The PASTA system. *Bioinformatics*. Accepted for publication.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. 1998. Information extraction: Identifying protein names from biological papers. In *Proc. of the Pacific Symp. on Biocomputing '98 (PSB'98)*, pp. 707–718, Hawaii, January.

R. Gaizauskas, P. Herring, M. Oakes, M. Beaulieu, P. Willett, H. Fowkes, and A. Jonsson. 2001. Intelligent access to text: Integrating information extraction technology into text browsers. In *Proc. of the Human Language Technology Conf. (HLT 2001)*, pp. 189–193, San Diego.

K. Humphreys, R. Gaizauskas, S. Azzam, C Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 1998. Description of the LaSIE-II system as used for MUC-7. In *Proc. of the 7th Message Understanding Conf. (MUC-7)*. Available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

K. Humphreys, G. Demetriou, and R. Gaizauskas. 2000. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proc. of the Pacific Symp. on Biocomputing '2000 (PSB'2000)*, pp. 505–516, Hawaii, January.

D. Proux, F. Rechenmann, L. Julliard, V. Pillet, and B. Jacq. 1998. Detecting gene symbols and names in biological texts. In *Proc. of the 9th Workshop on Genome Informatics*, pp. 72–80.

J. Pustejovsky, J. Castano, J. Zhang, M. Kotecki, and B. Cochran. 2002. Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proc. of the Pacific Symp. on Biocomputing 2002 (PSB'2002)*, pp. 362–373, Hawaii, January.

T. Rindflesh, L. Tanabe, J. Weinstein, and L. Hunter. 2000. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proc. of the Pacific Symp. on Biocomputing '2000 (PSB'2000)*, pp. 517–528, Hawaii, January.

T. Sekimizu, H. S. Park, and J. Tsujii. 1998. Identifying the interactions between genes and gene products based on frequently seen verbs in Medline abstracts. In *Proc. of Genome Informatics*, pp. 62–71, Tokyo.

J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. 2000. Automatic extraction of protein interactions from scientific abstracts. In *Proc. of the Pacific Symp. on Biocomputing '2000 (PSB'2000)*, pp. 541–551, Hawaii, January.