BIOINFORMATICS

Vol. 19 no. 1 2003
Pages 135–143

# Protein Structures and Information Extraction from Biological Texts: The PASTA System

R. Gaizauskas[1,*], G. Demetriou[1], P. J. Artymiuk[2] and P. Willett[3]

[1]Department of Computer Science, [2]Department of Molecular Biology and Biotechnology and [3]Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TU, UK

Received on January 31, 2002; revised on July 29, 2002; accepted on August 7, 2002

**ABSTRACT**

**Motivation:** The rapid increase in volume of protein structure literature means useful information may be hidden or lost in the published literature and the process of finding relevant material, sometimes the rate-determining factor in new research, may be arduous and slow.

**Results:** We describe the Protein Active Site Template Acquisition (PASTA) system, which addresses these problems by performing automatic extraction of information relating to the roles of specific amino acid residues in protein molecules from online scientific articles and abstracts. Both the terminology recognition and extraction capabilities of the system have been extensively evaluated against manually annotated data and the results compare favourably with state-of-the-art results obtained in less challenging domains. PASTA is the first information extraction (IE) system developed for the protein structure domain and one of the most thoroughly evaluated IE system operating on biological scientific text to date.

**Availability:** PASTA makes its extraction results available via a browser-based front end: http://www.dcs.shef.ac.uk/nlp/pasta/. The evaluation resources (manually annotated corpora) are also available through the website: http://www.dcs.shef.ac.uk/nlp/pasta/results.html.

**Contact:** r.gaizauskas@dcs.shef.ac.uk; g.demetriou@dcs.shef.ac.uk; p.artymiuk@shef.ac.uk; p.willett@shef.ac.uk

## INTRODUCTION

The explosive growth in the number of protein structures in the Protein Data Bank (PDB) (Berman *et al.*, 2000) has spurred the development of a range of powerful computational tools to prove structural or functional analogies between amino acids in different structures. However, the assessment of significance of such structural comparisons often requires the extensive investigation of references in the literature to work out the importance of particular residues in protein structures.

While traditional Information Retrieval (IR) techniques (Baeza-Yates and Ribeiro-Neto, 1999) can be used to facilitate access to 'relevant' documents within huge document collections, the user is left to read the retrieved documents to satisfy his/her information need. Such systems cannot provide much help in locating specific pieces of relevant information within each text.

Information Extraction (IE) (Cowie and Lehnert, 1996), in contrast to IR, aims to identify automatically the relevant fragments of information in unstructured text sources and to extract these fragments into a structured representation that can subsequently be stored in a searchable database, used for content-based indexing or clustering of texts, summarized for the user, or input to a data mining algorithm.

In this paper, we describe an application of IE in the domain of three-dimensional protein structures. The Protein Active Site Template Acquisition (PASTA) system performs automatic extraction of filled templates relating to the roles of specific amino acid residues in protein molecules from online scientific articles and abstracts, with particular reference to proteins whose structure coordinates have been deposited in PDB. A structured database has been built from the extracted information and has been made searchable via a Web browser-based interface.

In previous work (Humphreys *et al.*, 2000), we presented preliminary results relating to the identification of terminological information in scientific papers. Here, we describe the full application of IE in the protein structure domain by discussing the methodology and the resources used for PASTA system development, and the results obtained after blind evaluation for both terminology identification and template filling tasks.

The PASTA system's novel contributions are: (1) development of a fast, accurate terminological processor for term classes not previously addressed; (2) application of template filling technology to a novel domain—protein structures; (3) application of extensive quantitative scoring methodology to a bioinformatics application and the publication of human-annotated datasets for use by the

*To whom correspondence should be addressed.

community; (4) development of a web-based front end to extraction results, made available via a publicly accessible web-site.

## INFORMATION EXTRACTION

The term *information extraction* is used to denote the activity of identifying information about specific entities, relationships, and events in natural language texts and extracting this information into one or more structured representations, or schema, called *templates*. A template typically consists of a number of slots that store information about the entities, relations and scenarios of interest.

IE has emerged as a technology incorporating some of the newest developments in natural language processing (NLP) and has matured through the Message Understanding Conferences (MUCs), a 10-year series of open competitive IE system evaluations sponsored by DARPA in the U.S. (MUC-7, 1998). IE tasks defined in the MUC competitions aimed primarily at the extraction of information from newswire texts about events such as terrorist attacks or joint venture announcements. A set of subtasks was specified to provide insight into different aspects of systems' performance. These subtasks ranged from the recognition of named entities (NEs), e.g. persons, locations, and organizations, through coreference resolution, the linking together of two or more textual expressions that refer to the same extralinguistic entity, e.g. pronouns and their antecedents, to the filling of templates.

A standard methodology has emerged for IE system development and evaluation through the MUC competitions (Gaizauskas and Wilks, 1998). Firstly, each extraction task is specified and explained as clearly as possible in a task definition document. This includes, for each template filling task, the precise specification of the template, both its syntax and semantics (the intended meaning of the slots). A set of relevant texts is selected for development, and another is chosen for blind evaluation. Each set is analysed by human experts to produce data that serves as the 'gold standard', i.e. correctly annotated texts and template extraction results. An IE system is then developed using the development texts along with their associated manually annotated texts and the extracted templates as a target. Systems may use any sort of approach: handcrafted rules, machine learning techniques, or both. To evaluate progress the system's answers are scored against the human-produced data using the measures of *precision*, which records the proportion of correct answers in the system's output, and *recall*, which records the proportion of correct answers with respect to the total number of answers in the texts. The final step is a blind evaluation against the held-out evaluation dataset.

While current state-of-the-art IE systems cannot guarantee perfect solutions and lag behind human performance overall, good results that approach human performance are achievable for some tasks, such as NE recognition. For the other tasks, performance levels are sufficient to make the technology effective in applications where the cost of collecting information manually is prohibitive and where some errors can be tolerated, because the results will be post-validated by a human.

## 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.

### PASTA Extraction Tasks

*Terminological Tagging.* One of the lessons from MUC is that template filling cannot be carried out well unless textual references to primary entities in the domain can be identified and classified. In biological texts the relevant entities are proteins, residues, species, etc., which are referred to via a bewildering profusion of technical terminology. We 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[†].

*PASTA Template Design.* The PASTA template, like the MUC templates, is *object oriented*. Each *template object* stores information about a specific *entity* (e.g. protein or residue), a *relation* between two entities (e.g. `in_protein`) or a stereotypical event or *scenario* (e.g. a metabolic reaction). Each object contains one or more *slots* each filled with information extracted from the text. Slot fillers may be of three types: (1) *string fill*—a string excised directly from the text (e.g. `Pseudomonas cepacia`); (2) *set fill*—a normalized form selected from a predefined set (e.g. the expressions `Ser` or `serine` are mapped to `SERINE`, one of a set of normalized 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. Pointer fill slots allow template objects to include linked and embedded objects to arbitrary levels of complexity.

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`,

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

```
<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"
```

**Fig. 1.** PASTA template examples

holding between a protein and the species in which it occurs. In contrast to the MUC IE tasks which revolved around event-based scenarios (e.g. terrorist attacks, joint venture announcements), the topics of interest for PASTA turned out to be static, not event-based, and hence no scenario template was specified (see, e.g. Humphreys *et al.*, 2000 for an example of a scenario-based template in a bioinformatics domain—enzyme interactions).

Examples of templates produced by PASTA for a Medline abstract are 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[‡]. 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.

[‡] A residue may belong to more than one region.

**Table 1.** Annotated PASTA Corpora: size in texts

| Task | Dev | Int | Blind |
|---|---|---|---|
| Terminology tagging | 52 | 20 | 61 |
| Template filling | 25 | 10 | 30 |

## The PASTA Corpus

Following the IE system development methodology described above in the section on **Information Extraction**, 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.

The corpus served three purposes: (1) *Extraction task refinement*. Both the definition of the terminology classes and the initial template sketch underwent a process of iterative refinement in order to cope with ambiguities uncovered as new cases were considered. A sample of corpus texts was analysed to help in this refinement process. (2) *System development*. Corpus analysis was used to identify the different ways in which the biological entities and relationships are expressed and the typical contexts in which they occur. (3) *System evaluation*. Once stable, the finalised template specification and terminology class definitions were used by domain experts to produce manually a set of filled templates/terminology-tagged texts for a corpus sample. These manually produced resources, divided into two disjoint sets, served as the gold standard against which the system-generated results were compared and scored, both during development (Dev) and final blind evaluation (Blind).

The sizes of these annotated subcorpora are summarised in Table 1. To further ensure the distinctness of the development and evaluation process, the test texts were taken from articles in the years 1999 and 2000 whereas the development texts were from the period 1994–98. Some of the Blind texts were independently annotated by two annotators (Int), so that inter-annotator agreement measures could be calculated [§].

## System Architecture

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

[§] The PASTA annotated resources are freely available at: http://www.dcs.shef.ac.uk/nlp/pasta/results.html.

*Text Preprocessing.* Text preprocessing consists of three activities. The first is *section analysis* in which a set of regular expressions is used to identify those sections of a text that are considered relevant for IE and exclude those that are not. This saves processing time and reduces the probability of errors in subsequent modules. For documents whose structure is clearly marked, such as MEDLINE abstracts, sections such as the title of the article, the authors' names, the journal reference and the body of the abstract are marked for further processing.

The second preprocessing activity is *tokenisation*, the segmentation of the text into the character sequence units which form the atoms of further processing. This involves determining, e.g. word, punctuation and number boundaries, and may, in this domain, result in the splitting of a single complex 'word' into its constituents. For example, in the case of `Cys128`, tokenisation yields two tokens—the three-letter residue abbreviation `Cys` and the number `128`.

The final activity in this stage is *sentence splitting*. A rule-based sentence splitter is used to identify sentence-terminating full stops and hence to segment the text into sentences.

*Terminological Processing.* The aim of terminological processing is to identify and correctly classify instances of the term classes described above in the section on **PASTA Extraction Tasks**. Three component modules contribute to this goal. The first is *morphological analysis* in which individual tokens are analysed to see if they contain interesting biochemical affixes such as `-ase` or `-in` that indicate candidate enzyme or protein names. Such evidence is taken as indicative only, and is not treated as conclusive (consider, e.g. `decrease`).

The second stage is *lexical lookup* in terminological lexicons which we have compiled from biological databases such as CATH[¶] and SCOP[‖] and have been augmented with terms produced by statistical corpus processing techniques (Demetriou and Gaizauskas, 2000). The set of lexicons includes approximately 20 000 terms classified into the 12 main categories introduced above; each term may consist of multiple tokens. In addition to the twelve main term categories, additional subcategories are defined for terms that may occur as part of larger multi-token constructions. Lexical lookup is implemented as a series of fast finite-state recognisers which are applied at each token boundary in the text to determine the category or categories of token-sequences starting from that point. Because of lexical ambiguities[**] the lexical lookup process cannot provide definitive term type classification.

The final stage of terminological processing is *terminology parsing* in which a rule-based terminology parser analyses the tokenisation, morphological and lexical properties of component terms and attempts to combine them into a single multi-token unit. For example, in the protein name `casein kinase 1`, the term `kinase` is identified as a `protein_head` term [††] either based on the morphological affix `-ase` or by lookup in the protein lexicon; the term `casein` is categorised as a `protein_modifier` due to its occurrence in non-terminal position in other compound protein terms. A grammar rule such as

```
protein -->  protein_modifier, protein_head,
             numeral.
```

is then used to recognise this multi-word term as a protein.

During parsing, ambiguities at the lexical level are resolved by the longest match principle (for example, *`Drosophila melanogaster`* will be preferred over just *`Drosophila`*). Using such a grammar rule-based approach gives the term processing system generative capability, enabling it to recognise terms not previously seen in a corpus or stored in the lexicons.

*Syntactic and Semantic Processing.* The main aim at this stage is to build a 'semantic' representation of the text on a sentence-by-sentence basis. This is done using the conventional NLP approach of syntactic analysis followed by transduction of the grammatical form into a semantic representation—a predicate-argument representation akin to predicate logic. Syntactic analysis consists of *part-of-speech tagging*, in which grammatical class information is assigned to each token, followed by *phrasal parsing*, in which a general grammar of English is used to derive a phrase structure analysis of each sentence (in terms of noun phrases, verb phrases etc.). In this process previously recognised terms are treated as non-decomposable noun phrases.

Given a phrase structure analysis of a sentence, a predicate argument representation can be derived from this. For example, given the sentence *Ser154, Tyr167 and Lys171 are found at the active site*, syntactic analysis identifies a co-ordinated noun phrase as the logical object of a passive verb phrase modified by a prepositional phrase. Transduction to a predicate argument representation yields the following (simplified) expression:

```
residue(e1), name(e1,"Ser154"),
residue(e2), name(e2,"Tyr167"),
residue(e3), name(e3,"Lys171"),
set(e4), set_member(e4, e1),
set_member(e4,e2), set_member(e4,e3),
find(e5), lobj(e5,e4),
active_site(e6), at(e5,e6)
```

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

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

[**] A particular term may occur in several lexicons or a term which occurs in one lexicon may also occur as part of a longer term occurring in another lexicon.

[††] 'Head' here is used by analogy with the linguistic designation 'head noun' in a noun phrase—usually the rightmost in English.

*Discourse Processing and Template Extraction.* The penultimate processing stage is *discourse processing* in which information from multiple sentences is linked, potentially by making inferences using a limited predefined *domain model*, which consists of a concept hierarchy, or ontology, in which inheritable properties and inference rules may be associated with the concepts. Sentence-by-sentence, semantic representations, as shown above, are integrated into the domain model and a coreference mechanism attempts to merge new with previously added instances. In addition, specific instances of entities that are required to fill the template may be hypothesized and the coreference mechanism will attempt to merge these hypothesized instances with instances from the text. For example, a residue may be assumed to be found in a protein and if a semantic representation mentions a residue but no source protein then a 'dummy' source protein is hypothesized and an attempt made to corefer it with a protein mentioned elsewhere in the text. Such a mechanism is needed (i) to deal with failures of the syntactic/semantic processor to detect explicit semantic relations and (ii) because in many cases semantic relations are implicit, part of the assumed knowledge of the reader, and hence cannot be extracted from the text alone.

Consider these sentences, slightly simplified, from a Medline abstract (sentences 2–4, contain no mention of Endo H).

*S1* *The three-dimensional structure of Endo H has been determined . . .*
*S5* *A shallow curved cleft runs across the surface of the molecule from . . .*
*S6* *This cleft contains the putative catalytic residue Asp130 . . .*

Jointly these sentences allow us to infer that Endo H contains the residue Asp130. PASTA determines this as follows.

From S1, *Endo H* is identified as a protein, say `protein(e1),name(e1,"Endo H")`, and `e1` is added to the discourse model as an instance of the `protein` concept.

From S5, *a . . . cleft* is identified as, say, `cleft(e23)` and *the molecule* as, say, `molecule(e25)`. The domain ontology records that proteins are molecules and coreference resolves the definite reference to `e25` with `e1`. The domain model also records that clefts are regions and that regions are located in proteins. So, a protein, say `e42`, is hypothesized, as is the relation `located_in(e23,e42)`. In the absence of full semantic analysis of *runs across the surface of*, the coreference mechanism picks the closest protein and resolves `e42` with `e1`/`e25`, i.e. the cleft is assumed to be in Endo H.

Finally, from S6, *the cleft* is identified as, say `e52`, and *the . . . residue*, as `e61`. Syntactic and semantic analysis yields that `e52` is the logical subject of a `contain` event whose object is `e61`. The domain model permits the inference that if a region contains a residue then the residue is located in the region, i.e. `located_in(e61,e23)` is inferred—the residue is located in the cleft. Next, coreference resolves `e52` with the preceding discourse referent for a cleft, `e23`. Lastly, the transitivity of `located_in` permits the desired conclusion: `located_in(e61,e1)`, i.e. Asp130 is in EndoH. A template writing module scans the final discourse model for any instances that are relevant to the template filling task, ensures that the minimum requisite slot-fill information is present, and then formats and outputs the templates.

### Accessing PASTA Templates

Templates of the form shown in Figure 1 or terminology-tagged texts are of little direct use to biologists. To explore ways in which PASTA-extracted information could be usefully presented to biologists, we have built a prototype web browser-based interface to the extraction results, accessible at http://www.dcs.shef.ac.uk/nlp/pasta/. This allows a user to browse alphabetically ordered indices of residues, proteins, or species detected in a corpus of Medline abstracts and from them to access the related terminology-tagged texts and extracted templates created by PASTA. Terminology tagged texts are displayed with terms colour-coded by class and hyperlinked to other documents containing the same term, permitting very rapid navigation by terms through a text collection. Templates are displayed in tables which are of two kinds: (i) document-centred tables combining extracted information for all residues, proteins and species mentioned in a single text, and (ii) protein-centred tables combining all extracted information for a single protein from the entire text collection. Further discussion of this interface may be found in Demetriou and Gaizauskas (2002).

## RESULTS

### Evaluation Results

Performance evaluation of the PASTA system was carried out using the manually annotated resources described above in the section on **The PASTA Corpus**. Results for the terminology processing component portion of the system are shown in Table 2 and results for template filling are shown in Table 3. Both tables show results for the development corpus (Development) and the unseen final evaluation corpus (Blind) and in addition they show results of comparing two human annotators (Interannotator) (here one annotator is taken to be the gold standard and the other scored against him). For each of these evaluations the tables show the number of instances of the particular term class or template slot that were to be found and

**Table 2.** Evaluation results for term recognition and classification

| | Development | | | Interannotator | | | Blind | | |
|---|---|---|---|---|---|---|---|---|---|
| Term type | No | Recall | Precision | No | Recall | Precision | No | Recall | Precision |
| Protein | 410 | 87 | 97 | 235 | 93 | 98 | 788 | 81 | 91 |
| Species | 133 | 83 | 97 | 70 | 92 | 94 | 223 | 82 | 83 |
| Residue | 179 | 98 | 93 | 88 | 93 | 98 | 269 | 96 | 90 |
| Site | 87 | 61 | 84 | 36 | 77 | 94 | 133 | 71 | 83 |
| Region | 43 | 44 | 100 | 66 | 70 | 73 | 183 | 77 | 70 |
| Secondary struct. | 79 | 99 | 99 | 27 | 81 | 78 | 60 | 90 | 82 |
| Supersecondary struct. | 84 | 100 | 94 | 6 | 50 | 100 | 23 | 52 | 52 |
| Quaternary struct. | 120 | 96 | 97 | 16 | 65 | 69 | 76 | 83 | 84 |
| Base | 38 | 100 | 97 | 21 | 83 | 90 | 28 | 96 | 71 |
| Atom | 44 | 95 | 81 | 23 | 79 | 96 | 53 | 75 | 71 |
| Nonprotein | 107 | 100 | 84 | 80 | 84 | 88 | 233 | 85 | 81 |
| Interaction | 13 | 77 | 91 | 14 | 67 | 86 | 50 | 62 | 82 |
| ALL | 1337 | 88 | 94 | 682 | 86 | 92 | 2119 | 82 | 84 |

**Table 3.** Evaluation results for template extraction

| | Development | | | Interannotator | | | Blind | | |
|---|---|---|---|---|---|---|---|---|---|
| TEMPLATE slot | No | Recall | Precision | No | Recall | Precision | No | Recall | Precision |
| ARTICLE | | | | | | | | | |
| Title | 25 | 100 | 100 | 10 | 100 | 100 | 30 | 97 | 97 |
| Author | 25 | 100 | 100 | 10 | 100 | 100 | 30 | 100 | 100 |
| Source | 25 | 100 | 100 | 10 | 100 | 100 | 30 | 100 | 100 |
| RESIDUE | | | | | | | | | |
| Name | 91 | 98 | 83 | 30 | 100 | 97 | 102 | 88 | 80 |
| No | 77 | 90 | 83 | 22 | 95 | 91 | 80 | 86 | 93 |
| Secondary_struct. | 34 | 85 | 48 | 2 | 100 | 67 | 9 | 44 | 40 |
| Quaternary_struct. | 19 | 79 | 43 | 17 | 100 | 76 | 37 | 59 | 42 |
| Site | 142 | 55 | 89 | 31 | 74 | 62 | 80 | 63 | 50 |
| Region | 165 | 86 | 58 | 20 | 88 | 67 | 34 | 53 | 46 |
| Interaction | 67 | 78 | 60 | 7 | 46 | 75 | 28 | 46 | 38 |
| PROTEIN | | | | | | | | | |
| Name | 71 | 65 | 61 | 27 | 70 | 52 | 71 | 62 | 56 |
| SPECIES | | | | | | | | | |
| Name | 38 | 89 | 85 | 14 | 69 | 64 | 43 | 74 | 57 |
| IN_PROTEIN | | | | | | | | | |
| Residue | 91 | 92 | 80 | 29 | 83 | 100 | 119 | 76 | 81 |
| Protein | 91 | 63 | 54 | 28 | 69 | 86 | 118 | 53 | 56 |
| IN_SPECIES | | | | | | | | | |
| Protein | 35 | 57 | 49 | 9 | 44 | 78 | 48 | 34 | 43 |
| Species | 35 | 49 | 41 | 9 | 40 | 67 | 47 | 33 | 43 |
| ALL | 1031 | 82 | 68 | 275 | 81 | 77 | 906 | 68 | 65 |

the Recall and Precision rates, as percentages, of the evaluated system or annotator. The final row in each table shows total terms/template slots to be extracted and mean recall and precision figures, averaged over all term/slot occurrences. All results were computed using the MUC scoring software; the algorithm underlying the scoring, which for templates involves a rather complex alignment process, is described in MUC-7 (1998).

## Discussion

The evaluation results indicate that the system is capable of performing at close to, but not quite at, the state-of-the-art for IE systems as obtained at the last open IE system evaluation—MUC-7 (see Table 4). This can be accounted for by differences in the tasks (e.g. 5 NE categories in MUC-7, 12 term classes in PASTA), differences in domains (persons, artifacts, organisations and locations in

**Table 4.** MUC-7 best scores versus PASTA

| Task | MUC-7 Best Rec | Pre | PASTA Rec | Pre |
|---|---|---|---|---|
| NE/Term. | 92 | 95 | 82 | 84 |
| TE | 86 | 87 | 75 | 66 |
| TR | 67 | 86 | 68 | 65 |

MUC-7 versus proteins, residues and species here), poorer task definition/annotation consistency[‡‡], and less resource for system development (MUC-7 best scores are pooled results from more than 15 participants).

Comparing term recognition with template extraction results for the `residue` term class and the `RESIDUE:name` slot, there is obvious correlation both for precision and recall. This is to be expected as these values are extracted mostly from simple expressions such as 'aspartate' or 'asp123'. However, for proteins and species there is a drop, especially in precision. This is due in part to these terms being harder to recognise, but also to template slots being single-valued, i.e. the system must produce a single value from all possible references to the term in the text. Currently, the system returns the longest term equivalent, but this is not always the case for the human generated answers.

With respect to the template extraction scores (Table 3), the interannotator agreement results provide useful clues about the difficulty of some tasks and this is reflected in the Blind evaluation results. For instance, the precision and recall scores for `RESIDUE:name` and `:number` are again quite high but other more complex tasks, such as the extraction of interactions, are more prone to error since the information may be distributed over long distances in the text and be more difficult to capture. Some part of the lower scores for precision is due to peculiarities to the MUC scoring software which requires exact string matching for the 'string fill' type of slots and, in addition, severely penalises multiple response values for a slot when the equivalent key slot contains a single answer. For example, while PASTA produces multiple values for the `SITE`, `REGION`, `SEC_STRUCT`, `QUAT_STRUCT` and `INTERACTION` slots, so as to provide more information about the residue, the annotators preferred to produce only a single description in most cases.

## RELATED WORK

Research on the application of IE techniques to biological texts has increased rapidly in the past few years. This work varies both with respect to the intended *task* and the language processing *methods* adopted. Unfortunately, at this time precise comparative evaluation of existing IE systems developed for the biomedical domain cannot be made, since the tasks and text collections addressed by researchers vary widely.

The identification of biomedical terms has proved to be the easiest extraction task. Techniques used for this task include rule-based methods (Fukuda *et al.*, 1998), statistical methods (Collier *et al.*, 2000) and statistical-rule-based hybrids (Proux *et al.*, 1998). As described above our approach is rule-based, but contrasts with Fukuda *et al.* (1998) in addressing a wider set of term classes and in using lexicons which are first manually created, then extended semi-automatically from corpora using statistical techniques.

More complex IE tasks involving the extraction of relational information have also been addressed by the bioinformatics community. These include protein–protein interactions (Blaschke *et al.*, 1999; Thomas *et al.*, 2000; Park *et al.*, 2001; Yakushiji *et al.*, 2001; Pustejovsky *et al.*, 2002), relations between genes and drugs (Rindflesh *et al.*, 2000) and identification of metabolic pathways (Humphreys *et al.*, 2000; Leroy and Chen, 2002). The approaches employed vary widely and may be characterised by the type of linguistic processing they attempt. PASTA, although it does not aim at full syntactic, semantic and discourse processing, performs deeper natural language analysis than most current systems. We briefly contrast it with some of these systems here.

Perhaps the simplest approach is that of Blaschke *et al.* (1999) who do not aim to extract structured representations, but rather to identify sentences carrying protein–protein interaction information. They suppose a pre-specified set of protein names and interaction-signalling verbs and select sentences which contain these key verbs and proteins within a fixed proximity. Their domain, task and methods are different from PASTA, which aims to extract structured representations of protein structure information and assumes no fixed set of proteins.

Next in terms of complexity of language processing are systems which use some syntactic analysis, e.g. part-of-speech (POS) tagging and shallow parsing, on a sentence-by-sentence basis to identify entities of interest, features of these entities and, possibly, relations between these entities. The EDGAR system (Rindflesh *et al.*, 2000) uses stochastic POS tagging and 'underspecified parsing' to identify phrasal chunks (e.g. NPs and PPs). Gene and cell names are identified in two passes using local context rules triggered by, e.g. NPs with *gene* or *cell* as headword, and lookup in lexical resources. Various cell features, such as organ or cancer type, are then extracted using contextual rules triggered by signal words and semantic type checking in UMLS; no relational information is currently extracted.

[‡‡] Interannotator scores below 80% in MUC were taken as indication that further task refinement was required.

Slightly richer are systems which use the same general approach but also identify key verbs (e.g. those suggesting interaction) and their arguments (e.g. subject and object) using manually encoded rules or heuristics and then map these arguments into a normalized relational structure or template. Examples are Thomas *et al.* (2000) who have developed mapping rules for about 30 verbs in the domain of protein-protein interaction and Sekimizu *et al.* (1998) who have modelled eight verbs heavily used in describing gene–gene product interactions (e.g. *activate, bind, regulate, encode*). Both of these systems use shallow parsing techniques. However, both less sophisticated approaches, such as Leroy and Chen (2002), who use heuristics triggered by prepositions to extract strings in the local context into templates, and more sophisticated approaches, such as Park *et al.* (2001), who use a powerful combinatory categorial grammar parser, and Yakushiji *et al.* (2001), who employ principled XHPSG parsing, have also been explored. PASTA differs from these systems in two key respects. First it defers template filling until after discourse processing so that information from multiple sentences may merged into the extracted templates; secondly, PASTA employs a limited domain model that supports restricted reasoning about entities and relations in the domain, as part of the process of sentence and discourse interpretation.

The most linguistically complex systems attempt to relate information across sentence boundaries. So far only a handful of systems try this, although most authors recognise that this capability is essential (cf. the example above in the seciton on **Discourse Processing**). Pustejovsky *et al.* (2002) employ POS tagging with a UMLS-enhanced lexicon and shallow parsing, followed by relation extraction using semantic automata developed for particular verbal and nominal forms (they concentrate on *inhibit* in the paper). Following sentence analysis, a frame of discourse referents is built and an anaphora resolution module searches backwards through preceding frames, attempting to resolve sortal and pronominal anaphors. In the MEDSYNDICATE system (Hahn *et al.*, 2002) sentence level processing involves parsing into a dependency graph from which a semantic interpretation is produced using conceptual knowledge of the domain (a description logic representation of UMLS) and semantic knowledge which constrains how the dependency graph may be interpreted given the conceptual model. Text level understanding involves tracking reference relations across sentences using a 'center list'. The authors apply their system to a corpus of German histopathology reports. With regard to both these systems PASTA differs in the domain addressed and fine details of the technologies and resources employed (e.g. PASTA works in a grammatical framework of phrase structure rather than dependency analysis; PASTA uses application-tailored domain models rather than generic models such as UMLS, which while increasing coverage can also increase noise.

Given the range of tasks and text resources which biomedical IE systems have addressed, comparing quantitative results can only be vaguely indicative of relative merit. Nonetheless, PASTA results compare favourably with most other IE systems in the biomedical domain. 'Best' recall/precision figures reported for terminology recognition are: 99/95% Fukuda *et al.* (1998) for protein names only, 73% (combined score) (Collier *et al.*, 2000) for ten term classes, 94/91% Proux *et al.* (1998) for gene names only. Best figures reported for template extraction are: 58/77% (Thomas *et al.*, 2000), 48/80% (Park *et al.*, 2001), 47/70% (Leroy and Chen, 2002) and 59/90% (Pustejovsky *et al.*, 2002), although the latter was based on the extraction of *inhibit* relations only. Perhaps, a pleasing feature of PASTA's performance is that the discrepancy between precision and recall is quite small, indicating system maturity and tolerance to unseen data.

## CONCLUSIONS

We have described an advanced text processing system, PASTA, that analyses Medline abstracts, identifies occurrences of terms in 12 term classes, and extracts information about the role of residues in proteins from the abstracts. The system has been developed and evaluated against a corpus of Medline abstracts, representative of a wide range of literature relevant to the study of protein structure. A browser-based interface has been constructed that allows the extracted information to be directly accessed by a biologist as a natural extension to browsing a collection of abstracts.

We believe that PASTA demonstrates that IE from biological texts is feasible. Future work must concentrate on improving the accuracy of extracted information, facilitating the adaptation of systems to new domains, and investigating ways to refine the delivery mechanism to biologists, so as to best support them in their research.

## ACKNOWLEDGEMENTS

This work was supported by the UK BBSRC/EPRSC BioInformatics Initiative research grant 50/BIF08754. The authors would like to thank Ruth Spriggs and Darren South for their expert assistance in tagging terminology and filling templates.

## REFERENCES

Baeza-Yates,R. and Ribeiro-Neto,B. (1999) *Modern Information Retrieval*. ACM Press Books.

Berman,H., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T., Weissig,H., Shindyalov,I. and Bourne,P. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Blaschke,C., Andrade,M.A., Ouzonis,C. and Valencia,A. (1999) Automatic extraction of biological information from scientific text: protein-protein interactions. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*. pp. 60–67.

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

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

Demetriou,G. and Gaizauskas,R. (2000) Automatically augmenting terminological lexicons from untagged text. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000)*. pp. 861–867.

Demetriou,G. and Gaizauskas,R. (2002) Utilizing text mining results: The PastaWeb system. In *Proceedings of the Association for Computational Linguistics Workshop on Natural Language Processing in the Biomedical Domain*. pp. 77–84.

Fukuda,K., Tsunoda,T., Tamura,A. and Takagi,T. (1998) Information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing 1998*. pp. 707–718.

Gaizauskas,R. and Wilks,Y. (1998) Information Extraction: Beyond Document Retrieval. *J. Documentation*, **54**, 70–105.

Hahn,U., Romacker,M. and Schulz,S. (2002) Creating knowledge repositories from biomedical reports: The MEDSYNDIKATE text mining system. In *Proceedings of the Pacific Symposium on Biocomputing 2002*. pp. 338–349.

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

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

Leroy,G. and Chen,H. (2002) Filling preposition-based templates to capture information from medical abstracts. In *Proceedings of the Pacific Symposium on Biocomputing 2002*. pp. 350–361.

MUC-7 (1998) *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Defense Advanced Research Projects Agency. Available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Park,J.C., Kim,H.S. and Kim,J.J. (12001) Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. In *Proceedings of the Pacific Symposium on Biocomputing 2001*. pp. 529–540.

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

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

Rindflesh,T., Tanabe,L., Weinstein,J. and Hunter,L. (2000) Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proceedings of the Pacific Symposium on Biocomputing 2000*. pp. 517–528.

Sekimizu,T., Park,H. and Tsujii,J. (1998) Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. In *Proceedings of the 9th Workshop on Genome Informatics*. pp. 62–71.

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

Yakushiji,A., Tateisi,Y., Miyao,Y, and Tsujii,J. (2001) Event extraction from biomedical papers using a full parser. In *Proceedings of the Pacific Symposium on Biocomputing 2001*. pp. 529–540.