Answering questions in the genomics domain

Rinaldi, Fabio ; Dowdall, J ; Schneider, G

Answering Questions in the Genomics Domain

Fabio Rinaldi, James Dowdall, Gerold Schneider
Institute of Computational Linguistics,
University of Zurich, CH-8057 Zurich
Switzerland
{rinaldi, dowdall, gschneid}@cl.unizh.ch

Andreas Persidis
Biovista, 34 Rodopoleos Str.,
Ellinikon, GR-16777 Athens,
Greece
andreasp@biovista.com

Abstract

In this paper we describe current efforts aimed at adapting an existing Question Answering system to a new document set, namely research papers in the genomics domain. The system has been originally developed for another restricted domain, however it has already proved its portability. Nevertheless, the process is not painless, and the specific purpose of this paper is to describe the problems encountered.

1 Introduction

One of the core problems in exploiting scientific papers in research and clinical settings is that the knowledge that they contain is not easily accessible. Although various resources which attempt to consolidate such knowledge are being created (e.g. UMLS[1], SWISS-PROT, OMIM, GeneOntology, GenBank, LocusLink), the amount of information available keeps growing exponentially (Stapley and Benoit, 2000).

There is accordingly a pressing need for intelligent systems capable of accessing that information in an efficient and user-friendly way. Question Answering systems aim at providing a focused way to access the information contained in a document collection. Specific research in the area of Question Answering has been prompted in the last few years in particular by the Question Answering track of the Text REtrieval Conference (TREC-QA) competitions (Voorhees, 2001). The TREC-QA competitions focus on open-domain systems, i.e. systems that can (potentially) answer any generic question. As these competitions are based on large volumes of text, the competing systems (normally) resort to a relatively shallow text analysis.[2] In contrast a question answering system working on a restricted domain can take advantage of the formatting and style conventions in the text, can make use of the specific domain-dependent terminology, and of full parsing.

In many restricted domains, including technical documentation and research papers, terminology plays a pivotal role. This is in fact one of the major differences between restricted domains and open domain texts. While in open domain systems Named Entities play a major role, in technical documentation, as well as in research papers, they have a secondary role, by contrast a far greater role is played by domain terminology. Terminology is a major obstacle for processing research papers and at the same time a key access path to the knowledge encoded in those papers. Terminology provides the means to name and access domain-specific concepts and objects.

Restricted domains present the additional problem of "domain navigation". Users of the system cannot always be expected to be completely familiar with the domain terminology. Unfamiliarity with domain terminology might lead to questions which contain imperfect formulations of domain terms. It becomes therefore essential to be able to detect terminological variants and exploit the relations between terms (like synonymy, meronymy, antonymy). The process of variation is well investigated in terminological research (Daille et al., 1996). In the Biomedical domain, an example of a system that deals with terminological variants (also called "aliases") can be found in (Pustejovsky et al., 2002).

In the rest of this paper we will first briefly describe our existing Question Answering system, ExtrAns (section 2). In the following section (3) we detail the specific problems encountered in the new domain and the steps that we have taken to solve them. We conclude the paper with an overview of related research (section 4).

[1] http://www.nlm.nih.gov/research/umls/

[2] With some notable exception, e.g. (Harabagiu et al., 2001).

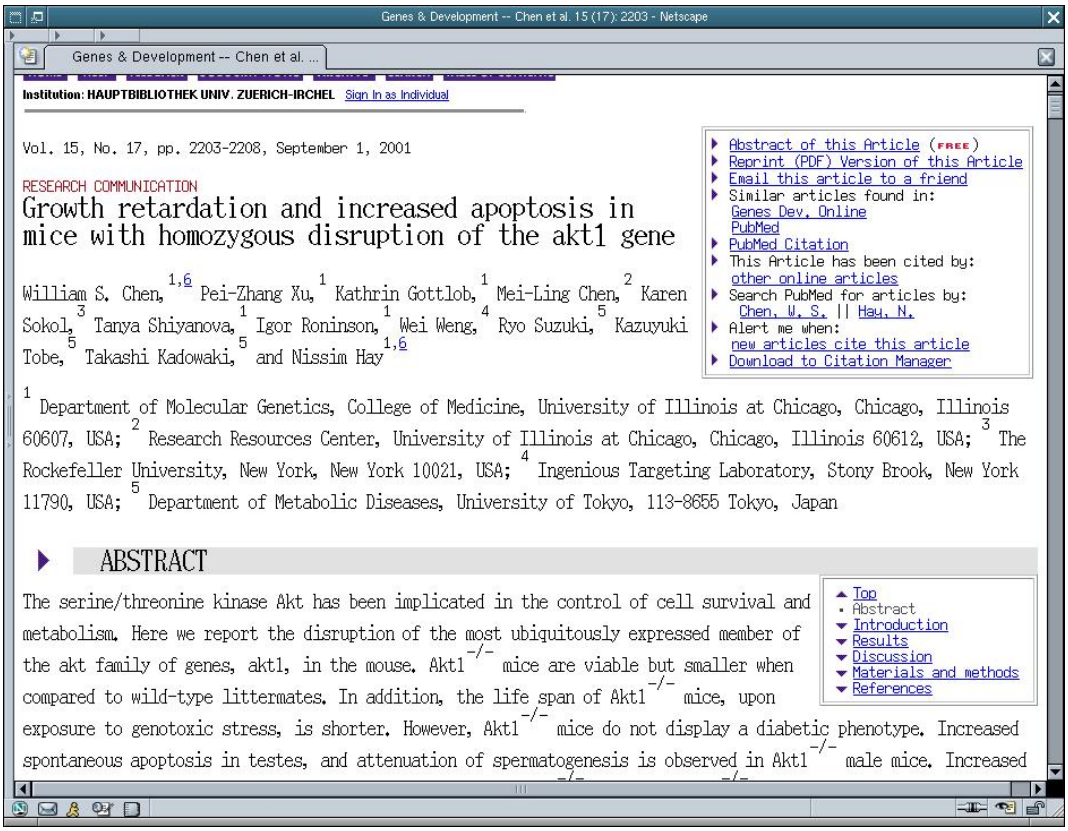

Figure 1: Example of document to be analyzed

2 The original Question Answering system

ExtrAns is a Question Answering system aimed at restricted domains, in particular terminology-rich domains. While open domain Question Answering systems typically are targeted at large text collections and use relatively little linguistic information, ExtrAns answers questions over such domains by exploiting linguistic knowledge from the documents and terminological knowledge about a specific domain. Various applications of the ExtrAns system have been developed, from the original prototype aimed at the Unix documentation files (Mollá et al., 2000) to a version targeting the Aircraft Maintenance Manuals (AMM) of the Airbus A320 (Mollá et al., 2003; Rinaldi et al., 2004). In the present paper we describe current work in applying the system to a different domain and text type: research papers in the genomics area.

Our approach to Question Answering is particularly computationally intensive; this allows a deeper linguistic analysis to be performed, at the cost of higher processing time. The documents are analyzed in an off-line stage and transformed in a semantic representation (called 'Minimal Logical Forms' or MLFs), which is stored in a Knowledge Base (KB). In an on-line phase (see fig. 2) the user queries are analyzed using the same basic machinery (however the cost of processing them is negligible, so that there is no visible delay) and their semantic representation is matched in the KB. If a match is encountered, the sentences that gave origin to the match are presented as possible answer to the question.

Documents (and queries) are first tokenized, then they go through a terminology-processing module. If a term belonging to a synset in the terminological knowledge base is detected, then the term is replaced by a synset identifier in the logical form. This results in a canonical form, where the synset identifier denotes the concept that each of the terms in the synset names. In this way any term contained in a user query is automatically mapped to all its variants. This approach amounts to an implicit 'terminological normalization' for the domain, where the synset identifier can be taken as a reference to

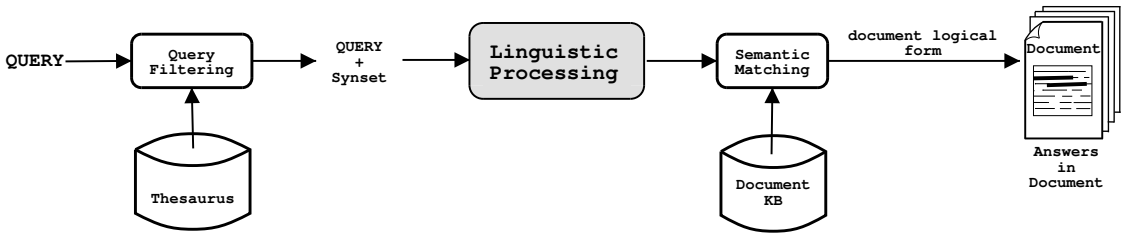

Figure 2: Schematic representation of the core QA engine

the 'concept' that each of the terms in the synset describes (Kageura, 2002).

ExtrAns depends heavily on its use of logical forms, which are designed so that they are easy to build and to use, yet expressive enough for the task at hand (Mollá, 2001). The logical forms and associated semantic interpretation methods are designed to cope with problematic sentences, which include very long sentences, even sentences with spelling mistakes, and structures that are not recognized by the syntactic analyzer. An advantage of ExtrAns' Minimal Logical Forms (MLFs) is that they can be produced with minimal domain knowledge. This makes our technology easily portable to different domains. The only true impact of the domain is during the preprocessing stage of the input text and during the creation of a thesaurus that reflects the specific terms used in the chosen domain, their lexical relations and their word senses.

Unlike sentences in documents, user queries are processed on-line and the resulting MLFs are proved by deduction over the MLFs of document sentences stored in the KB. When no direct answer for a user query can be found, the system is able to relax the proof criteria in a stepwise manner. First, hyponyms are added to the query terms. This makes the query more general but maintains its logical correctness. If no answers can be found or the user determines that they are not good answers, the system will attempt approximate matching, in which the sentence that has the highest overlap of predicates with the query is retrieved. The matching sentences are scored and the best matches are returned.

The MLFs contain pointers to the original text which allow ExtrAns to identify and highlight those words in the retrieved sentence that contribute most to a particular answer. An example of the output of ExtrAns can be seen in fig. 3. When the user clicks on one of the answers provided, the corresponding document will be displayed with the relevant passages highlighted. Another click displays the answer in the context of the document and allows the user to verify the justification of the answer.

3 Moving to the new domain

The first step in adapting the system to a new domain is identifying the specific set of documents to be analyzed. We have experimented with two different collections in the genomics domain. The first collection (here called the 'Biovista' corpus) has been generated from Medline using two seed term lists of genes and pathways (biological process) to extract an initial corpus of research papers (full articles). The second collection is constituted by the GENIA corpus (Kim et al., 2003)[3], which contains 2000 abstracts from Medline (a total of 18546 sentences). The advantage of the latter is that domain-specific terminology is already manually annotated. However focusing only on that case would mean disregarding a number of real-world problems (in particular terminology detection).

3.1 Formatting information

An XML based filtering tool has been used to select zones of the documents that need to be processed in a specific fashion. Consider for instance the case of bibliography. The initial structure of the document allows to identify easily each bibliographical item. Isolating the authors, titles and publication information is then trivial (because it follows a regular structure). The name of the authors (together with the html cross-references) can then be used to identify the citations within the main body of the paper. If a preliminary zone identification (as described) is not performed, the names of the authors used in the citations would appear as spurious elements within sentences, making their analysis very difficult.

Another common case is that of titles. Normally they are Nominal Phrases rather than sentences. If

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

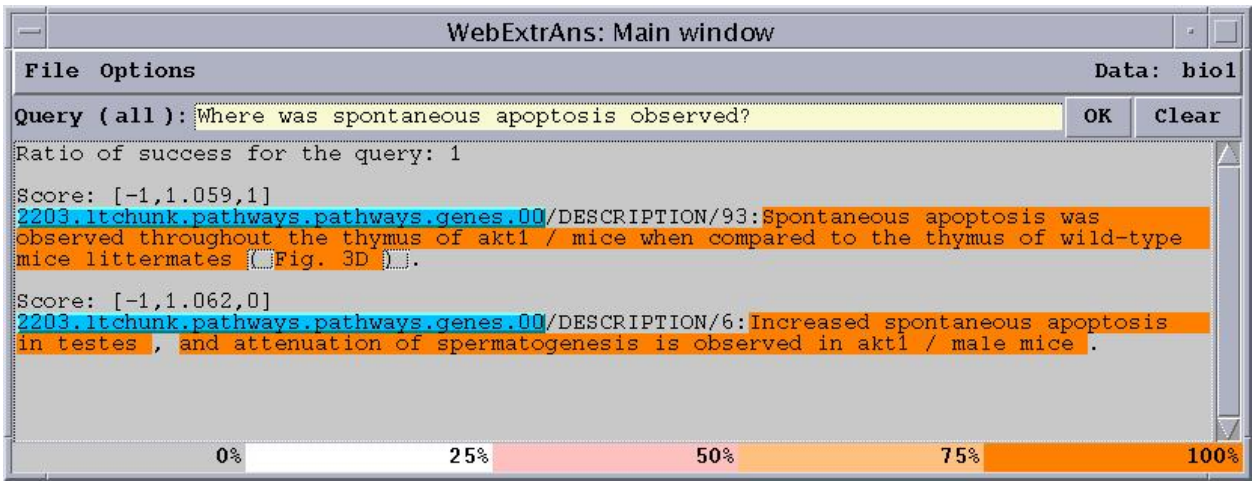

Figure 3: Example of interaction with the system

the parser was expecting to find a sentence it would fail. However using the knowledge that a title is being processed, we can modify the configuration of the parser so that it accepts an NP as a correct parse.

3.2 Terminology

The high frequency of terminology in technical text produces various problems when locating answers. A primary problem is the increased difficulty of parsing text in a technical domain due to domain-specific sublanguage. Various types of multi-word terms characterize these domains, in particular referring to specific concepts (e.g. genome sequences, proteins). These multi-word expressions might include lexical items which are either unknown to a generic lexicon (e.g. "*argentine methylation*"), have a specific meaning unique to this domain or deverbal adjectives (and nouns) are often mistagged as verbs (e.g. "*mediated activation*", "*cell killing*"). Abbreviations and acronyms, often complex (e.g. bracketed inside NPs, like "*adenovirus (ad) infection*") are another common source of inconsistencies. In such cases the parser might either fail to identify the compound as a phrase and consequently fail to parse the sentence including such items. Alternatively a parser might attempt to 'guess' their lexical category (in the set of open class categories), leading to an exponential growth of the number of possible syntactic parses and often incorrect decisions. Not only the internal structure of the compound can be multi-way ambiguous, also the boundaries of the compounds are difficult to detect and the parsers may try odd combinations of the tokens belonging to the compounds with neighboring tokens.

We have described in (Rinaldi et al., 2002) some approaches that might be taken towards terminology extraction for a specific domain. The GENIA corpus removes these problems completely by providing pre-annotated terminological units. This allows attention to be focused on other challenges of the QA task, rather than getting 'bogged down' with terminology extraction and organization.

In the case of the Biovista corpus, we had to perform a phase of terminology discovery, which was facilitated by the existence of the seed lists of genes and pathways. We first marked up those terms which appear in the corpus using additional xml tags. This identified 900 genes and 218 pathways that occur in the corpus - represented as boxed tokens in fig. 4. Next the entire corpus is chunked into nominal and verbal chunks using LT Chunk (Finch and Mikheev, 1997). Ignoring prepositions and gerunds the chunks are a minimal phrasal group - represented as the square braces in fig. 4. The corpus terms are then expanded to the boundary of the phrasal chunk they appear in. For example, NP3 in fig. 4 contains two terms of interest producing the new term "*IFN-induced transcription*". The 1118 corpus terms were expanded into 6697 new candidate terms. 1060 involve a pathway in head position and 1154 a gene. The remaining 4483 candidate terms involve a novel head with at least one gene or pathway as a modifier.

Once the terminology is available, it is necessary to detect relations among terms in order to exploit

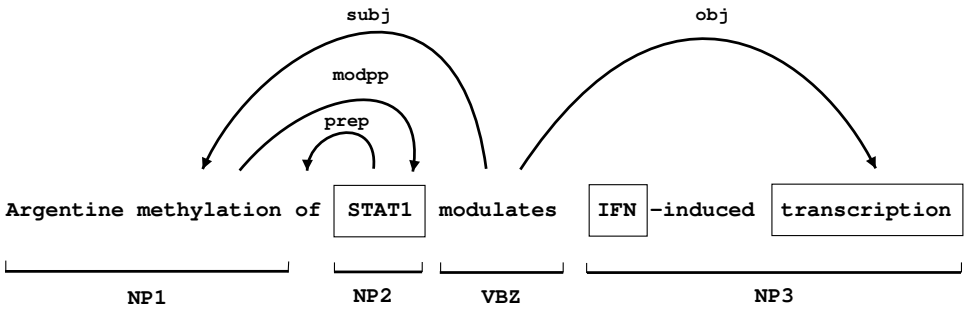

Figure 4: An example of syntactic analysis

it. We have focused our attention in particular to the relations of synonymy and hyponymy, which are detected as described in (Dowdall et al., 2003) and gathered in a Thesaurus. The organizing unit is the WordNet style synset which includes strict synonymy as well as three weaker synonymy relations. These sets are further organized into a isa hierarchy based on two definitions of hyponymy.

One of the most serious problems that we have encountered in working in restricted domains is the syntactic ambiguity generated by multi-word units, in particular technical terms. Any generic parser, unless developed specifically for the domain at hand, will have serious problems dealing with those multi-words. The solution that we have adopted is to parse multi-word terms as single syntactic units. The tokenizer detects the terms (previously collected in the Thesaurus) as they appear in the input stream, and packs them into single lexical tokens prior to syntactical analysis, assigning them the syntactic properties of their head word. In previous work this approach has proved to be particularly effective, bringing a reduction in the complexity of parsing of 46% (Rinaldi et al., 2002).

3.3 Parsing

The deep syntactic analysis builds upon the chunks to identify sentence level syntactic relations between the heads of the chunks. The output is a hierarchical structure of syntactic relations - functional dependency structures - represented as the directed arrows in fig. 4. The parser (*Pro3Gres*) uses hand-written declarative rules to encode acknowledged facts, such as verbs typically take one but never two subjects, combined with a statistical language model that calculates lexicalized attachment probabilities, similar to (Collins, 1999). Parsing is seen as a decision process, the probability of a total parse is the product of probabilities of the individual decisions at each ambiguous point in the derivation.

Probabilistic parsers generally have the advantage that they are fast and robust, and that they resolve syntactic ambiguities with high accuracy. Both of these points are prerequisites for a statistical analysis that is feasible over large amounts of text and beneficial to the Q&A system's performance.

In comparison to shallow processing methods, parsing has the advantage that relations spanning long stretches of text can still be recognized, and that the parsing context largely contributes to the disambiguation. In comparison to deep linguistic, formal grammar-based parsers, however, the output of probabilistic parsers is relatively shallow, pure context-free grammar (CFG) constituency output, tree structures that do not include grammatical function annotation nor co-indexation and empty nodes annotation expressing long-distance dependencies (LDD). In a simple example sentence "*John wants to leave*", a deep-linguistic syntactic analysis expresses the identity of the explicit matrix clause subject and implicit subordinate clause subject by means of co-indexing the explicit and the empty implicit subject trace t: "*[John$_1$ wants [t_1 to leave]]*". A parser that fails to recognize these implicit subjects, so-called control subjects, misses very important information, quantitatively about 3 % of all subjects.

Although LDD annotation is actually provided in Treebanks such as the Penn Treebank (Marcus et al., 1993) over which they are typically trained, most probabilistic parsers largely or fully ignore this information. This means that the extraction of LDDs and the mapping to shallow semantic representations such as MLF is not always possible, because first co-indexation information is not available, second a single parsing error across a tree fragment containing an LDD makes its extraction impossible, third some syntactic relations cannot be recovered

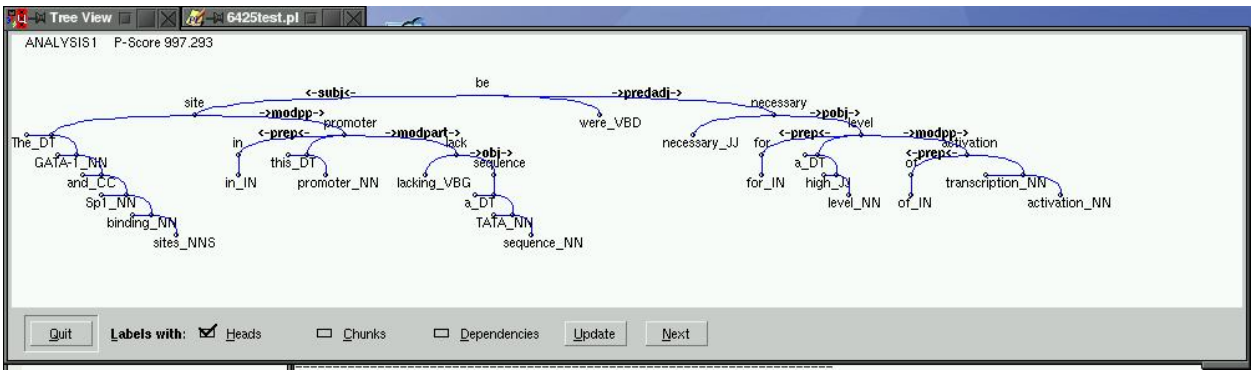

Figure 5: Dependency Tree output of the SWI Prolog graphical implementation of the parser

on configurational grounds only.

We therefore adapt ExtrAns to use a new statistical broad-coverage parser that is as fast as a probabilistic parser but more deep-linguistic because it delivers grammatical relation structures which are closer to predicate-argument structures and shallow semantic structures like MLF, and more informative if non-local dependencies are involved (Schneider, 2003). It has been evaluated and shown to have state-of-the-art performance.

The parser expresses distinctions that are especially important for a predicate-argument based shallow semantic representation, as far as they are expressed in the Penn Treebank training data, such as PP-attachment, most LDDs, relative clause anaphora, participles, gerunds, and the argument/adjunct distinction for NPs.

In some cases functional relations distinctions that are not expressed in the Penn Treebank are made. Commas are e.g. disambiguated between apposition and conjunction, or the Penn tag *IN* is disambiguated between preposition and subordinating conjunction. Other distinctions that are less relevant or not clearly expressed in the Treebank are left underspecified, such as the distinction between PP arguments and adjuncts, or a number of types of subordinate clauses. The parser is robust in that it returns the most promising set of partial structures when it fails to find a complete parse for a sentence. For sentences syntactically more complex than this illustrative example, as many hierarchical relations are returned as possible. A screenshot of its graphical interface can be seen in fig. 5. Its parsing speed is about 300,000 words per hour.

Fig. 4 displays the three levels of analysis that are performed on a simple sentence. Term expansion yields `NP3` as a complete candidate term. However, `NP1` and `NP2` form two distinct, fully expanded noun phrase chunks. Their formation into a noun phrase with an embedded prepositional phrase is recovered from the parser's syntactic relations giving the maximally projected noun phrase involving a term: "*Argentine methylation of STAT1*" (or juxtaposed "*STAT1 Argentine methylation*"). Finally, the highest level syntactic relations (`subj` and `obj`) identifies a transitive predicate relation between these two candidate terms.

3.4 MLFs

The deep-linguistic dependency based parser partly simplifies the construction of MLF. First, the mapping between labeled dependencies and a surface semantic representation is often more direct than across a complex constituency subtree (Schneider, 2003), and often more accurate (Johnson, 2002). Dedicated labels can directly express complex relations, the lexical participants needed for the construction are more locally available.

Let us look at the example sentence *"Adenovirus infection and transfection were used to model changes in susceptibility to cell killing caused by E1A expression"*. The control relation (*infection* is the implicit subject of *model*) and the PP relation (including the description noun) are available locally. The reduced relative clause *killing caused by* is expressed by a local dedicated label (modpart). Only the conjunction *infection and transfection*, expressed here by bracketing, needs to be searched across the syntactic hierarchy.

This leads to the following MLFs:

```
object(infection, o1, [o1]).
object(transfection, o2, [o2]).
object(change, o3, [o3]).
```

```
object(susceptibility, o4, [o4]).
object(killing, o5, [o5]).
object(expression, o6, [o6]).
object(cell, o7, [o7]).
evt(cause, e3, [o6]).
evt(model, e1, [(o1,o2), o3]).
evt(use, e2, [(o1,o2), e1]).
by(e3, o6).
in(o5, o7).
to(o4, o5).
in(o3, o4).
```

4 Related Work

Question Answering in Biomedicine is surveyed in detail in (Zweigenbaum, 2003), in particular regarding clinical questions. An example of a system applied to such questions is presented in (Niu et al., 2003), where it is applied in a setting for Evidence-Based Medicine. This system identifies specific 'roles' within the document sentences and the questions, determining the answers is then a matter of comparing the roles in each. To this aim, natural language questions are translated into the PICO format (Sackett et al., 2000).

Automatic knowledge extraction (or strategies for improving these methods) over Medline articles are numerous. For example, (Craven and Kumlien, 1999) identifies possible drug-interaction relations (predicates) between proteins and chemicals using a 'bag of words' approach applied to the sentence level. This produces inferences of the type: drug-interactions (protein, pharmacologic-agent) where an agent has been reported to interact with a protein.

(Sekimizu et al., 1998) uses frequently occurring predicates and identifies the subject and object arguments in the predication, in contrast (Rindflesch et al., 2000) uses named entity recognition techniques to identify drugs and genes, then identifies the predicates which connect them. This type of 'object-relation-object' inference may also be implied (Cimino and Barnet, 1993). This method uses 'if then' rules to extract semantic relationships between the medical entities depending on which MeSH headings these entities appear under. For example, if a citation has "Electrocardiography" with the subheading "Methods" and has "Myocardial Infarction" with the subheading "Diagnosis" then "Electrocardiography" diagnoses "Myocardial Infarction".

(Spasić et al., 2003) uses domain-relevant verbs to improve on terminology extraction. The co-occurrence in sentences of selected verbs and candidate terms reinforces their termhood. But where such linguistic inferences are stored in a KB as facts, statistical inferences are only used to visualize possible relations between objects for further investigation. (Stapley and Benoit, 2000) measures statistical gene name co-occurrence and graphically displays the results for an expert to investigate the dominant patterns. The PubMed[4] system uses the UMLS to relate metathesaurus concepts against a controlled vocabulary used to index the abstracts. This allows efficient retrieval of abstracts from medical journals, but it makes use of hyponymy and lexical synonymy to organize the terms. It collects terminologies from differing sub-domains in a metathesaurus of concepts.

All such inferences (especially statistical) need to be verified by an expert to ensure their validity. Syntactic parsing, if any, is reserved to shallow NP identifying strategies (Sekimizu et al., 1998), or possibly supplemented with PP information (Rindflesch et al., 2000). Semantic interpretation of the documents is only attempted through their MeSH headings (Mendonca and Cimino, 1999).

5 Conclusion

This paper documents our approach towards QA in the genomics domain. Although some aspects of the work described in this paper are still experimental, we think that the description of the problems that we have encountered and the specific solutions adopted or planned will provide an interesting contribution to the workshop. We conclude by observing that Question Answering is currently seen as an "advanced" topic in the Genomics Track of TREC[5], due to be targeted for the first time in Year 2 (2005).

Acknowledgments

The authors wish to thank the organizers of the workshop and the anonymous reviewers for their helpful comments and suggestions.

References

J.J. Cimino and G.O. Barnet. 1993. Automatic Knowledge Acquisition from Medline. *Methods of Information in Medicine*, 32(2):120–130.

Michael Collins. 1999. *Head-Statistical Models for Natural Language Processing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, USA.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge bases by extracting information from

[4] http://www.ncbi.nlm.nih.gov/pubmed/

[5] http://medir.ohsu.edu/~genomics/roadmap.html

text sources. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*.

B. Daille, B. Habert, C. Jacquemin, and J. Royauté. 1996. Empirical observation of term variations and principles for their description. *Terminology*, 3(2):197–258.

James Dowdall, Fabio Rinaldi, Fidelia Ibekwe-Sanjuan, and Eric Sanjuan. 2003. Complex Structuring of Term Variants for Question Answering. In *Proc. of the ACL 03, Workshop on Multiword Expression: Analysis, Acquisition and Treatment*, Sapporo, Japan, July.

Steve Finch and Andrei Mikheev. 1997. A Workbench for Finding Structure in Texts. In *Proceedings of Applied Natural Language Processing*, Washington, DC, April.

Sanda Harabagiu, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Gîrju, Vasile Rus, and Paul Morarescu. 2001. FALCON: Boosting knowledge for answer engines. In Ellen M. Voorhees and Donna Harman, editors, *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Meeting of the ACL*, University of Pennsylvania, Philadelphia.

Kyo Kageura. 2002. *The Dynamics of Terminology, A descriptive theory of term formation and terminological growth*. Terminology and Lexicography, Research and Practice. John Benjamins Publishing.

J.D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus - a semantically annotated corpus for biotextmining. *Bioinformatics*, 19(1):180–182.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313–330.

E. A. Mendonca and J. J. Cimino. 1999. Automated Knowledge Extraction from Medline Citations. *Medical Informatics*.

Diego Mollá, Rolf Schwitter, Michael Hess, and Rachel Fournier. 2000. ExtrAns, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*, pages 495–522.

Diego Mollá, Fabio Rinaldi, Rolf Schwitter, James Dowdall, and Michael Hess. 2003. Answer Extraction from Technical Texts. IEEE Intelligent Systems.

Diego Mollá. 2001. Ontologically promiscuous flat logical forms for NLP. In Harry Bunt, Ielka van der Sluis, and Elias Thijsse, editors, *Proceedings of IWCS-4*, pages 249–265. Tilburg University.

Yun Niu, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. Answering clinical questions with role identification. In Sophia Ananiadou and Jun'ichi Tsujii, editors, *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 73–80.

J. Pustejovsky, J. Castaño, R. Saurí, A. Rumshisky, J. Zhang, and W. Luo. 2002. Medstract: Creating Large-scale Information Servers for Biomedical Libraries. In *ACL 2002 Workshop on Natural Language Processing in the Biomedical Domain. Philadelphia, PA*. Available at `http://www.medstract.org/publications.html`.

Fabio Rinaldi, James Dowdall, Michael Hess, Kaarel Kaljurand, Mare Koit, Kadri Vider, and Neeme Kahusk. 2002. Terminology as Knowledge in Answer Extraction. In *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE02)*, pages 107–113, Nancy, 28–30 August.

Fabio Rinaldi, Michael Hess, James Dowdall, Diego Mollá, and Rolf Schwitter. 2004. Question answering in terminology-rich technical domains. In Mark Maybury, editor, *New Directions in Question Answering*. AAAI Press.

T.C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter. 2000. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing*, pages 514–25.

D. L. Sackett, S. E. Straus, W. S. Richardson, W. Rosenberg, and R. B. Haynes. 2000. *Evidence Based Medicine: How to Practice and Teach EBM*. Churchill Livingstone.

Gerold Schneider. 2003. Extracting and Using Trace-Free Functional Dependencies from the Penn Treebank to Reduce Parsing Complexity. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Växjö, Sweden, November 14-15.

T. Sekimizu, H. Park, and J Tsujii. 1998. Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. *Genome Informatics, Universal Academy Press.*

Irena Spasić, Goran Nenadić, and Sophia Ananiadou. 2003. Using domain-specific verbs for term classification. In Sophia Ananiadou and Jun'ichi Tsujii, editors, *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 17–24.

B.J. Stapley and G. Benoit. 2000. Bibliometrics: information retrieval and visualization from co-occurrence of gene names in medline abstracts. In *Proceedings of the Pacific Symposium on Biocomputing (Oahu, Hawaii)*, pages 529–540.

Ellen M. Voorhees. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.

Pierre Zweigenbaum. 2003. Question answering in biomedicine. In *Proc. of EACL 03 Workshop: Natural Language Processing for Question Answering*, Budapest.