GoPubMed: ontology-based literature search applied to Gene Ontology and PubMed

Ralph Delfs[1], Andreas Doms[1], Alexander Kozlenkov[2], and Michael Schroeder[1]

[1] Biotec/Dept. of Computing, TU Dresden, Germany, ms@mpi-cbg.de
[2] Dept. of Computing, City University, London, UK

Abstract. The biomedical literature grows at a tremendous rate, so that finding the relevant literature is becoming more and more difficult. To address this problem we introduce ontology-based literature search, which structures search results thorugh the categories of an ontology. We develop and implement GoPubMed, which submits keywords to PubMed, extracts GeneOntology-terms from the retrieved abstracts, and presents the relevant sub-ontology for browsing. For GoPubMed we develop a novel term extraction algorithm and evaluate its performance. GoPubMed is available at www.gopubmed.org
Keywords: Text mining, term extraction, ontologies

1 Introduction

The biomedical literature grows at a tremendous rate and PubMed comprises over 14.000.000 abstracts. Text mining of biomedical literature aims to manage this data avalange [2]. There have been a number of approaches using literature databases such as PubMed to extract relationships such as protein interactions [1], [13], pathways [4], and micro array data [12]. Mostly, these approaches aim to improve literature search by going beyond mere keyword search by providing natural language processing capabilities. While these approaches are successful in their remit, they do not mimic human information foraging, so that finding relevant literature is still an important problem.

Current approaches based on keyword searches and results presented as lists have two shortcomings: (a) in order to choose the keywords leading to good results users need to understand the domain very well; (b) list presentations are linear and thus do not capture multiple views of the results. Furthermore, list-based presentation encourages users to pursue only a few top hits, while more relevant papers may remain unexplored.

We investigate how keyword search can be enhanced through the use of the GeneOntology (GO) to structure large amounts of relevant literature. Our main contribution is the introduction and realisation of ontology-based literature search by realising GoPubMed, which submits keywords to PubMed, extracts GO-terms from the retrieved abstracts, and presents the relevant sub-ontology for browsing. To implement this system, we analyse the structure and content of GeneOntology and derive a novel term extraction algorithm. From the extracted terms, we derive an induced ontology, which is used by the user to explore the PubMed search results. In a sample of 500.000 PubMed abstracts our algorithm detects 29% of all GO-terms in the Gene Ontology, which is in line with other approaches. Our algorithm finds GO-terms in 56% of the abstracts,

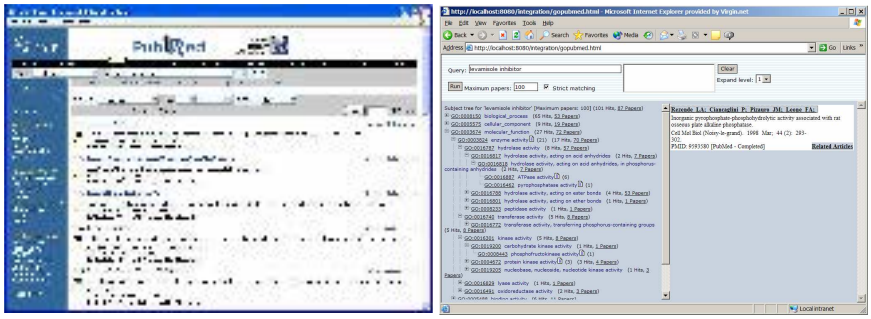


Fig. 1. Left: We want to know which enzymes are inhibitited by levamisole. A search on PubMed returns over 100 relevant papers: too many to get a quick answer. While alkaline phosphatase appears in titles of top hits, phosphofructokinase is buried in the 84th abstract. Right: User interface of GoPubMed. It displays the results for "levamisole inhibitor" limited to 100 papers. A number of relevant enzyme activies such as alkaline phosphatase and phosphofructokinase are found and directly displayed to the user.

which is a good result as only 78% of the abstracts do have words occurring in GO-terms and not being part of a list of very common words.

Our GoPubMed approach has a number of advantages: First, instead of pursuing only top hits, users get a high-level overview of the whole search result. Second, users are not forced to view multi-dimensional and thus often incomparable articles in a one-dimensional list. Instead they explore different search result "categories", which are hierarchically ordered and therefore allow for fast navigation from general to specific concepts. Third, the use of GO enables one to derive general keywords relevant to the search although they are not mentioned in the article at all, as they are derived indirectly from the GO. Forth, our approach and system are general and can be applied to other ontologies and text bodies.

Consider the following example: A researcher wants to know which enzymes are inhibited by levamisole. A keyword search for "levamisole inhibitor" produces well over 100 hits in PubMed (see Fig. 1). To find out about specific functions, we have to go through all these papers! We are interested in the relevant enzymatic functions. From the first titles it immediately is evident that levamisole inhibits alkaline phosphatase. A less well-known fact is however still buried in the abstracts. The 84th abstract (not the title) states that levamisole also inhibits phosphofructokinases. Without knowing specific activities, which a user is interested in, refining a keyword is also difficult. E.g. a search for "levamisole inhibitor enzymatic activity" produces only 5 hits: enough to learn about alkaline phosphatase, not enough to learn about the phosphofructokinase.

Often scientists search the literature to discover new and relevant articles. They provide keywords and usually get a possibly very long list of papers sorted by relevance. The search can be broken down into three steps: First, a query may be pre-processed (e.g. keywords may be stemmed, synonyms may be included and in general terms may be expanded (as done e.g. in PubMed)), second the search is carried out (this can range

from a simple keyword search to refined concepts such as using document link structure as implemented in Google) and finally post-processing of relevant results (in most cases presentation of results as a list). While such lists are useful, when looking up specific references, they are inadequate to get an overview over a large amount of literature and they do not provide a principled approach to discover new knowledge.

Our main contribution in this paper is the idea of ontology-based literature search. We propose to explore abstracts by categorising them according to an ontology, i.e a vocabulary to hierarchically structure a given domain. Prominent examples are GeneOntology (GO) [6] for molecular biology (www.geneontology.org), MeSH, SNOMED, UMLS for the medical domain (umlsinfo.nlm.nih.gov), which are related and overlap to some extent [7,3]. The hierarchical nature of ontologies allows one to quickly navigate from an overview to very detailed terms. As an example, even with over 19.000 terms in GO, it takes maximally 16 terms from the root of the ontology to the deepest and most refined leave concept. To illustrate the power of our approach let us consider the levamisole example again.

Consider the right figure of Fig. 1, which shows a screenshot of the GoPubMed prototype. The user wants to learn about enzymatic functions relevant to levamisole inhibitors. He/she submits "levamisole inhibitor" with GoPubMed limiting the classification and exploration to 100 articles. GoPubMed classifies the papers with GO and the user can explore the ontological classification of the papers: Out of the 100 papers 54 papers have ontology terms, which are biological processes, 20, which are cellular components, and 72, which are molecular functions. Selecting molecular function and enzymatic activities, for which terms are occurring in 70 papers, the user finds the terms hydrolase (relevant to 57 papers), transferase (8 papers), kinase (8 papers), lyase (1 paper), and oxidoreductase (3 papers). Opening the hydrolases, the user quickly finds alkaline phosphatase mentioned in 52 papers. The titles of these abstracts such as e.g. "Effects of alkaline phosphatase and its inhibitor levamisole..." immediately sustain that levamisole inhibits alkaline phosphatase. More interesting opening the kinase activity and carbohydrate kinase activity entries, the user discovers the phosphofructokinase. The abstract of the article from where this was extracted verifies that "levamisole directly inhibits tumor phosphofructokinase".

The main problem that needs to be solved before we can use ontologies for literature exploration is term extraction. Finding terms exactly in the literature is rarely possible and so we present a novel algorithm for this task, which is based on our analysis of GO. We first try to find short matching seed terms, which are then iteratively extended. The seed terms ignore uninformative term endings.

In the remainder we analyse GO and PubMed and derive from this our term extraction algorithm. We then show how to generate the induced ontologies for such terms and how to integrate these induced ontologies. We then evaluate our approach by collecting statistics on a sample of some 500.000 PubMed abstracts and by presenting an example in the implemented system.

2 GeneOntology and PubMed

Depth and breadth of GeneOntology. GeneOntology provides a structured vocabulary for molecular biologists[1]. GeneOntology's 13826 terms are organised in a hierarchy. The top level of the ontology breaks it down into three branches covering function, processes, and localisation. Overall, the GeneOntology has a depth[2] of 13 levels and starting from the top level it broadens up to the sixth level (3885 terms wide) and then shrinks again. The same holds for the distribution of leaves, which peaks at level 6 with 1223 leaves. Besides an Isa-relationship(pronounced *is a*), GO defines a $PartOf$ relationship.

Vocabulary. GeneOntology has 5320 different words, of which 2255 appear only once. The number of words per term follows a lognormal distribution; the average term consists of 4.01 words with a standard deviation of 2.11. The longest term has 29 words such as e.g. "oxidoreductase activity, acting on paired donors, with incorporation or reduction of molecular oxygen, 2-oxoglutarate as one donor, and incorporation of one atom each of oxygen into both donors". The most frequent words in GO-terms and at the end of GO-terms are all very general (activity, metabolism, biosynthesis, transport, etc.). The word "activity" is the most frequent word and appears mostly at the end of GO-terms. Overall, there are 1308 different endings, of which 708 occur only once. 8315 of the GO-terms, i.e. $> 50\%$, end with a word in the Reuter's stoplist of words. This means that the endings of GO-terms have a very low information content as they are extremely common, not only in the GO, but also in any text corpus. This will have implications for term extraction, as such terms should not be required for matches.

Hierarchical nature of GO. GO is not only a hierarchy by definition, but many GO-terms are substrings of another GO-term. 9794 GO-terms have at least one subterm, which is a GO-term and 7612 GO-terms end with another GO-term. An extreme case is the GO-term "positive regulation of retinal programmed cell death (sensu Drosophila)", which has the nine subterms "cell", "death", "cell death", "programmed cell death", "retinal programmed cell death", "regulation of retinal programmed cell death", "positive regulation of retinal programmed cell death", "regulation of retinal programmed cell death (sensu Drosophila)", "retinal programmed cell death (sensu Drosophila)".

Conclusion. From the above statistics we can draw some conclusions for term extraction of GO-terms from free text: First, often the endings of GO-terms are very general and thus of low information content. When trying to extract GO-terms these endings can safely be dropped. Second, often GO-terms are subsequences of each other. As a consequence, one strategy for extraction of GO-terms will first consider short matching GO-terms, which are then iteratively expanded.

PubMed. PubMed is a collection of over 14 million abstracts of the biomedical literature dating back to the 1960s. For the experiments below we selected a random corpus with 531958 papers. It turns out that 119831 (23%) do not have an abstract and will be of limited use for term detection. On average the sample papers have 170 words in their abstract with a standard deviation of 118 words. The longest abstract has 1700 words.

[1] We used the May 2003 edition of GeneOntology

[2] Please note that there is a difference between the minimal and maximal length path between a term and the root. The depth, as formally below, is the minimal length path.

PubMed provides an intelligent keyword search, which uses the MESH ontology, which is geared towards the medical domain in contrast to GO, which covers molecular biology. Essentially, PubMed does not only search for submitted keywords, but also for synonyms and expansions according to the ontology.

3 GoPubMed

The idea of GoPubMed is to use the GeneOntology to search and browse PubMed. To do this, two problems need to be solved: First, how to extract GO-terms from a PubMed abstract and second, how to construct the relevant sub-ontology of GO.

Definitions. To proceed we need some basic definitions such as word sequences (or sequences for short), subsequences, segments, terms, an ontology, and patterns. Note, that for convenience we enumerate word sequences from n to 1, as this reflects the following algorithms, which move from right to left, more appropriately.

Definition 1: Let $w_n, .., w_1$ be words, then $s = (w_n, ..., w_1)$ is a word sequence of length n and $s' = (w_{i_m}, ..., w_{i_1})$ is a subsequence of s iff $n \geq i_m \geq ... \geq i_1 \geq 1$. A contiguous subsequence is called a segment.

For convience, we define an abstract as a typically long word sequence containing free text. Similarly, a term in an ontology is a typically short word sequence. We will sometimes refer to a segment of a term as a subterm. Next we define a minimal notion of an ontology, which is however sufficient to capture the GeneOntology and other simple taxonomies. Essentially, such a minimal ontology comprises a set of terms, which provide the basic vocabulary, a root of the tree, which is a term, and the relationship Isa, which defines the inheritance relationship of the terms. Isa can be any rooted, connected and directed acyclic graph. The root of this graph is the top term, from which all other terms inherit. The relationship has to be acyclic to ensure that there are no self-referential definitions of terms. Additionally, it is useful to require that the Isa relationship is minimally defined in that the inheritance $t_1\ Isa\ t_3$ should not be present if it can be inferred from $t_1\ Isa\ t_2$ and $t_2\ Isa\ t_3$.

Definition 2. Let $O = (T, root, Isa)$, where T is a finite set of finite length terms, $root \in T$, $Isa \subseteq T \times T$. Let $t_i \in T$ for $1 \leq i \leq n$ and $t_i\ Isa\ t_{i+1}$ for $1 \leq i < n$, then $p = [t_1, ..., t_n]$ is called a path of length n from t_1 to t_n. O is called an **ontology** iff for all $t \in T$, $t \neq root$, there exists a path from t to $root$ and there is no path from t to t. The length of the longest path to the root is called the height of ontology O. The depth of a term t is the length of the minimal path from t to the root. The breadth of the ontology at level l is the number of terms with depth l. The breadth of the ontology is the maximal breadth across all levels.

For term extraction we will use regular expressions as a notation to describe the algorithm. The patterns use symbols as defined below.

Definition 3. The expression `\w` matches a word, `\s` a space, the dot `.` any single character. To repeatedly match an expression there are three operators: `?` requires the preceding pattern to appear once or not at all, `+` requires it to appear once at least once, and `*` requires it to appear any number of times (including 0). Brackets () are used to group expressions.

Step 1: Term extraction. Term extraction should reflect the following constraints: 1. Terms in the ontology have a structure, as they specialise from right to left. A term is e.g. specialised by prepending an adjective. Often the rightmost word of the term and hence the most general will not be present in the abstract, as it can be dropped in natural language without compromising the reader's understanding (e.g. "activity", "biosynthesis"). Term extraction should ignore these very general words. 2. Extraction of terms should start by first matching short (and hence general) terms. These seed matches are then extended to build maximal matches, which may comprise gaps. 3. If more than one term matches a pattern for a term, then the shortest and highest in the ontology should be returned. This will usually be the most general match as it is higher up in the ontology. Let us consider concrete examples in the GeneOntology and PubMed for the above requirements:

Example 1. PMID 12747961 contains the segment "cAMP-dependent kinase", which does not occur as such in GO. However, "kinase activity" is a GO-term. Therefore it should be matched and act as a seed. This seed GO-term has a child "cAMP-dependent protein kinase activity", which indeed is the best fitting match. A general method to find matches this way searches first for the pattern `kinase \w+` and then for the pattern `cAMP-dependent .* kinase activity`. Three more examples are a) "cyclin-dependent kinase" (PMID 12702569), which matches GO-term "kinase activity" as seed and then GO-term "cyclin-dependent protein kinase activating kinase activity" as extension, b) "protein phosphorylation" (PMID 10218114) with seed " phosphorylation" and full match "protein amino acid phosphorylation" and c) "alpha-amino-3-hydroxy-5-methyl-4-isoxazole glutamate receptor" (PMID 12204345) with seed "glutamate receptor activity" and full match "alpha-amino-3-hydroxy-5-methyl-4-isoxazole propionate selective glutamate receptor activity". However, the requirement of a seed also means that some GO terms cannot be discovered even though they appear in the text. The GO-term "extracellular matrix structural constituent conferring tensile strength" cannot be discovered as "strength" is in the stop-list and as "tensile" does not appear in any other GO-term and hence cannot serve as a seed.

This means that term extraction should move from right to left. However, the very general rightmost word of many GO-terms can be ignored. The term `activity` occurs e.g. in over 33% of GO-terms as last word. The above terms also illustrate that term extraction should not attempt to match highly specialised terms directly, but first to find seeds, which can then be expanded. As a consequence, we will always aim to match GO-terms of the form $t = (w_n, ..., w_1, c)$ where c is an optional word that can be empty or ignored if it is from a list of frequently occurring words.

The term extraction (algorithm 1) proceeds as follows: It iterates through the words a_j of the abstract from right to left (line 6-26). If the word a_j is part of a stop list of common words it is not considered (line 8). Otherwise, the algorithm tries to find an initial seed for a matching ontology term by searching for terms starting with a_j followed by space, dash, slash and another word (line 9 and 10). This pattern-based search is done via the method *FindExpr(pattern,T)*, which applies the pattern *pattern* to the terms *T*. If the method finds more than one matching term, it returns the match that has the highest level in the ontology's hierarchy and is the shortest among those at the

Algorithm 1 $F = ExtractTerms(a)$

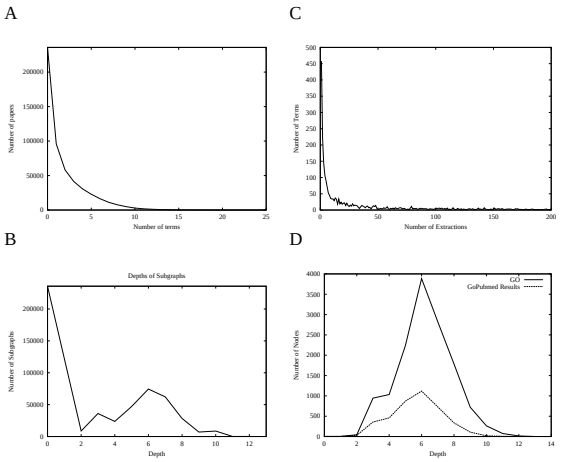
```
1: Let stoplist be a list of common words
2: Let T be the ontology's terms
3: F := ∅ {set of terms t ∈ T found in a}
4: a = (a_m, ..., a_1) {Split abstract a into words}
5: j := 1 {variable iterating through a}
6: while j ≤ m do
7:    k := 1 {variable to expand matching term}
8:    if a_j ∉ stoplist then
9:       pattern := a_j([\s - \]\w+)? {Pattern to look for a_j optionally followed by either space, minus, or slash
         and then a word}
10:      t = (a_j, c) := FindExpr(pattern, T) {construct the initial seed GO-term}
11:      if a term t is found and (c is empty or in the stoplist) then
12:         while j + k ≤ m do
13:            pattern := a_{j+k} .* t {pattern to expand current match: look for a_{j+k}, any insertion, and then the term
               t found so far}
14:            t' := FindExpr(pattern, T)
15:            if no such t' is found then
16:               exit the while loop
17:            else
18:               t := t'
19:               k := k + 1 {next word}
20:            end if
21:         end while
22:         add t to F {store the discovered term}
23:      end if
24:   end if
25:   j ← j + k
26: end while
27: return F
```

same level. If such a seed term t is found for abstract word a_j, then it is extended and refined if possible (line 12 to 21). This is again done with *FindExpr* with the pattern searching for a term consisting of the next word in the abstract a_{j+k} followed by any other possible word matches and then the already found term t. The above algorithm has complexity linear in the number of words of the abstract. The required pattern matching can be implemented very fast by using hash tables.

Step 2: Induced Ontology. Once terms have been extracted from abstracts, we wish to use the ontology to browse the given abstracts. To avoid unnecessary parts of the ontology not relevant to the given abstracts, we are seeking the minimal sub-ontology that covers all the terms extracted from the abstracts. Given an ontology $O = (T, root, Isa)$ and a set of terms $T'' \subseteq T$ extracted from free text, we want to construct the minimal sub-ontology of O, which is induced by the extracted terms T''. This means that we have to find all intermediate terms, which are contained in all paths from terms in T'' to the root. Essentially, we start at terms $t'' \in T''$ and iteratively look up parent terms until the root is reached. Overall, we have to maximally consider all terms in T, so that the complexity for the sub-ontolgoy construction is $O(|T|)$.

4 Evaluation

To evaluate ontolgoy-based literature search with GoPubMed, we collect statistics on the term extraction and present example queries and interaction with the implemented system, which is available online.



E

Freq	term
59428	protein
29342	growth
23025	receptor activity
17544	enzyme activity
14575	membrane
13900	mRNA metabolism
10492	inflammatory response
10147	antigen processing
10086	intracellular
9971	behavior
9877	peptide metabolism
9626	apoptosis
9118	physiological processes
8463	pathogenesis
8439	kinase activity
8345	RNA
8065	mitochondrial electron transport chain
8005	electron transport

Fig. 2. A) Top left: Number of terms in papers. The maximal number of GO-terms found in an abstract is 25. **B) Bottom left:** Distribution of the depths of the induced ontologies of the sample abstracts. On average induced ontologies are 3.22 terms deep (standard deviation 3.18) and 2.03 term wide (standard deviation 2.61). **C) Top middle:** Distribution of extracted terms (x-axis trimmed at 200). Most terms are extracted only once. However some very general ones are extracted many times, such as "protein" (59428 times). **D) Bottom middle:** level by level breadth of GeneOntology as a whole and the induced ontology of all sample abstract. The induced ontology mimics the breadth distribution of GO well. **E) Right:** Most frequently extracted GO-terms from sample abstracts.

Number of terms. For the distribution of GO-terms per abstract/title consider the top left figure in Fig. 2. Out of the 531958 random sample abstracts we considered, at least one term appears in the abstract and/or title of 296202 (56%) and in the title of 147265 (28%). At first glance, this figure appears to be low, but only 78% of the abstracts do actually contain any words occurring in GO-terms and not being part of a list of common words. On average 1.76 terms are found per abstract with a standard deviation of 2.40. The maximum number of terms discovered is 25 (in PUBID 12244448).

At first glance, 44% of abstracts without GO-terms appears high implying that the term extraction is too restrictive. To verify this, we examined some of the papers without GO-terms. It turns out that PubMed comprises a lot of literature not relevant to molecular biology and GeneOntology. Examples are papers entitled "Relative efficacy of the proposed Space Shuttle antimotion sickness medications", "Point-counterpoint: should physicians accept gifts from their patients? No: Gifts debase the true value of care", "The why and wherefore of empowerment: the key to job satisfaction and professional advancement", and "Naming, labelling, and packaging of pharmaceuticals".

Extracted terms. In the same way that not all of PubMed is relevant to GeneOntology, not all of GeneOntology is appearing in PubMed abstracts. In fact, only 29% (4036 terms) of GO are included in the induced ontology of all sample abstracts. Consider the top right figure in Fig. 2. 458 of the occurring GO-terms occur only once. A number of very general GO-terms such as "protein", "growth", "receptor activity" appear very frequently as the table on the right of Fig. 2 shows.

Induced ontologies. Consider the bottom left figure of Fig. 2. On average an induced ontology of a single abstract has 8.60 terms (standard deviation 10.98) and 9.13 edges

(standard deviation 12.58). On average it is 3.22 terms deep (standard deviation 3.18) and 2.03 terms wide (standard deviation 2.61). The bottom right figure in Fig. 2 shows the breadth of GeneOntology as a whole and the induced ontology of all sample abstract for each level of the ontology. The induced ontology mimics the breadth distribution of GO well.

5 Comparison and Conclusion

Little is published so far on ontology-based literature search. The most similiar systems to GoPubMed are XplorMed and goKDS.

XplorMed. [9] have developed XplorMed, which maps PubMed results to 8 main MeSH categories and then extracts topic keywords and their co-occurrences. We tested XplorMed with "levamisole inhibitor". XplorMed returned 22 relevant words (e.g. activity, protein, cell), which are, however, very general (only 3 of the 22 appear with frequency below the average word frequency in GO) and cannot really be considered as topics. Overall it was not possible to extract new knowledge from XplorMed, as is possible in GoPubMed. Also, GoPubMed uses the full ontology with 4000 terms actually detected in our sample documents as opposed to XplorMed, which limits categorisation to the top layer of MeSH (8 terms).

goKDS. goKDS (www.reeltwo.com/go) [11] is a commercial system, which extracts GO-terms from PubMed articles using naive bayes learning and support vector machines. GoKDS uses 30.000 GO-annotated documents for training to learn 3500 categories - around a quarter of the GO. Often there are, however, only a few training documents per category, which causes problems. For some 21 categories, the authors evaluated accuracy and obtain around 70%, which is in line with results reported in [10] using a maximal entropy approach. Term extraction in GoKDS and in our own approach differ in that GoKDS uses machine learning, whereas we parse abstracts directly applying our matching heuristic. GoKDS's classification acts as a blackbox and therefore makes it difficult to assess the quality of a classification. GoPubMed does not suffer from this problem. GoPubMed also deploys GO to browse and explore search results, a feature not catered for in GoKDS.

Extracting Protein Names. [5] investigate how to extract protein names from free text and establishes some general principles. In particular, the authors follow an incremental approach, in which various heuristics are applied after tokenizing to extend basic matches to full ones. The approach is based on some basic natural language processing. In contrast, our own incremental approach is motivated by the ontology structure being reflected in the substring relationship. The structure of GeneOntology terms has been analysed by [7]. They found that 79% of GO are useful for natural language processing applications and that 35% of GO-terms can be found in Medline. This is in line with our findings (29%) and also GoKDS (25% (3500/13826)). The hierarchical structure and substring relationships between GO terms, which is fundamental to our algorithm, has been independently investigated by [8]. By comparing this substring relationship to the `Isa`-relationship, the authors are able to suggest a number of terms currently missing from the ontology. In contrast to [8], our study of substring relationships within GO also includes term endings, which is a vital basis for our novel term extraction algorithm.

Summary. Let us summarise: We analysed GO and collected statistics on word composition and the hierarchical nature of GO-terms. We found that most term endings have low information content and can therefore be ignored for term extraction and that 2/3 of GO-terms have GO-subterms (when considered as strings). From these observations we derived a novel term extraction algorithm, which first finds a matching seed GO-term ignoring uninformative term endings and which then iteratively extends the seed.

We took a sample of more than 500.000 abstracts. The average length is 170 words, the maximum 1700. The abstracts mapped to 29% of GO-terms, a rate similar to other approaches. 56% of the abstracts could be annotated with GO-terms. The latter is not surprising as 23% have no abstract and as only 78% of abstracts do contain words occurring in GO-terms and not being part of a list of common words. Furthermore, there is much literature in PubMed not relevant to molecular biology and hence GO.

Finally, we developed an online system, which supports ontology-based exploration of PubMed search results and showed how it can discover novel findings through the GO-classification. Our approach and system are general and can be applied to other ontologies and text bodies.

Acknowledgment: We gratefully acknowledge support of the EU projects BioGrid (IST-2002-38344) and REWERSE (IST-2004-06779).

References

1. C. Blaschke, MA. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: protein-protein interaction. In *Proc. of ISMB*, 1999.
2. C. Blaschke, L. Hirschman, and A. Valencia. Information extraction in molecular biology. *Briefings in Bioinformatics*, 3:154–65, 2002.
3. M.N. Cantor, I.N. Sarakr, R. Gelman, F. Hartel, O. Bodenreider, and Y.A. Lussier. An evaluation of hybrid methods for matching biomedical terminologies: Mapping the GeneOntology to the UMLS. *Stud Health Technol Inform.*, 95:62–7, 2003.
4. C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. Genies: a NLP system for the extraction of molecular pathways from journal articles. In *Proc. of ISMB*, 2001.
5. K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proc. of PSB*, 1998.
6. GeneOntologyConsortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.*, 1(32):D258–61, 2004.
7. Alexa T. McCray, Allen Browne, and Olivier Bodenreider. The lexical properties of the Gene Ontology (GO). In *Proc. of AMIA Symp.*, pages 504–8, 2002.
8. P.V. Ogren, K.B. Cohen, G.K. Acquaah-Mensah, J. Erberlein, and L. Hunter. The compositional structure of Gene Ontology terms. In *Proc. of PSB*, 2004.
9. C. Perez-Iratxeta, A.J. Perez, P. Bork, and M.A. Andrade. Update on xplormed: A web server for exploring scientific literature. *Nucleic Acids Res*, 31(13):3866–8, 2003.
10. S. Raychaudhuri, J.T. Chang, P.D. Sutphin, R.B. Altman. Associating genes with GO codes using a maximum entropy analysis of biomedical literature. *Gen. Res.*, 1:203–14, 2002.
11. Tony Smith and John Cleary. Automatically linking medline abstracts to the geneontology. In *Proc. of Bio-Ontologies Meeting*, Brisbane, Australia, 2003.
12. L. Tanabe, U. Scherf, LH. Smith, JK. Lee, L. Hunter, and JN. Weinstein. Medminer: internettext-mining tool for biomedical information, with application to gene expression profiling. *BioTechniques*, 27(6):1210–4,1216–7, 1999.
13. James Thomas, David Milward, Christos Ouzounis, Stephen Pulman, and Mark Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Proc. of PSB*, 2002.