Protein Annotation as Term Categorization in the Gene Ontology

Karin Verspoor, Judith Cohn, Cliff Joslyn, Sue Mniszewski,
Andreas Rechtsteiner, Luis M. Rocha, Tiago Simas

Los Alamos National Laboratory
PO Box 1633, MS B256
Los Alamos, NM 87505

We addressed BioCreAtIvE Task 2, the problem of annotation of a protein with a node in the Gene Ontology (GO). We approached the task as a problem of categorizing terms derived from the document neighborhood of the given protein in the given document into nodes in the GO based on the lexical overlaps with terms on GO nodes and terms identified as related to those nodes. The system incorporates NLP components such as a morphological normalizer, a named entity recognizer, a statistical term frequency analyzer, and an unsupervised method for expanding words associated with GO ids based on a probability measure that captures word proximity (Rocha, 2002). The categorization methodology uses our novel Gene Ontology Categorizer (GOC) methodology (Joslyn et al. 2004) to select GO nodes as cluster heads for the terms in the input set based on the structure of the GO.

Pre-processing

Swiss-Prot and TrEMBL IDs were provided as input identifiers for the protein, so we needed to establish a set of names by which that protein could be referenced in the text. We made use of both the gene name and protein names that are in Swiss-Prot itself, when available, and a collection of synonyms constructed by Procter & Gamble Company. The fallback case was to use the name filled in from the EBI TrEMBL human data[1]. The resulting database tables were used to construct a list which was dynamically loaded from the database into a GATE (Cunningham et al. 2002) gazetteer processing module (which in turn compiles it into a finite state recognizer).

Additional pre-processing was performed on the document corpus. First, the original SGML documents were parsed to extract the Title, Abstract, and Body components, to normalize SGML character entities to corresponding ASCII characters (for instance, converting "&prime;" to an apostrophe), and to remove all formatting tags apart from the paragraph markers. Subsequently, we morphologically normalized the documents using a tool called "BioMorpher"[2]. We performed frequency analysis on the resulting terms, and selected representative terms for each document using a TFIDF filter (term frequency inverse document frequency, Witten et al 1994).

Unsupervised Methodology for Expanding Words Associated with GO ids

The (protein, document, GO id) triples provided for training purposes, as well as those given for the evaluation of Task 2.1, were used to determine sets of words related to GO ids following a methodology developed for the *Active Recommendation Project*[3] at Los Alamos (Rocha and Bollen, 2001). After document pre-processing, we divided each document into paragraphs and calculated for each document a matrix of word occurrence in the paragraphs: R: $P \times W$, where P is the set of all m paragraphs in a document, and W is the set of all n words. This is a Boolean matrix ($r_{i,j} \in \{0, 1\}$) that specifies if a given word occurred at least once in a given paragraph.

$$wpp(w_i, w_j) = \frac{\sum_{k=1}^{m}\left(r_{i,k} \wedge r_{j,k}\right)}{\sum_{k=1}^{m}\left(r_{i,k} \vee r_{j,k}\right))}$$

From the R matrices, we calculated a *word in paragraph proximity* matrix, WPP, for each document, using the co-occurrence probability measure shown at right, as defined in Rocha (2002). WPP denotes the association strength between pairs of words (w_i, w_j) , based on how often they co-occur in the paragraphs of a given document. A value of

[1] A script was applied to the TrEMBL names that generated variants of strings containing mismatched punctuation and parentheticals such as "(precursor)" or "(fragment)" which were felt not to be likely to occur directly in the text.

[2] BioMorpher is a morphological analysis tool built on the Morph tool originally developed at the University of Sheffield by Kevin Humphreys and Hamish Cunningham for general English, extended to include large exception lists for biological text as well as to handle some morphological patterns not handled by the original tool.

[3] http://arp.lanl.gov

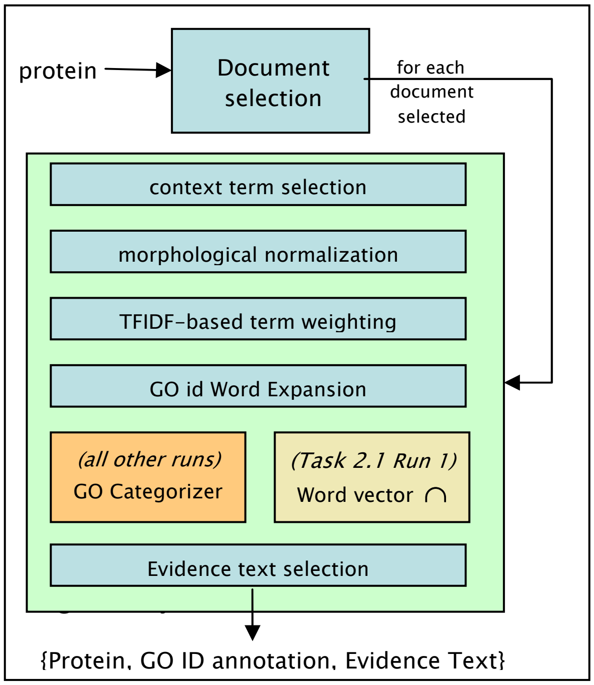

Figure 1: System Architecture

$wpp\ (w_i, w_j)$ = 0.3, means that words w_i and w_j co-occur in the same paragraphs 30% of the time that either one of them occurs. To avoid artificially high values of WPP, we computed this value only if the total number of paragraphs in which either of the words occurs (the denominator of the formula) is at least 3.

From the GO ids in the provided triples we retrieved the words from the GO node label. Let us refer to this set of words as W_{GO}. For each document, we then retrieved a set of words highly associated in WPP with the words in W_{GO}. Specifically, we returned the top 5 to 10 additional words with highest average value of WPP to all words in W_{GO}. The additional words thus discovered were used to expand W_{GO}. Let us refer to the expanded set of words as W_{GOPRox}; the additional words are not found in the respective GO node label, but co-occur highly in a given document with those words.

In Run 1 submitted for Task 2.1 (see below), which yielded arguably the best result of any submission for this task, for each (GOid, document) pair we used its respective matrix R and set W_{GOPRox} to recommend paragraphs as evidence text for the GOid. This was done using a vector intersection operation. The columns of R are vectors of words occurring in a paragraph. We choose as evidence text for the GO id the paragraphs associated with the columns of R that yield the largest intersection with W_{GOPRox}. That is, paragraphs containing the largest number of words also found in W_{GOPRox} are selected.

System Operation

The architecture of the system is shown in Figure 1. The system is built around a technology called the Gene Ontology Categorizer (GOC, Joslyn *et al.* 2003, 2004), which utilizes the structure of the Gene Ontology to find the best covering nodes given a set of node “hits”. GOC uses pseudo-distances between comparable nodes to score each node with respect to a given query, balancing *coverage* – covering as many inputs as possible – and *specificity* – covering inputs at the “lowest level” in the GO as possible.

GOC was originally designed (Joslyn *et al.* 2004) to take as a query a list of gene products, which are then mapped to the set of GO nodes to whch they are annotated. For BioCreAtIvE Task 2, GOC was extended first to accept weighted query items, and additionally extended to take terms, which are then again mapped to the set of GO nodes in which the terms appear, as query items.

Terms are collected through analysis of the sentential context of the given protein, morphologically normalized, and weighted using a normalized TFIDF value derived during pre-processing. Weights represent the contentfulness of each term. Internally, GOC looks for lexical matches between the input term set and (morphologically normalized) terms associated with each individual node in the GO. A match between an input term and a term associated with a GO node counts as a “hit” on that node, with the strength of a hit determined by the weight of the term.

Terms are associated with GO nodes via one of three mechanisms:

- **Direct**: The term occurs in the node label of GO node
- **Definitional**: The term occurs in the definition text associated with GO node
- **Proximity**: Additional terms are identified as closely related to each GO node following the Proximity GO id Word Expansion as described above (Rocha 2002).

Direct and indirect associations are counted as distinct “hits” on a node and can be weighted differently. After transforming the input query into a set of node hits, GOC traverses the structure of the Gene Ontology, percolating hits upwards, and calculating scores for each GO node (see Joslyn et al 2003, 2004). GOC returns a set of GO nodes representing “cluster heads”[4] for the weighted term input set, as well as data on which of the input terms contributed to the selection of each cluster head. This information is used to select the evidence text for the GO assignment associated with the cluster head. To address this, we again bring in proximity measurement – in this case, the proximity of terms to individual paragraphs in the document. The set of terms which contributes to an annotation is judged to be close to one or more paragraphs in the document, and finally the closest match is selected as the evidence.

Evaluation Results

We submitted 3 runs for each of tasks 2.1 and 2.2 (as well as a run for task 2.3 which was not scored). The results for the two runs are shown in Figure 2. One of the runs we submitted for task 2.1 had arguably the best result of any submission. This run (user 7, run 1) utilized a configuration of our system which bypassed GOC, utilizing only the Proximity GOid Word Expansion followed by vector intersection of the columns of R and the expanded set of words associated with a GOid, W_{GOPRox}, to discover paragraphs. We achieved a score of either perfect or generally good for 413 of the results; this corresponds to a good result for 38% of the 1076 queries, and the highest combined score (the next closest was user 14 with 357). Focusing just on perfect results, our result of 263 was in the top echelon. In this configuration, we ignored the protein altogether and focused on the GO node-paragraph relationship. Nonetheless, we received a score of “high” on the protein mention measurement for 638 of the 1050 (61%) answers we submitted. This result reflects a high coherence between the GO nodes and the given target proteins in the given documents, at least at the level of paragraphs.

Our results for the other runs we submitted for Task 2.1 were less good, achieving a perfect or

User, Run	# results	“perfect”	“generally”
4, 1	1048	268 (25.57%)	74 (7.06%)
5, 1	1053	166 (15.76%)	77 (7.31%)
5, 2	1050	166 (15.81%)	90 (8.57%)
5, 3	1050	154 (14.67%)	86 (8.19%)
7, 1	1050	263 (25.05%)	150 (14.29%)
7, 2	1856	43 (2.32%)	40 (2.16%)
7, 3	1698	59 (3.47%)	27 (1.59%)
9, 1	251	125 (49.80%)	13 (5.18%)
9, 2	70	33 (47.14%)	5 (7.14%)
9, 3	89	41 (46.07%)	7 (7.87%)
10, 1	45	36 (80.00%)	3 (6.67%)
10, 2	59	45 (76.27%)	2 (3.39%)
10, 3	64	50 (78.12%)	4 (6.25%)
14, 1	1050	303 (28.86%)	69 (6.57%)
15, 1	524	59 (11.26%)	28 (5.34%)
15, 2	998	125 (12.53%)	69 (6.91%)
17, 1	412	0 (0.00%)	1 (0.24%)
17, 2	458	1 (0.22%)	0 (0.00%)
20, 1	1048	300 (28.63%)	57 (5.44%)
20, 2	1050	280 (26.72%)	60 (5.73%)
20, 3	1050	239 (22.76%)	59 (5.62%)

User, Run	# results	“perfect”	“generally”
4, 1	661	78 (11.80%)	49 (7.41%)
7, 1	153	1 (0.65%)	1 (0.65%)
7, 2	124	1 (0.81%)	1 (0.81%)
7, 3	263	2 (0.76%)	10 (3.80%)
9, 1	28	9 (32.14%)	3 (10.71%)
9, 2	41	14 (34.15%)	1 (2.44%)
9, 3	41	14 (34.15%)	1 (2.44%)
10, 1	120	35 (29.17%)	8 (6.67%)
10, 2	86	24 (27.91%)	6 (6.98%)
10, 3	116	37 (31.90%)	11 (9.48%)
15, 1	502	3 (0.60%)	8 (1.59%)
15, 2	485	16 (3.30%)	26 (5.36%)
17, 1	94	1 (1.06%)	0 (0.00%)
17, 2	55	1 (1.82%)	0 (0.00%)
17, 3	99	1 (1.01%)	1 (1.01%)
20, 1	673	20 (2.97%)	30 (4.46%)
20, 2	672	38 (5.65%)	26 (3.87%)
20, 3	673	58 (8.62%)	27 (4.01%)

Figure 2: Results of system runs. The table to the left contains the results for Task 2.1; the table to the right contains results for Task 2.2. We were User 7.

[4] Note that we are *not* using “cluster” here in the sense of traditional clustering, e.g. *k*-means (Joslyn *et al.* 2003, 2004).

generally good score for 83/86 (runs 2/3, respectively) of the queries, or about 8%. These two runs used the full architecture as shown in Figure 1; run 2 used a very basic sentence text selection algorithm, in which the sentence containing the most number of terms from the set of input terms while run 3 used the proximity-based paragraph selection algorithm.

Our Task 2.2 results were poor, at the bottom of the sets of runs along with user 17. However, we have been informed of a problem with the evaluation of our Task 2.2 results, and it is as yet unclear what the impact of that problem is.

Results Discussion

There are several important general issues in the evaluation that impacted our performance.

Unknown proteins: The strategy that we follow for identifying the "context window" of a protein (in runs other than Task 2.1, Run 1) depends on recognizing references to the protein in the text. We depend on a list of known names associated with the protein IDs to pick out sentences or paragraphs of particular relevance to the protein. We chose this strategy as it was straightforward to implement, and because the problem of protein reference identification was being addressed in Task 1 and we felt the focus in Task 2 would be on the GO annotation tasks. The training data for Task 2 bore this out – a large majority (about 70%) of the queries contained proteins that were known to us. However, we discovered that the test data contained many protein IDs that were not yet available in SwissProt; we assume that these are recently identified proteins. Only 58 of the 286 (20%) proteins referenced in all subtasks of Task 2 were in our database of known proteins; 29/138 (21%) for Task 2.1 and 19/138 (14%) for Task 2.2. The statistics for the impact this had on the queries was even worse: only 153/1076 (14%) of the queries in Task 2.1 and 44/435 (10%) of the Task 2.2 queries included proteins for which we had names. We were able to fall back to the names in the TrEMBL database, but these are of poor quality and usually there is only one name, not a full set of synonyms for a protein. This issue had a huge impact on our ability to focus in on text within documents that was directly relevant to the protein of interest and effectively placed a very low upper bound on our evidence text selection scores.

Assessment criteria: The methodology followed by the evaluators of Task 2.2 focused on the evidence text, measuring whether the selected evidence text for a given query mentioned both the protein of interest, and the function/process/component indicated by the target GO node. The GO node prediction was not evaluated independently of the evidence text. Our interpretation of the task was that there were two results, prediction of the GO node and selection of the exidence text. Our understanding is that the primary task of an annotator is to correctly annotate a protein; in fact the GO and other biological knowledge repositories rarely reference anything more specific than a Pubmed ID as evidence for an annotation. Hence we considered the two results separately and focused our energy more on the GO node annotation component. Our analysis of our results strictly for annotation prediction (Table 1) shows that we achieved an F-score of 0.24 for runs 1 and 2 (these results are identical as expected since the runs only varied with respect to the evidence text selected) and 0.29 for run 3, across all desired answers. The scores were calculated in terms of direct hits, i.e. queries for which we returned exactly the GO node that was desired, and indirect hits, or queries for which we returned a node which was either a sibling, aunt, cousin, ancestor, or descendent of the desired answer.

There were also some issues specific to our algorithms that led to poor results.

Discussion, GOC-based runs:

Due to the "unknown proteins" problem described above, we were unable to focus on a context window around the protein of interest, and the "protein neighborhood" terms input to GOC were in most instances the top TFIDF-ranked terms for the document as a whole, rather than coming

Run	Precision, Direct	Precision, Indirect	Precision, Total	Recall, Direct	Recall, Indirect	Recall, Total	F-score, Total
Run 1	0.061	0.185	0.246	0.059	0.181	0.241	0.243112
Run 2	0.061	0.185	0.246	0.059	0.181	0.241	0.243112
Run 3	0.057	0.228	0.285	0.059	0.238	0.298	0.291323

Table 1: Assessment of Precision/Recall over the GO annotation prediction component of Task 2.2.

from a coherent textual neighborhood around the protein. This had several implications. First, GOC may have been “overseeded” – since the input terms were derived from across the document, they may have matched very dispersed nodes in the GO. This would make it difficult for the GOC algorithm to confidently select a covering node for the input terms. Second, evidence text selection on the basis of overlap with or proximity to terms from across the document is difficult; it is unlikely that any single sentence/paragraph matches more than a few of these terms.

The overseeding may have led to an additional difficulty. The number of terms from the GOC input set used to rank a GO node was very small – normally 1-3 terms – and only this subset of terms was passed on to the two evidence selection algorithms. The motivation underlying this approach was to enable the evidence text selection for a GO annotation to proceed on the basis of only those document terms relevant to that annotation. In practice, given the small and weakly coherent sets of terms that were generated, this created great difficulty for reliably selecting a contiguous chunk of text focused on that GO node. This problem may have ameliorated by incorporating the strategy from Task 2.1, Run 1, utilizing all available information about the selected GO node, rather than limiting ourselves to terms from the context window.

Finally, we would like to explore the interaction between TFIDF weights and the importance of a term in the GO. Preliminary analysis suggests that there are very frequent terms in the GO with relatively high TFIDF scores in the corpus; this would unfairly value those terms in GOC and exacerbate the overseeding problem. Some adjustment of the weighting scheme to better take into consideration the terminological structure of the GO is perhaps warranted.

Discussion, Proximity-based Word Expansion and Evidence Text selection:

While the proximity-based word expansion proved to be a very useful technique, responsible for arguably the best run of the entire competition for task 2.1, the evaluator comments indicated that they were often unhappy with paragraphs as the basic unit for evidence text. To address this, we envision several changes. We could apply the proximity measurements at the sentence level, rather than the word level; we could explore metrics for recognizing excessively long paragraphs and splitting them at positions of subtle topic change; or we could try to use more linguistic (structural) analysis to focus in on the core information expressed and narrow the text returned.

There are some additional ways to build on our results. We could calculate a global word proximity matrix, rather than one matrix per document, which should strengthen our confidence in the relationships between words, as well as relating any given word to more words due to consideration of its occurrence across the document corpus. We could also incorporate semi-metric analysis of the word proximities (Rocha 2002) to find additional related words, even if they do not co-occur in the corpus.

Acknowledgements

This work was sponsored in part by a cooperative research agreement with the Procter & Gamble Company, who also supplied us with the list of protein synonyms. We would like to thank Andy Fulmer and Jun Xu for their contributions to this work.

References

H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.

Joslyn, C., S. Mniszewski, A. Fulmer, G. Heaton (2003). “Structural Classification in the Gene Ontology”. In Proceedings of the Sixth Annual Bio-Ontologies Meeting (Bio-Ontologies 2003), Brisbane, Australia, June 28, 2003.

Joslyn, C., S. Mniszewski, A. Fulmer, G. Heaton (2004). “The Gene Ontology Categorizer”, to appear in *Bioinformatics*.

Rocha, Luis M. and Johan Bollen [2001]. "Biologically Motivated Distributed Designs for Adaptive Knowledge Management". In: *Design Principles for the Immune System and other Distributed Autonomous Systems*. L. Segel and I. Cohen (Eds.) *Santa Fe Institute Series in the Sciences of Complexity*. Oxford University Press, pp. 305-334

Rocha, Luis M. (2002). "Semi-metric Behavior in Document Networks and its Application to Recommendation Systems". In: Soft Computing Agents: A New Perspective for Dynamic Information Systems. V. Loia (Ed.) International Series Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 137-163.

I.H. Witten, A. Moffat, and T. Bell [1994]. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, New York.