# *Preliminary Report on the BioCreative Experiment: Task Presentation, System Description and Preliminary Results.*

**Frédéric Ehrler**$^{ac}$ **and Patrick Ruch**$^{1ab}$

[a] *SIM, University Hospital of Geneva, Geneva*
[b] *Theoretical Computing Laboratory, Swiss Federal Institute of Technology, Lausanne*
*AI Laboratory, University of Geneva*

## Summary

For this first edition of the BioCreative challenge, two tracks were organized: named entity recognition (track 1) and information extraction/retrieval (track 2). We participated only in track 2. For subtasks were proposed: 2.1, 2.2, 2.3 and 2.4. 2.3 and 2.4 can be seen as a traditional ad hoc information retrieval tasks in two different (small and medium) size document collections. Task 2.1 is an information extraction tasks: given a protein (SwissProt entry) and a Gene Ontology (GO) annotation, provide a segment of text that can support this annotation. Task 2.2 is somehow similar but add a text categorization task: the GO terms are missing and must be provided. Our participation concerns tasks 2.1 and 2.2: this report describes 1) the information extraction module, which attempts to find the textual segment that support the GO annotation and 2) the text categorization system used in task 2.2.

All tests and developments were done on Intel platforms with 1 GB of memory and 60 GB of disks.

## Introduction and Purposes

While task 1 concentrated on named-entity recognition, task 2 of the BioCreative challenge gathered a larger set of traditional tasks: from text categorization (2.2), information extraction (2.1 and 2.2) up to information retrieval (2.3 and 2.4). Because, information retrieval has been earlier adressed in other forum [6], our work concentrates on tasks 2.1 and 2.1. Task 2.1 consists in selecting a text segment to support the GO annotation as supplied by SwissProt: the idea is to find a textual evidence that can support the GO annoation for example for quality checking purposes.Task 2.2 is a more classical automatic categorization tasks, where categories are GO terms.

## Preprocessing

As preliminary observations we noted that applying our tools on full text articles rather than on abstracts did require improving our pre-processing tools, especially to detect sentence boundaries. However, because pre-processing of full articles was more complex, we were not able to take advantage of it within the short time frame of the competition and therefore the following experiments were mostly conducted using abstracts.

For tasks 2.1 and 2.2 the same text-processing module is used in order to extract a textual segment likely to support the GO annotation. In both subtasks, the question of the length of the appropriate segment to be considered was crucial. Following what was learned at TREC for the information extraction task of the Genomic track, we assumed that sentences were likely to be relevant segments [1] for general information extraction. As result of this effort, frequent biomedical phenomena and specificities are now more carefully handled:

1. acronyms: “e.g.”;
2. person names: “J.P. Wallace” ;
3. decimal numbers and staging measures: “3.0”;
4. other frequent micro-grammatical phenomena such as sentences starting with lower case letters.

Because clean training data were missing, we decided not to investigate the use of machine learning approaches to solve the sentence pre-processing problem (as in [2]), and instead we decided to use simple manually crafted regular expressions. The tool relies on a set of finite-state automata, which are applied sequentially. Although the system is simple, it offers satisfying maintaining skills and results, which are about 97%, are similar to more advanced sentence boundary detectors.

[1] Contact author. Email: patrick.ruch@sim.hcuge.ch

## Information Extraction

Once sentence boundaries are identified, the segmented abstract is sent to the information extraction module. The title of the article is also added as if it would be a regular sentence. For this task, we decided to experiment a totally new method: we rely on a string edit distance ranking system. First the system computes a lexical distance (Dice-like) between each candidate sentence and the considered GO term. Then, this simple pattern matching step is completed by a fuzzy one (based on the Levenshtein distance: see for example [7] for a short introduction) in order to avoid simple string variations (like those allowed by stemming). Finally, sentences are ranked by similarity to the GO term, and the most similar sentence is chosen to support the GO annotation.

## Text Categorization

Usual text categorization systems (see [3] for a survey) use large sets of training data in order to induce a classification model. But again considering the lack of training data for GO annotation, we decided to rely -as far as possible- on data-independent approaches.

We used the text categorization system described in [4]. This tool indexes the collection of GO terms as if they were documents and then it treats each document (MedLine abstract) as if it was a query to be categorized in GO categories. Then, we use the score attributed to each GO categories to rank them. The system combined two retrieval engines: a vector space model (with tf.idf parameters) and a pattern-matcher. Indexes are built on the target collection of terms (GO). Two types of indexing units are used: stems (Porter-like) and linguistically motivated phrases (noun phrases). The UMLS thesaural resources are also used for string normalization. Formally, the GO annotation is seen as a retrieval task: the top n terms are attributed for each of the considered SwissProt entry (n is given for each protein).

Unfortunately, we were neither able to adapt the system for the GO categorization nor able to evaluate the system with the official BioCreative metrics, which were not known in detail yet [8] were results were submitted. In this context, the original system -i.e. fine-tuned for mapping MedLine abstracts to MeSH terms with ltc.lnn settings- was used for the GO annotation. The only differences concerned the indexed targets: GO terms were used instead of MeSH terms. In addition, some problematic features (all the “tc” and “ec” codes, which contains numerical features) were discounted because they were overweighed by regular tf.idf weighting schemas.

Preliminary evaluations with the GO ontology showed an important loss of precision when measured with 11-point average precision in comparison to MeSH categorization as shown in Table 1b; results at different point of recall are given in Table 1a for the GO categorization. While a dynamic threshold could have been used for the GO categorization task because the number of GO term per axes was a priori known, experiments reported in Table 1 were done selecting a static number of term: for MeSH categorization, the top-15 terms are selected; for GO categorization the top-5 terms are selected.

| | |
|---|---|
| Queryid (Num) for GO: | 622 |
| Total number of documents over all queries | |
| Retrieved: | 3110 |
| Relevant: | 1642 |
| Rel_ret: | 214 |
| Interpolated Recall - Precision Averages: | |
| at 0.00 | 0.1651 |
| at 0.10 | 0.1651 |
| at 0.20 | 0.1570 |
| at 0.30 | 0.1211 |
| at 0.40 | 0.0833 |
| at 0.50 | 0.0797 |
| at 0.60 | 0.0349 |
| at 0.70 | 0.0291 |
| at 0.80 | 0.0291 |
| at 0.90 | 0.0291 |
| at 1.00 | 0.0291 |

**Table 1a: precision at different recall values for GO categorization.**

| |
|---|
| Average precision for GO categorization |
| 0.0785 |
| |
| Average precision for MeSH categorization |
| 0.1904 |

**Table 1b: average precision for GO and MeSH terms, with similar settings: these optimal settings for MeSH categorization are reused for GO categorization without tuning.**

## Preliminary Conclusion and Future Work

For the text categorization task (subtask 2.2), temporary observations suggest that although MeSH terms and GO terms show a conceptual overlap [5], GO categorization is more complex than MeSH categorization. As for the subtasks (i.e. both 2.1 and 2.2) dedicated to the extraction of a relevant piece of text to support the GO annotation, we are now trying to merge the approach described in this report with the argumentative one [1] applied at TREC for the extraction of GeneRIFs (Gene Reference Into Functions) in LocusLink.

## Acknowledgments

We would like to thank the BioCreative team in general and Christian Blaschke in particular, who has been very responsive to our questions.

hThe study reported in this paper has been partially supported by the Swiss National Foundation within the IM2 NCCR; by a separate SNF grant (MedTag); and by the European Union via the SemanticMining Network of Excellence (FP6-507505).

Finally, we would like to thank Christine Chichester (GeneBio SA) for the assistance she provided along the competition.

## References

1. P Ruch, C Chichester, G Cohen, G Coray, F Ehrler, H Ghorbel, H Müller, V Pallotta. Report on the TREC 2003 Experiment: Genomic Track, *TREC* 2003, TREC Notebook Paper, Gaithersburg, 17-21 November 2003.

2. D Beeferman, A Berger, and J Lafferty, Statistical models for text segmentation. Machine Learning, (34):177-210, 1999. Special Issue on Natural Language Learning (C. Cardie and R. Mooney, eds).

3. F Sebastiani, Machine learning in automated text categorization. ACM Computing Surveys 34(1): 1-47, 2002.

4. P Ruch, R Baud, and A Geissbühler. Learning-free Text Categorization, *AIME* 2003, LNCS/LNAI 2780, Dojat M; Keravnou E; Barahona P (Eds.).

5. AT McCray, AC Browne, O Bodenreider. The lexical properties of the gene ontology. 504-8, AMIA 2002

6. Hersh, W., and Bhupatiraju, R.T. TREC genomics track overview. In *Notebook of the TREC-2003*, Gaithersburg, MD, 2003, 148-157.

7. P Ruch and R Baud. Evaluating and Reducing the Effect of Data Corruption when Applying Bag of Words Approaches to Medical Records. *Int J Med Inf*, 67 (1-3):75-83, 2002.

8. P.W.Lord, R.D. Stevens, A. Brass, and C.A.Goble. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275-83, 2003.