Building the Scientific Knowledge Mine (SciKnowMine[1]): a community-driven framework for text mining tools in direct service to biocuration

Cartic Ramakrishnan[3], William A. Baumgartner Jr.[1], Judith A. Blake[2], Gully APC Burns[3], K. Bretonnel Cohen[1], Harold Drabkin[2], Janan Eppig[2], Eduard Hovy[3], Chun-Nan Hsu[3], Lawrence E. Hunter[1], Tommy Ingulfsen[3], Hiroaki 'Rocky' Onda[2], Sandeep Pokkunuri[4], Ellen Riloff[4], Christophe Roeder[1], Karin Verspoor[1]

Affiliation information:
[1] University of Colorado Denver, PO Box 6511, MS 8303, Aurora, CO 80045, USA
[2] The Jackson Laboratory, 600 Main Street, Bar Harbor, Maine 04609 USA
[3]Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292, USA
[4] University of Utah, 50 S. Central Campus Drive, Rm 3190 MEB, Salt Lake City, UT 84112-9205

E-mail: cartic@isi.edu, William.Baumgartner@ucdenver.edu, judith.blake@jax.org, gully@usc.edu, kevin.cohen@gmail.com, hjd@informatics.jax.org, jte@informatics.jax.org, hovy@isi.edu, chunnan@isi.edu, Larry.Hunter@ucdenver.edu, tommying@isi.edu, honda@informatics.jax.org, sandeepp@cs.utah.edu, riloff@cs.utah.edu, Chris.Roeder@ucdenver.edu, Karin.Verspoor@ucdenver.edu

Abstract

Although there exist many high-performing text-mining tools to address literature biocuration (populating biomedical databases from the published literature), the challenge of delivering effective computational support for curation of large-scale biomedical databases is still unsolved. We describe a community-driven solution (the SciKnowMine Project) implemented using the Unstructured Information Management Architecture (UIMA) framework. This system's design is intended to provide knowledge engineering enhancement of pre-existing biocuration systems by providing a large-scale text-processing pipeline bringing together multiple Natural Language Processing (NLP) toolsets for use within well-defined biocuration tasks. By working closely with biocurators at the Mouse Genome Informatics[2] (MGI) group at The Jackson Laboratory in the context of their everyday work, we break down the biocuration workflow into components and isolate specific targeted elements to provide maximum impact. We envisage a system for classifying documents based on a series of increasingly specific classifiers, starting with very simple surface-level decision criteria and gradually introducing more sophisticated techniques. This classification pipeline will be applied to the task of identifying papers of interest to mouse genetics (primary MGI document triage), thus facilitating the input of documents into the MGI curation pipeline. We also describe other biocuration challenges (gene normalization) and how our NLP-framework based approach could be applied to them.

1. Introduction

In biomedical research, organizations such as the NIH funded model organism databases or the Cochrane Collaboration systematically scan, read, evaluate and organize the published literature to provide formally structured resources that summarize individual fields of research. This effort, termed 'literature biocuration', is widely recognized as important to the scientific community (Bourne et al. 2006), and the promise that text-mining systems will be able to assist biocuration is long standing and well supported (Rebholz-Schuhmann et al. 2005).

Although suitable natural language processing (NLP) methods that can support biocuration (Hersh et al. 2005) exist, we assert that authentic computational support for biocuration work has not yet been delivered to the places where it is most needed. In this position paper, we describe the possible role that a framework-based approach might play in accomplishing this goal. In collaboration with the Mouse Genome Informatics group (MGI) (Bult et al. 2010), we seek to provide the necessary scalable computational support to speed up MGI biocuration. Every month, the biocuration staff process the contents of roughly 200 separate scientific journals in order to determine if each paper needs to be read in more depth or can be discarded as irrelevant to MGI's mission (this process is known as 'document triage'). Despite some success in measures of utility, systems developed in shared evaluations (Hersh, Cohen et al. 2005) have not been incorporated into the MGI curation workflow. We describe a community-based cyberinfrastructure project (called 'SciKnowMine') specifically designed *to accelerate biocuration*.

2. Goals of BioNLP development

Biomedical NLP (BioNLP) development should involve the creation of novel algorithms, systems and solutions that satisfy well-established global metrics. Such metrics permit the community to evaluate which methods are the most effective for a given task (as it does now).

[1] The SciKnowMine project is funded by NSF grant #0849977 and supported by U24 RR025736-01, NIGMS: RO1-GM083871, NLM: 2R01LM009254, NLM:2R01LM008111, NLM:1R01LM010120-01, NHGRI:5P41HG000330
[2] http://www.informatics.jax.org

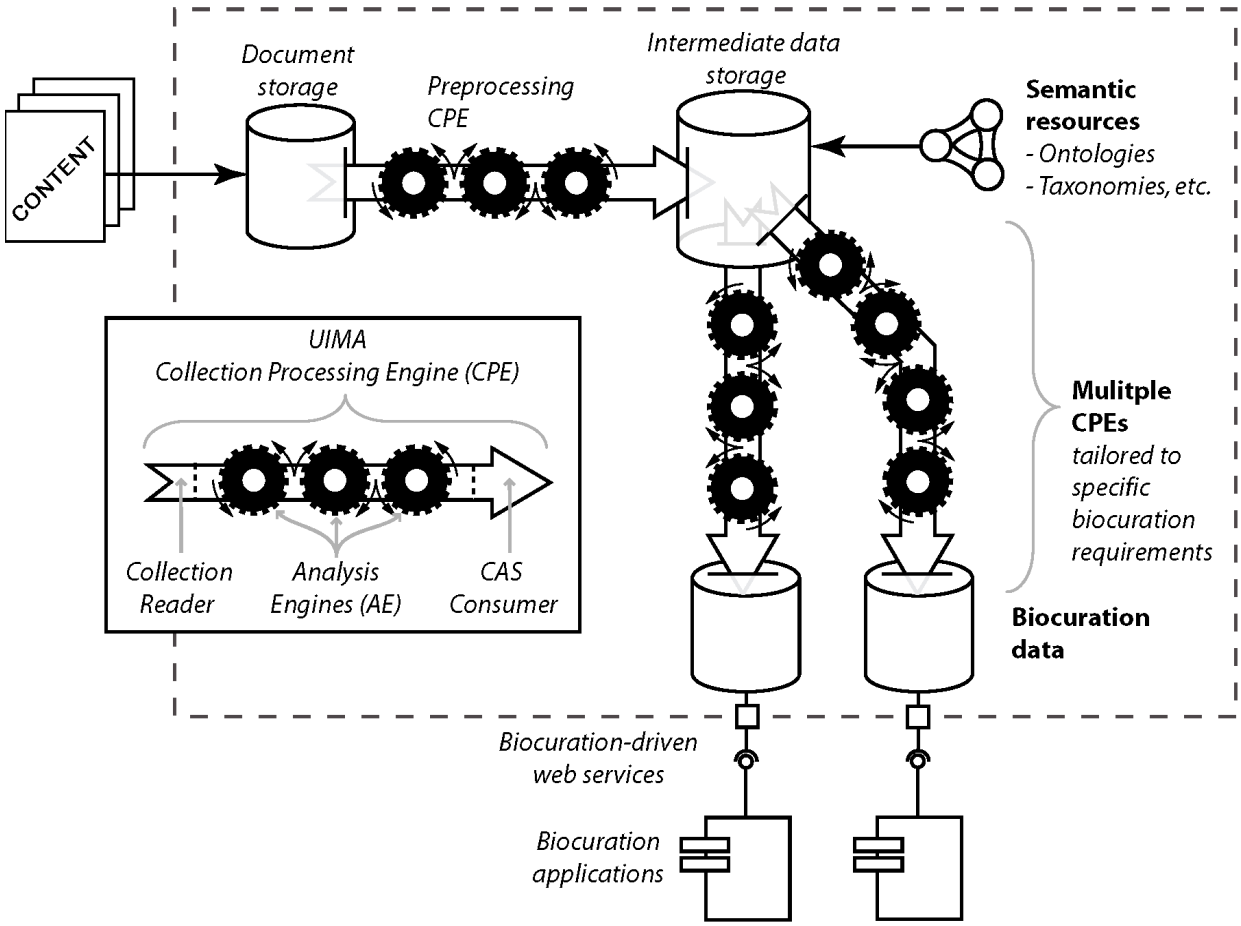

Figure 1 High-level design of SciKnowMine

We argue that this is not enough. It is essential that there exist a practical methodology for these systems to be used *in-situ* within biomedical databases' biocuration teams. There are a variety of individual components with similar high-level functions (*e.g.,* the classification of text spans within documents) from different research groups. Examples of such tools are the NCBO Annotator (Jonquet et al. 2009) that enables annotation of arbitrary concepts in text from existing biomedical ontologies, GOPubMed which organizes PubMed abstracts according to MeSH and Gene Ontology terms contained within them (Doms et al. 2005), the proprietary ProMiner system for terminology recognition in text (Hanisch et al. 2005), and the TextPresso system which recognizes terms based on regular expression (Müller et al. 2004). While each of these tools is potentially useful, various challenges have been encountered when trying to integrate them into a biocuration workflow, ranging from technical or licensing difficulties to inherent limitations in the applicability of the tools to the relevant full-text publications the biocurators work with.

A reliable middleware infrastructure providing a common platform where different components can easily be deployed together to develop a complete tool for an individual biocuraton task is therefore necessary.

The infrastructure used to support community evaluations is closely related to this idea, since these similar components must run on a shared system in order to be compared accurately. For example, the BioCreative meta-server[3] (Leitner et al. 2008) and the U-compare framework[4] (Kano et al. 2009) used in the BioNLP'09 evaluation each provide a mechanism for evaluating multiple systems against a common task, However, it has been recently argued that systems that perform well in tasks with intrinsic measures (such as F-Score, Precision and Recall) do not necessarily accelerate biocuration (Alex et al. 2008; Caporaso et al. 2008). This situation is being addressed in future evaluations (such as the BioCreative III workshop), but this raises the question of how performance can be measured. Extrinsic measures require direct measurement of usefulness within the context of a real-world use case whereas evaluation measures used in these shared tasks must be a computable substitute for the 'true measure'. In developing such a measure, it is therefore imperative that, at some stage, these systems are made to run within the direct context of a live biocuration task in a biomedical database.

A crucial barrier to success in this project is access to the full text biomedical literature for automated processing (Dowell et al. 2009). Although electronic access to individual publications is widely available through resources such as the National Library of Medicine's PubMedCentral and commercial subscription

[3] http://bcms.bioinfo.cnio.es/
[4] http://u-compare.org/bionlp2009.html

servers such as Elsevier ScienceDirect®, these mechanisms do not provide bulk access to entire collections, as is necessary for automated processing of the full range of publications relevant to biocuration. Attempts to use APIs or web interfaces to get systematic access to entire collections are blocked, despite the apparent rights of subscribers, or in the case of PubMedCentral, the public, to this material. A critical need is to reconcile the requirements for automated processing of publications in bulk with the concerns and technological capabilities of publishers and the National Library of Medicine. Our approach of reusing existing NLP components as open-source software for use by the BioNLP community will make NLP expertise available for publishers who would otherwise not have access to this technology. Since access to full-text articles for these text-mining efforts is a stumbling block, discussions regarding these issues are underway with Elsevier labs. We are building a single demonstration implementation that can produce measurable acceleration of the biocuration process at MGI coupled with active development of novel NLP technology. This work requires a collaborative effort between multiple groups (U. Colorado, USC ISI, U. Utah and JAX).

3. Goals of BioNLP development

SciKnowMine is a large-scale text-processing pipeline based on the BioNLP-UIMA project (Baumgartner et al. 2008). UIMA is the Unstructured Information Management Architecture (Ferrucci et al. 2004), and is available as an Apache open source project. As shown in Figure 1, SciKnowMine brings together multiple BioNLP toolsets in a UIMA implementation. UIMA provides a way of defining 'Collection Processing Engines' (CPEs). Each CPE is defined in a three stage cascade consisting of a 'Collection Reader' (which iterates over documents to initiate processing), a series of 'Analysis Engines' (which add meta-data, often in the form of text annotations, to the CAS or 'Common Analysis Structure' UIMA document representation), and finally a 'CAS Consumer' (which formats the final CAS data and writes it to output). The first step of our processing is designed to upload remote files, store them locally, execute a set of standard preprocessing tasks (*e.g.* tokenization, sentence splitting, etc.) and then store a local set of partially annotated data. We propose to wrap several different implementations of these standard preprocessing tasks as AEs (see also Section 3.4). We will then develop a library of CPEs specifically tailored for biocuration tasks that operate on this set of pre-annotated texts. These CPEs will be made available as web-services that specifically deliver biocuration functionality to end-user systems in a structured way. Our objectives are to prototype, develop and scale up this infrastructure in order to convert the entire primary research literature into an online resource that can then be uniformly accessed by both BioNLP researchers and biocuration teams, enabling the development of powerful new accelerated methods of biocuration.

3.1 A Repository of UIMA Analysis Engines

SciKnowMine takes a community-centered approach to building its biocuration workflows. We use publicly available tools when applicable and construct new tools when needed and release these new tools as open source software. We take advantage of existing public repositories containing UIMA components including (1) the BioNLP-UIMA framework (2) resources from the Julie lab[5] (3) the UIMA sandbox and (4) U-Compare (Kano, Baumgartner et al. 2009). Every UIMA component is dependent on a defined set of data types that are specified by that component's 'type system'. One challenge is that we must integrate across multiple UIMA type systems since components that use different type systems cannot work with each other. The SciKnowMine infrastructure uses a generic, domain-independent type system capable of expressing the wide range of types necessary to support biocuration (Verspoor et al. 2009). It is easily adaptable to external type systems, and is already compatible with U-Compare. With this approach, as SciKnowMine progresses, it will not only result in a series of biocuration workflows, but will also amass a collection of UIMA components compatible with a single type system that could be made publicly available.

Almost all the components discussed in this paper already exist, mostly taken from well-known external resources. Our planned contribution is assembling them into a single framework and extending the modules where necessary to obtain a useful, scalable and seamless end-to-end support system for MGI. This will be made available to the MGI biocuration team as a web service. We will make public Collection Processing Engines (CPEs) for certain exemplar text processing tasks along with the component Analysis Engines (AE) and collection readers for such tasks. These will serve as templates for the community to deploy and test their implementations of individual AEs on a large corpus of biomedical text relevant to focused biomedical research groups like MGI.

3.2 Scaling up SciKnowMine

We will explore three ways to scale up processing in order to meet the goal of analyzing large collections of text in reasonable amount of time. Methods for scaling up UIMA processing range from adding more threads to the CPE processor (the Collection Processing Manager, CPM), to scaling out to more hardware using UIMA-AS, to Hadoop cloud computing[6]. Given a large, shared-memory, multi-processor machine, a lot can be accomplished by specifying more threads in a CPE. The pipeline's primary data structure, the CAS, is shared between engines in memory. Running more than one instance of the pipeline allows for parallel execution of each engine, including any relatively slow engines. As a scaling method in UIMA, this is effective, but limited.

[5] http://www.julielab.de/Resources/Software
[6] http://hadoop.apache.org

Each engine in the pipeline is duplicated for each thread, so engines that consume a large amount of memory would also be duplicated.

Since the discrete nature of document processing allows for independent processing, a small cluster of single CPU, multi-core machines can be as effective as a single more powerful shared-memory machine, at a fraction of the cost. Using UIMA-AS (Asynchronous Scale-out) allows different analysis engines from a single CPE to be located on different machines. In this way, slow engines that need multiple instances can be duplicated, while faster engines can be run with single instances with engines connected through the use of message queues. This allows for asynchronous access and makes distributing work and collecting results easy. UIMA has been adapted to Hadoop, a MapReduce (Dean et al. 2004) implementation, in a project called Behemoth[7]. It makes use of UIMA Processing Engine Archive (PEAR) packaging so that Hadoop and the Hadoop Distributed File System (HDFS) can manage distributing the code and data files across a much larger cluster. Not everyone has access to thousands of nodes, but Hadoop cluster time is available for rent on Amazon's Elastic Compute Cloud (EC2)[8] as Elastic MapReduce[9]. The economics make it worth considering.

3.3 Access to PDF content

Given the ubiquity and familiarity of PDF documents, building effective methods for extracting and processing text from PDF files is a high priority. We use a combination of machine-learning and rule-based approaches to render and extract text as a UIMA Collection Reader. This approach is an open-source component that has been used in text mining studies of neuroanatomical experiments (Burns et al. 2007). We plan to make this PDF extraction technology available as an open-source UIMA analysis engine.

3.4 Incorporating new NLP research

We intend to incorporate novel NLP approaches into our system by using UIMA as a central representational framework and working with NLP researchers to build methods to wrap their tools as UIMA components. We follow the general approach of having each NLP system produce annotations that are attached to the text(s) in appropriate places, resulting in a steady accretion of information within and around each text. We follow the stand-off model of annotation which is inherent to the UIMA data structures. Annotations produced by multiple components – even annotations of the same type, e.g. different tokenizations or gene mention annotations – can exist in parallel and be made available for downstream analysis. Each downstream component can choose to use whichever annotations it believes to be the most useful for its task, perhaps even using multiple sets of annotations of the same kind (e.g., a component could utilize the annotations produced by three different named entity recognizers to maximize coverage).

[7] http://code.google.com/p/behemoth-pebble
[8] http://aws.amazon.com/ec2/
[9] http://aws.amazon.com/elasticmapreduce/

4. Knowledge Engineering Study of the MGI Biocuration Workflow

A key feature of this project is our approach to understanding the biocuration workflow being used at MGI. This approach is modeled loosely after the CommonKADS methodology (Schreiber et al. 1999). Using the UIMA framework it is possible to deploy an automated biocuration engine with relative ease. However, given the well-known shortcomings of automating biocuration shown in previous work, we plan to use the system as a human aide, and thus to integrate it with the human curators' workflow. In order to minimally disrupt the existing well-honed procedures and to obtain as much guidance for automated processing as possible, this integration requires careful consideration of several issues.

- At which point during the manual biocuration should the intermediate results of automated curation be made known to the biocurators?
- How should the system inform the biocurator of these results so as to be least intrusive?

These issues have motivated us to conduct studies in modeling workflows of manual biocuration. We used UML 2.0 activity diagrams to model the activities of different curator teams to extract information from the literature. Although this approach is not strictly formal, it does provide a useful framework for exploring questions such as: 'Which tasks take the longest?' 'Where are the most prominent curation bottlenecks?' Our preliminary investigations have helped us identify three MGI curation operations that are candidates for acceleration via computational support.

4.1 MGI Triage Automation Tools

We view the triage task as a document classification task that ranks documents in order of likelihood of interest for further analysis. Biocurators can then vary parameters to learn how they characterize the likelihood thresholds to include documents in the system or not. Our approach is to build a series of increasingly specific classifiers, starting with very simple surface-level decision criteria and gradually introducing more sophisticated NLP. The current baseline is that a document is included if it contains the words *mouse, mice* or *murine* (unless the words appear in the Bibliography section only). Subsequent levels involve setting zone-specific classification decisions (such as the presence of 'stigma words' within methods sections, *etc.*), the use of word combinations (bigrams, trigrams, *etc.*) in these decisions, the use of topic model signatures derived from language modeling, and at the highest level the development of structured linguistic information extraction frames. In keeping with our objective of minimal disruption of the manual biocuration process, our automated triage system will rank the documents downloaded and provide for

each one its classification suggestion(s), together with an indication of its confidence. Human curators will use this to determine the confidence level at which the system's judgments are trustworthy.

4.2 Gene Normalization Tools

'Gene normalization' refers to a mapping of mentions of genes or proteins in text to an appropriate database identifier. This is challenging due to species ambiguity in the text (genes in different organisms often share names) and the widespread use of acronyms and abbreviations. Solutions to this problem could be integrated into a biocuration process to help curators assess the relevance of a particular paper to their target area, as well as focus the curator's attention to specific parts of the text that mention particular genes. Gene normalization has been the focus of several recent challenge tasks in BioCreative II (Krallinger et al. 2008) and II.5 (Mardis et al. 2009). The state-of-the-art performance is currently achieved by the GNAT system (Hakenberg et al. 2008). Currently, MGI is incorporating gene normalization tools independently of the triage process. Our task would be to incorporate such a tool into the triage task.

4.3 Event Recognition Tools

Protein-protein interactions have been the most common candidate for biological event extraction from the earliest studies (Blaschke et al. 1999; Craven et al. 1999), to the latest competitions like BioCreative II and II.5. Research has also extended to other types including those focused on in the recent BioNLP'09 challenge (Kim et al. 2009): (a) gene expression, (b) transcription, (c) protein catabolism, (d) protein localization, (e) binding, (f) phosphorylation, (g) regulation, (h) positive regulation, and (i) negative regulation. The needs of MGI will require extension to novel composite semantic types, such as 'phenotype'. Phenotypes are observable attributes of an organism caused by myriad underlying factors. Identifying them requires extracting information on chromosomal locations, polymorphisms, Gene Ontology terms, protein domains, and experimental assays; all of these information extraction tasks are either novel or demonstrably difficult but if solved, could have a large impact. We have begun experiments with information extraction pattern learning (Riloff 1996) in order to address some of these tasks.

5. Conclusion

The work described in this paper is currently in the preliminary stages. We have collected a representative corpus of documents to serve as training data for classifiers within the biocuration pipeline, and begun the design of the classifier. We have also engaged the MGI biocurators in a requirement elicitation process to build models of their workflows. Experiments are also underway to tune our PDF extraction system to extract text from the MGI journals.

We have described a fundamental (even formative) unsolved challenge in the field of BioNLP and present a community driven approach that directly leverages NLP Frameworks to solve it. SciKnowMine is an effort to leverage the BioNLP community's expertise to solve that challenge in a general way that can be used across different biocuration systems.

6. References

Alex, B., C. Grover, et al. (2008). "Assisted curation: does text mining really help?" Pacific Symposium Biocomputing: 556-67.

Baumgartner, W., B. Cohen, et al. (2008). "An open-source framework for large-scale, flexible evaluation of biomedical text mining systems." Journal of Biomedical Discovery and Collaboration 3: 1.

Blake, J., J. Eppig, et al. (2006). "The Mouse Genome Database (MGD): updates and enhancements." Nucleic Acids Research 34(suppl_1): D562-567.

Blaschke, C., M. A. Andrade, et al. (1999). "Automatic extraction of biological information from scientific text: protein-protein interactions." ISMB: 60-67.

Bourne, P. E. and J. McEntyre (2006). "Biocurators: contributors to the world of science." PLoS Computational Biology 2(10): e142.

Bult, C. J., J. A. Kadin, et al. (2010). "The Mouse Genome Database: enhancements and updates." Nucleic Acids Research 38(Database issue): D586-92.

Burns, G., D. Feng, et al. (2007). Infrastructure for Annotation-Driven Information Extraction from the Primary Scientific Literature: Principles and Practice. 1st IEEE Intl. Workshop on Service Oriented Technologies for Biological Databases and Tools (SOBDAT 2007), Salt-Lake City.

Caporaso, J. G., N. Deshpande, et al. (2008). "Intrinsic evaluation of text mining tools may not predict performance on realistic tasks." Pacific Symposium Biocomputing: 640-51.

Clement Jonquet, Nigam H. Shah, Mark A. Musen, The Open Biomedical Annotator, AMIA Summit on Translational Bioinformatics, p. 56-60, March 2009, San Francisco, CA, USA.

Craven, M. and J. Kumlien (1999). Constructing biological knowledge-bases by extracting information from text sources. Proceedings of the Seventh ISMB.

Dean, J. and S. Ghemawat (2004). MapReduce: Simplified Data Processing on Large Clusters. OSDI 2004.

Doms, A. and M. Schroeder (2005). "GoPubMed: exploring PubMed with the Gene Ontology." Nucleic Acids Research 33(suppl_2): W783-786.

Dowell, K. G., M. S. McAndrews-Hill, et al. (2009). "Integrating text mining into the MGI biocuration workflow." Database : The Journal of biological databases and curation 2009(0).

Ferrucci, D. and A. Lally (2004). "Building an example application with the unstructured information management architecture." IBM Syst. J. 43(3): 455-475.

Hakenberg, J., C. Plake, et al. (2008). "Inter-species normalization of gene mentions with GNAT." Bioinformatics 24(16): i126-132.

Hanisch, D., K. Fundel, et al. (2005). "ProMiner: rule-based protein and gene entity recognition." BMC Bioinformatics 6(Suppl 1): S14.

Hersh, W., A. Cohen, et al. (2005). "TREC 2005 Genomics Track Overview."

Kano, Y., W. Baumgartner, et al. (2009). "U-Compare: share and compare text mining tools with UIMA." Bioinformatics 25(15): 1997-1998.

Kim, J.-D., T. Ohta, et al. (2009). Overview of BioNLP'09 shared task on event extraction. Proceedings of the Workshop on BioNLP: Shared Task. Boulder, Colorado, ACL.: 1-9.

Krallinger, M., A. Morgan, et al. (2008). "Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge." Genome Biology 9(Suppl 2).

Leitner, F., M. Krallinger, et al. (2008). "Introducing meta-services for biomedical information extraction." Genome Biology **9**(Suppl 2): S6.

Müller, H.-M., E. Kenny, et al. (2004). "Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature." PLoS Biol **2**(11): e309

Mardis, S., F. Leitner, et al. (2009). BioCreative II.5: Evaluation and ensemble system

Rebholz-Schuhmann, D., H. Kirsch, et al. (2005). "Facts from text--is text mining ready to deliver?" PLoS Biol 3(2): e65.

Riloff, E. (1996). Automatically Generating Extraction Patterns from Untagged Text. (AAAI-96), 1996, pp. 1044-1049.

Schreiber, G., H. Akkermans, et al. (1999). Knowledge Engineering and Management: The CommonKADS Methodology, {The MIT Press}.

Verspoor, K., W. Baumgartner, et al. (2009). Abstracting the Types away from a UIMA Type System. From Form to Meaning: Processing Texts Automatically. C. Chiarcos, Eckhart de Castilho, Stede, M.: 249-256.