The Role of Information Extraction in the Design of a Document Triage Application for Biocuration

Sandeep Pokkunuri
School of Computing
University of Utah
Salt Lake City, UT
sandeepp@cs.utah.edu

Cartic Ramakrishnan
Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA
cartic@isi.edu

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT
riloff@cs.utah.edu

Eduard Hovy
Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA
hovy@isi.edu

Gully APC Burns
Information Sciences Institute
Univ. of Southern California
Marina del Rey, CA
burns@isi.edu

Abstract

Traditionally, automated triage of papers is performed using lexical (unigram, bigram, and sometimes trigram) features. This paper explores the use of information extraction (IE) techniques to create richer linguistic features than traditional bag-of-words models. Our classifier includes lexico-syntactic patterns and more-complex features that represent a pattern coupled with its extracted noun, represented both as a lexical term and as a semantic category. Our experimental results show that the IE-based features can improve performance over unigram and bigram features alone. We present intrinsic evaluation results of full-text document classification experiments to determine automatically whether a paper should be considered of interest to biologists at the Mouse Genome Informatics (MGI) system at the Jackson Laboratories. We also further discuss issues relating to design and deployment of our classifiers as an application to support scientific knowledge curation at MGI.

1 Introduction

A long-standing promise of Biomedical Natural Language Processing is to accelerate the process of literature-based 'biocuration', where published information must be carefully and appropriately translated into the knowledge architecture of a biomedical database. Typically, biocuration is a manual activity, performed by specialists with expertise in both biomedicine and the computational representation of the target database. It is widely acknowledged as a vital lynch-pin of biomedical informatics (Bourne and McEntyre, 2006).

A key step in biocuration is the initial triage of documents in order to direct to specialists only the documents appropriate for them. This classification (Cohen and Hersh, 2006)(Hersh W, 2005) can be followed by a step in which desired information is extracted and appropriately standardized and formalized for entry into the database. Both these steps can be enhanced by suitably powerful Natural Language Processing (NLP) technology. In this paper, we address text mining as a step within the broader context of developing both infrastructure and tools for biocuration support within the Mouse Genome Informatics (MGI) system at the Jackson Laboratories. We previously identified 'document triage' as a crucial bottleneck (Ramakrishnan et al., 2010) within MGI's biocuration workflow.

Our research explores the use of information extraction (IE) techniques to create richer linguistic features than traditional bag-of-words models. These features are employed by a classifier to perform the triage step. The features include lexico-syntactic patterns as well as more-complex features, such as a pattern coupled with its extracted noun, where the noun is represented both as a lexical term and by its semantic category. Our experimental results show that the IE-based enhanced features can improve performance over unigram and bigram features alone.

Proceedings of the 2011 Workshop on Biomedical Natural Language Processing, ACL-HLT 2011, pages 46–55,
Portland, Oregon, USA, June 23-24, 2011. ©2011 Association for Computational Linguistics

Evaluating the performance of BioNLP tools is not trivial. So-called *intrinsic* metrics measure the performance of a tool against some gold standard of performance, while *extrinsic* ones (Alex et al., 2008) measure how much the overall biocuration process is benefited. Such metrics necessarily involve the deployment of the software in-house for testing by biocurators, and require a large-scale software-engineering infrastructure effort. In this paper, we present intrinsic evaluation results of full-text document classification experiments to determine automatically whether a paper should be considered of interest to MGI curators. We plan in-house deployment and extrinsic evaluation in near-term work.

Our work should be considered as the first step of a broader process within which (a) the features used in this particular classification approach will be re-engineered so that they may be dynamically recreated in any new domain by a reusable component, (b) this component is deployed into reusable infrastructure that also includes document-, annotation- and feature-storage capabilities that support scaling and reuse, and (c) the overall functionality can then be delivered as a software application to biocurators themselves for extrinsic evaluation in any domain they choose. Within the 'SciKnowMine' project, we are constructing such a framework (Ramakrishnan et al., 2010), and this work reported here forms a prototype component that we plan to incorporate into a live application. We describe the underlying NLP research here, and provide context for the work by describing the overall design and implementation of the SciKnowMine infrastructure.

1.1 Motivation

MGI's biocurators use very specific guidelines for triage that continuously evolve. These guidelines are tailored to specific subcategories within MGI's triage task (phenotype, Gene Ontology[1] (GO) term, gene expression, tumor biology and chromosomal location mapping). They help biocurators decide whether a paper is relevant to one or more subcategories. As an example, consider the guideline for the phenotype category shown in Table 1.

This example makes clear that it is not sufficient to match on relevant words like 'transgene' alone.

[1] http://www.geneontology.org/

'Select paper **If**: it is about transgenes where a gene from any species is inserted in mice **and** this results in a phenotype. **Except**: if the paper uses transgenes to examine promoter function'.

Table 1: Sample triage guideline used by MGI biocurators

To identify a paper as being '*within-scope*' or '*out-of-scope*' requires that a biocurator understand the context of the experiment described in the paper. To check this we examined two sample papers; one that matches the precondition of the above rule and another that matches its exception. The first paper (Sjögren et al., 2009) is about a transgene insertion causing a pheotype and is a positive example of the category phenotype, while the second paper (Bouatia-Naji et al., 2010) is about the use of transgenes to study promoter function and is a negative example for the same category.

Inspection of the negative-example paper illustrates the following issues concerning the language used: (1) This paper is about transgene-use in studying promoter function. Understanding this requires the following background knowledge: (a) the two genes mentioned in the title are transgenes; (b) the phrase 'elevation of fasting glucose levels' in the title represents an up-regulation phenotype event. (2) Note that the word 'transgene' never occurs in the entire negative-example paper. This suggests that recognizing that a paper involves the use of transgenes requires annotation of domain-specific entities and a richer representation than that offered by a simple bag-of-words model.

Similar inspection of the positive-example paper reveals that (3) the paper contains experimental evidence showing the phenotype resulting from the transgene insertion. (4) The 'Materials and Methods' section of the positive-example paper clearly identifies the construction of the transgene and the 'Results' section describes the development of the transgenic mouse model used in the study. (3) and (4) above suggest that domain knowledge about complex biological phenomena (events) such as phenotype and experimental protocol may be helpful for the triage task.

Together, points (1)–(4) suggest that different sections of a paper contain additional important context-specific clues. The example highlights the complex nature of the triage task facing the MGI biocurators. At present, this level of nuanced 'understanding' of content semantics is extremely hard for machines to replicate. Nonetheless, merely treating the papers as a bag-of-words is unlikely to make nuanced distinctions between positive and negative examples with the level of precision and recall required in MGI's triage task.

In this paper we therefore describe: (1) the design and performance of a classifier that is enriched with three types of features, all derived from information extraction: (a) lexico-syntactic patterns, (b) patterns coupled with lexical extractions, and (c) patterns coupled with semantic extractions. We compare the enriched classifier against classifiers that use only unigram and bigram features; (2) the design of a biocuration application for MGI along with the first prototype system where we emphasize the infrastructure necessary to support the engineering of domain-specific features of the kind described in the examples above. Our application is based on Unstructured Information Management Architecture (UIMA) (Ferrucci and Lally, 2004), which is a pipeline-based framework for the development of software systems that analyze large volumes of unstructured information.

2 Information Extraction for Triage Classification

In this section, we present the information extraction techniques that we used as the basis for our IE-based features, and we describe the three types of IE features that we incorporated into the triage classifier.

2.1 Information Extraction Techniques

Information extraction (IE) includes a variety of techniques for extracting factual information from text. We focus on pattern-based IE methods that were originally designed for event extraction. Event extraction systems identify the role fillers associated with events. For example, consider the task of extracting information from disease outbreak reports, such as ProMed-mail articles (http://www.promedmail.org/). In contrast to a named entity recognizer, which should identify all mentions of diseases and people, an event extraction system should only extract the diseases involved in an outbreak incident and the people who were the victims. Other mentions of diseases (*e.g.*, in historical discussions) or people (*e.g.*, doctors or scientists) should be discarded.

We utilized the Sundance/AutoSlog software package (Riloff and Phillips, 2004), which is freely available for research. Sundance is an information extraction engine that applies lexico-syntactic patterns to extract noun phrases from specific linguistic contexts. Sundance performs its own syntactic analysis, which includes morphological analysis, shallow parsing, clause segmentation, and syntactic role assignment (*i.e.*, identifying subjects and direct objects of verb phrases). Sundance labels verb phrases with respect to active/passive voice, which is important for event role labelling. For example, *"Tom Smith was diagnosed with bird flu"* means that Tom Smith is a victim, but *"Tom Smith diagnosed the elderly man with bird flu"* means that the elderly man is the victim.

Sundance's information extraction engine can apply lexico-syntactic patterns to extract noun phrases that participate in syntactic relations. Each pattern represents a linguistic expression, and extracts a noun phrase (NP) argument from one of three syntactic positions: Subject, Direct Object, or Prepositional Phrase. Patterns may be defined manually, or they can be generated by the AutoSlog pattern generator (Riloff, 1993), which automatically generates patterns from a domain-specific text corpus. AutoSlog uses 17 syntactic 'templates' that are matched against the text. Lexico-syntactic patterns are generated by instantiating the matching words in the text with the syntactic template. For example, five of AutoSlog's syntactic templates are shown in Table 2:

(a) <SUBJ> PassVP
(b) PassVP Prep <NP>
(c) <SUBJ> ActVP
(d) ActVP Prep <NP>
(e) Subject PassVP Prep <NP>

Table 2: Five example syntactic templates (PassVP means passive voice verb phrase, ActVP means active voice verb phrase)

Pattern (a) matches any verb phrase (VP) in a passive voice construction and extracts the Subject of the VP. Pattern (b) matches passive voice VPs that are followed by a prepositional phrase. The NP in the prepositional phrase is extracted. Pattern (c) matches any active voice VP and extracts its Subject, while Pattern (d) matches active voice VPs followed by a prepositional phrase. Pattern (e) is a more complex pattern that requires a specific Subject[2], passive voice VP, and a prepositional phrase. We applied the AutoSlog pattern generator to our corpus (described in Section 3.1) to exhaustively generate every pattern that occurs in the corpus.

As an example, consider the following sentence, taken from an article in PLoS Genetics:

> *USP14 is endogenously expressed in HEK293 cells and in kidney tissue derived from wt mice.*

<SUBJ> PassVP(expressed)
<SUBJ> ActVP(derived)
PassVP(expressed) Prep(in) <NP>
ActVP(derived) Prep(from) <NP>
Subject(USP14) PassVP(expressed) Prep(in) <NP>

Table 3: Lexico-syntactic patterns for the PLoS Genetics sentence shown above.

AutoSlog generates five patterns from this sentence, which are shown in Table 3:

The first pattern matches passive voice instances of the verb *'expressed'*, and the second pattern matches active voice instances of the verb *'derived'*.[3] These patterns rely on syntactic analysis, so they will match any syntactically appropriate construction. For example, the first pattern would match *'was expressed', 'were expressed', 'have been expressed'* and *'was very clearly expressed'*. The third and fourth patterns represent the same two VPs but also require the presence of a specific prepositional phrase. The prepositional phrase does not need to be adjacent to the VP, so long as it is attached to the VP syntactically. The last pattern is very specific and will only match passive voice instances of *'expressed'* that also have a Subject with a particular head noun (*'USP14'*) and an attached prepositional phrase with the preposition *'in'*.

The example sentence contains four noun phrases, which are underlined. When the patterns generated by AutoSlog are applied to the sentence, they produce the following NP extractions (shown in **boldface** in Table 4):

<**USP14**> PassVP(expressed)
<**kidney tissue**> ActVP(derived)
PassVP(expressed) Prep(in) <**HEK293 cells**>
ActVP(derived) Prep(from) <**wt mice**>
Subject(USP14) PassVP(expressed) Prep(in) <**HEK293 cells**>

Table 4: Noun phrase extractions produced by Sundance for the sample sentence.

In the next section, we explain how we use the information extraction system to produce rich linguistic features for our triage classifier.

2.2 IE Pattern Features

For the triage classification task, we experimented with four types of IE-based features: *Patterns*, *Lexical Extractions*, and *Semantic Extractions*.

The *Pattern* features are the lexico-syntactic IE patterns. Intuitively, each pattern represents a phrase or expression that could potentially capture contexts associated with mouse genomics better than isolated words (unigrams). We ran the AutoSlog pattern generator over the training set to exhaustively generate every pattern that appeared in the corpus. We then defined one feature for each pattern and gave it a binary feature value (*i.e.*, 1 if the pattern occurred anywhere in the document, 0 otherwise).

We also created features that capture not just the pattern expression, but also its argument. The *Lexical Extraction* features represent a pattern paired with the head noun of its extracted noun phrase. Table 5 shows the Lexical Extraction features that would be generated for the sample sentence shown earlier. Our hypothesis was that these features could help to distinguish between contexts where an activity is relevant (or irrelevant) to MGI because of the combination of an activity and its argument.

The Lexical Extraction features are very specific, requiring the presence of multiple terms. So we

[2]Only the head nouns must match.

[3]Actually, the second clause is in reduced passive voice (*i.e.*, tissue that was derived from mice), but the parser misidentifies it as an active voice construction.

PassVP(expressed), **USP14**
ActVP(derived), **tissue**
PassVP(expressed) Prep(in), **cells**
ActVP(derived) Prep(from), **mice**
Subject(USP14) PassVP(expressed) Prep(in), **cells**

Table 5: Lexical Extraction features

also experimented with generalizing the extracted nouns by replacing them with a semantic category. To generate a semantic dictionary for the mouse genomics domain, we used the Basilisk bootstrapping algorithm (Thelen and Riloff, 2002). Basilisk has been used previously to create semantic lexicons for terrorist events (Thelen and Riloff, 2002) and sentiment analysis (Riloff et al., 2003), and recent work has shown good results for bioNLP domains using similar bootstrapping algorithms (McIntosh, 2010; McIntosh and Curran, 2009).

As input, Basilisk requires a domain-specific text corpus (unannotated) and a handful of seed nouns for each semantic category to be learned. A bootstrapping algorithm then iteratively hypothesizes additional words that belong to each semantic category based on their association with the seed words in pattern contexts. The output is a lexicon of nouns paired with their corresponding semantic class. (e.g., *liver :* BODY PART).

We used Basilisk to create a lexicon for eight semantic categories associated with mouse genomics: BIOLOGICAL PROCESS, BODY PART, CELL TYPE, CELLULAR LOCATION, BIOLOGICAL SUBSTANCE, EXPERIMENTAL REAGENT, RESEARCH SUBJECT, TUMOR. To choose the seed nouns, we parsed the training corpus, ranked all of the nouns by frequency[4], and selected the 10 most frequent, unambiguous nouns belonging to each semantic category. The seed words that we used for each semantic category are shown in Table 6.

Finally, we defined *Semantic Extraction* features as a pair consisting of a pattern coupled with the semantic category of the noun that it extracted. If the noun was not present in the semantic lexicons, then no feature was created. The Basilisk-generated lexicons are not perfect, so some entries will be incorrect. But our hope was that replacing the lexical terms with semantic categories might help the classifier learn more general associations. For example, "PassVP(expressed) Prep(in), CELLULAR LOCATION" will apply much more broadly than the corresponding lexical extraction with just one specific cellular location (e.g., 'mitochondria').

[4]We only used nouns that occurred as the head of a NP.

BIOLOGICAL PROCESS: expression, activity, activation, development, function, production, differentiation, regulation, reduction, proliferation
BODY PART: brain, muscle, thymus, cortex, retina, skin, spleen, heart, lung, pancreas
CELL TYPE: neurons, macrophages, thymocytes, splenocytes, fibroblasts, lymphocytes, oocytes, monocytes, hepatocytes, spermatocytes
CELLULAR LOCATION: receptor, nuclei, axons, chromosome, membrane, nucleus, chromatin, peroxisome, mitochondria, cilia
BIOLOGICAL SUBSTANCE: antibody, lysates, kinase, cytokines, peptide, antigen, insulin, ligands, peptides, enzyme
EXPERIMENTAL REAGENT: buffer, primers, glucose, acid, nacl, water, saline, ethanol, reagents, paraffin
RESEARCH SUBJECT: mice, embryos, animals, mouse, mutants, patients, littermates, females, males, individuals
TUMOR: tumors, tumor, lymphomas, tumours, carcinomas, malignancies, melanoma, adenocarcinomas, gliomas, sarcoma

Table 6: Seed words given to Basilisk

Information extraction patterns and their arguments have been used for text classification in previous work (Riloff and Lehnert, 1994; Riloff and Lorenzen, 1999), but the patterns and arguments were represented separately and the semantic features came from a hand-crafted dictionary. In contrast, our work couples each pattern with its extracted argument as a single feature, uses an automatically generated semantic lexicon, and is the first application of these techniques to the biocuration triage task.

3 Results

3.1 Data Set

For our experiments in this paper we use articles within the PubMed Central (PMC) Open Access Subset[5]. From this subset we select all articles that are published in journals of interest to biocurators at MGI. This results in a total of 14,827 documents out of which 981 have been selected manually by MGI biocurators as relevant (referred to as **IN** documents). This leaves 13,846 that are presumably out of scope (referred to as **OUT** documents), although it was not guaranteed that all of them had been manually reviewed so some relevant documents could be included as well. (We plan eventually to present to the biocurators those papers not included by them but nonetheless selected by our tools as **IN** with high confidence, for possible reclassification. Such changes will improve the system's evaluated score.)

As preprocessing for the NLP tools, we split the input text into sentences using the Lingua::EN::Sentence perl package. We trimmed non-alpha-numerics attached before and after words. We also removed stop words using the Lingua::EN::StopWords package.

3.2 Classifier

We used SVM Light[6](Joachims, 1999) for all of our experiments. We used a linear kernel and a tolerance value of 0.1 for QP solver termination. In preliminary experiments, we observed that the cost factor (C value) made a big difference in performance. In SVMs, the cost factor represents the importance of penalizing errors on the training instances in comparison to the complexity (generalization) of the model. We observed that higher values of C produced increased recall, though at the expense of some precision. We used a tuning set to experiment with different values of C, trying a wide range of powers of 2. We found that C=1024 generally produced the best balance of recall and precision, so we used that value throughout our experiments.

[5] http://www.ncbi.nlm.nih.gov/pmc/about/openftlist.html

[6] http://svmlight.joachims.org/

3.3 Experiments

We randomly partitioned our text corpus into 5 subsets of 2,965 documents each.[7] We used the first 4 subsets as the training set, and reserved the fifth subset as a blind test set.

In preliminary experiments, we found that the classifiers consistently benefitted from feature selection when we discarded low-frequency features. This helps to keep the classifier from overfitting to the training data. For each type of feature, we set a frequency threshold θ and discarded any features that occurred fewer than θ times in the training set. We chose these θ values empirically by performing 4-fold cross-validation on the training set. We evaluated θ values ranging from 1 to 50, and chose the value that produced the highest F score. The θ values that were selected are: 7 for unigrams, 50 for bigrams, 35 for patterns, 50 for lexical extractions, and 5 for semantic extractions.

Finally, we trained an SVM classifier on the entire training set and evaluated the classifier on the test set. We computed Precision (P), Recall (R), and the F score, which is the harmonic mean of precision and recall. Precision and recall were equally weighted, so this is sometimes called an F1 score.

Table 7 shows the results obtained by using each of the features in isolation. The lexical extraction features are shown as 'lexExts' and the semantic extraction features are shown as 'semExts'. We also experimented with using a hybrid extraction feature, 'hybridExts', which replaced a lexical extraction noun with its semantic category when one was available but left the noun as the extraction term when no semantic category was known.

Table 7 shows that the bigram features produced the best Recall (65.87%) and F-Score (74.05%), while the hybrid extraction features produced the best Precision (85.52%) but could not match the bigrams in terms of recall. This is not surprising because the extraction features on their own are quite specific, often requiring 3-4 words to match.

Next, we experimented with adding the IE-based features to the bigram features to allow the classifier to choose among both feature sets and get the best of both worlds. Combining bigrams with IE-based

[7] Our 5-way random split left 2 documents aside, which we ignored for our experiments.

Feature	P	R	F
unigrams	79.75	60.58	68.85
bigrams	84.57	**65.87**	**74.05**
patterns	78.98	59.62	67.95
lexExts	76.54	59.62	67.03
semExts	72.39	46.63	56.73
hybridExts	**85.52**	59.62	70.25
bigrams + patterns	84.87	62.02	71.67
bigrams + lexExts	85.28	**66.83**	**74.93**
bigrams + semExts	85.43	62.02	71.87
bigrams + hybridExts	**87.10**	64.90	74.38

Table 7: Triage classifier performance using different sets of features.

features did in fact yield the best results. Using bigrams and lexical extraction features achieved both the highest recall (66.83%) and the highest F score (74.93%). In terms of overall F score, we see a relatively modest gain of about 1% by adding the lexical extraction features to the bigram features, which is primarily due to the 1% gain in recall.

However, precision is of paramount importance for many applications because users don't want to wade through incorrect predictions. So it is worth noting that adding the hybrid extraction features to the bigram features produced a 2.5% increase in precision (84.57% $\rightarrow$ 87.10%) with just a 1% drop in recall. This recall/precision trade-off is likely to be worthwhile for many real-world application settings, including biocuration.

4 Biocuration Application for MGI

Developing an application that supports MGI biocurators necessitates an application design that minimally alters existing curation workflows while maintaining high classification F-scores (intrinsic measures) and speeding up the curation process (extrinsic measures). We seek improvements with respect to intrinsic measures by engineering context-specific features and seek extrinsic evaluations by instrumenting the deployed triage application to record usage statistics that serve as input to extrinsic evaluation measures.

4.1 Software Architecture

As stated earlier, one of our major goals is to build, deploy, and extrinsically evaluate an NLP-assisted curation application (Alex et al., 2008) for triage at MGI. By definition, an extrinsic evaluation of our triage application requires its deployment and subsequent tuning to obtain optimal performance with respect to extrinsic evaluation criteria. We anticipate that features, learning parameters, and training data distributions may all need to be adjusted during a tuning process. Cognizant of these future needs, we have designed the SciKnowMine system so as to integrate the various components and algorithms using the UIMA infrastructure. Figure 1 shows a schematic of SciKnowMine's overall architecture.

4.1.1 Building configurable & reusable UIMA pipelines

The experiments we have presented in this paper have been conducted using third party implementations of a variety of algorithms implemented on a wide variety of platforms. We use SVMLight to train a triage classifier on features that were produced by AutoSlog and Sundance on sentences identified by the perl package Lingua::EN::Sentence. Each of these types of components has either been reimplemented or wrapped as a component reusable in UIMA pipelines within the SciKnowMine infrastructure. We hope that building such a library of reusable components will help galvanize the BioNLP community towards standardization of an interoperable and open-access set of NLP components. Such a standardization effort is likely to lower the barrier-of-entry for NLP researchers interested in applying their algorithms to knowledge engineering problems in Biology (such as biocuration).

4.1.2 Storage infrastructure for annotations & features

As we develop richer section-specific and context-specific features we anticipate the need for provenance pertaining to classification decisions for a given paper. We have therefore built an Annotation Store and a Feature Store collectively referred to as the Classification Metadata Store[8] in Figure 1. Figure 1 also shows parallel pre-processing populating the annotation store. We are working on developing parallel UIMA pipelines that extract expensive (resource & time intensive) features (such as depen-

[8]Our classification metadata store has been implemented using Solr `http://lucene.apache.org/solr/`

dency parses).The annotation store holds features produced by pre-processing pipelines. The annotation store has been designed to support query-based composition of feature sets specific to a classification run. These feature sets can be asserted to the feature store and reused later by any pipeline. This design provides us with the flexibility necessary to experiment with a wide variety of features and tune our classifiers in response to feedback from biocurators.

5 Discussions & Conclusions

In this paper we have argued the need for richer semantic features for the MGI biocuration task. Our results show that simple lexical and semantic features used to augment bigram features can yield higher classification performance with respect to intrinsic metrics (such as F-Score). It is noteworthy that using a hybrid of lexical and semantic features results in the highest precision of 87%.

In our motivating example, we have proposed the need for sectional-zoning of articles and have demonstrated that certain zones like the 'Materials and Methods' section can contain contextual features that might increase classification performance. It is clear from the samples of MGI manual classification guidelines that biocurators do, in fact, use zone-specific features in triage. It therefore seems likely that section specific feature extraction might result in better classification performance in the triage task. Our preliminary analysis of the MGI biocuration guidelines suggests that experimental procedures described in the 'Materials and Methods' seem to be a good source of triage clues. We therefore propose to investigate zone and context specific features and the explicit use of domain models of experimental procedure as features for document triage.

We have also identified infrastructure needs arising within the construction of a biocuration application. In response we have constructed preliminary versions of metadata stores and UIMA pipelines to support MGI's biocuration. Our next step is to deploy a prototype assisted-curation application that uses a classifier trained on the best performing features discussed in this paper. This application will be instrumented to record usage statistics for use in extrinsic evaluations (Alex et al., 2008). We hope that construction on such an application will also engender the creation of an open environment for NLP scientists to apply their algorithms to biomedical corpora in addressing biomedical knowledge engineering challenges.

6 Acknowledgements

This research is funded by the U.S. National Science Foundation under grant #0849977 for the SciKnowMine project (`http://sciknowmine.isi.edu/`). We wish to acknowledge Kevin Cohen for helping us collect the seed terms for Basilisk and Karin Verspoor for discussions regarding feature engineering.

References

[Alex et al.2008] Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. 2008. Assisted curation: does text mining really help? *Pacific Symposium On Biocomputing*, 567:556–567.

[Bouatia-Naji et al.2010] Nabila Bouatia-Naji, Amélie Bonnefond, Devin A Baerenwald, Marion Marchand, Marco Bugliani, Piero Marchetti, François Pattou, Richard L Printz, Brian P Flemming, Obi C Umunakwe, Nicholas L Conley, Martine Vaxillaire, Olivier Lantieri, Beverley Balkau, Michel Marre, Claire Lévy-Marchal, Paul Elliott, Marjo-Riitta Jarvelin, David Meyre, Christian Dina, James K Oeser, Philippe Froguel, and Richard M O'Brien. 2010. Genetic and functional assessment of the role of the rs13431652-A and rs573225-A alleles in the G6PC2 promoter that are strongly associated with elevated fasting glucose levels. *Diabetes*, 59(10):2662–2671.

[Bourne and McEntyre2006] Philip E Bourne and Johanna McEntyre. 2006. Biocurators: Contributors to the World of Science. *PLoS Computational Biology*, 2(10):1.

[Cohen and Hersh2006] Aaron M Cohen and William R Hersh. 2006. The TREC 2004 genomics track categorization task: classifying full text biomedical documents. *Journal of Biomedical Discovery and Collaboration*, 1:4.

[Ferrucci and Lally2004] D Ferrucci and A Lally. 2004. Building an example application with the Unstructured Information Management Architecture. *IBM Systems Journal*, 43(3):455–475.

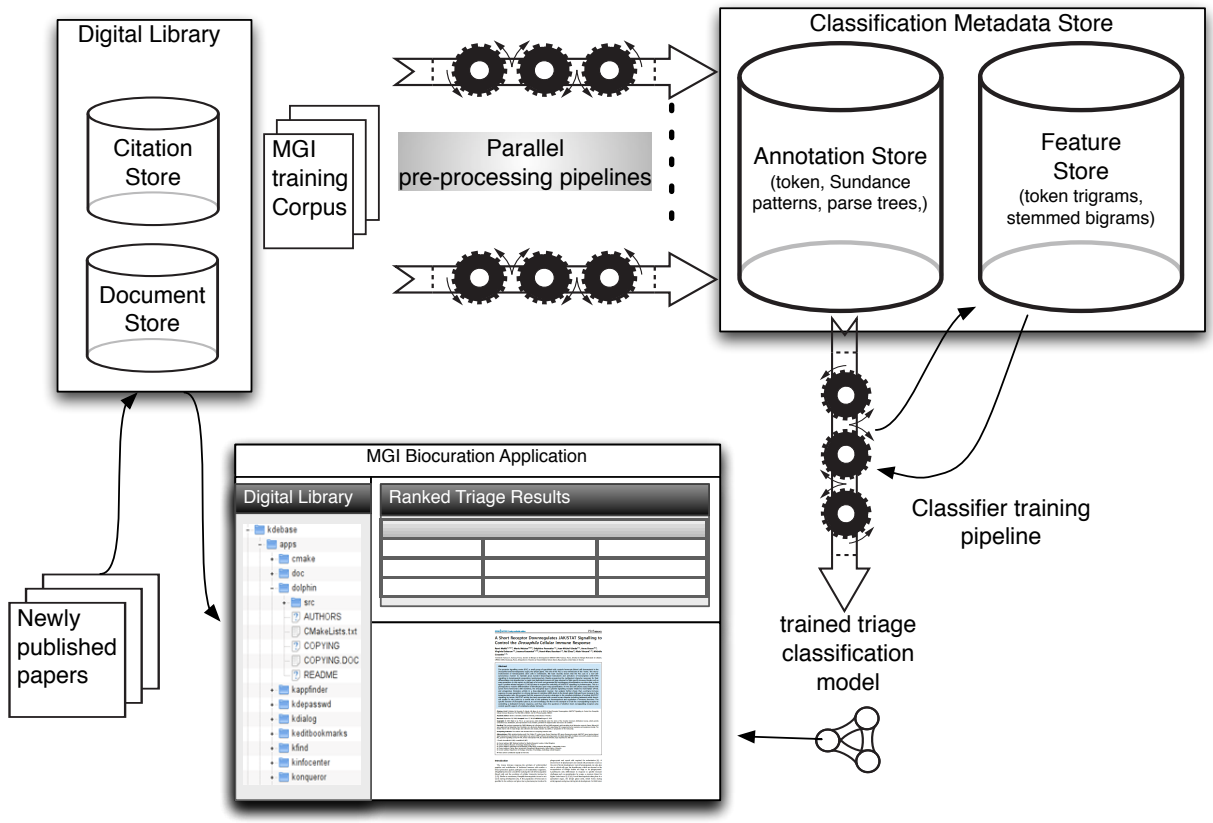

Figure 1: Design schematic of the MGI biocuration application. The components of the application are: (A) Digital Library composed of a citation store and document store. (B) Pre-processing UIMA pipelines which are a mechanism to pre-extract standard features such as parse trees, tokenizations *etc.* (C) Classification Metadata Store which is composed of an Annotation Store for the pre-extracted standard features from (B), and a Feature Store to hold derived features constructed from the standard ones in the Annotation Store. (D) Classifier training pipeline. (E) MGI Biocuration Application.

[Hersh W2005] Yang J Bhupatiraju RT Roberts P M. Hearst M Hersh W, Cohen AM. 2005. TREC 2005 genomics track overview. In *The Fourteenth Text Retrieval Conference*.

[Joachims1999] Thorsten Joachims. 1999. Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods Support Vector Learning*, pages 169–184.

[McIntosh and Curran2009] T. McIntosh and J. Curran. 2009. Reducing Semantic Drift with Bagging and Distributional Similarity. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*.

[McIntosh2010] Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, number October, pages 356–365. Association for Computational Linguistics.

[Ramakrishnan et al.2010] Cartic Ramakrishnan, William A Baumgartner Jr, Judith A Blake, Gully A P C Burns, K Bretonnel Cohen, Harold Drabkin, Janan Eppig, Eduard Hovy, Chun-Nan Hsu, Lawrence E Hunter, Tommy Ingulfsen, Hiroaki Rocky Onda, Sandeep Pokkunuri, Ellen Riloff, and Karin Verspoor. 2010. Building the Scientific Knowledge Mine (SciKnowMine 1): a community- driven framework for text mining tools in direct service to biocuration. *In proceeding of Workshop "New Challenges for NLP Frameworks" collocated with The seventh international conference on Language Resources and Evaluation (LREC) 2010*.

[Riloff and Lehnert1994] E. Riloff and W. Lehnert. 1994. Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information*

Systems, 12(3):296–333, July.

[Riloff and Lorenzen1999] E. Riloff and J. Lorenzen. 1999. Extraction-based text categorization: Generating domain-specific role relationships automatically. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer Academic Publishers.

[Riloff and Phillips2004] E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

[Riloff et al.2003] E. Riloff, J. Wiebe, and T. Wilson. 2003. Learning Subjective Nouns using Extraction Pattern Bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 25–32.

[Riloff1993] E. Riloff. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence*.

[Sjögren et al.2009] Klara Sjögren, Marie Lagerquist, Sofia Moverare-Skrtic, Niklas Andersson, Sara H Windahl, Charlotte Swanson, Subburaman Mohan, Matti Poutanen, and Claes Ohlsson. 2009. Elevated aromatase expression in osteoblasts leads to increased bone mass without systemic adverse effects. *Journal of bone and mineral research the official journal of the American Society for Bone and Mineral Research*, 24(7):1263–1270.

[Thelen and Riloff2002] M. Thelen and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pa ttern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 214–221.