Extraction of Named Entities from Tables in Gene Mutation Literature

Wern Wong[2], *David Martinez*[1,2], *Lawrence Cavedon*[1]

[1]NICTA Victoria Research Laboratory
[2]Dept of Computer Science and Software Engineering
The University of Melbourne

{*wongwl,davidm,lcavedon*}*@csse.unimelb.edu.au*

Abstract *Information extraction and text mining are receiving growing attention as useful techniques for addressing the crucial information bottleneck in the biomedical domain. We investigate the challenge of extracting information about genetic mutations from tables, an important source of information in scientific papers. We use various machine learning algorithms and feature sets, and evaluate performance in extracting fields associated with an existing hand-created database of mutations. We then show how this technique can be leveraged to improve on existing named entity detection systems for mutations.*

1 Introduction and Background

Biomedical science is a large, fast-paced and rapidly growing field. The volume of papers being written every year presents a serious bottleneck to researchers in the field, both in terms of keeping pace with discoveries and with checking for connections to be made with observations made in laboratories. A large amount of biomedical researchers' and workers' time is spent searching and reading the literature for information salient to a particular experimental result or observation in a clinical or diagnostic laboratory.

Information extraction and *text mining* techniques are garnering much interest in the biomedical space due to their potential for alleviating the information bottleneck experienced by researchers and clinical workers (e.g. [2]). We are interested in applying such methods to aiding the construction of databases of biomedical information, in particular information about genetic mutations. Such databases are currently constructed by hand: a long, involved, time-consuming and human-intensive process. Each paper considered for inclusion in the database must be read, the interesting data identified and then entered by hand into a database.[1]

In this paper, we focus on the task of extracting information from *tables* in biomedical research papers. Tables present a succinct and information-rich format for providing information, and are particularly important when reporting results in biological and medical research papers: the important results in such papers may be reported only in tabular form and not in the main text at all. Table processing presents its own challenges, especially that of detecting tables in text and dealing with structure—see [8] for a survey on table recognition and [3] for a discussion on processing tables in web documents. While interesting approaches to detecting and processing tables have been used in various applications—e.g. Wei *et al* [5] perform question-answering over tables extracted from financial documents—we know of no previous attempt to process tables in biomedical documents.

Our extraction task is grounded in the specific context of the *Mismatch Repair (MMR) Database* compiled at the Memorial University of Newfoundland [7]—a database of known genetic mutations related to hereditary non-polyposis colorectal cancer (HNPCC), a hereditary form of bowel cancer. The MMR Database contains information on genetic mutations known to be related to HNPCC, along with links to the research papers from which the database has been constructed.[2] From the database and its links to papers, we were able to construct a collection of tables related to HNPCC mutations, and then use the MMR database records themselves as a gold standard for evaluating our techniques. As at May 2008, the MMR database contained a total of 5,491 records on mutations that occur on any one of four genes that have been identified as related to colon cancer. An example record from the MMR database is the following:

```
MLH1 | Exon13 | c.1491delG | Yamamoto et al.  | 9500462
```

Respectively, this record contains: the gene; exon; mutation; citation of the paper the information was sourced from;[3] and the paper's PubMedID (PubMedID is a unique identifier assigned to papers whose abstract is contained in the MEDLINE collection). These fields are important because they contain information

[1]Karamis *et al* [4] illustrate how even simple tools can have an impact on improving the database-curation process.

Proceedings of the 13th Australasian Document Computing Symposium, Hobart, Australia, 8 December 2008.

[2]I.e. a team of geneticists manually trawled the biomedical literature for information on HNPCC-related mutation information, and added links to any papers relevant to those mutations in the context of HNPCC.

[3]This field has been abbreviated. We have also omitted fields such as "internal id".

researchers are directly interested in (gene, exon, mutation) and the paper said information was found in. Note that if a gene/mutation pair is referenced in multiple papers, then there are correspondingly multiple entries in the database. Conversely, if a single paper mentions multiple (relevant) genes, then that paper is mentioned in multiple database records. Our goal in this paper is to work towards automatic aids for the curation of this database.

2 Experimental Setting

In this section, we describe the process of creating our experimental dataset and the task design.

2.1 Creating the Dataset

Our collection of tables was extracted via the MMR database, leveraging the MEDLINE collection of biomedical abstracts. We first collected all information available in the hand-curated MMR records, obtaining a total of 5,491 mutations linked to 719 distinct PubMedIDs[4]. We next used a crawler to retrieve the corresponding full-text articles (identified by PubMedIDs stored in the MMR records) by following links from the PubMed interface, and downloaded those papers that had a full-text HTML version, and which contained at least one content table. Tables were then extracted from the full-text HTML files. It is worth noting that the tables were already present as links to separate HTML files rather than being presented as inline tables, making this process easier. Papers that did not contain tables in HTML format were eliminated.

Our final collection consisted of 70 papers from the original 719 PubMedIDs. The articles are linked to 784 MMR records (mutations), which constitutes our gold standard hand-curated annotation. The collection contains 197 tables in all.[5]

The tables in the collection were then pre-processed into a form that more readily allowed experimentation. The tables were split into three parts: column headers, row headers, and data cells. This was done based on the HTML formatting, which was consistent throughout the data set as the tables were automatically generated: cells were replicated when they spanned multiple rows or columns; `img` tags were replaced by the *alternate* text (when available); and `hr` tags were used to separate out column headers from the cells themselves. Row headers were detected by checking if the top left cell of the table was blank, a pattern which occurred in all row-major tables. We acknowledge that this processing may be specific to the vagaries of the particular format of the HTML generation used by PubMed (from which we sourced the tables). However, our whole task is specific to this domain; further, our focus is on the data extraction task rather than the actual detection of row/column headers.

[4]Data was downloaded from the web interface in May 2008.

[5]This collection could be increased in size by more putting more effort into retrieving documents linked to MMR records.

2.2 Task Design

In order to extract mutations from tables, we first performed classification of full columns/rows into relevant entities. Since the content of a column (or row, depending on whether the table was row- or column-oriented) tends to be homogeneous, this allowed us to build classifiers that can identify full vectors of relevant entities in a single step. We refer to this task as *table vector classification*.

We identified the following entities as relevant: Gene, Exon, Mutation, Codon, and Statistic. The first four were chosen directly from the MMR Database. We decided to include "Statistic" after inspecting the tabular dataset, since we found that this provides relevant information about the importance of a given mutation. From the five entities, Mutation is the most informative for our final information extraction goal.

The next step was to hand-annotate the headers of the 197 tables in our collection by using the five entities and the class "Other" as the tagset. Some headers belonged to more that one class, because the entities were collapsed into a single field of the table.

We performed two tasks: vector classification, and mutation extraction. The evaluation for the vector classification step was done using precision, recall and f-score, micro-averaged among the classes. For the machine learning (ML) algorithms, we used stratified 10-fold cross-validation. For mutation extraction we focus on the mutation class, and produce precision and recall against the subset of the hand-curated MMR database.

3 Table Vector Classification

We describe here heuristic and ML approaches to vector classification, along with an analysis of their performance.

3.1 Heuristic Approach

As a baseline method, we approached the task of classifying headers by matching the header string to the names of the classes in a case-insensitive manner. When the class name was found as a substring of the header, the class would be assigned to it. For example, a header string such as "Target Mutation" would be assigned the class "Mutation". Some headers had multiple annotations (e.g. "Gene/Exon").

For better recall, we also matched synonyms for the class "Mutation" (the terms "Variation" and "Missense") and the class "Statistic" (the terms "No.", "Number" and "%"). For the remaining classes we did not identify other obvious synonyms.

Results are shown in Table 1. Precision was very low for the "Mutation" class, illustrating that different types of information are provided under this heading; e.g. the heading "Mutation detected" above a "Gene"

Class	Precision	Recall	FScore
Gene	0.537	0.620	0.575
Exon	0.762	0.615	0.681
Codon	0.850	0.654	0.739
Mutation	0.283	0.301	0.292
Statistic	0.911	0.324	0.478
Other	0.581	0.903	0.707
Micro Avg.	**0.693**	**0.614**	**0.651**

Table 1: Naive Baseline results across the different classes and micro-averaged

Class	Precision	Recall	FScore
Gene	0.537	0.611	0.571
Exon	0.762	0.615	0.681
Codon	0.850	0.654	0.739
Mutation	0.600	0.452	0.515
Statistic	0.911	0.340	0.495
Other	0.579	0.910	0.708
Micro Avg.	**0.715**	**0.633**	**0.672**

Table 2: Results integrating MutationFinder across the different classes and micro-averaged

vector. Recall was low for most classes, suggesting that more sophisticated approaches are required.

Our second step was to build a more informed classifier for the "Mutation" class. We applied the mutation NER tool MutationFinder [1] to the text in cells to identify which table-vectors contained at least one mutation mention. Any such vectors were classified as mutations. The results are shown in Table 2. This approach caused the "Mutation" results to improve, but the overall f-scores leave room for improvement.

3.2 Machine Learning Methods

For the ML experiments we used the Weka [6] toolkit, as it contains a wide selection of in-built algorithms. As a baseline, we applied the majority class from the training data to all test instances. We applied the following ML algorithms from Weka[6]: Naive Bayes (NB), Support Vector Machines (SVM), Propositional Rule Learner (JRip), and Decision Trees (J48).

In order to define our feature sets, we used the text in both the headers and cells of the tables. Other sources of information, such as captions or the running text referring to the table where not employed at this stage, but may also provide valuable information. We used four feature sets:

- **Basic (Basic):** header string, the average and median cell lengths, and a binary feature indicating whether the data in the cells was numeric;
- **Cell Bag-of-Words (C_bow):** Bag of words over the tokens in the table cells;
- **Header Bag-of-Words (H_bow):** Bag of words over the tokens in the header strings;
- **Header + Cell Bag-of-Words (HC_bow):** Bags of words formed by the tokens in headers and cells, represented as different feature types.

[6]We applied a number of other ML algorithms as well, but these showed significantly lesser performance.

Algorithm	Feature Sets			
	Basic	**C_bow**	**H_bow**	**HC_bow**
Maj. Class	0.288			
NB	0.614	0.454	0.678	0.581
SVM	**0.717**	**0.599**	**0.839**	**0.816**
JRip	0.564	0.493	0.790	0.749
J48	0.288	0.532	0.793	0.782

Table 3: Micro-Averaged FScores for ML algorithms. The best results per column are given in bold.

Class	Precision	Recall	FScore
Gene	0.778	0.737	0.757
Exon	0.786	0.707	0.745
Codon	0.833	0.882	0.857
Mutation	0.656	0.679	0.667
Statistic	0.919	0.853	0.885
Other	0.820	0.884	0.850
Micro Avg	**0.839**	**0.841**	**0.839**

Table 4: Results for SVM and the feature set *H_bow* per class and micro-averaged.

The micro-averaged results of the different learning methods and feature sets are shown in Table 3. Regarding the feature sets, we can see that the best performance is obtained by using the headers as bag-of-words, while the content of the cells seems to be too sparse to guide the learning methods. SVM is clearly the best algorithm for this dataset, with JRip and J48 following, and NB performing worst of the four in most cases. Only for the basic feature set (with very few features) does NB outperform JRip and J48.

Overall, the results show that the ML approach is significantly superior to the baselines when relying on the header bag of words[7]; SVM is able to reach a high f-score of 83.9% in predicting the relevant entities.

We break down the results per class in Table 4, using the outputs from SVM and feature-set *H_bow*. We can see that all classes improve over the heuristic baselines. There is a big increase for the classes "Gene" and "Statistic", and all classes except mutation are above 70% f-score. "Mutation" is the most difficult class to predict, but it still reaches 66.7% f-score, which can be helpful for some tasks, as we explore in Section 4.

4 Mutation Extraction

We applied the results of our classifier to a real-world application: the detection of mutations in the literature for the MMR Database project. Table vector classification allows us to extract lists of candidate mutation names from tables to be added to the database. In order to test the viability of this approach, we measured the precision and recall of the system in detecting the existing hand-curated mutations in MMR. Recall is more important in this setting, since it shows the proportion of mutation mentions that we are able to obtain with our technique. Precision will give an indication of the rate of false positives, but note that we also consider as false positives those valid mutations that were not interesting

[7]As shown by a paired *t-test* at 99% confidence.

System	Precision	Recall
MF (full text)	0.01 (6 / 438)	0.01 (4 / 717)
Table Vector classifier	**0.09 (153 / 1702)**	**0.21 (153 / 717)**
Gold standard heads	0.11 (198 / 1847)	0.28 (198 / 717)

Table 5: Mutation detection results, Table Vector classifier in bold.

for MMR, and therefore the reported precision will be artificially low.

As state-of-the-art mutation detection system, we apply MutationFinder (MF) [1] to the full text (including tables) of the journal collection. This allows us to compare the results of our table-processing approach over an existing tool that parses the full text.

The evaluation results are shown in Table 5.[8] We can see that the goldstandard table vector annotation retrieves 28% of the mutations, at a precision of 11%. This means that by looking only at the tables we have an upper-bound of 28% on the percentage of relevant mutations that we can extract. In comparison, MF is only able to retrieve 1% of the mutations by looking at full articles. This happens because MF targets mutation mentions that follow a specific nomenclature, and the mentions that it is able to detect are not the ones covered in the MMR Database. Finally, our automatic table vector classifier is able to retrieve 21% of the gold standard mutations at a precision of 9%, which is 75% of the upperbound.

The precision figures are low for the goldstandard and table vector classifier. The reason for this is that we do not discriminate automatically for mutations of interest for the MMR Database, and valid mutation mentions are often classified as negative. All in all, the vector classifier discriminates 1,702 mutation cells out of a total of 27,700 unique cells in the collection, and it effectively identifies 153 out of the 198 relevant mutations present in the tabular data.

Finally, after the evaluation process we observed that many false mutation candidates could be removed by discarding those that do not contain two consecutive digits or any of the following n-grams: “c.”, “p.”, ’>’, “del”, “ins”, “dup”. This heuristic raises the precision of the system to 15.5% (153 true positives out of 989) with no cost in recall, which would result in greater saved time for database curators in a practical setting.

5 Discussion

Our preliminary results on the task of identifying relevant entities from gene mutation literature show that targeting tables can be a fruitful approach for text mining. By relying on ML methods and simple bag-of-words features, we were able to achieve good performance over a number of selected entities, well above header word-matching baselines. This allowed us to identify lists of mentions of relevant entities with minimal effort, reaching 21% of recall over a hand-curated database.

Another advantage of our approach is that the annotation of examples for training and evaluation is considerably easier, since many entities can be annotated in a single step. This opens the way to faster annotation of other entities of interest in the biomedical domain, which can present a wide variety of forms and non-standard terminology. However, since a table vector of homogeneous information may include representatives of the heterogeneous nomenclature schemes, classification of a whole column or row potentially helps nullify the effect of the terminological variability.

For future work, we plan to study different types of features for better representing the entities targeted in this work. Especially for mutation mentions, we observed that the presence of certain ngrams (e.g. ”del”) can be a strong indicator for this class. Another goal is to increase the size of our dataset of articles by improving our retrieval process, and by hand-annotating the retrieved table vectors for further experimentation.

Acknowledgements NICTA is funded by the Australian government as represented by Dept. of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme. Thanks to Mike Woods and his colleagues at the Memorial University of Newfoundland for making the MMR database available to us. Eric Huang wrote several of the scripts mentioned in Section 2 for creating the table collection.

References

[1] J. G. Caporaso, W. A. B. Jr., D. A. Randolph, K. B. Cohen, and L. Hunter. Mutationfinder: A high-performance system for extracting point mutation mentions from text. *Bioinformatics*, 23(14):1862–1865, 2007.

[2] A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining : Annual progress in bioinformatics. *Briefings in Bioinformatics*, 6(1), 2005.

[3] M. Hurst. Layout and language: Challenges for table understanding on the web. Technical report, WhizBang!Labs, 2001.

[4] N. Karamanis, R. Seal, I. Lewin, P. McQuilton, A. Vlachos, C. Gasperin, R. Drysdale, and T. Briscoe. Natural language processing in aid of flybase curators. *BMC Bioinformatics*, 9:193–204, 2008.

[5] X. Wei, W. Croft, and D. Pinto. Question answering performance on table data. *Proceedings of National Conference on Digital Government Research*, 2004.

[6] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[7] M. Woods, P. Williams, A. Careen, L. Edwards, S. Bartlett, J. McLaughlin, and H. B. Younghusband. A new variant database for mismatch repair genes associated with lynch syndrome. *Hum. Mut.*, 28, 2007.

[8] R. Zanibbi, D. Bolstein, and J. R. Cordy. A survey of table recognition. *Int’l J. on Document Analysis and Recognition*, 7(1), 2004.

[8] Because of the different mutation nomenclature formats used, comparison to gold standard required manual checking.