Automatic Extraction of ICD-O-3 Primary Sites from Cancer Pathology Reports

Ramakanth Kavuluru, Ph.D[1], Isaac Hands, B.S[2], Eric B. Durbin, DrPH[2], and Lisa Witt, A.S[2]

[1]Division of Biomedical Informatics, University of Kentucky, Lexington, KY
[2]Kentucky Cancer Registry, Cancer Research Informatics Shared Resource Facility, Markey Cancer Center, Lexington, KY

Abstract

Although registry specific requirements exist, cancer registries primarily identify reportable cases using a combination of particular ICD-O-3 topography and morphology codes assigned to cancer case abstracts of which free text pathology reports form a main component. The codes are generally extracted from pathology reports by trained human coders, sometimes with the help of software programs. Here we present results that improve on the state-of-the-art in automatic extraction of 57 generic sites from pathology reports using three representative machine learning algorithms in text classification. We use a dataset of 56,426 reports arising from 35 labs that report to the Kentucky Cancer Registry. Employing unigrams, bigrams, and named entities as features, our methods achieve a class-based micro F-score of 0.9 and macro F-score of 0.72. To our knowledge, this is the best result on extracting ICD-O-3 codes from pathology reports using a large number of possible codes. Given the large dataset we use (compared to other similar efforts) with reports from 35 different labs, we also expect our final models to generalize better when extracting primary sites from previously unseen reports.

1 Introduction

International Classification of Diseases for Oncology [1], Third Revision (ICD-O-3) is an extension of the ICD coding standard for tumor diseases. It is an international coding standard that constitutes a dual classification system for both topography and morphology of a neoplasm. ICD-O-3 codes are used [2] to capture information on cancer incidence, prevalence, and survival in the US by the Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI) and the National Program of Cancer Registries (NPCR) of the Centers for Disease Control and Prevention (CDC). In addition to being central to the reporting purposes for cancer registries, the codes are indirectly used by healthcare providers for quality control and by researchers for biospecimen annotations and clinical trial recruitment.

The topography axis of the ICD-O-3 standard consists of codes for *generic anatomical sites* [1] and *subsites* of neoplasms. Each such code takes the form `Cab.x`, where `C` is always the lead character, the two digit code `ab` specifies a generic anatomical site and the single digit `x` determines a subsite. For example, in the code `C34.1`, the two digit code `34` indicates the generic site *lung*, while the `1` after the period indicates the subsite *upper lobe*. There are total of 70 distinct two digit codes after the leading character, thus 70 generic sites, and a total of 316 unique codes when considering the subsite code.

ICD-O-3 codes are in general extracted by trained coders, more specifically, certified tumor registrars (CTRs), from free text pathology reports and other artifacts in patient medical records. Pathology reports are primarily dictated and transcribed by pathologists describing their interpretations after examining tissue samples under a microscope. Although pathology reports are not the only source for code extraction, they are still the major source of coding information. The time consuming manual coding task can be significantly expedited by complementing manual efforts with automatic approaches to code extraction which is the main purpose of this paper. We are particularly interested in extracting the generic site code for primary cancers reported in pathology reports.

2 Background

Automatic approaches to information extraction from cancer pathology reports have been extensively published in informatics literature (see Section 5 in [3] for a recent survey). A few results focus on extracting TNM staging [4, 5] information for a particular type of cancer (so primary site is fixed) or identifying cancer related named entities in pathology reports, which are later used for cohort selection and case registration [6]. Here we note that we are

interested in the primary tumor generic site (that is, the top level two digit ICD-O-3 main site code) and not all mentions of anatomical sites in pathology reports. Identifying all tumor site mentions can be addressed by state-of-the-art cancer information extraction systems such as caTIES [7], which in addition can perform complex queries to retrieve specific pathology reports. Although many mentions of anatomical sites occur in pathology reports, in the majority of cases there is only one primary tumor site, and other non-primary sites referred to within in the report provide the context of patient's history or other metastatic progressions of the primary tumor.

Previous attempts at extracting primary tumor sites are discussed in few cases. Coden et al. [8] present a comprehensive structured cancer disease knowledge representation model and populate it using a system MEDTAS/P that they developed to extract cancer elements from pathology reports. They demonstrate their results on a dataset of 302 pathology reports of patient diagnosed with colon cancer. Their F-score in primary tumor identification is 0.82 using the generic site *colon*; however, they also extract histology and other elements under the primary tumor class and hence deal with a related but different task of information extraction. Martinez and Li [9] achieve an average F-score of 0.58 to classify among 11 generic sites with named entities as features using a Naïve Bayes classifier; their dataset size is 217 pathology reports. Jouhet et al. [10] report an average F-score of 0.72 in classifying French language pathology reports among 26 ICD-O-3 topography codes using support vector machines (SVMs) with ngrams as features and a dataset of 5121 documents. To compute balanced average F-scores, micro-averaging is used in [9] but the method of averaging (micro or macro) is not discussed in [10]. As detailed in the next section, using a much larger dataset and three different representative algorithms from different statistical learning paradigms, we achieve a micro average F-score of 0.9 and a macro average F-score of 0.72 when classifying among 57 generic sites. To our knowledge, this is the best result on ICD-O-3 topography code extraction for a large number of sites.

3 Methods

We used a dataset consisting of 56,426 free text pathology reports associated with reported cancer cases to the Kentucky Cancer Registry from 35 different labs. Only one primary tumor site (so one ICD-O-3 code) was linked with each pathology report. Although a pathology report can lead to multiple primary site codes, each registry record reports only one primary site which was linked to the corresponding pathology report in the report database. So in our dataset, each report is associated with only one primary site. In this dataset, the top five cancer generic sites with their counts in parentheses are: breast (9491), lung (9361), prostate gland (4162), colon (3738), and skin (2294). The top 10 sites account for nearly 70% of reports in the dataset. Although we have examples for all 70 generic sites in the ICD-O-3 standard, only 57 of them had at least 50 examples which we determined to be the minimum acceptable number of examples for our experiments. We also tested thresholds of at least 100 examples (a total of 42 codes), and then a threshold of at least 1000 examples (a total of 14 codes).

Text Classification with Statistical Approaches: As noted earlier, named entity recognition techniques can identify mentions of anatomical sites with high accuracy. Since multiple sites can be mentioned in a pathology report and as generally there is a very small number (usually one or two*) of primary tumors, these techniques are not suitable by themselves for extracting the primary tumor site. Hence we resort to machine learning techniques for text classification.

We use two types of document features as input to our techniques. The first type are the unigrams and bigrams (for simplicity we just refer to them as ngrams [11]) found in the reports. After removing stop words and using regular expressions to prune those ngrams that contain words that are either just a single character or those that start with a non-alphabetic character (such as "00am"), we observed $\approx$ 240,000 ngrams. We further pruned them by imposing a minimum document frequency of 20; that is, we only considered ngrams that occurred in at least 20 reports. With this, our final ngram feature space was reduced to $\approx$ 90,000 features. We also used named entities from the reports as the second type of features. We used NLM's named entity recognition tool MetaMap [12] and extracted named entities that belong to the disorders, procedures, or chemicals & drugs semantic groups [13] and also come from a popular set of 14 vocabularies [14] that provide significant coverage of the clinical text domain. Since MetaMap also handles negation, we also considered negated occurrences of concepts as separate features. Using a minimum document threshold of 3, we obtained $\approx$ 7800 Unified Medical Language System (UMLS) concepts represented using UMLS Metathesaurus concept unique identifiers (CUIs). We conducted experiments with ngram and CUI features separately and also using a combined set.

Using ngram and CUI features, we experiment with representative algorithms from three different learning paradigms. Naïve Bayes (NB) is a *generative* classifier that computes joint probabilities of classes and documents and derives posterior probabilities of a class given a document based on the established joint probability of classes and documents from training data. The Multinomial version of NB [15] also takes into account the frequencies of ngrams and CUIs instead of just considering their presence as a Boolean value. Logistic Regression (LR) belongs to a class

*In our dataset, we assume only one primary tumor per report.

of *discriminative* classifiers that directly compute conditional probabilities over the classes given document features without computing the joint and class prior probabilities. Support vector machines (SVMs) are *discriminative but non-probabilistic* classifiers that compute hyperplanes (in the case of linear SVMs) that separate feature vectors of documents into different classes. We use the Java machine learning framework Weka 3.7 [16] for representing training reports and their features. We also use Weka's implementation of Multinomial NB based on [15] and also the Weka wrappers for the LIBLINEAR [17] package that implements fast linear SVMs and LR. Multinomial NB can directly distinguish documents of multiple classes, but SVM and LR as implemented by LIBLINEAR are binary classifiers and hence use a *one-vs-all* approach, where a binary classifier is trained for each class treating the examples of all other classes as negative examples. Finally, the target class is chosen based on the highest confidence score output by each binary classifier for all classes. Multinomial NB requires frequency counts of features, while LR and SVMs work with both Boolean and numerical (frequencies) features. In addition to using Boolean features for LR and SVM, we also tried using frequencies of features. Finally, we also tried different regularization parameters for LR and SVM that help reduce the over fitting issues of discriminative models.

In the next section we report results using the full content of each pathology report for feature extraction, but we also conducted experiments by considering only the final diagnosis portion of the report and as a separate experiment considered reports excluding the clinical history portion. All experiments use the 10-fold cross validation approach where the training examples are divided into 10 equal folds such that examples of each class are distributed evenly across all folds. Next in each of the 10 iterations, one of the previously unused 10 folds is used for testing and the remaining 9 folds are using for training. In the Weka framework, the aggregated results over all folds give the precision and recall measures for the model.

4 Results

		Logistic Regression		Linear SVM		Multinomial NB	
		Macro-F	Micro-F	Macro-F	Micro-F	Macro-F	Micro-F
Top 14 Sites	ngrams	**0.93**	**0.93**	0.92	**0.93**	0.80	0.82
	CUIs	0.91	0.92	0.90	0.91	0.84	0.85
	ngrams+CUIs	**0.93**	**0.93**	0.92	**0.93**	0.81	0.83
Top 42 Sites	ngrams	**0.78**	**0.91**	**0.78**	0.90	0.56	0.80
	CUIs	0.71	0.88	0.67	0.86	0.53	0.82
	ngrams+CUIs	**0.78**	**0.91**	**0.78**	0.90	0.56	0.81
Top 57 Sites	ngrams	0.70	**0.90**	0.71	**0.90**	0.43	0.80
	CUIs	0.63	0.88	0.60	0.86	0.41	0.81
	ngrams+CUIs	0.71	**0.90**	**0.72**	**0.90**	0.43	0.80

Table 1: Topography Generic Site Based Micro and Macro F-Scores (top scores shown in **bold**)

We evaluate our methods using both micro and macro averaging of F-scores. Before we present the results, we briefly discuss these measures. For each topography code T_j in the set of classes T being considered, we have code based precision $P(T_j)$, recall $R(T_j)$, and F-score $F(T_j)$ defined as

$$P(T_j) = \frac{TP_j}{TP_j + FP_j},\ R(T_j) = \frac{TP_j}{TP_j + FN_j},\ \text{and}\ F(T_j) = \frac{2P(T_j)R(T_j)}{P(T_j) + R(T_j)},$$

where TP_j, FP_j, and FN_j are true positives, false positives, and false negatives, respectively, of code T_j. Given this, the code based macro average F-score is defined as

$$\text{Macro-F} = \frac{1}{|T|} \cdot \sum_{j=1}^{|T|} F(T_j).$$

Finally, the code based micro precision, recall, and F-score are defined as

$$P^{micro} = \frac{\sum_{j=1}^{|T|} TP_j}{\sum_{j=1}^{|T|}(TP_j + FP_j)},\ R^{micro} = \frac{\sum_{j=1}^{|T|} TP_j}{\sum_{j=1}^{|T|}(TP_j + FN_j)},\ \text{and Micro-F} = \frac{2P^{micro}R^{micro}}{P^{micro} + R^{micro}},$$

respectively. While the macro measures consider all codes as equally important, micro measures tend to give more importance to codes that are more frequent. With these definitions, we present our results in Table 1. Although we conducted several experiments with different parameters, feature types (Boolean vs frequencies), and different

portions of reports, we did not find significant improvements in the average F-scores over all sites. Hence we report results obtained using the default parameters and using all portions of the reports. To see how the techniques perform when we consider fewer sites based on their frequencies, as explained in Section 3, we ran experiments with sites having at least 1000 (top 14 sites), 100 (top 42 sites), and 50 (top 57 sites) training examples separately. In each of these cases, we create an "other" class that groups together all examples that do not belong to the classes being selected for classification. So if we are running the classifiers for the top k sites, we actually have a total of $k + 1$ sites as input to the classifiers.

5 Discussion

From Table 1 we see that LR and SVMs significantly outperform Multinomial NB in all cases. This seems to be in agreement with the general observation by the scientific community that NB tends to perform well with fewer training examples compared with discriminative classifiers, but is quickly overtaken by them given more training data [18]. Roughly, there is a 30 point difference in macro averages and 10 point difference in micro averages between NB and the discriminative models. However, there is no major performance difference between LR and SVM approaches. Also, the performance with the default regularization parameter $C = 1$ for these algorithms did not improve when changes were made with increments of 0.1 up to $C = 2$, and next with increments of 1 up to $C = 10$. Using UMLS concepts does not seem to improve the performance significantly. CUIs alone, although an order of magnitude smaller in number than ngrams (7,800 Vs 90,000), prove very effective when classifying among top 14 sites, with a negligible performance difference. But with the top 57 sites, while they still do well in micro averages, in general suffer a 7–8 percentage point difference in macro averages when ngrams and combinations are introduced. For the top 14 classes, in the case of NB, CUIs outperform ngrams and the combination of ngrams+CUIs. The CUI+ngram combination shows just one point gain in macro average F-score over using only ngrams. Furthermore, extraction of CUIs from textual documents is a computationally expensive task compared with ngram processing. So further investigations based on feature selection might be needed to assess their predictive value in ICD-O-3 code extraction.

Another interesting observation is that, in all experiments, individual site level precision (not shown in the table) is consistently greater than the recall values especially for codes with fewer than 100 examples. This is observed for all three algorithms and more so when using NB. When using LR the difference between precision and recall was as high as 50 percentage points, compared to about 85 points for NB. Further investigation is needed to understand these poor recall values that are resulting in lower F-scores.

To conduct qualitative error analysis we analyzed the confusion matrix of the best performing model and randomly selected several site pairs that were often mutually misclassified (these are pairs that correspond to large values in the non-diagonal cells in the confusion matrix). The most prominent site pairs that caused a high number of errors are: (colon, rectum), (lung, pancreas), (prostate gland, urinary bladder), and (lymph node, hemaptopoietic system).

We randomly selected a total of 16 misclassified pathology reports for the site pairs described above. These were provided to a certified tumor registrar (CTR) in the Kentucky Cancer Registry who was informed that they caused errors but not the nature nor the sites of the misclassification. The CTR came up with top 2 most probable sites by manually reviewing the reports. In cases when the evidence is overwhelmingly indicative of just one site, the reviewer just coded that one site and in cases when no clear primary site is mentioned, the reviewer coded it as "site information not found". Later the coded sites were compared with actual pairs (actual and misclassified sites) for each report. The reviewer then provided possible explanations as to the causes of confusion when she was reading the reports. Using this information, we then qualitatively classified the causes of errors into four categories:

1. not enough evidence in the report to identify a primary cancer site
2. more than one primary site discussed in the report
3. an actual error in the registry (wrong reports linked to registry records), and
4. algorithmic classification error

Using this list, we found 11 of the 16 example reports selected for review fell into the first category where a trained human reviewer could not disambiguate between the multiple (not necessarily primary) sites discussed or could not find a reportable primary site purely based on the report content. For example, a report that had an actual primary site as lymph node was predicted as affecting the hematopoietic system. The reviewer actually assigned what the techniques predicted, hematopoietic system. We glean from the reviewer comments that solely based on the report, the predicted site would be correct, but because the rules for establishing the primary site of lymphomas take into account clinical information not available in the path reports, automated techniques solely based on the path reports might not be able to identify these sites. For a couple of examples corresponding to the misclassified pair (lung, pancreas), the reviewer could only find metastatic diseases and could not ascertain a primary tumor site. For example reports of rectum misclassified as colon, the reviewer found that the phrase "colon polyp" was inaccurately used to characterize an actual polyp in the rectum.

Both examples corresponding to misclassification of the pair (urinary bladder, prostate gland) result from the fact that both primary sites are discussed in the reports. Because in the current project we only consider one primary site, the algorithms only selected one site and missed the other, which lead to a classification error. In the case of a report of primary site colon misclassified as rectum, upon manual inspection, we found a large number of negated occurrences of rectal polyps. This is the only error that fell into the algorithmic classification error category as there was clear evidence that this was a primary colon cancer case. Although we modeled negative occurrences of UMLS concepts as separate features (using MetaMap's negation detection), it is not clear how negative evidence could lead to a positive classification, which needs further investigation.

Although we improved on the state-of-the-start in both micro and macro average F-scores by considering a larger set of 57 generic sites, we still have at least two ways to improve and extend our results in the immediate future. The first is to address the low recall values for sites that have fewer than 100 examples. The second is to extend this framework to extract full codes including subsite information, instead of just the generic codes.

References

[1] Fritz AG, Jack A, Parkin D, Percy C, Shanmugarathan S, Sobin L, et al. International classification of diseases for oncology: ICD-O, Third Edition. World Health Organization; 2000.

[2] Adamo MB, Johnson CH, Ruhl JL, Dickie LA. SEER Program Coding and Staging Manual 2012. National Cancer Institute, NIH Publication number 12-5581; 2012.

[3] Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? Journal of Biomedical Informatics. 2009;42(5):760–772.

[4] McCowan I, Moore D, Nguyen AN, Bowman RV, Clarke BE, Duhig EE, et al. Application of Information Technology: Collection of Cancer Stage Data by Classifying Free-text Medical Reports. JAMIA. 2007;14(6):736–745.

[5] Nguyen AN, Lawley M, Hansen DP, Bowman RV, Clarke BE, Duhig EE, et al. Symbolic rule-based classification of lung cancer stages from free-text pathology reports. JAMIA. 2010;17(4):440–445.

[6] Hanauer DA, Miela G, Chinnaiyan AM, Chang AE, Blayney DW. The Registry Case Finding Engine: An Automated Tool to Identify Cancer Cases from Unstructured, Free-Text Pathology Reports and Clinical Notes. Journal of the American College of Surgeons. 2007;205(5):690 – 697.

[7] Crowley RS, Castine M, Mitchell K, Chavan G, McSherry T, Feldman M. caTIES: a grid based system for coding and retrieval of surgical pathology reports and tissue specimens in support of translational research. JAMIA. 2010;17(3):253–264.

[8] Coden A, Savova GK, Sominsky IL, Tanenblatt MA, Masanz JJ, Schuler K, et al. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. Journal of Biomedical Informatics. 2009;42(5):937–949.

[9] Martinez D, Li Y. Information extraction from pathology reports in a hospital setting. In: Proceedings of the 20th ACM intl. conf. on Inf. and knowledge mgmt. CIKM '11. ACM; 2011. p. 1877–1882.

[10] Jouhet V, Defossez G, Burgun A, le Beux P, Levillain P, Ingrand P, et al. Automated classification of free-text pathology reports for registration of incident cases of cancer. Methods of Information in Medicine. 2012;51(3):242–251.

[11] Peng F, Schuurmans D. Combining naive bayes and n-gram language models for text classification. In: Proceedings of the 25th European conference on IR research. ECIR'03. Berlin, Heidelberg; 2003. p. 335–350.

[12] Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. JAMIA. 2010;17(3):229–236.

[13] Bodenreider O, McCray AT. Exploring semantic groups through visual approaches. J of Biomedical Informatics. 2003 Dec;36(6):414–432.

[14] Wu ST, Liu H, Li D, Tao C, Musen MA, Chute CG, et al. Unified Medical Language System term occurrences in clinical notes: a large-scale corpus analysis. JAMIA. 2012;19(1):149–156.

[15] Mccallum A, Nigam K. A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI-98 Workshop on Learning for Text Categorization; 1998. .

[16] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. SIGKDD Explor Newsl. 2009 Nov;11(1):10–18.

[17] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A Library for Large Linear Classification. J Mach Learn Res. 2008;9:1871–1874.

[18] Ng AY, Jordan MI. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In: NIPS; 2001. p. 841–848.