Machine Learning for Clinical Text: NLP for Abstracting Breast Pathology Results

David Cronkite, Scott Halgrim, David Carrell
Group Health Research Institute, Seattle, WA

Introduction

The Need

Abstracting structured data from free-text pathology reports (Figure 1) is valuable for research, quality assurance, and patient care.

The Challenge

Manual abstraction is time-consuming, costly, and limits the quantity of information available.

The Opportunity

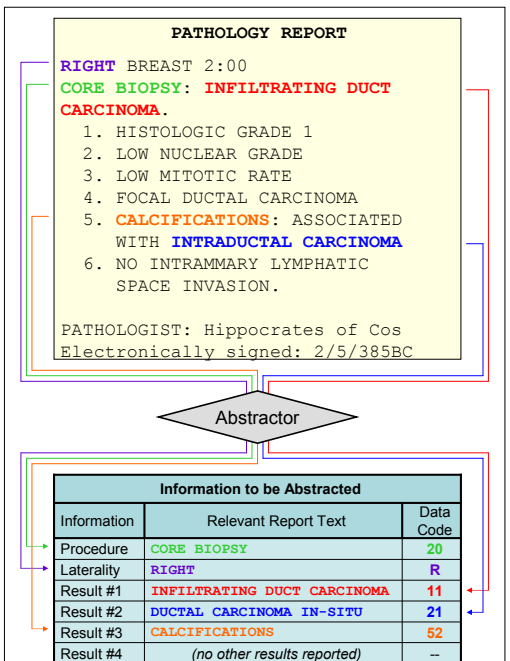
Use **natural language processing (NLP)** to make manual abstraction more efficient, catch errors.

The Approach

Determining a report's results is treated as a set of individual **classification tasks** using the Support Vector Machine (SVM) algorithm.

SVMs belong to the larger class of **machine learning (ML)** algorithms. ML algorithms rely on features to analyze data and recognize underlying distributions (patterns). The SVM is a binary classifier, judging whether or not a report describes (for example) *ductal carcinoma in-situ*.

For each classification task, an SVM model is trained on features derived from training documents with known results (Figure 2, A-D). Trained models are used to classify previously unseen documents (Figure 2, E-G).



Information to be Abstracted

Information	Relevant Report Text	Data Code
Procedure	CORE BIOPSY	20
Laterality	RIGHT	R
Result #1	INFILTRATING DUCT CARCINOMA	11
Result #2	DUCTAL CARCINOMA IN-SITU	21
Result #3	CALCIFICATIONS	52
Result #4	*(no other results reported)*	--

Figure 1. Information in pathology reports (top) is abstracted (middle) and stored as coded data (bottom).

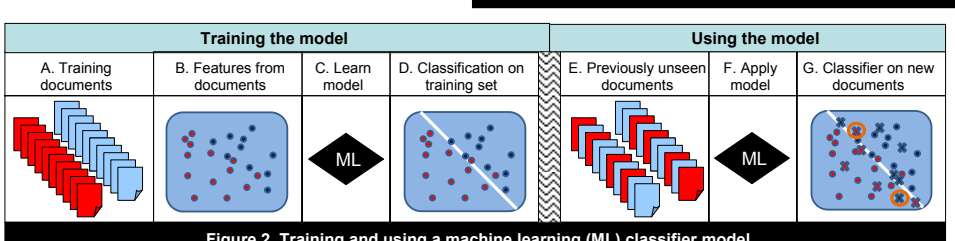


Figure 2. Training and using a machine learning (ML) classifier model.

Source: G:\CTRHS\NLP_Interns\Tiger\Cronkite_Carrell_GHRI_202_IT_PosterSession_ver03.ppt

Methods

Documents

6,965 manually abstracted Group Health breast pathology reports (2009-2011), randomly divided:
- 80% training (N = 5,575)
- 20% test (N = 1,390)

Training set is for developing ML models; test set for assessing performance.

Model Features

1) Lemmas (word stems)
2) Unigrams (single "words" e.g., "duct" or "2:00")
3) Bigrams (e.g., "infiltrating duct" and "duct carcinoma")
4) Trigrams (e.g., "infiltrating duct carcinoma" and "invasive duct carcinoma")
5) Keywords, recognize negation, ignoring misspellings (terms known to be important, such as "DCIS" for "ductal carcinoma in-situ")

Feature Selection

Calculate relevancy of features using Mutual Information and Chi-squared statistical tests.
Retain the more relevant features.

Model Development Process

One model per classification task (e.g., Fig. 1, Result #1).
Start with simple models, gradually adding features.
Iterative use of error analysis to tune the model.
Choose best performing model based on training set.
Test model, once, on the test set.

Model Evaluation

Compare model results to human-abstracted gold standard results on a document-by-document basis.

Future Work

Employ the algorithm on classifying procedure and laterality.

Obtain additional training instances for infrequent codes by drawing from 2001 to present.

Incorporate the classifier into production workflow.

Results

Table 1. Evaluation results: 4 most frequent categories and two infrequent ones.

Result Code	%	Precision	Recall	F-score
60-Benign	19.67	81.82	85.5	83.62
11-Invasive Ductal	13.78	97.02	97.67	97.43
21-Ductal in-situ	11.68	92.16	98.00	94.99
53-Microcalcification	10.56	95.44	97.67	96.54
32-Lob. Atyp. Hyper.	0.55	100.0	88.89	94.12
17-Sarcoma	0.03	100.0	33.33	50.0

Preliminary results from 100/1391 testing instances:

Precision	Recall	F-score
77.0	82.7	80.0

Discussion

Various post-processing rules have not been included:
1) only the top 5 results are retained (regardless of the actual number of results).
2) certain codes are combined into a single code

Model performance improves with increased training data, as the effect of data inconsistencies is diminished.

Models for codes designating broad categories (e.g., 60) perform worse. Additional training data will improve their coverage.

New codes, like 17, partially replace old codes.

Conclusions

Approach is effective on frequently occurring codes.

Some codes are too infrequent to support effective models.

Contacts: David Cronkite (cronkite.d@ghc.org) and David Carrell (carrell.d@ghc.org)