Three Approaches to Automatic Assignment of ICD-9-CM Codes to Radiology Reports

Ira Goldstein, M.B.A., Anna Arzumtsyan, M.L.S., and Özlem Uzuner, Ph.D.
State University of New York, Albany, NY 12222

Abstract

We describe and evaluate three systems for automatically predicting the ICD-9-CM codes of radiology reports from short excerpts of text. The first system benefits from an open source search engine, Lucene, and takes advantage of the relevance of reports to one another based on individual words. The second uses BoosTexter, a boosting algorithm based on n-grams (sequences of consecutive words) and s-grams (sequences of non-consecutive words) extracted from the reports. The third employs a set of hand-crafted rules that capture lexical elements (short, meaningful, strings of words) derived from BoosTexter's n-grams, and that are enhanced by shallow semantic information in the form of negation, synonymy, and uncertainty. Our evaluation shows that semantic information significantly contributes to ICD-9-CM coding with lexical elements. Also, a simple hand-crafted rule-based system with lexical elements and semantic information can outperform algorithmically more complex systems, such as Lucene and BoosTexter, when these systems base their ICD-9-CM predictions only upon individual words, n-grams, or s-grams.

Introduction

International Code for Diseases (ICD) provides a standard for coding the diagnoses and procedures associated with hospital utilization. The Ninth Revision of ICD (ICD-9)[1] provides a standard for coding clinical records. Although primarily used for billing purposes[1], ICD-9 codes can also be useful for detection of epidemics[2] and for the development of patient problem lists[3].

The ICD-9 codes of a clinical record are determined based on the narrative of that record. "On arrival [to the hospital], the patient or a family member is interviewed by a […] nurse who writes the chief complaint in free text on a paper form. A registration clerk then enters the complaint, using ICD-9, into the computerized registration system"[2]. Manual assignment of ICD-9 codes to records is a laborious and error-prone process. Automatic and accurate determination of these codes can reduce the labor involved; it can also help resolve the inconsistencies in coding that arise due to human error. In this paper, we present and evaluate three different approaches to automatically assigning ICD-9-CM codes to radiology reports. ICD-9-CM is the Clinical Modification to ICD-9 and is the standard coding system used in the United States[4].

The systems described in this paper were developed in response to the University of Cincinnati Computational Medicine Center's *International Challenge on Classifying Clinical Free Text Using Natural Language Processing*[5]. Our rule-based ICD-9-CM coder was submitted to the challenge for evaluation and placed second out of 44 systems.

Related Work

Automatically predicting ICD-9-CM codes of medical records requires recognizing the most salient disease(s) or symptom(s) asserted to be present in the patient. This task relates to the body of literature on automatic coding of key concepts in clinical records.

Controlled vocabularies such as those found in the Unified Medical Language System (UMLS)[6,7,8], e.g., SNOMED[9,10], are widely used for coding clinical records. MetaMap[8] identifies candidate phrases through shallow parsing and maps these phrases (or their substrings) to the UMLS Metathesaurus. Elkin, et al.[9] map noun phrases to SNOMED-RT codes. Delbecque[11] first maps phrases to UMLS semantic types and then to more specific semantic categories, e.g., diagnoses and procedures. Lussier, et al.[12] map records to ICD-9 codes.

Many automatic coding systems employ string matching mechanisms[13]; others enrich string features with syntactic information such as parts of speech and phrase tags[9]. For example, Nadkarni, et al.[7] apply phrase analysis to indexing the information in discharge summaries and surgical notes. Friedman, et al. develop a method for mapping entire clinical records to UMLS codes. For this, they use the MedLEE system that fully parses the records[6].

Deviating from coding with controlled vocabularies, Hersh, et al. index findings and diagnoses[14], Sibanda, et al. identify the semantic categories of key concepts[15], and Averbuch uses context to identify negative/positive instances of various symptoms[16].

We use information about diseases, symptoms, and findings mentioned in radiology reports and explore

three algorithmically different approaches to automatically assigning ICD-9-CM codes to these reports. The first approach applies a search engine, Lucene. The second uses boosting implemented by BoosTexter. The third is a rule-based ICD-9-CM coder consisting of hand-crafted rules. The Lucene and BoosTexter approaches employ only individual or strings of words; the rule-based ICD-9-CM coder utilizes additional semantic information. Unlike the efforts in literature, we do not try to capture any syntactic information but find semantic information that can be extracted through surface processing. Using our hand-crafted rules, we explore the contribution of semantic features, i.e., negations, synonyms, and uncertainty, to lexical elements when predicting ICD-9-CM codes.

Data

The data for this study consisted of radiology reports and came from the 2007 Computational Medicine Center challenge[5]. These reports had been preprocessed in two ways: they had been fully de-identified and stripped of most of the "superfluous" content that did not relate to their ICD-9-CM codes. After preprocessing, the narratives of these reports consisted of two fields marked "Clinical_History" and "Impression" (see Figure 1). A representative "Clinical_History" field stated the complaints of the patients, e.g., "Cough", or important medical history, e.g., "Family history of …". A representative "Impression" field noted the findings of the doctors, e.g., "Normal chest radiograph". Typically, each field of a report consisted of one or two sentences.

Three independent coding companies hand-labeled the preprocessed reports with ICD-9-CM codes. These codes were obtained by a majority vote. The codes in each report reflected only the definite diagnoses mentioned in that report; multiple codes per report were allowed.

```
<doc id="99590311" type="RADIOLOGY_REPORT">
<codes>
<code origin="CMC_MAJORITY" type="ICD-9-CM">593.70</code>
<code origin="COMPANY3" type="ICD-9-CM">593.70</code>
<code origin="COMPANY1" type="ICD-9-CM">593.70</code>
<code origin="COMPANY1" type="ICD-9-CM">V13.02</code>
<code origin="COMPANY2" type="ICD-9-CM">593.70</code>
<code origin="COMPANY2" type="ICD-9-CM">599.0</code>
</codes>
<texts>
<text origin="CCHMC_RADIOLOGY"
type="CLINICAL_HISTORY">This patient had a history of urinary
tract infection. This is a followup study. The patient had prior grade II
left vesicoureteral reflux.</text>
<text origin="CCHMC_RADIOLOGY"
type="IMPRESSION">Interval growth of the right kidney. The left
kidney appears stable in size and has not grown significantly since last
exam.</text>
</texts>
</doc>
```

Figure 1. Example XML ICD-9-CM report.

The challenge organizers initially released a training set of 978 labeled radiology reports. We split those reports into two sets. We used 90% (880 reports) for training, parameter tuning, cross-validation, and development of our systems. The remaining 10% (98 reports) were used for initial testing. We refer to this set of 98 reports as the held-out set. The challenge organizers then released a test set of 976 reports. We refer to this set as the challenge test set. 218 of the challenge test set reports were assigned multiple codes, for a total of 1205 code assignments across the 976 reports. 34 of the 45 codes were assigned to more than one report. The Institutional Review board of SUNY Albany approved this study.

Methods

We built and evaluated three different approaches to determining the ICD-9-CM codes of radiology reports. The density of the information in the reports guided our system designs, positing that with minimal "superfluous" text in each report, systems that took advantage of individual or strings of words would be worth pursuing.

Lucene: Our first system used an open source search engine, Apache Lucene[17]. The Lucene library includes text processing utilities, e.g., tokenization tools, which enabled rapid deployment and testing. We used this library, filtered the words that appear in Lucene's default stopword list, i.e., a list of words such as "a", "the", "of", etc. that are usually not useful for searching, and indexed the remaining text of the narratives of the reports in the training set. We then queried the generated index, using the narrative of a target report, i.e., the report to be coded, as the query. We thus determined those reports that were similar to the target based upon their narratives.

Lucene uses the relative importance of words in two reports to compute their similarity to each other, e.g., two reports that overlap in high-weight words are treated as being more similar than two that overlap in low-weight words. For ICD-9-CM coding, we hypothesized that similarity based on term frequency-inverse document frequency (tf-idf)-weighted words of two reports would imply similarity in ICD-9-CM codes. tf-idf is often considered a measure of the relative importance of a word in a document in a corpus. Term frequency (tf) is the number of times that a word appears in the document, divided by the total number of words in the document. Document frequency (df) is the number of documents that contain the given word, divided by the total number of documents in the corpus. Inverse document frequency (idf) is (1/df). tf-idf multiplies tf by idf, capturing the intuition that the more frequently a word appears in a

document (tf), the better it represents the content of that document, and the less frequently the word appears in the rest of the corpus (idf), the more accurately it represents the unique content of that document.

Given tf-idf-weighted words of the narratives of reports, for each of the target reports in the test set, we identified the three training reports most similar to it as computed by Lucene, i.e., the top three reports retrieved by the system. We assigned to each target the ICD-9-CM codes that were used in two or more of the retrieved training reports. In cases where the top three retrieved reports did not reveal a majority code, the fourth training report was also used.

Failure to find a majority vote among the top four retrieved training reports for a target resulted in a NULL code for that target. Through tenfold cross-validation on the training set, we determined that the ICD-9-CM codes of the top three or four retrieved reports were useful for coding the target. The reports that were not retrieved within the top four did not contribute to coding.

BoosTexter: Our second system used BoosTexter[18]. BoosTexter implements boosting, a machine learning algorithm for boosting the performance of supervised learning systems. In general, a boosting algorithm performs several iterations of two steps:
1. it breaks data into subsamples, and
2. it trains a "weak learner", i.e., a classifier that performs slightly better than chance, for the set of subsamples.
At each iteration, the algorithm gives more weight to the samples that had been misclassified in the previous iterations and increases the probability that those samples will be trained on by the next weak learner. At the end of a predetermined number of iterations, the final classifier is created by combining the votes of the weak learners.

BoosTexter classifies text using strings of words that may or may not be consecutive, without giving consideration to semantics. We subdivide these strings into three categories: single words (unigrams), sequences of consecutive words (n-grams), and sequences of non-consecutive words, where some of the words are specified and others are allowed to vary (s-grams). For example, the s-gram "health#?#technology", where the pound sign (#) represents a space and the question mark (?) represents an optional intervening word, can match "health information technology" as well as "health and technology" and "health technology".

Before boosting, we preprocessed the data to remove punctuation and stemmed it using the Porter stemmer[19]. We cross-validated BoosTexter (tenfold) on the training set for parameter tuning. We determined the optimal parameters to be 100 rounds of boosting, with n-grams and s-grams of up to four words. We then trained BoosTexter on the training set once with n-grams and once with s-grams. Analyzing the generated classifiers showed that the n-gram classifier contained just 20 n-grams and 80 unigrams, while the s-gram classifier produced three s-grams, 21 n-grams, and 76 unigrams. While some of the unigrams and n-grams were of value (e.g., "urinary#tract"), we felt the utility of many of the n-grams and s-grams (e.g., "of#?#and", "a#history", and "in#?#of#the") to be questionable.

Rule-based ICD-9-CM Coder: The limited number of predictive unigrams, n-grams, and s-grams used for ICD-9-CM coding by BoosTexter led us to believe in the potential of a rule-based ICD-9-CM coder. Our third system, therefore, implemented a set of simple rules, consisting of four subsets, that we developed on the training set.

The first subset of rules identified lexical elements based upon those useful unigrams and n-grams generated by BoosTexter, and expanded the resulting set of rules with additional lexical elements that helped identify ICD-9-CM codes that were not otherwise addressed. We omitted 17 ICD-9-CM codes from this process. 16 of these codes included six or fewer reports and generating rules for them would potentially overfit the rules to the training data; for the remaining one ICD-9-CM code (see evaluation section regarding V13.02) no unique rule could be created. The rules we created using only lexical elements constituted our base rule-based (BRB) system.

The remaining three rule subsets constituted our semantic components, i.e., they captured shallow semantics. In particular, our second subset of rules was based loosely upon NegEx's pre-UMLS negation phrases[20] and captured the explicitly negated information in the reports. The third subset of rules marked uncertainty with respect to the diagnosis. Following the practice that discourages "over-coding" of reports, i.e., assignment of unnecessary ICD-9-CM codes to a report[5], we treated uncertainty phrases in the Impression field as negations. For example, we took the phrase "may represent atelectasis" to mean that atelectasis (the full or partial collapse of a lung) could not be definitively diagnosed; therefore, the system should not code this report as 518.0 (the ICD-9-CM code for atelectasis). These uncertainty phrases included "most consistent with", "likely", and even "probable". The fourth subset of rules extended the terminology used in the

system to synonyms of disease names. We obtained these synonyms by manually examining the definitions of ICD-9-CM codes, the alternate terminology used for describing the codes, and the index entries containing the codes from www.icd9data.com. Synonyms allowed us, for example, to extend the scope of 788.30 (the ICD-9-CM code for loss of bladder control) from "incontinence" to also include alternatives such as "enuresis" and "wetting".

Our final rule-based ICD-9-CM coder, as submitted to the 2007 Computational Medicine Center challenge, incorporated all four rule subsets (the BRB system and the three semantic components) and is referred to as the full rule-based (FRB) system. The FRB system, in sequence, applied uncertainty, negation, and synonymy rules prior to employing the BRB system. This sequential application prevented conflicts between rule sets.

Evaluation

We evaluated all of our systems on the held-out set and on the challenge test set. We evaluated a total of nine coding systems: the FRB system, the BRB system, the BRB system combined with each of the three semantic components, the two BoosTexter systems, the BoosTexter system with just unigrams, and Lucene. As evaluation metrics, we used micro-averaged precision, recall, and F-measure[21,22] which are derived from true positive (TP), false positive (FP), and false negative (FN) counts. In order to measure the performance on a data set that includes multiple labels for some reports, we followed the scoring system used by the challenge organizers[5] and counted each correctly assigned code as a TP, each missed code as an FN, and each erroneously assigned code as an FP. We compared all of the systems to the baseline of assigning the most frequent ICD-9-CM code, i.e., 786.2, to all of the reports.

Results and Discussion

The results in Table 1 show that the BoosTexter, Lucene, and rule-based systems all perform significantly better than the baseline. We also found significant differences in system F-measures, at α=0.05, between the FRB system and each of the other systems, showing that the FRB system significantly outperforms the BoosTexter, Lucene, and BRB systems. Given the inclination of ICD-9-CM codes to mark only the definite diagnoses, improvement of the FRB system over the BRB system confirms that the lexical elements present in a report do not positively indicate the presence of a disease or symptom in the patient. Studying specific assertions about the presence or absence of these diseases and symptoms can help performance.

Analyzing the results of the BRB system on the challenge test set combined with each of the three semantic components, we found that each of the semantic components significantly improved upon the results of the BRB system. Adding negation to the BRB system corrected 37 FPs and 19 FNs, while inducing additional incorrect codes in just one report. Rules encoding synonyms corrected 1 FP and 85 FNs (59 of which previously had no code assigned by the BRB system). The synonym rules also introduced 19 incorrect codes; six of these incorrect codes were due to an inconsistency in the coding of the ground truth. The annotators used two different codes, 599.0 and V13.02, interchangeably to mark Urinary Tract Infections (UTI). This resulted in some otherwise indistinguishable reports to be marked with two different codes. It also caused the majority vote to fail to identify UTI as a diagnosis for some reports. For example, report number 99590311 (Figure 1) was coded simply as reflux (593.70) instead of both reflux and UTI. Finally, we determined that the uncertainty rules corrected 78 FPs and 40 FNs while introducing coding errors into eight reports.

Conclusion

We presented three different approaches to predicting the ICD-9-CM codes of radiology reports. We showed that a simple algorithm based on individual words taken from the reports and implemented

System	Challenge Test Set			Held-Out Set		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Lucene	0.6946	0.6456	0.6692	0.6417	0.7333	0.6844
BoosTexter unigram	0.8524	0.7477	0.7966	0.7750	0.8611	0.8158
BoosTexter n-gram	0.8562	0.7510	0.8002	0.7833	0.9216	0.8468
BoosTexter s-gram	0.8727	0.7452	0.8039	0.7833	0.9216	0.8468
BRB	0.7992	0.7859	0.7925	0.7667	0.7603	0.7635
BRB + Negation	0.8245	0.7992	0.8116	0.8583	0.8175	0.8374
BRB + Synonyms	0.8011	0.8556	0.8274	0.8000	0.8727	0.8348
BRB + Uncertainty	0.8524	0.8149	0.8333	0.8417	0.8145	0.8279
FRB	0.8758	0.8954	0.8855	0.8667	0.8739	0.8703
Baseline	0.2695	0.2183	0.2412	0.2250	0.2755	0.2477

Table1. Performance Statistics

through the Lucene search engine can give an F-measure of 66.9%. We showed that a more sophisticated learning system, BoosTexter, trained with n-grams and s-grams of words, gives better performance than Lucene. Learning algorithms automatically pick up some features that are predictive of ICD-9-CM codes; however, the explanation behind the predictive power of some of these features can be unclear to humans. Rule-based systems are immune to this problem. Our rule-based ICD-9-CM coder employs lexical elements that build on n-grams and enhances lexical elements with semantic information. Our experiments show that studying explicit negations and uncertainty helps eliminate false positives while synonyms of disease names and disease descriptions improve true positives. We conclude that negation, synonymy, and uncertainty information play key roles in determining ICD-9-CM codes. The resulting rule-based system outperforms the algorithmically more complex BoosTexter (88.5% vs. 80.4%) when BoosTexter is limited to n-grams or s-grams of words.

Future Work

Given our findings, we plan on combining the algorithmically more complex systems such as BoosTexter with the highly informative negation, synonymy, and uncertainty features. We believe that this combination may further improve performance.

REFERENCES

1. Puckett CD. 2004 Annual physician version: The educational annotation of ICD-9-CM. Fifth edition. Reno, NV: Channel Publishing; 2003.
2. Tsui F, Wagner MM, Dato V, and Chang CCH. Value of ICD-9-coded chief complaints for detection of epidemics. J Am Med Inform Assoc. 2002 Nov–Dec; 9(6 Suppl 1): s41-s47.
3. Bui AAT, Taira RK, El-Saden S, Dordoni A, Aberle DR. Automated medical problem list generation: towards a patient timeline. MedInfo, 2004;11(Pt 1):587-91.
4. National Center for Health Statistics. International Classification of Diseases, ninth revision, Clinical Modification (ICD-9-CM). cdc.gov/nchs/about/otheract/icd9/abticd9.htm
5. Pestian, JP, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Bretonnel Cohen K, Duch W. A shared task involving multi-label classification of clinical free text, Proceedings ACL:BioNLP, Prague, June 2007;:97-104.
6. Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004;11(5):392-402.
7. Nadkarni P, Chen R, Brandt C. UMLS concept indexing for production databases: a feasibility study. J Am Med Inform Assoc. 2001;8(1):80-91.
8. Aronson A, Effective mapping of biomedical text to the UMLS Metathesaurus: the Metamap program. AMIA, 2001;:17-21.
9. Elkin PL, Tuttle MS, Keck K, Campbell K, Atkin G, Chute C. The role of compositionality in standardized problem list generation. MedInfo, 1998;:660-4.
10. Wasserman H, Wang J. An applied evaluation of SNOMED CT as a clinical vocabulary for the computerized diagnosis and problem list. AMIA, 2003;:699-703.
11. Delbecque T, Jacquemart P, and Zweigenbaum P. Indexing UMLS semantic types for medical question-answering. Studies in Health Technology and Informatics. 2005; 116:805-10.
12. Lussier Y, Friedman C Shagina L, Eng P. Automated ICD-9 encoding using medical language processing: A feasibility study. AMIA, 2000;:1072.
13. Long W. Extracting diagnoses from discharge summaries. AMIA, 2005;:470-4.
14. Hersh W, Mailhot M, Arnott-Smith C, Lowe H. Selective automated indexing of findings and diagnoses in radiology reports. J Biomed Inform. 2001;34(4):262-73.
15. Sibanda T, He T, Szolovits P, Uzuner Ö. Syntactically-informed semantic category recognizer for discharge summaries. AMIA, 2006 ;:714-8.
16. Averbuch M, Karson T, Ben-Ami B, Maimon O, and Rokach L. Context-sensitive medical information retrieval. MedInfo, 2004;11(Pt 1):282-6.
17. Hatcher E and Gospodnetic O. Lucene in action. Manning Publications, December 2004.
18. Schapire RE and Singer Y. BoosTexter: A boosting-based system for text categorization. Machine Learning, 2000;39(2/3):135-168.
19. Porter MF. An algorithm for suffix stripping, program, 1980;14(3):130-7.
20. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform. 2001;34(5):301-10.
21. Yang Y and Liu X. A re-examination of text categorization methods. Proceedings of SIGIR: International Conference on R&D in Information Retrieval. SIGIR '99. ACM Press; 1999;:42-9.
22. Salton G, McGill MJ. Introduction to Modern Information Retrieval. McGraw-Hill; 1983.