Machine Learning to Automate the Assignment of Diagnosis Codes to Free-text Radiology Reports: a Method Description

Hanna Suominen HANNA.SUOMINEN@UTU.FI
Turku Centre for Computer Science (TUCS) and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Filip Ginter FILIP.GINTER@UTU.FI
TUCS and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Sampo Pyysalo SAMPO.PYYSALO@UTU.FI
TUCS and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Antti Airola ANTTI.AIROLA@UTU.FI
TUCS and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Tapio Pahikkala TAPIO.PAHIKKALA@UTU.FI
TUCS and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Sanna Salanterä SANNA.SALANTERA@UTU.FI
University of Turku, Department of Nursing Science, 20014 University of Turku, FINLAND

Tapio Salakoski TAPIO.SALAKOSKI@UTU.FI
TUCS and University of Turku, Department of Information Technology, 20014 University of Turku, FINLAND

Abstract

We introduce a multi-label classification system for the automated assignment of diagnostic codes to radiology reports. The system is a cascade of text enrichment, feature selection and two classifiers. It was evaluated in the Computational Medicine Center's 2007 Medical Natural Language Processing Challenge and achieved a 87.7% micro-averaged F1-score and third place out of 44 submissions in the task, where 45 different ICD-9-CM codes were present in 94 combinations. Especially the text enrichment and feature selection components are shown to contribute to our success. Our study provides insight into the development of applications for real-life usage, which are currently rare.

1. Introduction

The application of natural language processing (NLP) methods to clinical free-text is of growing interest for both health care practitioners and academic researchers; the motivation being the potential of these applications to support the use of the gathered information in decision-making, administration, science, and education. However, applications used in direct care are still rare.

In spring 2007, an international challenge on the development of machine learning and NLP-based methods for this domain was organized. The task was to automate the assignment of International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM, National Center for Health Statistics (2007)) codes to free-text radiology reports (Computational Medicine Center, 2007). All applicable codes were to be assigned to each report. The challenge organizers motivated the task with its practical importance for hospital administration and health insurance because the codes serve as justification for having specific procedures performed.

Appearing in the Proceedings of the ICML/UAI/COLT 2008 Workshop on Machine Learning for Health-Care Applications, Helsinki, Finland, 2008.

This application paper describes our successful submission: a system that combines feature engineering and two complementary classifiers. In Section 2, we briefly review related work. Sections 3–5 specify the data, introduce our system, and describe the measures used in performance evaluation. In Section 6, we present and discuss our results. Section 7 concludes the study.

2. Background

The transition from paper documentation to electronic patient information systems has enabled machine learning and NLP methods to support the use of the gathered clinical information. This support includes, for example, generating statistics, trends and alerts, but is typically limited to the numerical and structured parts of patient records. However, a considerable amount of clinical information is documented as free-text.

Developing automated tools for free-text patient documents is of growing interest. Although text mining applications taken into clinical practice are rare in particular for minority languages, some success stories exist: As examples relevant to the ICD-9-CM coding topic, we refer to Medical Language Extraction and Encoding System (MedLEE) and Autocoder. MedLEE is routinely used in the New York Presbyterian Hospital to parse English clinical documents and map them to Unified Medical Language System (Bodenreider, 2004) (UMLS) codes (Mendonça et al., 2005). Adapting it for ICD-9-CM coding has also been studied (Lussier et al., 2000). Autocoder is implemented at the Mayo Clinic in Rochester, Minnesota to assign unit specific ICD-9-CM-related codes to patient documents. It has changed the nature of the coding personnel's duties to code verification and consequently resulted in an 80% workload reduction. (Pakhomov et al., 2007.)

When building our ICD-9-CM coding system, we drew on much of our prior experiences in machine learning and NLP method development. Before the challenge we have developed clinical language technology in our on-going project with a focus on supporting the use of the gathered intensive care documentation by identifying text pieces relevant to a given topic (see, e.g., Suominen et al. (2006) and Hiissa et al. (2007)). To lay groundwork for this, we have studied information extraction tasks in the related domain of biomedical scientific publications (Pyysalo et al., 2007). We have also derived efficient methods (see, e.g., Pahikkala et al. (2006; 2007)) that enabled us to fast test which strategies improve classification performance. Further, we have introduced a document classifier supported by domain terminological resources (Ginter et al., 2007), showing the potential of these resources to notably improve performance.

a)
CLINICAL HISTORY
Eleven year old with ALL, bone marrow transplant on Jan. 2, now with three day history of cough.
IMPRESSION
1. No focal pneumonia. Likely chronic changes at the left lung base. 2. Mild anterior wedging of the thoracic vertebral bodies.
ICD-9-CM CODING

786.2	*Cough*

b)
CLINICAL HISTORY
This is a 7-month - old male with wheezing.
IMPRESSION
Borderline hyperinflation with left lower lobe atelectasis versus pneumonia. Clinical correlation would be helpful. Unless there is clinical information supporting pneumonia such as fever and cough, I favor atelectasis.
ICD-9-CM CODING

486	*Pneumonia, organism unspecified*
518.0	*Pulmonary collapse*
786.07	*Wheezing*

c)
CLINICAL HISTORY
7-year - old with history of reflux and multiple urinary tract infections.
IMPRESSION
Interval growth of normal appearing kidneys.
ICD-9-CM CODING

V13.02	*Personal history, urinary (tract) infection*

d)
CLINICAL HISTORY
One UTI. Siblings with reflux.
IMPRESSION
Normal renal ultrasound.
ICD-9-CM CODING

599.0	*Urinary tract infection, site not specified*

Figure 1. Illustration of the data.

3. Data

The anonymized challenge data set was collected from US radiology department for children. The free-text documents described chest x-ray and renal procedures, and each included two parts seen as fundamental for assigning the ICD-9-CM codes: clinical history provided by an ordering physician before the procedure and impression reported by a radiologist after the procedure. Their style was concise and highly domain specific (Figure 1).

The data was accompanied with ICD-9-CM code annotation obtained by a majority vote of three independent parties. The majority vote was selected as the coding task is ambiguous: unit-specific detailed instructions are used to complement the official coding guidelines (Moisio, 2000, pp. 69–126) stating generally, for example, that uncertain codes should not be assigned, a definite diagnosis should be specified when possible, and symptoms must not be coded if the definite diagnosis is available.

Altogether 45 different codes in 94 combinations were present in the data set of 1954 documents. The most common were

1. 786.2 *Cough* ($N = 310$),
2. 599.0 *Urinary tract infection, site not specified* ($N = 193$),
3. 593.70 *Vesicoureteral reflux, unspecified or without reflux nephropathy* ($N = 161$),
4. 780.6 *Fever* AND 786.2 *Cough* ($N = 151$), and
5. 486 *Pneumonia, organism unspecified* ($N = 132$).

An unlabeled test set of 976 reports was made available a month after the training set of 978 documents was released. The sets were restricted by requiring that any combination of codes occurs at least once both in the training and test data.

4. System description

We next introduce our method combining machine learning and NLP for automated assignment of ICD-9-CM codes to free-text radiology reports. It can be divided into feature engineering and classification phases (Figure 2). At the former phase, text is enriched and features improving performance are extracted from the input text. The latter phase contains a cascade of two classifiers.

4.1. Phase 1: Feature engineering

The documents are represented as a set of binary features. The text is initially represented using the simple bag-of-words (unigram) model — the addition of word bigrams and trigrams was tested during development but these features were omitted in favor of the simpler model as they provided no notable performance advantage. Additionally, the text is semantically enriched using concepts from the UMLS metathesaurus and their hypernyms. Further, features are marked for occurrence in a negative context. Finally, the training set is augmented with a small set of artificial examples. The feature engineering is next described in more detail.

Initially, in order to reduce sparseness problems due to inflection and synonymous expressions, the text is tokenized and UMLS concepts occurring in the text are recognized using the MetaMap program (Bodenreider, 2004). From the MetaMap output, the mapping of the text to concepts for which MetaMap assigned the highest score is selected to obtain a set of unique UMLS concept identifiers. For instance, the terms *pneumonia* and *superimposed pneumonia* are both represented by the concept identifier *C0032285*.

In addition, the data is enriched by including also all hypernyms of the directly occurring concepts in the UMLS vocabularies into the feature set. Thus, for an occurrence of the concept *pneumonia*, the feature set is augmented so that it also contains the concept codes for *respiratory tract infection*, *disease caused by microorganism*, *bacterial infection,* et cetera. This allows the similarity of distinct but closely related concepts to be recognized by the machine learning method: for example, all mentions of specific types of respiratory tract infections will introduce the *respiratory tract infection* feature into the feature set.

Negations and conditional statements signaling uncertain or negative findings are identified in the text using a list of common trigger expressions such as *no*, *possible*, *suggestive*, and *likely.* In accordance with the coding guidelines, the goal is to avoid assigning negative or uncertain codes. As a simple heuristic, we assume that the scope of negation is up to the end of the sentence in which it occurs; features extracted from text following a trigger expression up to the end of the sentence are marked, making them distinct from the same features occurring in an affirmative context. Hypernyms of marked concepts are excluded from the feature set so that, for example, *no pneumonia* does not imply *no respiratory tract infection*, as other respiratory tract infections may be present.

Finally, the training set is augmented with 45 instances obtained by catenating the textual description of each of the 45 codes used in the challenge with the descriptions of its parents in the ICD-9-CM tree. For example, the artificial instance corresponding to the code 593.70 *Vesicoureteral reflux, unspecified or without reflux nephropathy* is *Diseases Of The Genitourinary System. Other diseases of urinary system. Other disorders of kidney and ureter. Vesicoureteral reflux. Vesicoureteral reflux unspecified or without reflux nephropathy.* This strategy is based on the intuition that informative keywords appear both in the radiology reports corresponding to a given code and

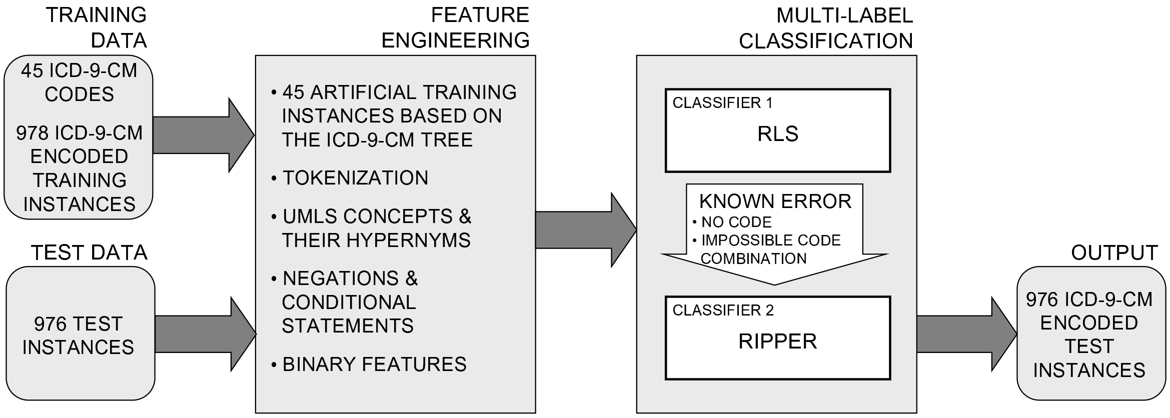
TRAINING DATA
45 ICD-9-CM CODES
978 ICD-9-CM ENCODED TRAINING INSTANCES
TEST DATA
976 TEST INSTANCES
FEATURE ENGINEERING
• 45 ARTIFICIAL TRAINING INSTANCES BASED ON THE ICD-9-CM TREE
• TOKENIZATION
• UMLS CONCEPTS & THEIR HYPERNYMS
• NEGATIONS & CONDITIONAL STATEMENTS
• BINARY FEATURES
MULTI-LABEL CLASSIFICATION
CLASSIFIER 1
RLS
KNOWN ERROR
• NO CODE
• IMPOSSIBLE CODE COMBINATION
CLASSIFIER 2
RIPPER
OUTPUT
976 ICD-9-CM ENCODED TEST INSTANCES

Figure 2. A flow chart of the system components.

in the description of this code in the ICD-9-CM tree (see, e.g., Figure 1a and 1c). Features are extracted from these artificial instances in the same manner as from the original training instances.

4.2. Phase 2: Classification

A machine-learning approach using a cascade of two classifiers trained on the same data is used to predict the codes. Both classifiers perform multi-label classification by decomposing the task into 45 binary classification problems, one for each code. In this setting, it is possible for a classifier to predict an empty, or impossible, combination of codes. Such recognizable mistakes are used to trigger the cascade: when the first classifier makes a known error, the output of the second classifier is used instead as the final prediction. No further correction of the output of the second classifier is performed, as preliminary experiments suggested that this would not further improve the performance. Next, we describe the two classifiers and explain why we chose them.

Classifier 1: Regularized Least Squares

The first classification method is a Regularized Least Squares (RLS) classifier (see, e.g., Rifkin et al. (2003)). This kernel-based classification method is closely related to Support Vector Machines (Suykens & Vandewalle, 1999) and has been shown to have comparable classification performance (Rifkin, 2002).

We formalize the RLS algorithm in the case of our binary classification problem as follows: Let

$$T = ((\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n))$$

be the training set of n input instances $\vec{x}_i \in \mathcal{X}$ and $y \in \mathcal{Y}$ the respective outputs. Further, let $f : \mathcal{X} \to \mathcal{Y}$ be the function that maps the input instances $\vec{x}_i \in \mathcal{X}$ to the outputs $y_i \in \mathcal{Y}$. Here $\mathcal{X}$ is the input space, that is, the sets of possible inputs and $\mathcal{Y} = \{0, 1\}$ the output space. Notice that while we call T a training set, we consider it as an ordered sequence. With this notation $f(\vec{x}_i) \in \mathcal{Y}$ is the hypothesized class for an input instance $\vec{x}_i$. The RLS algorithm can be defined as a minimization problem

$$\mathcal{A}(T) = \min_f \sum_{i=1}^{n} (y - f(\vec{x}))^2 + \lambda \|f\|_k^2, \tag{1}$$

where the regularization parameter $\lambda \in \mathbb{R}_+$ and $\|\cdot\|_k$ is a norm in a reproducing kernel Hilbert space defined by a positive definite kernel function k.

The minimizer of (1) has the form

$$f(\vec{x}) = \sum_{i=1}^{n} a_i k(\vec{x}, \vec{x}_i),$$

where parameters $a_i \in \mathbb{R}$. These parameters can be calculated from

$$\vec{a} = V(\Lambda + \lambda I)^{-1} V^{\mathrm{T}} \vec{y},$$

where $\vec{a} = (a_1, \dots, a_n)^{\mathrm{T}}$, $\vec{y} = (y_1, \dots, y_n)^{\mathrm{T}}$, and V and Λ consist of the eigenvectors and eigenvalues of

$$K = \begin{pmatrix} k(\vec{x}_1, \vec{x}_1) & \cdots & k(\vec{x}_1, \vec{x}_n) \\ \vdots & \ddots & \vdots \\ k(\vec{x}_n, \vec{x}_1) & \cdots & k(\vec{x}_n, \vec{x}_n) \end{pmatrix},$$

respectively. For the multi-label case, $\vec{a}$ and $\vec{y}$ are replaced with the matrices corresponding to the classification task in question. The kernel function which we used is the cosine of the input instances, that is,

$$k(\vec{x},\vec{x}_i) = \frac{\langle \vec{x},\vec{x}_i\rangle}{\sqrt{\langle \vec{x},\vec{x}\rangle\langle \vec{x}_i,\vec{x}_i\rangle}}.$$

We chose to use RLS in the challenge because it has the following computational advantages (Pahikkala et al., 2006; Pahikkala, 2008): Firstly, it is possible to calculate the cross-validation (CV) performance of RLS on the training data without retraining in each CV round. Secondly, the RLS solution can be computed for several different values of the regularization parameter as efficiently as calculating for only one. Thirdly, several learning problems on the same data set can be solved in parallel, provided that the same kernel function is used with each problem, as is the case in our multi-label classification task. Therefore, we can efficiently perform multi-label classification together with fast regularization parameter selection and cross-validation. These properties enabled us to test fast which strategies improve the system performance.

Classifier 2: RIPPER

The second method in the cascade is the RIPPER rule induction-based learning method (Cohen, 1995). The rules learned by the algorithm are formulated in propositional logic. Each individual rule is a conjunction of individual *conditions*. These conditions may be of the form $A = v$ (A being a nominal attribute), or $A \leq v$ or $A \geq v$ (A being a real valued attribute), where v denotes a value. A sequence of such rules is learned for recognizing positive examples. When predicting the class of a new example, each rule is applied. If any one of them matches the example, it is assigned the corresponding label.

The algorithm works as follows. In the initialization phase the training data is split to two sets, the *growing set* used for learning the rules and the *pruning set* used for removing overfitting rules. Each rule starts as an empty conjunction, to which new conditions are added based on an information gain criterion. Once all positive examples are covered, the rule is tested on the pruning set and overfitting conditions are removed from it. Based on the error of the rule on the test set and the total description length of the whole rule set the algorithm decides whether to include the pruned rule in the rule set and continue, or stop. Finally, a post-processing step, guided by a minimum description length-based heuristic, is performed to further optimize the rule set.

We selected the RIPPER algorithm to the cascade due to its excellent performance on the challenge data set and because it has quite a different learning principle than the one embodied by RLS. This may have allowed RIPPER to succeed in cases where RLS failed. The good performance of the conjunctive rules implies that the task of classifying the clinical documents can be reduced to recognizing certain groups of informative keywords from the text, as our intuition about the data was (see also Farkas and Szarvas (2008)). RIPPER was not, however, chosen as primary classifier because RLS performed slightly better in our preliminary experiments.

While the identification of all impossible code combinations might not be straightforward in a real-world setting, we note that the classifier cascade would still be applicable to cases where no code is assigned (approximately 50% of known errors) as well as to cases where, for example, codes to a disease and its symptom are both assigned, a combination excluded by the ICD-9-CM coding rules.

5. Performance evaluation measures

The primary performance measure used in the challenge was a micro-averaged F1-score ($\mathrm{F1}_{\mathrm{mi}}$). In addition, the organizers reported macro-averaged F1 ($\mathrm{F1}_{\mathrm{ma}}$) and cost-sensitive accuracy (CSA), but they had no effect on the submission ranking. All these measures compare the output of the classifier with the gold standard, which is in the challenge the majority annotation.

$\mathrm{F1}_{\mathrm{mi}}$ and $\mathrm{F1}_{\mathrm{ma}}$ are extensions of the F1-score for the multi-label case. The F1-score is a well-established classification performance measure. When only one class is considered, the standard F1-score is defined as the harmonic mean of precision P and recall R,

$$\mathrm{F1} = \frac{2PR}{P+R}, \tag{2}$$

where

$$\begin{aligned} P &= \frac{TP_i}{TP_i + FP_i}, \\ R &= \frac{TP_i}{TP_i + FN_i}, \end{aligned}$$

TP_i is the number of test instances correctly assigned to the class i (i.e., the number of true positives), FP_i the number of test instances the system predicts mistakenly to be a member of the class i (i.e., the number of false positives), and FN_i the number of test instances that belong to the class i in the gold standard but not in the system output (i.e., false negatives). The benefits of the F1-score include its independence on true negatives.

As multi-label classification can be decomposed into distinct binary classification problems, the F1-score (2) can also be calculated separately for each class. $\mathrm{F1}_{\mathrm{ma}}$ is achieved simply by averaging the scores over the classes, that is, if m is the number of classes and $\mathrm{F1}_i$ is the F1-score for the class $i \in \{1, \ldots, m\}$,

$$\mathrm{F1}_{\mathrm{ma}} = \frac{\sum_{i=1}^{m} \mathrm{F1}_i}{m}.$$

In contrast, $\mathrm{F1}_{\mathrm{mi}}$ evaluates the performance by computing the F1-score based on the global perspective of $m \times n_{\mathrm{test}}$ binary labeling decisions. Here m is the number of classes and n_{test} the size of the test set. Let $TP' = \sum_{i=1}^{m} TP_i$, $FP' = \sum_{i=1}^{m} FP_i$ and $FN' = \sum_{i=1}^{m} FN_i$. Then the micro-averaged precision and recall are

$$P_{\mathrm{mi}} = \frac{TP'}{TP' + FP'} \text{ and}$$
$$R_{\mathrm{mi}} = \frac{TP'}{TP' + FN'},$$

respectively. $\mathrm{F1}_{\mathrm{mi}}$ is calculated as in the formula (2) but replacing precision and recall with their micro-averaged variants:

$$\mathrm{F1}_{\mathrm{mi}} = \frac{2P_{\mathrm{mi}}R_{\mathrm{mi}}}{P_{\mathrm{mi}} + R_{\mathrm{mi}}}.$$

$\mathrm{F1}_{\mathrm{ma}}$ and $\mathrm{F1}_{\mathrm{mi}}$ were chosen to be used as they provide supplementary information: by definition, the former emphasizes the significance of performing well on all classes, including relatively ones, whilst the latter weights each code assignment decision equally. Because of the practical orientation towards an evaluation setting that is dominated by the performance in the common classes, $\mathrm{F1}_{\mathrm{mi}}$ is the primary performance measure.

The challenge organizers motivate the use of CSA by clinical regulations enforcing over and under-coding penalties (Pestian et al., 2007): The justification for these lie in the additional risks of possible prosecution for fraud and lost revenues. The proportion of the over-coding penalty p_o to the under-coding penalty p_u is known to be three. If B_i is the number of test instances that are assigned to the class i either in the gold standard or by the system and $B' = \sum_{i=1}^{m} B_i$, then

$$\mathrm{CSA} = \left(1 - \frac{p_u FN' + p_o FP'}{B'}\right)^{\alpha}.$$

In the challenge $\alpha = 1$, $p_o = 1$ and $p_u = 0.33$.

Table 1. An estimate of the effect of the different components on overall performance of our system. Relative decrease in the $\mathrm{F1}_{\mathrm{mi}}$ error (RDE) is given against the results of the previous rows.

Component	$\mathrm{F1}_{\mathrm{MI}}$	Error	RDE
RLS (initial)	79.3%	20.7%	-
Tokenization	80.7%	19.3%	7%
UMLS mapping	82.5%	17.5%	9%
UMLS hypernyms	83.4%	16.6%	5%
Context marking	84.7%	15.3%	8%
Cascaded RIPPER	86.5%	13.5%	12%
ICD-9 instances	86.6%	13.4%	1%

Table 2. Final ranking of top ten submissions (Computational Medicine Center, 2007).

Rank	$\mathrm{F1}_{\mathrm{MI}}$	$\mathrm{F1}_{\mathrm{MA}}$	CSA
1	89.1%	76.9%	91.8%
2	88.6%	72.9%	90.9%
3 (our system)	87.7%	70.3%	91.3%
4	87.6%	72.1%	90.9%
5	87.2%	77.6%	90.1%
6	87.1%	73.3%	89.8%
7	86.8%	73.2%	90.0%
8	85.9%	66.8%	90.5%
9	85.1%	68.2%	90.1%
10	85.0%	67.6%	87.8%

6. Results and discussion

We adopted a modular approach when developing our ICD-9-CM classifier. By following this policy, we evaluated not only the overall quality of the system but also the effect of its different components to the performance by using a 10-fold CV on the training set. We included only the components that led to a better performing system, although we tested also other strategies in addition to those mentioned in Table 1.

For the submission, we allowed our system to learn from all training data available. As expected, this improved the performance giving us on the test set the $\mathrm{F1}_{\mathrm{mi}}$ score of 87.7% and the third place (Table 2).

When developing our system, we focused on maximizing $\mathrm{F1}_{\mathrm{mi}}$. Different performance evaluation measures emphasize, however, different aspects of the problem, and our $\mathrm{F1}_{\mathrm{mi}} = 87.7\%$ illustrates a good quality in relation to the number of code assignments. In terms of the over and under-coding penalty-bearing CSA measure, our submission performed the second best whereas $\mathrm{F1}_{\mathrm{ma}}$ would have dropped us to the seventh place. The latter result is in line with our strategy of considering $F1_{mi}$ and not even aiming at performing well in the all 45 classes. The same trend is also observable in other submissions (Table 2).

According to the report of the challenge organizers (Pestian et al., 2007), initially about 150 registrations were made from six continents and more than 20 countries. When all 44 submissions were considered, the mean, standard deviation and median of $F1_{mi}$-scores were 76.7%, 13.3% and 79.9%, respectively. The organizers' review of all submissions suggests that for this particular task, the choice of the classifier was not crucial for success. However, use of negations, the structure of UMLS, hypernyms, synonyms, and symbolic processing seemed to contribute to performance. These characteristics were evident in our method development too. More information about other highly ranked submissions can be found, for example, in Farkas and Szarvas (2008) (the first place), Goldstein et al. (2007)(the second place) and Crammer et al. (2007) (the fourth place).

The experiences gained from the challenge encourage using machine learning and NLP-based methods to assign ICD-9-CM codes in clinical practice. Firstly, although further software development, integration and piloting is needed, the top systems perform well. Secondly, clinical experiences (Pakhomov et al., 2007) in semi-automated coding at the Mayo Clinic give evidence in support of the claim. Thirdly, the variation in human judgment advocates the use; the pairwise inter-annotator agreement rates measured by using $F1_{mi}$ in the test set were 67.3%, 72.7% and 75.8%, and when compared against the gold standard, the rate varied between 82.6% and 89.6% (Farkas & Szarvas, 2008). Notice that the stronger agreement between human annotators and the gold standard than individual human annotators is explained by the way the gold standard was created based on the majority vote. When comparing the performance of the challenge submissions with these numbers, we see that the top systems outperform some human annotators and are competitive even with the best.

7. Conclusion

We introduced a machine learning-based system for the automatic assignment of ICD-9-CM codes to radiology reports. Of its seven components, those related to feature engineering seemed to provide the competitive edge with respect to other systems submitted to the international challenge.

The experiences gained from the challenge are beneficial for developing human language technology for clinical use. As future work, we are particularly interested in Finnish intensive care and occupational health care domains.

Acknowledgments

This work has been supported by the Academy of Finland and the Finnish Funding Agency for Technology and Innovation, Tekes. We also thank Computational Medicine Center for organizing the challenge and providing anonymized real-world data for developing clinical applications.

References

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, *32S1*, D267–270.

Cohen, W. W. (1995). Fast effective rule induction. *Proceedings of the 12th ICML* (pp. 115–123). Tahoe City, CA: Morgan Kaufmann.

Computational Medicine Center (2007). *The Computational Medicine Center's 2007 Medical Natural Language Processing Challenge.* http://www.computationalmedicine.org/challenge.

Crammer, K., Dredze, M., Ganchev, K., & Talukdar, P. P. (2007). Automatic code assignment to medical text. *Proceedings of ACL BioNLP* (pp. 129–136). Prague, Czech Republic: Association for Computational Linguistics.

Farkas, R., & Szarvas, G. (2008). Automatic construction of rule-based ICD-9-CM coding systems. *BMC Bioinformatics*, *9S3*, S10.

Ginter, F., Pyysalo, S., & Salakoski, T. (2007). Document classification using semantic networks with an adaptive similarity measure. In N. Nicolov, K. Bontcheva, G. Angelova and R. Mitkov (Eds.), *Recent advances in natural language processing IV: Selected papers from RANLP 2005*, 137–146. Amsterdam, Netherlands: John Benjamins.

Goldstein, I., Arzumtsyan, A., & Uzuner, Ö. (2007). Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. *Proceedings of the Fall Symposium of the AMIA* (pp. 279–283). Chigaco, IL: American Medical Informatics Association.

Hiissa, M., Pahikkala, T., Suominen, H., Lehtikunnas, T., Back, B., Karsten, H., Salanterä, S., &

Salakoski, T. (2007). Towards automated classification of intensive care nursing narratives. *Int J Med Inform*, *76*, S362–S368.

Lussier, Y., Shaginai, L., & Friedman, C. (2000). Automating ICD-9-CM encoding using medical language processing: A feasibility study. *Proc AMIA Symp 2000* (p. 1072). American Medical Informatics Association.

Mendonça, E., Haas, J., Shagina, L., Larson, E., & Friedman, C. (2005). Extracting information on pneumonia in infants using natural language processing of radiology reports. *J Biomed Inform*, *38*, 314–321.

Moisio, M. (2000). *A guide to health care insurance billing.* Clifton Park, NY: Thomson Delmar Learning.

National Center for Health Statistics (2007). *International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM).* http://www.cdc.gov/nchs/about/otheract/icd9/abticd9.htm.

Pahikkala, T. (2008). *New kernel functions and learning methods for text and data mining.* TUCS Dissertations No 103. Turku, Finland: Turku Centre for Computer Science.

Pahikkala, T., Boberg, J., & Salakoski, T. (2006). Fast n-fold cross-validation for regularized least-squares. *Proceedings of SCAI 2006* (pp. 83–90). Espoo, Finland: Otamedia Oy.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., & Salakoski, T. (2007). Learning to rank with pairwise regularized least-squares. *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval* (pp. 27–33). Amsterdam, Netherlands.

Pakhomov, S., Buntrock, J., & Chute, C. (2007). Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *J Am Med Inform Assoc*, *13*, 516–525.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D., Johnson, N., Cohen, K. B., & Duch, W. (2007). A shared task involving multi-label classification of clinical free text. *Proceedings of ACL BioNLP* (pp. 97–104). Prague, Czech Republic: Association for Computational Linguistics.

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., & Salakoski, T. (2007). BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, *8*, 50.

Rifkin, R. (2002). *Everything old is new again: a fresh look at historical approaches in machine learning.* Doctoral dissertation, MIT.

Rifkin, R., Yeo, G., & Poggio, T. (2003). Regularized least-squares classification. In J. Suykens, G. Horvath, S. Basu, C. Micchelli and J. Vandewalle (Eds.), *Advances in learning theory: Methods, model and applications*, vol. 190 of *NATO Science Series III: Computer and System Sciences*, chapter 7, 131–154. Amsterdam, Netherlands: IOS Press.

Suominen, H., Pahikkala, T., Hiissa, M., Lehtikunnas, T., Back, B., Karsten, H., Salanterä, S., & Salakoski, T. (2006). Relevance ranking of intensive care nursing narratives. *Lecture Notes in Computer Science*, *4251*, 720–727.

Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Process Lett*, *9*, 293–300.