Procesamiento del Lenguaje Natural, Revista nº 53, septiembre de 2014, pp 77-84 recibido 15-04-14 revisado 30-06-14 aceptado 02-07-14

The aid of machine learning to overcome the classification of real health discharge reports written in Spanish

Aportaciones de las técnicas de aprendizaje automático a la clasificación de partes de alta hospitalarios reales en castellano

Alicia Pérez, Arantza Casillas, Koldo Gojenola, Maite Oronoz, Nerea Aguirre, Estibaliz Amillano
IXA Taldea, University of the Basque Country (UPV-EHU).
{*alicia.perez*, *arantza.casillas*, *koldo.gojenola*, *maite.oronoz*}@*ehu.es*

Resumen: La red de hospitales que configuran el sistema español de sanidad utiliza la Clasificación Internacional de Enfermedades Modificación Clínica (ICD9-CM) para codificar partes de alta hospitalaria. Hoy en día, este trabajo lo realizan a mano los expertos. Este artículo aborda la problemática de clasificar automáticamente partes reales de alta hospitalaria escritos en español teniendo en cuenta el estándar ICD9-CM. El desafío radica en que los partes hospitalarios están escritos con lenguaje espontáneo. Hemos experimentado con varios sistemas de aprendizaje automático para solventar este problema de clasificación. El algoritmo Random Forest es el más competitivo de los probados, obtiene un F-measure de 0.876.
Palabras clave: Procesamiento del Lenguaje Natural, Biomedicina, Aprendizaje Automático

Abstract: Hospitals attached to the Spanish Ministry of Health are currently using the International Classification of Diseases 9 Clinical Modification (ICD9-CM) to classify health discharge records. Nowadays, this work is manually done by experts. This paper tackles the automatic classification of real Discharge Records in Spanish following the ICD9-CM standard. The challenge is that the Discharge Records are written in spontaneous language. We explore several machine learning techniques to deal with the classification problem. Random Forest resulted in the most competitive one, achieving an F-measure of 0.876.
Keywords: Natural Language Processing, Biomedicine, Machine Learning

1 *Introduction*

Thousands of Discharge Records and, in general, Electronic Health Records (EHRs) are produced every year in hospitals. These records contain valuable knowledge sources for further diagnoses, association and allergy development reporting in a population. Apart from the documentation services from the hospitals there are other interests behind mining biomedical records, amongst others from the insurance services. In (Lang, 2007) it is stated that the cost of assigning ICD-9 codes to clinical free texts is $25 billion per year in the US. In particular, this work tackles EHR classification according to Diagnostic Terms (DT). The task deals with Spanish DTs written in spontaneous language.

1.1 Bridging the gap between spontaneous and standard written language

In this particular task we deal with real files written by doctors at the consultation time. The language is not the same as that found in the biomedical literature (e.g. PubMed), in the sense that the language in these records is almost free, including misspells and syntactically incorrect phrases. Being both natural language, we shall refer to the former as *spontaneous* and to the latter as *standard* jargon. At the consultation-time the doctor is devoted to the attention and care of the patient rather than filling the record. As a result, the spontaneous language used by doctors differs from the standardly accepted jar-

ISSN 1135-5948

gon (or the written language they would use in other more relaxed circumstances).

The language gap between these two language-varieties is self-evident in many ways:

- Acronyms: the adoption of non standard contractions for the word-forms.
- Abbreviations: the prefix of the words terminated with a dot.
- Omissions: often prepositions and articles are omitted in an attempt to write the word-form quickly. The verbs are often omitted.
- Synonyms: some technical words are typically replaced by others apparently more frequently used while possibly not that specific.
- Misspells: sometimes words are incorrectly written.

Examples of the aforementioned issues are gathered in Table 1.

1.2 Goals and challenges

In this work we tackle a particular classification problem associated with written spontaneous language processing. We devote to the classification of the discharge records. We are focusing on the records produced at the Galdakao-Usansolo Hospital (attached to the Spanish public hospital-system). These records convey information relative to:

- Personal details of the patient, admission and discharge date, counsellor, etc. In order to preserve the confidentiality, we do not count on this part of the records (it was removed beforehand).
- A narrative summary of the admission details, antecedents, referred main problems, treatment undergone, findings, recommended care plan, etc. This body-part is completely unstructured, since it does not count on sections to extract particular information from. Besides, not all the aforementioned pieces of information are necessarily present in all the records.
- The diagnostic terms together with their associated code in the International Classification of Diseases 9 Clinical Modification[1] (ICD-9-CM). Note that it is the ICD-9-CM that is being followed so far in the hospitals attached to the Spanish Ministry of Health, Social Services and Equality. Admittedly, in some countries the ICD-10 is being used.

453.40	Embolia y trombosis venosa aguda de vasos profundos no especificados de extremidad inferior	TVP MID TVP POPLITEO_FEMORAL MII
600.00	Hipertrofia (benigna) de próstata sin obstrucción urinaria ni otros síntomas del tracto urinario inferior (STUI)	HBP Hipertrofia de Prostata
530.81	Reflujo esofágico	E.R.G.E.
332	Enfermedad de Parkinson	Enf de Parkinson
536.8	Dispepsia y otros trastornos especificados del funcionamiento del estómago	Dispesia alta
185	Neoplasia maligna de la próstata	ca prostata

Table 1: Examples revealing the differences between standard and spontaneous writing. The ICD-9 code appears next to the standard DT, and below spontaneous forms that were assigned the same ICD-9 code are shown.

The aim of the text mining task in which we are focusing on is to get the discharge reports automatically classified by their diagnostic term (DT). That is, the goal is to de-

[1] The International Classification of Diseases 9 Clinical Modification in Spanish is accessible through the web in the Spanish Ministry `http://eciemaps.mspsi.es/ecieMaps/browser/index_9_mc.html`

sign a decision support system to assign an ICD-9-CM code to each DT in the records. So far, a set of experts are in charge of getting the records classified. Hence, all the work is carried out by hand, and our goal is to help to automatize this process. Our aim is to develop a computer aided classification system with very high precision. Addressing this process as a classification problem entails a major challenge: given that the entire ICD-9-CM is being considered, the problem conveys a very large-scale classification system (note that the ICD-9-CM gathers thousands of different classes). Moreover, precision is crucial in this process, and that is, indeed, why we do not aspire to get a fully automatic system.

1.3 State of the art and contributions

Since 1990 the task of extracting ICD-9 codes from clinical documents has become relevant. In 2007 the BioNLP workshop a shared task on multi-label classification of clinical texts was organised (Pestian et al., 2007). For this task it was developed the CMC dataset, consisting of 1954 radiology reports arising from outpatient chest X-ray and renal procedures, observed to cover a substantial portion of paediatric radiology activity. It covered a total of 45 unique codes. The best system of the competition achieved a micro-average F-score of 0.89 and 21 of the 44 participating systems scored between 0.8 and 0.9.

By contrast to the works presented in BioNLP, our work focuses on automatic generation of ICD-9 codes from DTs written in spontaneous Spanish language. We do not examine the whole document. Another relevant difference is that we deal with a problem of an order of magnitude bigger (we envisage more than 678 classes and achieve similar performance). We have also tried different inferred classifiers.

In (Ferrao et al., 2012), they propose a methodology encompassing EHR data processing to define a feature set and a supervised learning approach to predict ICD-9-CM code assignment. Four supervised learning models decision trees, naïve Bayes, logistic regression and support vector machines were tested and compared using fully structured EHR data. By contarst, our data lacks of structure.

The contribution of this work is to delve into real EHR classification on Spanish language. First, we collected a set of real EHRs written in Spanish, and got them fully anonimized. There are works in the literature aiming at overcoming the gap between spontaneous and standard language on the biomedical domain, yet, few of them deal with real EHRs. In this work we explore several machine learning techniques, train them on real EHRs, and assess their performance. Some machine-learning techniques have proven to be able to deal with this big-scale classification problem with quite high precision.

1.4 Arrangement

The rest of the paper is arranged as follows: Section 2 presents the inferred classifiers used in this work and also the means of representing the instances to get them inferred; Section 3 is devoted to present the experimental layout; finally, concluding remarks and some ideas for future work are given in Section 4.

2 Machine Learning

In brief, given a set of discharge records, we focus on the DTs and try to automatically assign the associated ICD-9 code. At first, we thought (and possibly the reader might do now) that this task could be neatly tackled by means of quite a naive system that would simply look up the given DT in the ICD-9-CM catalogue. Nevertheless, we were not aware yet of the aforementioned gap between spontaneous and standard jargon. Indeed, we proceed with this approach and extremely poor results were achieved: only 0.96% of the DTs within the evaluation set were found in the ICD-9-CM catalogue even after applying little modifications such as re-casing, accepting omission of write-accents, getting rid of multiple spaces and allowing to delete the punctuation marks (amongst others). As an alternative, we applied several machine learning techniques in this task.

Bearing in mind the language gap, we tried to approach this task by matching the spontaneous DTs not against the standard DTs from the ICD-9-CM catalogue, but against other sets of data in spontaneous language. That is, the system would learn from previously classified records. All together, this problem can be seen as a supervised classification process, and to that end, we count, in fact, on a set of previously classified set of data.

In this work, we explore four inferred classifiers that have proven successful in text mining problems. All of them were implemented using the libraries available in Weka-6.9 (Hall et al., 2009). Weka is an open-source software that implements a number of machine-learning algorithms, evaluation metrics and other helpful methods.

The machine learning approaches considered in this work are the following ones:

NB Naive Bayes

DT Decision Tree

RF Random Forest

SVM Support Vector Machines

Next, a full description of the machine learning schemes explored, as well as the motivation to do so are presented. The learning scheme and a few important details on the parameters selected for each of them are given. Also in this section, the operational description of the instances used to train the models are given.

2.1 Naïve Bayes

Within a general-framework on a probabilistic approach, the classification problem could be tackled as a maximum likelihood estimation problem:

$$\begin{aligned} \widehat{C} &= \arg\max_{C_k \in \mathcal{C}} p(C_k|\mathbf{x}) = & (1) \\ &= \arg\max_{C_k \in \mathcal{C}} \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_{C_j \in \mathcal{C}} p(\mathbf{x}|C_j)} & (2) \end{aligned}$$

Where the $\mathcal{C}$ is the set of possible classes (in our problem, the set of all the ICD-9 codes that can be given as output), and $\mathbf{x}$ represents the observations (in our problem, the operational representation of the input DT). In our context, each instance $\mathbf{x} \in \Sigma^N$ being Σ the input vocabulary. Besides, $\mathcal{C}$ comprises all the ICD-9 codes (since we are not restricting ourselves to any particular subset such as paediatrics as other works in the literature did).

Admittedly, we are dealing with a large-scale classification problem. In fact, if there are $D = |\mathbf{x}|$ inputs and each of them might take $|\Sigma|$ values, a general distribution would correspond to an application of Σ^D possible values for each class (with a constraint imposed by the total probability theorem). In an attempt to make this problem affordable, the naive-Bayes assumption is made: the features in $\mathbf{x}$ are conditionally independent given the class $C_k \in \mathcal{C}$.

These models were explored as a baseline, since they are efficient and besides they were successful in a number of text mining problems such as spam classifiers in short messages (Sriram et al., 2010; Peng et al., 2012) and also in biomedical classification (Soni et al., 2011; Rodríguez et al., 2012). Nevertheless, for our task it did not result to be competitive enough.

These models were implemented by means of the the `classifiers.bayes.NaiveBayes` library included in Weka (Hall et al., 2009).

2.2 Decision Tree

Decision Tree inference is based on the C4.5 algorithm (Quinlan, 1993). This technique follows a divide and conquer strategy recursively. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the Information Gain (IG), as described in eq. (3).

$$\begin{aligned} IG(\mathcal{X}, A) &= H(\mathcal{X}) - H(\mathcal{X}|A) = & (3) \\ &= H(\mathcal{X}) - \sum_{v \in Val(A)} \frac{|\mathcal{X}_v|}{|\mathcal{X}|} H(\mathcal{X}_v) \end{aligned}$$

where:

- $H(\mathcal{X})$ represents the entropy of the set of instances $\mathcal{X}$ with respect to the class. Likewise, $H(\mathcal{X}|A)$ represents the entropy of the set given the attribute A.
- $Val(A)$ is the set of all the possible values for attribute A.
- $\mathcal{X}_v = \{\mathbf{x} \in \mathcal{X} : \mathbf{x} \cdot A = v\}$ represents the set of instances that take the value v on the attribute A.

In plain words, IG measures the expected reduction in the entropy of the set $\mathcal{X}$ given an attribute A (Mitchell, 1997), and hence, it quantitatively measures the worth of keeping that attribute.

Once an attribute is selected, the set of training samples is divided into sub-sets (according to the value that the samples take for that attribute). The same criterion is recursively applied to each sub-set until convergence according to a an impurity measure (a threshold on the IG). As a result, a tree structure is generated, where the attribute with the highest IG is chosen to make the decision at each stage.

These models were implemented by means of the the `classifiers.trees.J48` library included in Weka (Hall et al., 2009). Besides, a parameter dealing with the impurity, the minimum number of instances in the leaf nodes, was fine-tuned so as to optimize the f-measure on the training set. As a result this parameter was set to 2.

2.3 Random Forest

Random Forest (RF) consists of a variety of ensemble models. RF combines a number of decision trees. The trees involved were close to the optimum tree, yet some randomness was introduced in the order in which the nodes are generated. Particularly, each time a node is generated in the tree, instead of choosing the attribute that minimized the error (instead of Information Gain), the attribute is randomly selected amongst the k best attributes. This randomness enhances the generalization ability of the trees, while the overfitting is avoided. Next, consensus is achieved to decide which class to vote.

These models were implemented by means of the the `classifiers . trees . RandomForests` library included in Weka (Hall et al., 2009). Besides, a parameter relative to the number of trees comprised in the forest was fine tuned so as to optimize the f-measure on the training set. As a result, 9 trees were selected.

2.4 Support Vector Machines

Support Vector Machines (SVMs) are kernel-based models that lay on sparse solutions. The predictions for new inputs rely upon the kernel function evaluated at a subset of the training data points. The parameters defining the model are chosen in a convex optimization problem (local solution is also a global optimum). In SVMs the decision boundary is chosen in such a way that the margin is maximized. That is, if there are multiple solutions that cope with the training data set without errors, the one with the smallest generalization error is chosen (Bishop, 2006).

These models were implemented by means of the the `classifiers.functions.SMO` library included in Weka (Hall et al., 2009). It implements John Platt's sequential minimal optimization algorithm for training a support vector classifier (Platt, 1999). Nevertheless, there exist other more powerful approaches such as LibSVM (Chang and Lin., 2001).

2.5 Operational description of the instances

As it is well-known, the success of the techniques based on Machine Learning relies, amongst others, upon the features used to describe the instances. In this work the operational description of the DTs was done in the same way for all the techniques explored. Admittedly, each technique would be favored by one or another sort of features. Thus, in order to make the most of each learning scheme, appropriate features should be adopted for each of them.

Originally, in the training set the samples are described using a string of variable length to define the DT and a nominal class. That is, while the set of DTs might be infinite, the classes belong to a finite-set of values (all of the ICD-codes admitted within the ICD-9-CM catalogue). In brief, each instance from the supervised set consists of a tuple $(\mathbf{s}, C) \in \Sigma^* \times \mathcal{C}$ being Σ the input vocabulary or a finite-set of words in the input language (hence, Σ^* represents its free monoid) and $\mathcal{C}$ a finite-set of classes.

First of all, a pre-processing was defined to deal with simple string formatting operations. This pre-processing is denoted as h in eq. (4). The application h defines an equivalence class between: lower/upper-case words; strings with and without written accents;...

$$\begin{aligned} h : \Sigma^* \times \mathcal{C} &\longrightarrow \Sigma^* \times \mathcal{C} \\ (\mathbf{s}, C) &\longrightarrow (\mathbf{s'}, C) \end{aligned} \tag{4}$$

The pre-processing defined by h enables the mapping of equivalent strings written in slightly different ways (as it is frequent in spontaneous writing).

Due to the fact that many methods are not able to deal with string-type of features, the transformation f, defined in eq. (5) was applied next.

$$\begin{aligned} f : \Sigma^* \times \mathcal{C} &\longrightarrow \mathcal{X} \times \mathcal{C} \\ (\mathbf{s}, C) &\longrightarrow (\mathbf{x}, C) \end{aligned} \tag{5}$$

Where $\mathcal{X} = 2^{|\Sigma|}$.

The application f acts as a filter. It transforms each string $\mathbf{s}$ (a sequence of words with precedence constraint) into a binary vector referred to the terms of the vocabulary Σ. That is, the element x_i is a binary feature that expresses whether the term $t_i \in \Sigma$ is present in the string $\mathbf{s}$ or not.

The application f is capable of describing each instance by their words as elements. This approach is also referred to as *Bag of Words* (BOW) in the sense that the string is mapped as a set of words without preserving the order, and thus, loosing the involved n-grams. While the precedence of the words is lost, the application allows a simple though effective representation for the instances. Besides, this approach enables a computationally efficient data structure representing the instances as sparse vectors. Hence, the dimensionality of Σ does not convey any computational course.

Note that the DTs consist of short strings with a high semantic component in each word and simple syntax. Intuitively, it is the keywords that matters above the syntax, this is the motivation behind using the BOW as operational description for the instances. Moreover, applying the filter f to the set of instances the dimension of the problem is made affordable since the free monoid comprises all the string whatever their length: $\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$

3 *Experimental framework*

3.1 Task and corpus

We count on a set of DTs written in spontaneous language extracted from real discharge records that were manually coded. The entire set of instances was randomly divided into two disjoint sets for training and evaluation purposes, referred to as Train and Eval respectively.

Table 2 provides a quantitative description of the Train and Eval sets. Each (DT, ICD-9) pair belongs to the pre-processed set of instances, formally denoted as $\mathcal{X} \times \mathcal{C}$ (with the preprocess described in Section 2.5). The first row shows the number of different instances, formally denote as $|\mathcal{X} \times \mathcal{C}|$; the second row, shows the number of different DTs, formally denoted as $|\mathcal{X}|$; the third row, shows the number of different ICD-9 codes, denoted as $|\mathcal{C}|$; the fourth row shows the number of features or relevant words in the vocabulary of the application, formally denoted as $|\Sigma|$.

Note that the number of instances is higher than the number of different ICD-9 codes. This means that some DTs are taken as equivalent, in the sense that different strings were assigned the same ICD-9 code.

	Train	Eval
Different instances	6,302	1,588
Different DTs	6,085	1,554
Different ICD-9 codes	1,579	678
$\lvert\Sigma\rvert$	4,539	

Table 2: Quantitative description of the training and evaluation sets.

On the other hand, since we are working with real data some diseases are more frequent than the others, this makes that some pairs appear more frequently. For example, there are around 3,500 pairs occurring only once, 500 occurring 3 times, and the ratio decreases exponentially, that is, there are very few pairs with high frequency. The distribution is not exactly the same for the DTs or for the ICD-codes, hence, the corpus shows some ambiguities. For example, the code `185` mentioned in Table 1, appears 22 times in the corpus, 17 DTs are different, amongst them, we can see:

- `Adenocarcinoma de próstata con bloqueo hormonal`
- `Ca. próstata metastásico`

3.2 Evaluation metrics

On what the evaluation metrics regards, the following evaluation metrics were considered:

Pr: precision

Re: recall

F1-m: f1-measure

It must be clarified that, given the large amount of classes, the results associated to each class are not provided, instead, a per-class average (weighted by the number of instances from each class) is given (as it is implemented in Weka libraries denoted as *weighted average* for each metric). Per-class averaging means that the number of instances in each class contributes as a weighting factor on the number of true-positives and negatives for that class.

3.3 Results

A twofold evaluation was carried out:

1. **Hold-out evaluation:** the model was trained on the Train set and the predictive power assessed on the Eval set.

2. **Re-substitution error:** the model was trained on the Train set and the predictive power assessed on the Train set. The quality of the training data, the difficulty and the ability of the learning techniques are limited and rarely provide an accuracy of 100%. We could not expect to overcome this threshold on an unseen evaluation set. Hence, in an attempt to get to know the maximum performance achievable on this task, we assessed the performance of the models on the Train set. That is, we explored the predictions exactly on the same set used to train the models. The error derived from this method are the so-called re-substitution error.

On account of this, Table 3 shows the performance of each model (the nomenclature for each model was given in Section 2) on either the Eval or the Train set.

Set	Model	Pr	Re	F1-m
Eval	NB	0.163	0.181	0.131
	DT	0.854	0.851	0.843
	RF	**0.883**	0.881	0.876
	SVM	0.880	**0.889**	**0.878**
Train	NB	0.328	0.394	0.312
	DT	0.905	0.909	0.902
	RF	**0.969**	**0.970**	**0.967**
	SVM	0.959	0.964	0.959

Table 3: Performance of different inferred classifiers on both the evaluation set and also on the training set itself as an upper threshold of the performance.

3.4 Discussion

We proposed the use of Naive Bayes as a baseline system (since it has proven useful in other text mining tasks such as in spam detection), yet, for this task with so many classes has resulted in very poor results. Amongst the explored ML techniques Random Forest presents the highest quality, yet with no significant difference with respect to Support Vector Machines. It is well worth mentioning that the highest f1-measure resulted in 0.876, satisfactorily enough, the upper threshold is not far from that (to be precise, the highest achievable f1-measure is 0.967).

Random Forest comprises 9 Decision Trees, and can be seen as an ensemble model made up of homogeneous classifiers (that is, Decision Trees). Note that the quality provided by a single Decision Tree is nearly the precision achieved by the Random Forest with substantially lower cost.

For this task it is crucial to achieve very high precision, and the Random Forest offers very high precision. Still, on a decision support system we would strive towards 100% precision. Hence, the presented system seems to be much benefitial as a computer aided decision support system, but not yet as an automatic classification system.

It is well-worth endeavoring towards an automatic classification system. Nevertheless, there are evident shortcomings, there are pragmatic limits on this task as it can be derived from the upper performance achievable (see Table 3). Admittedly, it is disappointing not to get an almost-null re-substitution error. A manual inspection of the Train set revealed that the corpus itself had several errors, in the sense that we observed that almost identical DTs had associated different ICD-9 codes. It is quite common not to get flawless datasets, and above all, when they are spontaneous. Moreover, we presented a source of ambiguity in Section 3.1. Possibly, the cause behind these errors might have to do with the conversion from electronic health records to the set of instances. Hence, for future work we will delve into the outlier detection in our training set.

4 Concluding remarks

4.1 Conclussions

This work tackles the classification of discharge records for their DT following the ICD-9-CM standard. The classification problem is quite tough for several reasons: 1) the gap between spontaneous written language and standard jargon; and 2) it is a large-scale classification system (being the number of possible classes the number of different diseases within the ICD-9-CM catalogue). There are few works facing this problem, and the authors are not aware of any in Spanish.

While a look-up in the standard ICD-9-CM provided very poor results, machine learning techniques, trained on spontaneous data resulted very competitive. Due to patient privacy it is difficult to find datasets of clinical documents for free use, this is most evident in the case of clinical text written in Spanish. We would like to remark the importance of harvesting this sort of corpus on the quality of the developed systems.

Amongst the techniques explored, Random Forest resulted the most competitive one (slightly over Support Vector Machines). The best system showed high-quality, an f1-measure of 0.876, being 0.967 the upper threshold for the expected achievable f-1 measure. It would be a great deal to strive towards improving both the hold-out evaluation and its upper boundary.

4.2 Future work

Currently we are working on enhancing the set of features by defining an equivalence class between synonyms derived from the SNOMED-CT (SNOMED-CT, 2012).

In the near future we will delve into the outlier detection in our training set so as to strive into 100% precision on the Train set. The aim will be to filter the outliers so that they do not do harm the inference process.

In this work we explored several ML schemes working alone. Nevertheless, ensemble learning has proven successful in recent research-challenges or competitions. For future work, we mean to double-check if the aforementioned classifiers complement each other and jointly get to improve the performance. Together with this, it could be useful to adapt the features to describe the DTs to each particular learning scheme and also to apply feature subset selection techniques.

As it is the case for speech recognition, we might try to overcome the spontaneous language gap by means of a language model trained on spontaneous data.

Acknowledgments

Authors would like to thank the Hospital Galdakao-Usansolo for their contributions and support, in particular to Javier Yetano, responsible of the Clinical Documentation Service. We would like to thank Jon Patrick for his kind comments on the feature transformation stages and assessment. This work was partially supported by the European Commission (SEP-210087649), the Spanish Ministry of Science and Innovation (TIN2012-38584-C06-02) and the Industry of the Basque Government (IT344-10).

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Chang, C. C. and C. J. Lin. 2001. Libsvm: a library for support vector machines.

Ferrao, J. C., M. D. Oliveira, F. Janela, and H.M.G. Martins. 2012. Clinical coding support based on structured data stored in electronic health records. In *Bioinformatics and Biomedicine Workshops, 2012 IEEE International Conference on*, pages 790–797.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

Lang, D. 2007. Natural language processing in the health care industry. Consultant report, Cincinnati Children's Hospital Medical Center.

Mitchell, T. 1997. *Machine Learning*. McGraw Hill.

Peng, H., C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy. 2012. Using probabilistic generative models for ranking risks of android apps. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 241–252. ACM.

Pestian, J. P., C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. Bretonnel Cohen, and W. Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*, pages 97–104. Association for Computational Linguistics.

Platt, J. C. 1999. Fast training of support vector machines using sequential minimal optimization. *MIT press*.

Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

Rodríguez, J. D., A. Pérez, D. Arteta, D. Tejedor, and J. A. Lozano. 2012. Using multidimensional bayesian network classifiers to assist the treatment of multiple sclerosis. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1705–1715.

SNOMED-CT. 2012. SNOMED CT User Guide. January 2012 International Release. Technical report, International Health Terminology Standards Development Organisation.

Soni, J., U. Ansari, D. Sharma, and S. Soni. 2011. Predictive data mining for medical diagnosis: An overview of heart disease prediction. *International Journal of Computer Applications*, 17.

Sriram, B., D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842. ACM.