# Automatic Diagnosis Coding of Radiology Reports: A Comparison of Deep Learning and Conventional Classification Methods

Sarvnaz Karimi[1], Xiang Dai[1,2], Hamed Hassanzadeh[3], and Anthony Nguyen[3]

[1]Data61, CSIRO, Sydney, Australia
[2]School of Information Technologies, University of Sydney, Sydney, Australia
[3]The Australian e-Health Research Centre, CSIRO, Brisbane, Australia

## Abstract

Diagnosis autocoding is intended to both improve the productivity of clinical coders and the accuracy of the coding. We investigate the applicability of deep learning at autocoding of radiology reports using International Classification of Diseases (ICD). Deep learning methods are known to require large training data. Our goal is to explore how to use these methods when the training data is sparse, skewed and relatively small, and how their effectiveness compares to conventional methods. We identify optimal parameters for setting up a convolutional neural network for autocoding with comparable results to that of conventional methods.

## 1 Introduction

Hospitals and other medical clinics invest in clinical coders to abstract relevant information from patients' medical records and decide which diagnoses and procedures meet the criteria for coding, as per coding standards such as International Statistical Classification of Diseases referred to as ICD Code. For example, *Multiple fractures of foot* is represented by the ICD-10 code 'S92.7'. These codes are used to find statistics on diseases and treatments as well as for billing purposes. Clinical coding is a specialized skill requiring excellent knowledge of medical terminology, disease processes, and coding rules, as well as attention to detail, and analytical skills. Apart from high costs of labor, human errors could lead to over and under-coding resulting in misleading statistics.

To alleviate the costs and increase the accuracy of coding, autocoding has been studied by the Natural Language Processing (NLP) community. It has been studied for a variety of clinical texts such as radiology reports (Crammer et al., 2007; Perotte et al., 2014; Kavuluru et al., 2015; Scheurwegs et al., 2016), surveillance of diseases or type of cancer from death certificates (Koopman et al., 2015a,b), and coding of cancer sites and morphologies (Nguyen et al., 2015).

Text classification using deep learning is relatively recent with promises to reduce the load of domain or application specific feature engineering. Conventional classifiers such as SVMs with well-engineered features have long shown high performance in different domains. We investigate if deep learning methods can further improve clinical text classification. Specifically, we investigate how and in what setting some of the most popular neural architectures such as Convolutional Neural Networks (CNNs) can be applied to the autocoding of radiology reports. The outcomes of our work can inform similar tasks with decision making on type and settings of text classifiers.

## 2 Related Work

In 2007 Pestian et al. (2007) organised a shared task which introduced a dataset of radiology reports to be autocoded with ICD9 codes. This multi-label classification task attracted a large body of research over the years—e.g., (Farkas and Szarvas, 2008; Suominen et al., 2008)—which tackled the problem with methods such as rule-based, decision trees, entropy and SVM classifiers. Text classification using SVM has long been known to be state-of-the-art.

|  |  |  |  | Best Values |  |  |
|---|---|---|---|---|---|---|
| **Parameter** | **Definition** | **Range** | **Default** | **ICD9** | **rICD9** | **IMDB** |
| Batch size | Number of samples that will be propagated through the network at each point of time | 8–256 | 8 | 16 | 16 | 32 |
| Number of epochs | Epoch is one forward pass and one backward pass of all training data | 1–40 | 30 | 30 | 30 | 3 |
| Activation function on convolution layer | Non-linearity function applied on the output of convolution layer neurons | linear, tanh, sigmoid, softplus, softsign, relu, relu6, leaky_relu, prelu, elu | relu | leaky_relu | relu6 | relu |
| Activation function on fully connected layer | Non-linearity function applied on the output of neurons in the fully-connected layer | linear, tanh, sigmoid, softmax, softplus, softsign, relu, relu6, leaky_relu, prelu, elu | softmax | softmax | softplus | softmax |
| Dropout rate | At each training stage, node can be dropped out of the network with probability $1 - p$. The reduced network is then trained on the data in that stage | 0.1 - 1.0 | 0.5 | 0.7 | 0.5 | 0.3 |
| Filter size | Receptive field of each neuron also known as local connectivity | 1–7 and all combinations | (3, 4, 5) | (2, 3, 4) | (2;3;4;5;6) | (3;4;5) |
| Depth | Number of filters per filter size | 40 - 5000 | 100 | 800 | 100 | 200 |
| Learning rate | Controls the size of weight and bias changes during training | 0.0001 - 0.03 | 0.001 | 0.001 | 0.001 | 0.001 |
| Word representation | How words in text are represented as input to the network | See Table 3 | random | Medline (300) | Medline (100) | Medline (40) |
| Vector size | Size of input vectors. When word embeddings are used, this represents the embedding size | 32–512 | 128 | 128 | 128 | 128 |
| Stride | Size of sliding window for moving filter over input | 1 | 1 | 1 | 1 | 1 |

Table 1: Hyperpameters, range of grid search for finding optimal values, initial and best values for three datasets.

Recently, neural network based learning methods have been investigated in generic NLP as well as domain-specific applications. For text classification, two dominant methods are: (1) Convolutional Neural Networks (CNNs) from the category of feed-forward neural networks; and (2) Long Short-Term Memory (LSTM) with a recurrent neural network (RNN) architecture. Also the use of word embeddings (Le and Mikolov, 2014)—which are to capture semantic representations of words in text—has been investigated in a variety of applications to replace one-hot (vector space) models which is the traditional method of text representation.

Text classification using CNNs has been increasingly studied in recent years (Kalchbrenner et al., 2014; Kim, 2014; Rios and Kavuluru, 2015). For example, Rios and Kavuluru (2015) applied CNN to classify biomedical articles for indexing, and Kavuluru et al. (2016) on suicide watch forums.

## 3 Method

We build a CNN network with the architecture proposed by (Kim, 2014). It consists of one convolutional layer using multiple filters and filter sizes followed by a max pooling and fully-connected layer to assign a label. This model is chosen based on its success in other tasks. This will set a base for what is achievable using this set of algorithms without using a very deep network or more complicated architecture.

Input text to the network is represented using two different settings: (1) a matrix of random vectors representing all the words in a document; or (2) word embeddings. We refer to word embeddings created from a corpus of medical text such as Medline citations as *in-domain*, and *out-of-domain* otherwise (i.e., using Wikipedia). We also experimented with *static* and *dynamic* embeddings. In static setting, the embedding vector values were pre-fixed based on the collection they were created on, whereas dynamic embeddings changed values during the training.

One goal of this work is to quantify the impact of CNN hyperparameters. Tuning hyperparameters can be considered equivalent of feature engineering in conventional machine learning tasks. We list some of the main hyperparameters to be set in a CNN in Table 1 (first two columns). Our experiments are focused on tuning these and investigate how they differ for different datasets.

| Classifier | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM | 80.52 | 66.02 | 67.69 | 65.63 |
| Random Forests | 68.22 | 50.85 | 49.38 | 48.66 |
| Logistic Regression | 79.43 | 66.08 | 66.15 | 64.63 |
| CNN (default) | 81.55 | 78.93 | 81.55 | 79.05 |
| CNN (optimal) | **83.84** | **81.44** | **83.84** | **81.55** |

Table 2: Comparison of conventional classifiers with CNN on ICD9.

## 4 Datasets

We experiment on two different datasets, *in-domain* and *out-of-domain*, in order to find common characteristics and domain specific properties of these datasets for text classification. These datasets are: (a) ICD9, a dataset of radiology reports, and (b) IMDB, a sentiment analysis dataset. These corpora are publicly available and are explained below.

ICD9 dataset is an open challenge dataset published by the Computational Medicine Center in 2007 (Pestian et al., 2007). The dataset consists of clinical free text which is a set of 978 anonymized radiology reports and their corresponding ICD-9-CM codes.[1] There are 38 unique ICD-9 codes present in the dataset. Given the imbalance of different disease categories in the dataset with some categories only having one or two instances, we created a revised subset rICD9. In rICD9 those codes with less than 15 instances are removed. This subset contains 894 documents with 16 unique codes. To measure how our grid search for hyper-parameters are robust and how much they are task and dataset dependent, we use an out-of-domain dataset. IMDB movie review dataset is a sentiment analysis dataset provided by Maas et al. (2011). It contains 100, 000 movie reviews from IMDB.

## 5 Experimental Setup

We treat this task as a multi-label classification problem. Our implementations use Tensorflow and Scikit-learn. For word embeddings we use Word2Vec (Mikolov et al., 2013). For SVM and other conventional methods, we used normalized tf-idf features similar to (Wang and Manning, 2012).

**Evaluation** For evaluations on ICD9 and its variant rICD9, we use stratified 10-fold cross-validation. We measure classification accuracy, precision, recall, and F-score by macro-averaging. Stratified cross-validation is used to make label distribution in each training and validation fold as consistent as possible. IMDB dataset has been divided into training data and testing data by its providers. We therefore train the model on the training data and evaluate the results on the test data. For all datasets, all experiments are run for 50 times, and reported results are averaged over repeated experiments.

| Word embedding | Vector Size | Dynamic | ICD9 | rICD9 | IMDB |
|---|---|---|---|---|---|
| Random embedding | | | 81.93 | 86.69 | 87.75 |
| Word2Vec<br>Wikipedia | 40 | Yes | 81.03 | 86.24 | 88.79 |
| | | No | 69.75 | 74.90 | 85.02 |
| | 100 | Yes | 82.04 | 86.86 | 88.55 |
| | | No | 75.93 | 81.40 | 86.98 |
| | 300 | Yes | 82.41 | 87.22 | 88.21 |
| | | No | 79.34 | 84.94 | 88.14 |
| | 400 | Yes | 82.60 | 87.24 | 88.10 |
| | | No | 80.03 | 85.53 | 88.19 |
| Word2Vec<br>Medline | 40 | Yes | 81.59 | 87.06 | **89.00** |
| | | No | 72.31 | 78.05 | 82.11 |
| | 100 | Yes | 82.55 | **87.76** | 89.00 |
| | | No | 78.66 | 84.06 | 85.70 |
| | 300 | Yes | **83.84** | 87.45 | 88.58 |
| | | No | 80.88 | 86.30 | 87.10 |
| | 400 | Yes | 82.55 | 87.56 | 88.62 |
| | | No | 81.39 | 86.66 | 87.21 |

Table 3: Impact of methods of generating word embeddings on classification accuracy.

**Hyperparameter Tuning** Effect of varying different hyperparameters on classification accuracy is examined by a grid search method that incrementally changes the values of hyperparameters. We start from a default setting as shown in Table 1 as a baseline. We also change one parameter at a time, according to a wide range given in column three, and analyze the results to find the optimal hyperparameter values. Based on the optimal parameter values, all experiments are repeated to measure the effects.

## 6 Experiments and Results

### CNN versus Conventional Classifiers

Classification accuracy was calculated varying values of different hyperparameters. Based on the best results we chose the optimal values for each hyper parameter as listed in columns 5 to 7 of Table 1. Table 2 compares three conventional classifiers, including SVM, Random Forests and logistics regression to CNNs. The results for CNN with default values as well as accuracy- optimized values on ICD9 dataset shows comparable re-

[1] Testing data for this dataset is no longer available.

sults to all the three conventional classifiers. That means the two sets of algorithms can achieve similar baselines with minimal feature engineering or parameter tuning.

**Effect of Pre-trained Word Vectors** Pre-trained word vectors can be considered as prior knowledge on meaning of words in a dataset. That is, instead of random values, the embedding layer can be initialized to values obtained from word embeddings. We investigated whether using word embeddings would improve classification accuracy in our coding task. Therefore, we created different word vectors trained using both Wikipedia and Medline with various vector sizes. We then compared the accuracy of random embeddings with these pre-trained embeddings. Our results, shown in Table 3, can be summarized as below: (1) Pre-trained word vectors improve the classification accuracy: The best accuracy achieved on all three datasets come from using pre-trained word vectors. It shows that pre-trained word vectors did improve the effectiveness of our model (t-test, p-value $< 0.05$); (2) Dynamic word vectors are better than static ones: Almost all dynamic word vectors achieve better accuracy than their corresponding static word vectors; (3) In-domain word vectors are better than generic ones: On ICD9 and its variant dataset, word vectors trained using Medline which is a collection of medical articles outperformed the word vectors trained using Wikipedia. It shows in-domain word vectors can better capture the meaning of medical terminology. On the other hand, for IMDB dataset, word vectors trained using Wikipedia were more effective than word vectors trained using Medline, but that was only if the word vectors were static. We believe that a dynamic word vector, regardless of what source it is built on, eventually leads to more accurate classification; and (4) Larger embedding size does not always lead to higher accuracy: For all three datasets, once the vector size was set to 100, the accuracy leveled with higher vector sizes. It means that the computation load associated with bigger vectors may not be necessary.

**Error Analysis** To identify how accurate our CNN classifier was and what mistakes it makes, we manually inspected some of these classification mistakes. We found two major sources of mistakes as below: (1) Not all the documents in ICD9 dataset have exactly one target label. 212 out of 978 documents (22%) have two target labels, and 14 documents have three. These multi-label annotations imply that even human experts cannot have full consensus on some of these coding tasks; and, (2) Companion diseases: Human experts may focus on different symptoms present on a patient report and therefore reach to different conclusions. For example, based on the following text: *"UTI with fever. Bilateral hydroureteronephrosis. Diffuse scarring lower pole right kidney."*, one expert labeled the instance as '591' (Hydronephrosis), and a second expert labeled it as both '591' and '599.0' (Urinary tract infection, site not specified), and a third expert labeled it as '591', '599.0' and '780.6' (Fever and other physiologic disturbances of temperature regulation). In this case, '591' is a majority vote, however, '599.0' may also be a reasonable target, since two of the experts agreed on that. Based on our experiments, accommodating this increases the overall accuracy on ICD9 by approximately 4%.

## 7 Conclusion

We explored the potential of machine learning methods using neural networks to compete with conventional classification methods. We used ICD9 coding of radiology reports. Our experiments showed that some of CNN hyperparameters such as depth are specific to a dataset or task and should be tuned, whereas some of the parameters (e.g., learning rate or vector size) can be set in advance without sacrificing the results. Our results also showed the value of using dynamic word embeddings. Our best classification results achieved comparable or superior results to SVM and logistic regression classifiers for autocoding of radiology reports. Our work is continuing in two major directions: (1) quantifying the relationships between hyperparameters using linear-regression analysis; and (2) applying CNN and LSTM models for ICD-10 autocoding of patient encounters in hospital settings.

## References

K. Crammer, M. Dredze, K. Ganchev, P. Pratim Talukdar, and S. Carroll. 2007. Automatic code assignment to medical text. In *Proceedings of the Workshop on BioNLP*. Prague, Czech Republic, pages 129–136.

R. Farkas and G. Szarvas. 2008. Automatic construction of rule-based ICD-9-CM coding systems. *BMC bioinformatics* 9(3):S10.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, pages 655–665.

R. Kavuluru, M. Ramos-Morales, T. Holaday, A. Williams, L. Haye, and J. Cerel. 2016. Classification of helpful comments on online suicide watch forums. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. Seattle, WA, pages 32–40.

R. Kavuluru, A. Rios, and Y. Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine* 65(2):155–166.

Y. Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pages 1746–1751.

B. Koopman, S. Karimi, A. N. Nguyen, R. McGuire, D. Muscatello, M. Kemp, D. Truran, M. Zhang, and S. Thackway. 2015a. Automatic classification of diseases from free-text death certificates for real-time surveillance. *BMC Mededical Informatics & Decision Making* 15:53.

B. Koopman, G. Zuccon, A. N. Nguyen, A. Bergheim, and N. Grayson. 2015b. Automatic ICD-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics* 84(11):956–965.

Q. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*. Beijing, China, pages 1188–1196.

A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, pages 142–150.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. pages 3111–3119.

A. Nguyen, J. Moore, J. O'Dwyer, and S. Philpot. 2015. Assessing the utility of automatic cancer registry notifications data extraction from free-text pathology reports. In *American Medical Informatics Association Annual Symposium*. pages 953–962.

A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad. 2014. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association* 21(2):231–237.

J. Pestian, C. Brew, P. Matykiewicz, D. Hovermale, N. Johnson, K.B. Cohen, and W. Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Prague, Czech Republic, pages 97–104.

A. Rios and R. Kavuluru. 2015. Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*. Atlanta, Georgia, pages 258–267.

E. Scheurwegs, K. Luyckx, L. Luyten, W. Daelemans, and T. Van den Bulcke. 2016. Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association* 23(e1).

H. Suominen, F. Ginter, S. Pyysalo, A. Airola, T. Pahikkala, S. Salantera, and T. Salakoski. 2008. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: a method description. In *Proceedings of the ICML/UAI/COLT 2008 Workshop on Machine Learning for Health-Care Applications*. Helsinki, Finland.

S. Wang and C. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju Island, Korea, pages 90–94.