Applying deep learning on electronic health records in Swedish to predict healthcare-associated infections

Olof Jacobson
Department of Computer and Systems Sciences, (DSV)
Stockholm University
P.O. Box 7003, 164 07 Kista
olofja@kth.se

Hercules Dalianis
Department of Computer and Systems Sciences, (DSV)
Stockholm University
P.O. Box 7003, 164 07 Kista
hercules@dsv.su.se

Abstract

Detecting healthcare-associated infections pose a major challenge in healthcare. Using natural language processing and machine learning applied on electronic patient records is one approach that has been shown to work. However the results indicate that there was room for improvement and therefore we have applied deep learning methods. Specifically we implemented a network of stacked sparse auto encoders and a network of stacked restricted Boltzmann machines. Our best results were obtained using the stacked restricted Boltzmann machines with a precision of 0.79 and a recall of 0.88.

1 Introduction

Healthcare-associated infections pose a major problem today within healthcare. In Sweden over ten percent of all in-patients suffer from Healthcare-associated infection (HAI). In Europe, this estimates to three million affected patients per year of which about 50,000 die, (Humphreys and Smyth, 2006).

HAI is defined as "an infection occurring in a patient in a hospital or other healthcare facility in whom the infection was not present or incubating at the time of admission." HAI causes patients to suffer while their healthcare periods are prolonged, it also lays an economic burden on the society. The Swedish National Board of Health and Welfare estimates that HAIs prolong the length of a patients stay at the hospital by an average of 4 days, (Burman, 2006).

A number of tools, using the electronic patient record of the patient to detect and predict HAI, have been developed. Freeman et al. (2013) contain a nice overview of over 44 different approaches, however the authors have not distinguished the technology behind the systems, only the accuracy of each system. Ehrentraut et al. (2014) have focused on describing the different systems in a more technical way, most systems use rules to detect HAI, but some few systems are using machine learning based approaches. Three classification algorithms, SVM, Random Forest and Gradient Tree Boosting algorithms, have been used but no attempt has been made using Deep Learning.

In order to distinguish between different types of documents a classifier may be required to recognize abstract concepts in the text. This is made difficult through the nature of human language.

The concepts themselves may not be directly mentioned in the text or may be abbreviated. For example *Pat was op. two days ago*, meaning *The patient was operated two days ago*, or *Patient has fever* meaning that a patient had fever (a symptom) and might have a infection (disease) that needs to be treated. For example *Penomax was given to pat*, meaning that a patient had an infection or had a suspected infection and therefore were given an antibacterial drug for profylactic reasons.

Deep learning architectures transform input data using several non-linear operations. Theoretically such transformations may enable a system to learn high level abstractions in the data, (Bengio, 2009). Deep learning systems are thereby interesting candidates for text classification tasks.

2 Previous research

A number of different methods have been used to detect healthcare associated infections using electronic patient records (EPRs). These range from manual methods, methods that use the structured fields of the EPRs, the clinical free text, as well as methods using both. Automatic methods have

Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pages 191–195,
Berlin, Germany, August 12, 2016.

used rules as well as machine learning methods. Proux et al. (Proux et al., 2009; Proux et al., 2011) describe a rule based system for French patient records. Tanushi et al. (2015) describe another rule based system for Swedish patient records.

Tanushi et al. detected urinary tract infections on 1,867 care episodes and they obtained a precision of 0.98, a specificity of 0.99 and negative predictive value 0.99, but a recall (sensitivity) of 0.60. One approach used machine learning based systems as SVM and Random Forest on the Stockholm EPR Detect-HAI Corpus in (Ehrentraut et al., 2014), where the authors obtained the best results using Random Forest with precision 0.83, recall 0.87 and F-score 0.85. In one other approach using GTB Gradient Tree Boosting the authors obtained a precision of 0.80, a recall of 0.94 and an F_1-score of 0.86, (Ehrentraut et al., 2016).

Deep learning has already been successfully used for natural language tasks. Wiriyathammabhum et al. (2012) applied a deep belief network for word sense disambiguation and found that it had better performance than many shallow machine learning architectures including SVMs. They showed that deep learning could be successfull even with few instances and high dimensionality of the data.

3 Materials and Methods

3.1 Materials

We have been using data from the Swedish Health Record Research Bank, (Dalianis et al., 2015) that contain over 2 million patient records from over 800 clinical units covering the year 2006-2014. and specifically a subset called Stockholm EPR Detect-HAI Corpus[1], (Ehrentraut et al., 2014), that contains 213 patient records written in Swedish that were classified by two different domain experts. The domain experts have reviewed and classified each record both separately and jointly and finally decided on the Gold Standard. The two classes into which the records were divided were patients that suffered from HAIs at some point during the time period covered by the record, and patients that did not. In general, the patient records describe patients that are very ill, see Table 1, for details on the corpus. Each record contains both free text but also structured fields as body temperatures, microbiological answers and drug prescriptions. The records of HAI positive patients are generally longer than those of patients not suffering from HAIs. This imbalance was not addressed by the methodology of the study.

	HAI	non-HAI
Number of records	128	85
Patient ages [years]	2 - 93	2 - 92
Total number of tokens	1,034,760	230,226
Time in hospital [days]	2 - 144	3 - 93
Total time in hospital [days]	3975	941

Table 1: A table detailing some statistics of the Stockholm EPR Detect-HAI Corpus. Tokens refer to space separated sequences of characters. HAI (Healthcare-associated Infection), non-HAI (no Healthcare-associated Infection)

3.2 Methods

The task of classifying texts through machine learning methods generally involve preprocessing and transforming the text to a numerical representation. Some sort of feature selection may then be used before the actual machine learning system attempts to classify the data.

The performance of the implemented systems were tested using a 10-fold cross validation. In each training fold a grid search, validated using 5-fold cross validation, was used to select the hyper parameters of the classifiers.

The varied parameters for the stacked sparse auto encoder were, the number of hidden layers and the sparsity parameter. The number of nodes in each hidden layer was set to be the mean of the number of nodes in the layers immediately before and after. For the stacked restricted Boltzmann machines the gridsearch was performed over several different network topologies with varying number of hidden layers and nodes in the hidden layers. The gridsearch was also performed over the learning rate in this case. All of the programming in this study was carried out in Python.

3.2.1 Preprocessing

The text of the patient records was preprocessed according to the following steps. First all characters not corresponding to Swedish letters, white space, or digits were removed from the corpus. Since accents were inconsistently used, they were removed from all letters in the corpus. The remaining text was converted to lowercase and all

[1]This research has been approved by the Regional Ethical Review Board in Stockholm (Etikprövningsnämnden i Stockholm), permission number 2012/1838-31/3.

stop words were removed using a stop word list from Python's natural language toolkit (NLTK)[2]. Lowercase conversion and stop word removal are standard preprocessing techniques that simplify the corpus and are usually assumed to not affect the classification (Uysal and Gunal, 2014). Stemming has been shown to work well for many tasks. Specifically for Swedish, that has a rich morphology, it has been shown that stemming improves the results for information retrieval tasks, (Carlberger et al., 2001). Therefore the text in the patient records was stemmed using the Python Snowball stemmer[3] for Swedish.

3.2.2 Conversion to numerical representation

In this study two different models for converting the records into numerical vectors were used. One of these models was a bag of words model where the tf-idf scores for the individual words in the patient records were calculated. Only the 1,000 most common words in the whole corpus were considered. Principal component analysis (PCA) was used to generate acompleted only the encodine tf-idf representation with reduced dimensionality, 99% of the variance was retained.

The other model used for conversion was the word2vec tool in Python's gensim package (Řehůřek and Sojka, 2010) using the skip-gram model, through which each word in a record could be converted into a vector. A record was represented by taking the mean of all such vectors in the record.

3.2.3 Artificial neural networks

Artificial neural networks is a biologically inspired type of machine learning architecture. The basic building blocks of these networks are inspired by the biological neuron and are usually referred to as nodes (Marsland, 2009).

The nodes are connected to other nodes forming a network which can be trained by various methods to perform different tasks (Amaral et al., 2013). In this study two different types of neural networks were used. Stacked sparse auto encoders and stacked restricted Boltzmann machines.

3.2.4 Stacked sparse auto encoders

A stacked sparse auto encoder is a neural network technology which consists of several sparse auto encoders. An auto encoder is a type of neural network that first encodes and then decodes input data. Auto encoders are trained to reproduce the data sent through them (Bengio, 2009). The cross entropy cost function was used for the auto encoders in this study. When the sparse auto encoders were stacked, each encoder was trained to reproduce the encoded data of the previous encoder. This unsupervised training constituted the pretraining phase of the network. After the pretraining was completed only the first half of the auto encoders, responsible for the encoding, was used in the resulting network. A softmax classifier was appended to the last auto encoder and the network was trained in a supervised manner through back propagation.

3.2.5 Stacked restricted Boltzmann machines

A restricted Boltzmann machine (RBM) is a special case of the more general Boltzmann machine. An RBM consists of one layer of visible nodes and one layer of hidden nodes. Nodes in the hidden layer are only allowed to connect to nodes in the visible layer (Bengio, 2009). One notable difference between auto encoders and RBMs is that RBMs have binary activation of the hidden layer nodes during the training phase. In the implemented stacked RBM topology the RBMs were trained to reproduce data input on the visible layer through the means of persistent contrastive divergence (Tieleman, 2008). The stack of RBMs was created in a similar way to the previously described stack of auto encoders. Each RBM was trained on the hidden layer activations of the previous RBM. A softmax classifier was appended at the end of the network and was trained in a supervised manner on the hidden layer representations of the previous RBM. Unlike the stack of auto encoders, the whole network was not fine tuned through supervised training.

4 Results

The results are presented in Table 2 where the acronyms used are the following: SSAE, *Stacked Sparse Auto Encoder classifier*. SRBM, *Stacked Restricted Boltzmann Machine classifier* and tf-idf, the *tf-idf* representation of the data was used. PCA, a version of the tf-idf data that had its dimensionality reduced through *Principal Component Analysis* was used. And finally word2vec, the *word2vec* representation of the data was used.

[2]Accessing Text Corpora and Lexical Resources, *http://www.nltk.org/book/ch02.html*

[3]Swedish Snowball stemming algorithm, *http://snowball.tartarus.org/algorithms/swedish/stemmer.html*.

	Precision	Recall	F_1-score
SSAE tf-idf	0.78	0.78	0.78
SSAE PCA	0.71	0.80	0.75
SSAE word2vec	0.78	0.84	0.81
SRBM tf-idf	0.79	0.88	0.83
SRBM word2vec	0.66	0.91	0.77

Table 2: The classification results for the different machine learning methods and preprocessing techniques.

The highest F_1-score was obtained for the stacked restricted Boltzmann machines and that classifier also had the highest precision. The word2vec version of the data gave lower F_1-score than the regular tf-idf data when the SRBM classifier was used but a higher F_1-score for the SSAE classifier. The dimensionality reduced version of the tf-idf data gave a slightly higher recall, and a lower precision as compared to the regular data.

5 Discussion

Deep learning models are computationally expensive to train. This is a disadvantage compared to simpler models. The disadvantage can in many cases be motivated by an increase in classification performance. However that was not observed in this study.

We could see that the SRBM with the tf-idf data gave the best results of our approaches. The results were comparable to those obtained by Ehrentraut et al. (2014), however they were slighly lower than the results presented by Ehrentraut et al. (2016). It was surprising that the classification accuracy for the word2vec representation of the data was higher than for the tf-idf version, for the SSAE classifier. The SRBM classification accuracy was lower for the word2vec representation than for the tf-idf representation. No appealing explanation was found for the difference.

In many cases the gridsearch parameter optimization preferred few layers in the neural networks. This implies that deeper architectures did not help the classifier identify useful abstract patterns in the data. This could be due to the chosen data representations that may not have retained some of the important features of the original data. The lack of training data may also have been contributing to the non efficacity of deeper architectures, since such architectures typically require a lot of training data. In this study the ratio of the number of features, and the number of training examples, may simply have been too large to enable the networks to learn abstract patterns.

6 Conclusion

The classification results were comparable to what has been obtained using more conventional classifiers in previous studies. The SRBM architecture using the bag of words, tf-idf data was the most successfull classifier in this study, see Table 2. SRBM gave a precision of 0.79, a recall of 0.88, and an F_1-score of 0.83. These results were slightly lower than those of the best classifier in previous studies, which had a precision of 0.80, a recall of 0.94 and an F_1-score of 0.86 (Ehrentraut et al., 2016). The word2vec representations scored worse than the bag of words model for the SRBM classifier while giving better scores for the SSAE classifier. The PCA version of the tf-idf data with reduced dimensionality gave worse classification results than the regular data.

In the future we would like to try out a wider range of preprocessing methods, similar the ones tried out by Ehrentraut et al. (2014), to detect negated symptoms or diseases, so called negation detection, for example to perform *negation detection*, or *remove negations*, or to carry out *stop word filtering* and finally *filtering out infection specific terms* to see if that will improve our results.

Acknowledgements

We would like to thank Mia Kvist and Elda Sparrelid both at Karolinska University Hospital. We would also like to thank Claudia Ehrentraut and Hideyuki Tanushi for their ground breaking work to construct the Stockholm EPR Detect-HAI Corpus.

References

Telmo Amaral, Luís M. Silva, Luís A. Alexandre, Chetak Kandaswamy, Jorge M. Santos, and Joaquim Marques de Sá. 2013. Using different cost functions to train stacked auto-encoders. In *MICAI (Special Sessions)*, pages 114–120. IEEE.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January.

Lars G Burman. 2006. Att förebygga vårdrelaterade infektioner - ett kunskapsunderlag. *Stockholm: Swedish National Board of Health and Welfare. ISBN*, 595547553, (In Swedish).

Johan Carlberger, Hercules Dalianis, Martin Hassel, and Ola Knutsson. 2001. Improving precision in information retrieval for Swedish using stemming. In *Proceedings of NODALIDA '01 - 13th Nordic Conference on Computational Linguistics*.

Hercules Dalianis, Aron Henriksson, Maria Kvist, Sumithra Velupillai, and Rebecka Weegar. 2015. Health bank - a workbench for data science applications in healthcare. In *Proceedings of the CAiSE-2015 Industry Track co-located with the 27th Conference on Advanced Information Systems Engineering - CAiSE 2015*, pages 1–18, Stockholm, Sweden, June. CEUR.

Claudia Ehrentraut, Maria Kvist, Elda Sparrelid, and Hercules Dalianis. 2014. Detecting healthcare-associated infections in electronic health records: Evaluation of machine learning and preprocessing techniques. In *Sixth International Symposium on Semantic Mining in Biomedicine (SMBM 2014), Aveiro, Portugal, October 6-7, 2014*, pages 3–10. University of Aveiro.

Claudia Ehrentraut, Markus Ekholm, Hideyuki Tanushi, Jörg Tiedemann, and Hercules Dalianis. 2016. Detecting hospital acquired infections: A document classification approach using support vector machines and gradient tree boosting. *Health Informatics Journal*, To be published.

Rachel Freeman, Luke S. P. Moore, Laura García Álvarez, Andre Charlett, and Alison Holmes. 2013. Advances in electronic surveillance for healthcare-associated infections in the 21st Century: a systematic review. *Journal of Hospital Infection*.

Hilary Humphreys and Edmund T. M. Smyth. 2006. Prevalence surveys of healthcare-associated infections: what do they tell us, if anything? *Clinical Microbiology and Infection*, 12(1):2–4.

Stephen Marsland. 2009. *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC, 1st edition.

Denys Proux, Pierre Marchal, Frédérique Segond, Ivan Kergourlay, Stéfan Darmoni, Suzanne Pereira, Quentin Gicquel, and Marie-Hélène Metzger. 2009. Natural Language Processing to detect Risk Patterns related to Hospital Acquired Infections. In *Proceedings of the Workshop Biomedical Information Extraction*, pages 35–41. Association for Computational Linguistics.

Denys Proux, Caroline Hagège, Quentin Gicquel, Suzanne Pereira, Stefan Darmoni, Frédérique Segond, and Marie-Hélène Metzger. 2011. Architecture and Systems for Monitoring Hospital Acquired Infections inside a Hospital Information Workflow. In *Proceedings of the Workshop on Biomedical Natural Language Processing. USA: Portland, Oregon*, pages 43–48. Citeseer.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Hideyuki Tanushi, Maria Kvist, and Elda Sparrelid. 2015. Detection of Healthcare-Associated Urinary Tract Infection in Swedish Electronic Health Records. *Innovation in Medicine and Healthcare 2014*, 207:330.

Tijmen Tieleman. 2008. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071.

Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information Processing and Management*, 50(1):104–112.

Peratham Wiriyathammabhum, Boonserm Kijsirikul, Hiroya Takamura, and Manabu Okumura. 2012. Applying deep belief networks to word sense disambiguation. *CoRR*, abs/1207.0396.