Detecting Low Back Pain from Clinical Narratives using Machine Learning Approaches

Michael Judd[1] , Farhana Zulkernine[2], Brent Wolfrom[3], David Barber[4], Akshay Rajaram[5]

[1,2,3,4,5]Queen's University, Kingston, Ontario, Kingston, ON K7L 2N8
{[1]m.judd, [2]farhana}@queensu.ca, {[3]brent.wolfrom,
[4]david.barber}@dfm.queensu.ca, [5]arajarem@qmed.ca,

Abstract. Free-text clinical notes recorded during the patients' visits in the Electronic Medical Record (EMR) system narrates clinical encounters, often using 'SOAP' notes (an acronym for subject, objective, assessment, and plan). The free-text notes represent a wealth of information for discovering insights, particularly in medical conditions such as pain and mental illness, where regular health metrics provide very little knowledge about the patients' medical situations and reactions to treatments. In this paper, we develop a generic text-mining and decision support framework to diagnose chronic low back pain. The framework utilizes open-source algorithms for anonymization, natural language processing, and machine learning to classify low back pain patterns from unstructured free-text notes in the Electronic Medical Record (EMR) system as noted by the primary care physicians during patients' visits. The initial results show a high accuracy for the limited thirty-four patient labelled data set that we used in this pilot study. We are currently processing a larger data set to test our approach.

Keywords: text-mining, natural language processing, machine learning, clinical decision support system, back pain.

1 Introduction

Physicians around the world use a variety of EMR systems to log data about patients' visits. These systems include patients' demography, physical and health-related information, laboratory tests, medications, billing information and unstructured text notes that the physicians enter in the EMR system based on their conversations with the patients. The patient-specific medical data, and more specifically, the free-text reports, provide an invaluable source of information for medical diagnosis. However, analyzing the unstructured text notes pose a difficult challenge due to the pervasive acronyms, spelling and typing errors, and dense author and domain-specific idiosyncrasies [1]. At the same time, correct extraction and appropriate presentation of the knowledge in the text notes and linking it with other structured health data can help physicians immensely in deciding about the right treatment plan. There is a strong movement towards analyzing free-text medical data to revolutionize the healthcare research not only for the sake of biomedical research such as diseases, comorbidities and drug interactions, but also for data-driven and evidence-based decision-making in healthcare [2]. With hospital

readmission penalties reaching an all-time high of over five billion dollars in the United States, the need for medical decision-support systems has never been greater [3].

With this global movement towards implementing EMRs, significant interest has been directed towards automating the processing and analyses of medical data [4]. The information within these EMRs enables us to build machine learning models for disease prediction which is also known as case-detection and suggest alternative treatments. Because the unstructured text notes provide a wealth of information, researchers are exploring a variety of Natural Language Processing (NLP) [5], Text Mining (TM) [4], and semantic knowledge management techniques [6] to accurately process this data. A recent review of text-mining algorithms for case-detection shows that they can be very effective when combined with the structured information within the EMR for case-detection; providing above 90% sensitivity and specificity in some studies [7]. Medical Decision Support Systems (DSS) such as IBM Watson Health [8] are applying cognitive computing techniques to ingest information from many sources along with statistical data mining and machine learning techniques to predict possible disease diagnosis and suggest alternative treatment plans. Medical data in the EMRs also contain hybrid data such as lab reports associated with image data. Researchers are working on extracting knowledge from these hybrid-distributed data sources and linking them to create efficient DSS [9].

Despite the potential, researchers are still facing many challenges and obstacles when applying NLP and TM on medical data. One major difficulty arises from the strict privacy and security requirements concerning medical data, which sets ethical and legal obstacles in accessing the data [10]. The free text data contains doctors' subjective opinions. Therefore, developing TM and machine learning algorithms to perform reliable case-detection based on doctors' chart notes poses yet another critical challenge. Furthermore, the text data are often incomplete or erroneous, contain domain specific terminology, and express negative, positive or neutral sentiments, which need to be interpreted correctly to develop a reliable DSS.

In this pilot study, we focus on applying unstructured text data analytics to the Open Source Clinical Application Resource (OSCAR) EMR [11] to diagnose chronic low back pain patterns. Low back pain is a common and costly health condition in Canada. A survey of 2400 Canadian's revealed an 83% lifetime prevalence of low-back pain, with 61% reporting low-back pain in the past year [12]. Canada's cost of medical expenditures for low back pain alone are estimated between $6 and $12 billion per year [13]. As low back pain is often first reported to primary care practitioners, these health professionals act as the gate keepers to referrals for further costly investigations [14], including imaging and specialist appointments. In alignment with guidelines from the College of Family Physicians of Canada, imaging for low back pain should not be recommended unless red flags are present [15]. Red flags can include: suspected epidural abscess or hematoma, suspected cancer, suspected infection, severe or progressive neurologic deficit, and a suspected compression fracture. However, literature shows that primary care physicians are in fact over imaging for non-specific low back pain [13]. Low back pain symptoms are not detectable from the various health metrics such as blood pressure, height, weight, body mass index, and lab values. The overall goal of this pilot project was to study the effectiveness of NLP in analyzing the free text notes

and identifying patients with low back pain by validating the results against the existing manual auditing process.

We explored existing open source de-identification tools to anonymize the text data while retaining necessary information in a format so that we can subsequently apply NLP techniques. We studied existing NLP tools and applied Apache cTAKES [16] to extract medical terms. Finally, we developed an information processing workflow and a DSS for the diagnosis and classification of low back pain. Therefore, the main contributions of the paper are: (i) a study of free and open-source software (FOSS) for de-identification and NLP of unstructured medical text notes, (ii) definition of an information processing workflow, and (iii) implementation and validation of a DSS for the classification of low back pain for a very small set of labelled data.

The remaining sections of the paper are organized as follows. Section 2 provides the background and lists some of the related work. Our framework is presented in Section 3. A discussion of the results is presented in Section 4. Section 5 concludes the paper stating our ongoing and future work.

2 Background and Related Work

2.1 Anonymization of Clinical Text

Anonymization of Protected Health Information (PHI) in clinical text is a crucial step within text-mining EMRs to preserve patient confidentiality. Anonymization involves the removal of patient identifiers defined by regulatory acts such as HIPPA. In most cases, only researchers with local authorization can analyze the data due to anonymization methods not being 100% accurate and as such, creates a bottleneck for research in the area [17].

Automating the process of de-identifying clinical text is a key to the success of finding a generalized way to text-mine EMRs. Currently, there are two main algorithmic methods used to anonymize PHI: rule-based and machine learning [18]. Both methods have their own unique sets of benefits and disadvantages to consider. Rule-based systems are a proven method that can guarantee that specific elements are always anonymized but requires time-consuming rule specification and customization for each dataset. Researchers must add entity specific information such as patient names or nearby locations to large dictionaries for the algorithm to work [19].

Conversely, most of the machine learning methods use supervised learning methods which attempt to identify words as PHI or not PHI. These methods do not require extensive customization; however, they require large datasets annotated by domain experts which are difficult to obtain. In a review of automatic de-identification methods for medical records, Meystre et al [20] stated that methods based on dictionaries performed better when PHI is rarely mentioned in text, but machine learning methods performed better over all. The review also concluded that hybrid methods using both machine learning and rule-based pattern matching performed best in terms of accurately removing PHI.

The major downside of all anonymization methods is the reduction in data quality for NLP processing. Medical text data is already highly irregular and difficult for NLP

tasks, and the anonymization of entities makes medical concepts recognition increasingly more difficult. Any anonymization algorithm would have to be carefully tuned not to over-scrub the data to withhold the accuracy of NLP algorithms.

2.2 Clinical NLP Systems

Numerous clinical NLP systems have been developed to solve many of the biomedical text-mining problems [1][4][16]. They work by using standard NLP tasks such as tagging, chunking, parsing, entity recognition, and summarization combined with the information from machine readable medical domain knowledge-base systems such as the Unified Medical Language System (UMLS) [23] to extract clinical concepts from free-text. Some of the most accurate systems being developed are unfortunately proprietary, however, there are successful FOSS systems including cTAKES, CLAMP Toolkit, MedEx, MedKAT/p, and more recently QuickUMLS [1]. These systems have all been used in separate studies with varied levels of success, however, little work has been done to compare the effectiveness of these existing clinical NLP tools.

Currently, cTAKES, the system implemented in this framework, seems to have the most comprehensive solution for clinical NLP. With a large user community and the Apache Software Foundation behind the project, the system should continue to grow and improve. Originally developed by the Mayo Clinic, cTAKES is based upon the Apache Unstructured Information Management Architecture (UIMA) framework. It utilizes the OpenNLP toolkit for NLP and the UMLS for medical concept annotation. The design of the system is pipeline-based, consisting of different modules that include a sentence boundary detector, tokenizer, normalizer, part-of-speech tagger, shallow parser, and named entity recognizer [21].

An alternative system that showed promise for building a generic framework for case-detection was QuickUMLS [1]. QuickUMLS is a tool for fast, unsupervised biomedical concept extraction from medical text. To find medical concepts, QuickUMLS takes advantage of Simstring [22], a fast algorithm for approximate string matching. The tool achieves precision and recall comparable to cTAKES for medical concept extraction at speeds up to 135 times faster than the comparable systems. This makes QuickUMLS a strong contender as a generalized text-mining tool for clinical decision support when scalability becomes increasingly important. The downside of exclusively using QuickUMLS is that it lacks the features of a full NLP suite such as negation and part of speech tagging for the possibility of deeper text analytics.

cTAKES and QuickUMLS, along with many other NLP systems rely heavily on the UMLS for clinical information extraction. The UMLS is a resource containing many health and biomedical vocabularies developed by the US National Library of Medicine to enable interoperability between computer systems. It integrates over two million names mapping to about 900,000 concepts from more than sixty families of biomedical vocabularies, and about twelve million relationships exist between these concepts [23]. The main UMLS tool that NLP systems use is called the Metathesaurus [23], which is a large biomedical thesaurus organized by concepts or meanings, and links to similar concepts from 200 vocabularies over various languages. It also identifies relationships between concepts with a tree structure, along with definitions and types for these concepts. For building a generic multilingual NLP framework for medical data, UMLS is

extremely useful as it provides many synonyms for medical terms and supports multiple languages.

2.3 Clinical NLP Systems

Based on our review of the literature on disease case-detection, previous research efforts were mostly geared towards diagnosing specific disease cases rather than building a general framework such as the framework proposed in this paper. Many case-detection frameworks applied custom-built NLP solutions or rule-based algorithms to diagnose cancer, heart disease, and mental health with reasonable success. However, they all lack the generalization to apply to other disease conditions [15][24].

In recent years, researchers have proposed some generalized machine learning-based case-detection frameworks [18][25]. D'Avolio et al. [25] developed a generalized framework which consisted of a modified version of cTAKES for the NLP component and the Machine Learning for Language Toolkit for case-detection. Like the method used in this paper, the algorithm tested a number of different machine learning algorithms, attempting to maximize the classification accuracy. The system chose various feature sets of cTAKES output to classify three types of cancer with 0.90 accuracy in two of three cases.

Szlosek and Ferret [18] created a clinical DSS using SpaCy, a general NLP library with linear time processing algorithms, and the Scikit-learn library of machine learning algorithms. SpaCy was used to clean, tokenize, and vectorize the text data. Case-detection was done using Scikit, which tested C-Support Vector Classification, Decision Tree Classifiers, and k-nearest neighbors classifiers. The DSS quickly managed to

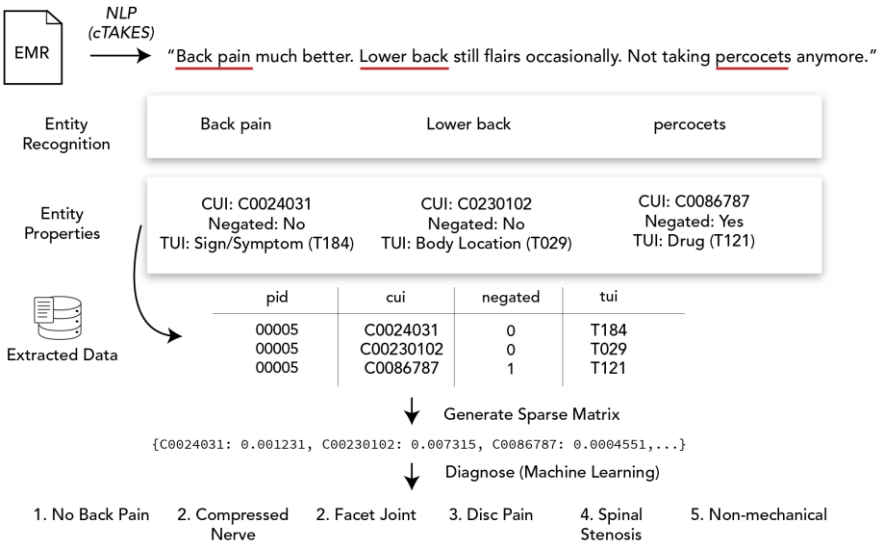

Fig. 1. Brief framework overview, focusing on Apache cTAKES output.

classify brain injuries with 0.98 sensitivity and 0.10 specificity. Although this system has an extremely high sensitivity rate, a DSS must have reasonable specificity since many people without brain injuries could be screened positive.
Combining clinical narratives and structured data from EMRs, including hospital billing codes and international classification of disease codes, also proved to be similarly effective to that of strictly using free-text data. Xu et al. [24] used the private medical NLP system MedLEE to first identify clinical concepts, then used those concepts combined with the hospital codes to encode medical conditions. These conditions were used as features to implement both a heuristic rule-based approach and machine learning approach to diagnose colorectal cancer cases. The Ripper machine learning algorithm with coded data and text data proved to be most effective, with 0.95 sensitivity and 0.96 specificity.

3 Our Framework

3.1 Data

We used data from the OSCAR EMR which included lab results, medical procedures, medications, and diagnoses including free-text data for social history, medical history, and clinical encounters. For this study, we only analyzed the encounter notes which contained seven-years of unstructured narrative text data recorded by the primary care provider during scheduled or walk-in appointments. For validation purposes, thirty-four patients of ages between 18-65 with no history of cancer were hand-selected with a complaint of back pain for greater than twelve months prior to data extraction. We attempted to classify each patient's pattern of low back pain as disc pain, facet joint pain, compressed nerve pain, symptomatic spinal stenosis, or without back pain. The output from the framework was then compared to the gold-standard results obtained from a manual review process conducted by a medical domain expert. The Centre for Effective Practice (CEP) Clinically Organized Relevant Exam (CORE) Back Tool was used as a guide throughout the manual diagnosis process [26].

3.2 Pipeline

The focus of this study was to build a generic clinical notes text mining system to provide physicians with clinical decision support. With that in mind, the success of the project has largely been made possible through the utilization of numerous FOSS projects which are actively being developed in the field of clinical free-text processing. The framework was designed to consist of four main components, which can be extended or replaced with different algorithms. The pipeline consists of:

1. *Anonymization:* The open-source de-identification tool Deid to anonymize PHI [27].
2. *NLP:* cTAKES to add rich semantic and medical concept annotations.
3. *Information extraction:* A custom-built cTAKES output parser to extract clinical concepts and metadata into a standardized SQL table.

4. *Case-detection algorithm:* Multiple supervised classifiers from Scikit-learn to classify the pattern of low back pain.

Anonymization. Anonymization of PHI is integrated in the data processing pipeline using an automated Perl-based de-identification software package for free-text medical records, Deid, that was designed by Neamatullah et al [27]. The software uses lexical look-up tables, regular expressions, and simple heuristics to locate both HIPAA PHI, and additional PHI including common names and date variations. In Neamatullah's study, the performance of the system was evaluated using a gold-standard manually annotated corpus which achieved an overall recall of 0.967 and precision of 0.749 [27].

Deid applies pattern matching to find PHI using numerous configuration parameters that can be set to meet the requirements of a country's regulations. This makes Deid, or any pattern matching for that matter, a strong contender for countries which have strict regulations for PHI. Before Deid is used, the user must manually fill in dictionary files for patient names, doctor's names, nearby hospitals, and places to use for de-identification. This step is required for guaranteed de-identification, however Deid does come with a basic American dictionary to find names and places, and can work without the manually created dictionary files. Once the dictionary files are prepared, a local text file has to be created with each patient's clinical chart notes extracted from the patient database.

NLP. After anonymization, we applied cTAKES in the NLP step to extract relevant medical terminology and relevant negation from the free-text. Fig 1 depicts a simple example of cTAKES called the default clinical pipeline. The pipeline uses the UMLS to annotate text with anatomical sites, signs/symptoms, procedures, diseases/disorders and medications. For every medical entity cTAKES recognizes, it provides a normalized UMLS Unique Concept Identifier (CUI), Type Unique Identifier (TUI), plus values for negation, and a subject. Negation is detected in cTAKES using part of speech tagging, where the program can identify cases such as "not taking Percocets" which will negate the Percocets entity as shown in Fig. 1.

It should be noted that the current case-detection algorithm uses a modified cTAKES pipeline consisting of medical entity recognition and negation, and not the rest of the NLP components. cTAKES provides the ability to not only define a custom pipeline with any of the features within cTAKES but also permits the creation of custom dictionaries to add into cTAKES to create additional domain knowledge.

Information Extraction. The annotations which cTAKES provides contain rich clinical context for each patient. The output from the NLP was parsed using a custom-built entity parser, which iterates through the XML output from cTAKES extracting and inserting all CUI occurrences and their associated negations into a structured data format to be used by the case-detection algorithm. CUIs with negation are recognized as a separate term with a "-" prepended to the CUI. CUIs are unique codes from the UMLS which reference to medical concepts. Each CUI in the UMLS is mapped to numerous medical dictionary concepts in a tree-like data structure and has relationships to other CUIs.

Case Detection Algorithm. The structured data extracted from cTAKES provides a wealth of information to mine and develop a case-detection algorithm. The data extracted presents itself as a bag-of-words model, which contains the multiset of all CUIs within a patient's record. This bag-of-words model in comparison to most text-classification approaches greatly reduces the feature-set size that a machine learning algorithm must account for. The medical term frequency $f_{t,p}$ is computed for each patient's record such that t is the number of times a medical term occurs in patient record p. The term frequency is then normalized by dividing by the sum S of CUIs contained within each patient record as shown in the equation below.

$$f_{t,p} = \frac{\sum_{t' \in p} f_{t',p}}{S} \tag{1}$$

From the calculated medical term frequencies, a sparse matrix was generated for machine learning using two different approaches. The first sparse-matrix created a superset of all CUIs, including every CUI for every patient. This method works well for small datasets but could cause scalability problems for large datasets. Additionally, this method accounts for medical terms that should not normally be considered in the case of a back pain diagnosis.

```
all_patients = {001:[C0000241,C0058291,...],...}
backpain_terms = [lumbar pain, sciatica, ...]
back_cuis = []
sparse_matrix = {}

# get back pain terms from umls api
for term in backpain_terms:
  back_cuis += get_umls_cui(term)

# generate sparse matrix
for id in all_patients:
  sparse_mat[id] = {}
  for cui in back_cuis:
    sparse_matrix[id][cui] =
  all_patients[id].count(cui) / len(all_patients[id])
```

Fig. 2. An overview of the second approach, creating a sparse matrix for each patient's CUIs associated with back pain

Using the manually diagnosed and labelled training data of the patients, four supervised machine learning models were trained using the Scikit-learn library to classify the five patterns of back pain. These four machine learning models included Bernoulli Naïve-Bayes (BernoulliNB), Multinomial Naïve-Bayes (MultinomialNB), Linear Support Vector Classifier (LinearSVC), and Perceptron neural network with stochastic gradient descent classification. To obtain optimal classification accuracy, each model was trained through a fit and score exhaustive search to tune the hyper-parameters of each classifier.

4 Results

From the manual diagnosis, the patient distribution of back pain patterns included twelve patients with disc pain, five with compressed nerve pains, one with facet joint pain, one with spinal stenosis, one with non-mechanical back pain, and fourteen others without any back pain. This variation presents a problem for any supervised machine learning approach. Using the single case patterns as training data would worsen the detection accuracy and using the single cases as test data would always fail. With that in mind, we chose to omit the three single back pain cases from the patient dataset and split the remaining data into 65% training and 35% testing dataset. With this dataset, both sparse matrix generation approaches led to an optimal precision and recall values of 100% for the Linear Support Vector, and slightly lower values for both Naïve-bayes algorithms. The linear perceptron performed rather poorly in all test cases possibly because of the very small size of training data. The results are shown in Fig. 3 where for each model the first column denotes the sensitivity and the second column denotes specificity.

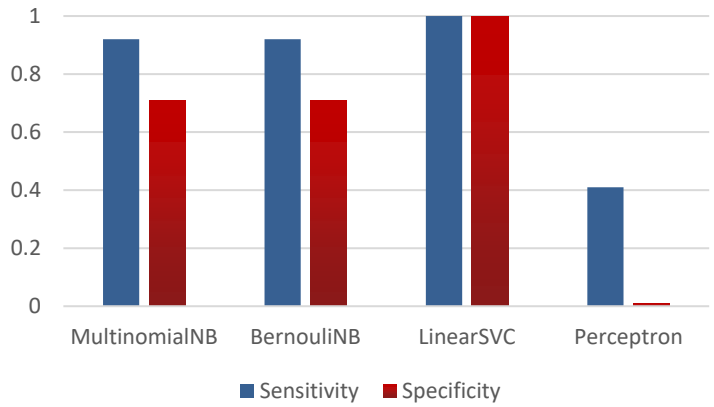

Fig. 3. Sensitivity and specificity for the four compared supervised learning algorithms.

Table 1 shows the performance measured and recorded for each component in the data processing pipeline. The processes were executed on an Intel Core i5-6200U 2.3GHz CPU with 8GB of RAM. For 3.27 Megabytes of plain-text clinical notes, the anonymization and cTAKES processing totalled five hours and ten minutes. For the small dataset used in this study, both approaches to generating sparse matrices had negligible impact on the performance. For both approaches, the case-detection models were by far the most time-efficient parts of the pipeline, with an average total training and testing time of less than 1 second.

Table 1. Running times for the core components of the framework

Framework Component	Computational Time
Anonymization (Read Database, preprocess, anonymize)	13.2 min (~54%)
NLP Processing (cTAKES)	3.4 min (~14%)
Information Extraction	7.6 min (~31%)
Case-Detection Algorithms (testing + training)	<1 sec (<1%)

4.1 Discussion

For this pilot NLP study on low back pain we had a limited set of labelled data. Therefore, these results are not a clear indicator of success. With the dataset, machine learning algorithms can be over or under trained, and generally modelling becomes inherently difficult due to the small sample size. The study provided valuable insights about the various processing steps in the pipeline, their relative performance with respect to time, existing FOSS tools and their performances. We are currently working on a larger, 1200 patient dataset for further evaluation.

As shown in Table 1, the computational performance of some of the system components such as Deid and cTAKES were underwhelming. However, the machine used for testing was not designed for powerful computation. We are currently exploring Apache Tika which supports integration with Apache cTAKES, and QuickUMLS as alternative tools to improve the performance of the NLP step. None of the system components applied parallel processing or even multi-core processing. The utilization of these methods would also greatly decrease the execution time of the complete analytic pipeline.

5 Conclusion and Future Work

Chronic low back pain deteriorates the quality of life. Medical imaging is an expensive procedure which can be effectively prescribed by physicians if the type of pain can be better diagnosed with the help of a medical DSS. Diseases such as low back pains and depression are difficult to diagnose from regular health metrics. Recent studies are exploring NLP and machine learning techniques to extract relevant diagnostic information from doctors' free text encounter notes data in the EMRs. In this paper we present our study on existing FOSS tools and algorithms, and the prototype of a simple medical DSS. The DSS can classify patients with low back pain based on unstructured text encounter notes data from an EMR system. We implemented a data processing pipeline utilizing FOSS anonymization algorithms, clinical NLP tools, and machine learning models to diagnose disc pain or compressed nerve pain using a limited number of patient data and achieved a 100% accuracy. The results also indicate that a generalized framework can be used to diagnose other medical conditions.

Since this is a preliminary study, there are numerous areas for improvement. Our ongoing work focuses on processing a larger dataset containing a greater number of

patients' data. The current framework has been designed with loosely integrated components such that each component can be replaced with a better option as one becomes available, or extended by contributing to the open-source projects.

The anonymization algorithm worked well for a project of this size, but the run times were too long to be used with larger datasets. Rudolf's [29] Clinical Records Anonymization and Text Extraction System showed massive improvements in anonymization performance in comparison to the current algorithm, which would add significant value to the current framework. It uses regular expressions as well, but runs at 14Mb/s.

Additionally, the data extracted from cTAKES for case-detection is also currently underutilized. cTAKES and information extraction together takes almost half of the total processing time as shown in Table 1 but only a fraction of the extracted information is used in the later part of the pipeline for diagnosis. The long processing time will cause scalability issues for larger datasets. Other state of the art NLP tools such as IBM's Watson Health, Apache Tika with cTAKES, spaCy, and QuickUMLS can be explored to address the above issues. Big data processing frameworks and parallelization of the data processing pipeline can also improve the processing time.

Finally, the current case-detection algorithm is relatively simple. Using the NLP data which cTAKES provides in combination with the structured data within the EMR such as age, weight and lab reports, and developing a more complex model such as the long short-term memory Recurrent neural network model to take into account the patient encounter times will allow creating a more robust DSS. Diagnosis of diseases is often not very straight forward. Having a confidence interval for each classification and using an ensemble learning model will increase users' trust on the system and help build a reliable DSS that can be trusted by medical practitioners.

6 References

1. Soldaini, L.,Goharian, N.: QuickUMLS: a fast, unsupervised approach for medical concept extraction (2016).
2. L. J. & Brunak, S.: Mining electronic health records: towards better research applications and clinical care. Nature Reviews Genetics (2012).
3. Rau, J.: Medicare's Readmission Penalties Hit New High, https://khn.org/news/more-than-half-of-hospitals-to-be-penalized-for-excess-readmissions/view/republish/, last accessed 2018/01/15.
4. Hassanzadeh, H., Nguyen, A. & Koopman, B.: Evaluation of Medical Concept Annotation Systems. Proceedings of Australasian Language Technology Association Workshop, pp. 15-24 (2016).
5. Buckley J, et al.: The feasibility of using natural language processing to extract clinical information from breast pathology reports. Journal of Pathology Informatics, 3(23) (2012).
6. Spasić, I., Livsey, J., Keane, J. & Nenadić, G.: Text mining of cancer-related information: Review of current status and future directions. International Journal of Medical Informatics, pp. 605-623 (2014).
7. Ford, E. et al.: Extracting information from the text of electronic medical records to improve case detection: a systematic review. Journal of the American Medical Informatics Association, 23(5), pp. 1007-1015 (2016.).
8. IBM: IBM Watson Health, https://www.ibm.com/watson/health, last accessed 2018/01/3

9. Sevenster, M., Ommering, R.V., Qjan, Y.: Bridging the Text-Image Gap: A Decision Support Tool for Real-Time PACS Browsing. Journal of Digital Imaging, vol. 25, pp. 227–239 (2012).
10. Jensen, K. et al.: Analysis of free text in electronic health records for identification of cancer patient trajectories. Scientific Reports, vol. 7 (2017).
11. OSCAR Canada: About OSCAR, http://oscarcanada.org/about-oscar/brief-overview, last accessed 2017/10/28.
12. Bone and Joint Canada: Low Back Pain, http://boneandjointcanada.com/low-back-pain/, last accessed 2018/01/10.
13. Canadian College of Family Physicians of Canada: Evidence-informed primary care management of low back pain, http://www.cfpc.ca/uploadedFiles/Directories/Committees_List/Low_Back_Pain_Guidelines_Oct19.pdf, last accessed 2018/01/11
14. Webster, B., Courtney, T., Huang, Y.H., Christiani, D.: Physicians' Initial Management of Acute Low Back Pain Versus Evidence-Based Guidelines. Journal of General Internal Medicine, vol 20., pp.1132–1135 (2005).
15. Devereaux, M.: Low back pain. Primary Care: Clinics in Office Practice, vol. 31, pp. 33–51 (2004).
16. Savova, G.K., et al.: Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association pp. 507–513 (2010).
17. Spasić, I., Livsey, J., Keane, J. & Nenadić, G.: Text mining of cancer-related information: Review of current status and future directions. International Journal of Medical Informatics, pp. 605-623 (2014).
18. Szlosek, D. A. & Ferrett, J.: Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems. Generating Evidence & Methods to improve patient outcomes, 4(3), pp. 1222 (2016).
19. Ferrández, O., South, B.R., Shen, S., et al.: Evaluating current automatic de-identification methods with Veteran's health administration clinical documents. BMC Medical Research Methodology (2012).
20. Meystre, S., Savova, G., Kipper-Schuler, K. & Hurdle, J. F.: Extracting information from textual documents in the electronic health record: a review of recent research. Yearbook of Medical Informatics, pp. 128-144 (2008).
21. Moja, L. et al.: Effectiveness of Computerized Decision Support Systems Linked to Electronic Health Records: A Systematic Review and Meta-Analysis. Am J Public Health, pp. 104-116 (2014).
22. Okazaki, N. & Tsujii, J.: Simple and efficient algorithm for approximate dictionary matching. Proceedings of the 23rd International Conference on Computational Linguistics, pp. 851-859 (2010).
23. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research, vol. 32, pp. D267-D270 (2004).
24. Xu, H., Fu, Z., Chen, Y., et al.: Extracting and Integrating Data from Entire Electronic Health Records for Detecting Colorectal Cancer Cases. AMIA Annual Symposium Proceedings, pp. 1564–1572 (2011).
25. D'Avolio, L.W. et al.: Evaluation of a generalizable approach to clinical information retrieval using the automated retrieval console (ARC). Journal of American Medical Informatics Association, 17(4), pp. 375-382 (2010).

26. Centre for Effective Practice: Clinically Organized Relevant Exam, http://www.cfpc.ca/uploadedFiles/Resources/Resource_Items/Health_Professionals/CEP_CoreBackTool_2016.pdf, last accessed 2017/08/20.
27. Neamatullah, I. et al.: Automated de-identification of free-text medical records. BMC Medical Informatics and Decision Making, pp. 8-32 (2008).
28. Finan, S. & Masanz, J.: Default Clinical Pipeline, https://cwiki.apache.org/confluence/display/CTAKES/Default+Clinical+Pipeline, last accessed 2017/09/02
29. Rudolf, C. N.: Clinical records anonymisation and text extraction (CRATE): an open-source software system. BMC Medical Informatics and Decision Making (2017).