

Triggering Clinical Decision Support Systems Alerts at the Point of Care

Gitanjali.K, S. Thenmalar

Abstract: Clinical decision support system (CDSS) is a software, that assists physicians with clinical decision support (CDS). The CDS alerts can be designed as rules using various health information technology codes and computerised guidelines. Many CDS systems do not abide by the FHIR structure or work with an older version of FHIR. Also, most CDS systems do not have a module that is able to extract information from free text areas (like patient notes). We propose a CDSS alerts system that adheres to the FHIR standards, and use three techniques (if-then conditions, SVM, and Logistic Regression) to extract free text from the patient notes and compare them to see which performs better. Amongst the three methods, Logistic Regression performs the best with an accuracy using cross validation method was 0.97.

Index Terms: Clinical Decision Support System, Clinical Decision System, FHIR

I. INTRODUCTION

Clinical Decision Support Systems (CDSS) can be used to help the physicians in making clinical decisions. They have been used for a variety of reasons such as creating preventive alerts and notifications, generating templates for forms such as consent forms, death certificates, prescription forms etc. [1,2]. They are often integrated with Electronic Health Records (EHRs) to reduce errors and delays to the workflow.

CDSS can be used in helping patient-related decision making, determining specific optimal treatment strategies, and aiding in the outlining of health policies by assessing and predicting the clinical and economic outcomes of different treatment methods and plans when randomized trials are not possible or cannot be done practically.

There are however a lot of problems in using CDSS effectively. Some general problems include alert fatigue, that is, notifying too many alerts to the physicians such that they get desensitized to the alerts. Another problem is the complexity of a human body and how it cannot be easily translated into machine understandable code.

In this paper we propose a CDSS that adheres to the FHIR (Fast Healthcare Interoperability Resources), standards and uses NLP techniques to extract free text from patient notes.

Revised Manuscript Received on June 05, 2019

Gitanjali. K, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.

Dr. S. Thenmalar, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.

II. BACKGROUND

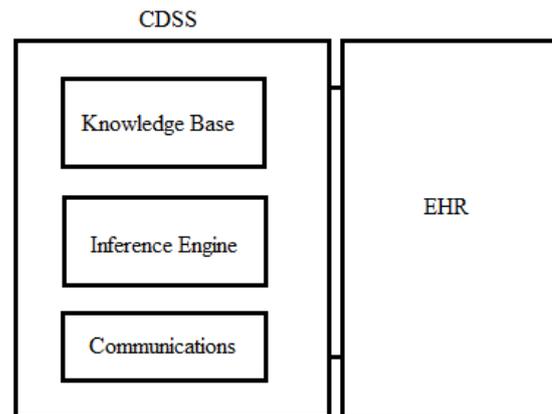


Fig.1 CDSS General Architecture integrated with an EHR

A. General CDSS Architecture

Figure 1 shows that CDSS generally have three main components: knowledge base, inference engine and communications [1,2]. The knowledge base stores information of the rules and guidelines with respect to the domain. The rules and guidelines are stored in accessible formats and are available to the inference engine. These guidelines maybe created by domain experts, or the data might be taken using machine learning techniques. The inference engine uses the information given by the knowledge base and implements the logic (i.e., the rules regarding the data). They can determine whether some action needs to be done (for example, alerting a physician to remind a patient to take a particular test). Inference engines have three main instructions to follow; match, select and execute rules. One common implementation of inference engines focus on forward chaining, which is working with the logic sequence, from the starting condition and coming to the end in that order. The communications module is used to display alerts to the user. It can be in the form of relaying information from the knowledge base to the inference engine and vice versa. They are often connected with an Electronic Health Records (EHR).

III. RELATED WORKS

For the research of this system, many papers were referred and their various models, techniques and datasets were identified and noted.

Based on models, Aihua Fan et al. [7] while building their CDSS, in the application



layer used the CLP-CPG Model (Constraint Logic Program Model). Tanvir Ahamed et al. [3] developed a model consisting of a three-stage approach that creates and assesses a meta-model that can manage the requirements of guideline-based clinical nursing-specific CDS systems. Anna M Lackerbauer et al. [14] chose the standard for the eConsent architecture as HL7 FHIR so as to maintain compatibility with the most current technology. Peng Zhang et al. [21], Suranga N. Kasthurirathne et al. [13], Crescenzo Diomaiuta et al. [6], Ryan A. Hoffman et al. [10], and Na Hong et al. [11] used FHIR as a means of structuring data elements (known as "resources") and also as an application programming interface (API) for transferring electronic health records. Based on techniques, Na Hong et al. [11] implemented a clinical NLP pipeline with an FHIR-based type system and performed a case study using medication data. Livvi Li Wei Sim et al. [18] used MySQL, HTML, CSS and Bootstrap were used to develop the web-based interface of the DiabetesDashboard. Henry Odikwa et al. [9] adopted the min-max method for normalizing the datasets. They also used genetic algorithm was used in initializing the training weights of the Multilayer Perceptron. K.E. Ravikumar et al. [17] used SQL queries, and a simple rule-based approach NLP algorithm in their CDSS. Sudhir Anakal et al. [4] used Multilayer Perceptron Neural Network using back propagation algorithm to build their Clinical Decision Support System for Chronic Obstructive Pulmonary Disease. Donald A Szlosek et al. [20] built CDSSs integrated into EMRs and can be assessed automatically by using NLP and machine learning techniques (SVC, Decision Tree Classifier; K-Neighbors Classifier). Yicheng Jiang et al. [22] developed a three-layer model for their knowledge base using a multi-symptom Naive Bayes algorithm, the three layers being disease-symptom-property. Yung-Fu Chen et al. [5] designed their CDSS with an integrated genetic algorithm (GA) and support vector machine (SVM). Dao Thi Anh Nguyen et al. [16] used Fuzzy Logic Inference System for their CDSS (which was knowledge based system). Felix Gräßer et al. [8] used Collaborative Demographic-based Recommenders to build their Therapy Decision Support System. Based on datasets, some used real-world data, while others had to manually create datasets to their preferences. Henry Odikwa et al. [9], K.E. Ravikumar et al. [17], Lars O. Karlsson et al. [12], Shervin Malmasi et al. [15], Na Hong et al. [11], Ryan Hoffman et al. [10], Shyamashree Sinha et al. [19], and Yung-Fu Chen et al. [5] used real-world data, taken from patient databases present in clinics. Donald A. Szlosek et al. [20] created dummy datasets which were generated with the help of clinicians and medical students so as not to disturb real world patient data.

IV. PROPOSED SYSTEM

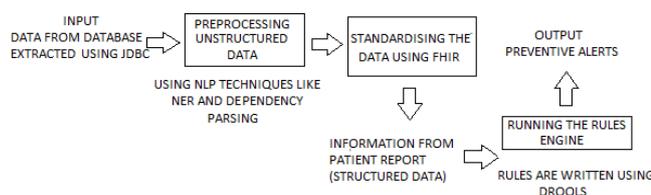


Fig. 2 Proposed CDSS

Figure 2 shows that the input of the proposed system consists of data extracted from the GlanceEMR database. The first part consists of extracting unstructured data using NLP techniques. The already structured data is mapped to the FHIR format. This is fed to the rules engine which checks all the conditions to see if any patient meets them. If any conditions are met, they create user alerts and will appear once the user clicks on the alerts icon.

A. Input Data From Database

The input to this system consists of patient data such as demographic history, investigations, diagnosis, medications, etc. which is stored directly in the database.

Data in database will be of two forms: structured and unstructured data. The structured data can be extracted by querying from the database by connecting to it. This involves writing a program using JDBC (Java Database Connectivity) to connect with the database, and reading the information from the relevant table.

B. Preprocessing Unstructured Data

Unstructured data will be present in text areas (in the patient notes). To extract this data, we can first read the data and perform pre-processing techniques such as stemming and parts-of-speech tagging. The processed data is then subjected to a named entity recognizer that recognizes if a noun is a finding, procedure, diagnosis, or medication. We then confirm that the patient does have these entities by performing dependency parsing and checking if the entity is affirmed or not (i.e. if the patient has them or not). For example, if the statement is "Patient does not complain of chest pain but confirms headaches", then through dependency parsing, we should find that the term *chest pain* is negated and that headache is affirmed and thus, should be extracted. The extracted data is then mapped to terminologies and classifications such as SNOMED CT and ICD-10. After mapping is done, it is stored in the database. This part of the system is not yet integrated into the current model.

C. Standardizing The Data Using FHIR

The data thus taken is mapped to data standards which are the rules by which data is described and recorded. Data must be standardized in order to be able to share, and to understand data better. This is achieved by the use of FHIR. It works by presenting the data elements as services via their own resource URLs. For example, the basic elements of healthcare like patient demographics, medications, procedures etc., can be retrieved using their specific URLs. The data, in the patient database, which is present in the database, is queried and then sent as JSON files which are present in the FHIR structure. These JSON files in FHIR structure are converted to FHIR objects. These are then used to find if the conditions set by the manually authored rules are fulfilled.

D. Authoring The Rules

The rules are written using a Business Rules Management System (BRMS), in this case Drools. A business rule is a statement that describes a business policy or procedure. Rule engines have Business Rule Repository, a database for storing the business rules as defined by the business users. Rules are authored in the following way:

rule "Name"
when



```
<some conditions>
then
  <take some action>
end
```

	Months to 5 Years, Screening	years
--	---------------------------------	-------

Table 1 Rules written

E. Rules Written

The following table gives the rules written and the parameters for each rule.

S.No	Rules	Rule Parameters
1	Abdominal Aortic Aneurysm Screening	Male patient Patient with smoking history Patient aged 65 to 75 years
2	Alcohol Unhealthy Use in Adolescents and Adults, Screening and Behavioral Counseling Interventions	Adults 18 years or older (not including pregnant women)
3	Asymptomatic Bacteriuria, Screening	Pregnant Women
4	Breastfeeding, Primary Care Interventions	Pregnant women, new mothers, and their children
5	Cardiovascular Disease Screening	Patients above 65 years
6	Folic Acid for all Women who are capable or are planning for pregnancy to prevent Neural Tube Defect	All Adult Women
7	Glaucoma, Screening	Patient aged 18 years Patient diagnosed with Vision symptom Glaucoma screening needs to be done atleast once for the patient
8	Chlamydia and Gonorrhea, Screening	Sexually active women
9	Hepatitis C Screening for adults	Adult patient (age > 18)
10	HIV, Screening	Adolescents and Adults: ages 15 to 65 years
11	Osteoporosis to Prevent Fractures, Screening	Screening for women less than 65 years and Postmenopausal
12	Osteoporosis to Prevent Fractures, Screening	Postmenopausal and women 65 and Older
13	Preeclampsia, Screening	For pregnant women
14	Prostate Cancer, Screening	Men, aged 50 and older
15	Rh (D) Incompatibility, Screening	Female, pregnant, only first visit
16	Skin Cancer Prevention	For persons aged : 6 months to 24 years
17	Syphilis Infection in Pregnant Women, Screening	For all pregnant women
18	Syphilis Infection in Non-pregnant Adults and Adolescents, Screening	For non pregnant women and people from the age of 12
19	Vision in Children Ages 6	Patient aged 3 to 5

The following is an example of a rule. We can use the example of Asymptomatic Bacteriuria Screening. For this alert to take place, the patient must be a pregnant woman. The name of the rule would be 'Asymptomatic Bacteriuria Screening' after which the conditions would be specified. For example (in pseudocode)

```
rule "Asymptomatic Bacteriuria Screening"
where
  <find if patient is female by accessing FHIR object >
  <find if patient is pregnant by accessing FHIR object >
then
  <set the rule status to true>
  <set rule id > //for database purpose
  <set grade> //grade : A
  <set message to be displayed once condition is satisfied>
  //"Patient is due for Asymptomatic Bacteriuria Screening
end
```

In this way, most of the rules are written. While writing these rules, we must constantly check the FHIR objects and refer to their structure. For example, to access a patient's age, we need to go to the FHIR object 'Patient' and access inside that the 'Patient Demographics', from which we can finally get the patient's age.

When the rules are run, if any conditions are met, a response in the form of a JSON form is sent to the User Interface, where the alert is shown.

F. Tools Used in the System

a) FHIR

Fast Healthcare Interoperability Resources (FHIR) is a draft standard when it comes to structuring and transferring clinical data. This standardization aids in improving the interoperability between the various systems CDSSs have to communicate. It uses base standards like JSON and XML. It is built from previous format standards from HL7 (Health Level 7), but has the advantage of using several updated web-technologies like RESTful protocol, HTML and CSS.

b) DROOLS

Drools is a Business Rules Management System (BRMS) tool. BRMSs are software systems that determine, employ, run, check-up on, and manage the pre-defined decision logic used by organizations or enterprises. In this project, we use Drools to create rules (i.e. the logic of the CDS).

c) spaCy

spaCy is a Python library specialising in implementing NLP techniques like training a custom Named Entity Recognition module and also performing dependency parsing. We use this tool to train a model to recognise medical entities and also to find the relationship between the entities and other parts of the sentence.



V. RESULTS AND DISCUSSION

The physician while accessing a patient chart, can click on the Alerts icon to see if there are any procedures due, or if a patient is due for some clinical test. The evaluation of the CDS system is usually measured by finding user satisfaction (i.e. by finding if the users found the system to be accurate or not) or by comparing the system's results to the expected results already using a dataset created for this purpose. User satisfaction and comparison to a proper dataset will be looked into in the future work.

The extraction of unstructured data from free text boxes is not yet integrated into the system. We have however implemented three different methods in order to find which one is more viable. We used if-else statements, and machine learning techniques such as Support Vector Machine and Logistic Regression. To find the accuracy, we use a manually created dataset, which consists of 500 rows and has four columns that are used in prediction (type of entity, if negative word is before, after or missing, position of entity, position of negative cue word).

The if-else conditions, are implemented as follows. The entity is recognised and its relationship to a negative cue word (which is present in a dataset named 'negative cue words' like 'not', 'does not suggest', etc.) is also found using spaCy. The decision of whether an entity is negated or not depends on if the entity is linked to the negative word, and if it has the co-ordinating conjunction 'and' the sentence is split, such that each entity can be evaluated separately. After the module has identified if the entity is negated or not, the output is printed. After running this module for 500 entity-containing statements, and checking if the entities are negated, we found that 450 entities were correctly identified. 112 entities were classified as negated by the if-then statements while originally there were only 106 entities negated. Thus its accuracy was 90% for this dataset.

We also tried machine learning algorithms such as Support Vector Machine and Logistic Regression to classify entities in sentences into negated and not negated. While applying SVM to the dataset that was created for this purpose, the cross-validation accuracy was 0.956, and the testing accuracy was 0.82. This was while using validation set approach, while leaving out 10% of the last part of the data as the test data. While using the k-fold cross validation repeated five times for $k = 5$, the average training accuracy was 0.902. While applying logistic regression, while using cross validation method, the cross validation accuracy came to 0.97 while the testing accuracy came to 1.0. While using k-fold cross

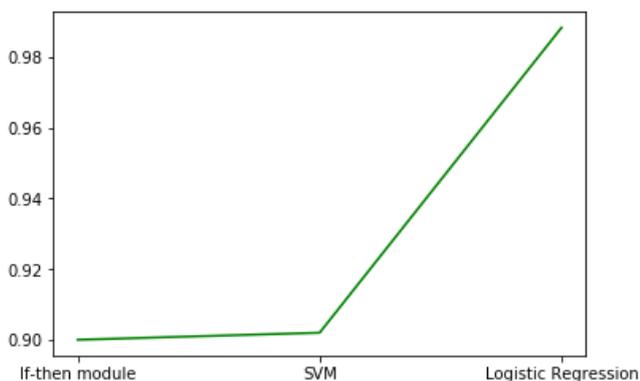


Fig. 3 Graph comparing the accuracies of the three methods used

validation repeated five times for $k = 5$, the average training accuracy was 0.9884.

Out of the three methods, Logistic Regression, being the more powerful machine learning algorithm of the two chosen, had the highest accuracy, while the if-else method had the least.

The graph (Fig. 3) shows the accuracy with the three methods. Table 2 shows the accuracy while finding if the sentence entity is negated or not.

Method	Accuracy
If-Then conditions	0.9
SVM (Cross-validation accuracy)	0.956
Linear Regression (Cross validation accuracy)	0.97
Linear Regression (K-Cross validation accuracy)	0.9884

Table 2 : Results

VI. LIMITATIONS

For all three methods, there are various limitations present. The module which classified the sentences based on if-else conditions, recognising entities and dependency parsing is limited and can only classify certain structures of sentences. For instance it can only recognise single sentences. Sentences come in a variety of lengths and grammatical structures. Also in real time, the data we will be extracting will have a large number of grammatical errors. The most significant of which would be the general lack of proper punctuation. spaCy's dependency parser depends on sentences following proper grammatical conventions at least to an extent and will not be able to correctly predict relationships if it cannot identify the end of one sentence and the beginning of the next.

Another important drawback is in the dataset used itself. It is limited to 500 rows, has only four parameters (type of entity, if entity is before/after the entity or completely absent, position of entity, and position of negative cue word) and each of the sentences provided is a single sentence. While running machine learning algorithms, the bigger the data set the better. Therefore smaller datasets may tend to over-fit models.

VII. CONCLUSION

In this paper, we have thus proposed a CDS system by building a rule engine using Drools tool and by using the FHIR standard to structure the data between different modules. We tried many NLP techniques to extract medical entities from the sample sentences. We created a dataset and applied machine learning algorithms to classify the sentences on whether the entities are negated or not. We found that among the three methods we used to classify the data, we found logistic regression to perform better amidst restraints present in the dataset. In the future we plan to increase the volume of the dataset to get more accurate view of the classifications thus performed and also to increase the number of parameters of the dataset. Glenwood Systems will be evaluating the CDSS based on user satisfaction and decision correctness among other methods.

REFERENCES



1. 'Understanding the Basics of Clinical Decision Support Systems' <https://healthitanalytics.com/features/understanding-the-basics-of-clinical-decision-support-systems>
2. 'CDS System Architecture' <https://confluence.ihtsdotools.org/display/DOCCDS/2.2.+CDS+System+Architecture>
3. T. Ahamed, R. Lederman, R. Bosua, K. Verspoor, W. Buntine, and G. Hart (2016). Towards a Methodology for Nursing-Specific Clinical Decision Support Systems (CDSS). *Journal of Decision Systems*, 25(sup1), pp.23-34.
4. Sudhir Anakal and Sandhya, P. (2017). Clinical decision support system for chronic obstructive pulmonary disease using machine learning techniques. 1-5. 10.1109/ICEECCOT.2017.8284601.
5. Y. Chen, C. Lin, K. Wang, L. Rahman, D. Lee, W. Chung, and H. Lin (2018). Design of a Clinical Decision Support System for Fracture Prediction Using Imbalanced Dataset. *Journal of Healthcare Engineering*, 2018, pp.1-14.
6. C. Diomaiuta, M. Sicuranza, M. Ciampi and G. De Pietro (2017). A FHIR-based System for the Generation and Retrieval of Clinical Documents. In *Proceedings of the 3rd International Conference on Information and Communication Technologies for Ageing Well and e-Health - Volume 1: ICT4AWE*, ISBN 978-989-758-251-6, pages 135-142. DOI: 10.5220/0006311301350142
7. A. Fan, D. Lin, and Y. Tang, (2017). Clinical Decision Support Systems for Comorbidity: Architecture, Algorithms, and Applications. *International Journal of Telemedicine and Applications*, 2017, pp.1-10.
8. F. Gräber, S. Beckert, D. Küster, J. Schmitt, S. Abraham, H. Malberg, and S. Zaunseder (2017). Therapy Decision Support Based on Recommender System Methods. *Journal of Healthcare Engineering*, 2017, pp.1-11.
9. H. Odikwa, U. Chidiebere, and I. Hycinth (2017). An Improved Model for Clinical Decision Support System. *International Journal of Artificial Intelligence & Applications*, 8(6), pp.37-55.
10. R. Hoffman, H. Wu, J. Venugopalan, P. Braun, and M. Wang (2018). Intelligent Mortality Reporting With FHIR. *IEEE Journal of Biomedical and Health Informatics*, 22(5), pp.1583-1588.
11. N. Hong, A. Wen, F. Shen, S. Sohn, S. Liu, H. Liu, and G. Jiang (2018). Integrating Structured and Unstructured EHR Data Using an FHIR-based Type System: A Case Study with Medication Data. *AMIA Joint Summits on Translational Science proceedings*. AMIA Joint Summits on Translational Science, 2017, 74-84.
12. L. Karlsson, S. Nilsson, M. Bång, L. Nilsson, E. Charitakis, and M. Janzon, (2018). A clinical decision support tool for improving adherence to guidelines on anticoagulant therapy in patients with atrial fibrillation at risk of stroke: A cluster-randomized trial in a Swedish primary care setting (the CDS-AF study). *PLOS Medicine*, 15(3), p.e1002528.
13. SN Kasthurirathne, B. Mamlin, G. Grieve, P. Biondich (2015) Towards standardized patient data exchange: integrating a FHIR based API for the open medical record system. *Stud Health Technol Inform.* 2015;216:932
14. A. Lackerbauer, A. Lin, O. Krauss, J. Hearn and E. Helm (2018). A Model for Implementing an Interoperable Electronic Consent Form for Medical Treatment Using HL7 FHIR. *European Journal for Biomedical Informatics*, 14(3), pp.37-47.
15. S. Malmasi, N. Hosomura, L. S. Chang, C. J. Brown, S. Skentzos, & A. Turchin (2018). Extracting Healthcare Quality Information from Unstructured Data. *AMIA Annual Symposium proceedings*. AMIA Symposium, 2017, 1243-1252.
16. D. Nguyen, I. Won, K. Kim, and J. Kwon (2018). A Development of Clinical Decision Support System for Video Head Impulse Test Based on Fuzzy Inference System. *Journal of Sensors*, 2018, pp.1-10.
17. K.E. Ravikumar, K. MacLaughlin, M. Scheitel, M. Kessler, K. Waghlikar, H. Liu, and R. Chaudhry (2018). Improving the Accuracy of a Clinical Decision Support System for Cervical Cancer Screening and Surveillance. *Applied Clinical Informatics*, 09(01), pp.062-071
18. L. Sim, K. Ban, T. Tan, S. Sethi, and T. Loh (2017). Development of a clinical decision support system for diabetes care: A pilot study. *PLOS ONE*, 12(2), p.e0173021.
19. S. Sinha, M. Jensen, S. Mullin, and P. Elkin (2017). Safe Opioid Prescribing: A SMART on FHIR Approach to Clinical Decision Support. *Online Journal of Public Health Informatics*, 9(2).
20. D. Szlosek, and J. Ferretti (2016). Using Machine Learning and Natural Language Processing Algorithms to Automate the Evaluation of Clinical Decision Support in Electronic Medical Record Systems. *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, 4(3), p.5.
21. P. Zhang, J. White, D. Schmidt, G. Lenz and S. Rosenbloom (2018). FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data. *Computational and Structural Biotechnology Journal*, 16, pp.267-278.
22. Y. Jiang, B. Qiu, C. Xu, and C. Li (2017). The Research of Clinical Decision Support System Based on Three-Layer Knowledge Base Model. *Journal of Healthcare Engineering*, 2017, pp.1-8.