pHealth 2024
M. Giacomini et al. (Eds.)

doi:10.3233/SHTI240065

Development of a Method for Automatic Matching of Unstructured Medical Data to ICD-10 Codes

Bogdan Volkov[a] and Georgy Kopanitsa[b, 1]
[a] *ITMO University, Saint-Petersburg, Russia*
[b] *Almazov National Medical Research Centre, Saint-Petersburg, Russia*

Abstract. Inconsistent disease coding standards in medicine create hurdles in data exchange and analysis. This paper proposes a machine learning system to address this challenge. The system automatically matches unstructured medical text (doctor notes, complaints) to ICD-10 codes. It leverages a unique architecture featuring a training layer for model development and a knowledge base that captures relationships between symptoms and diseases. Experiments using data from a large medical research center demonstrated the system's effectiveness in disease classification prediction. Logistic regression emerged as the optimal model due to its superior processing speed, achieving an accuracy of 81.07% with acceptable error rates during high-load testing. This approach offers a promising solution to improve healthcare informatics by overcoming coding standard incompatibility and automating code prediction from unstructured medical text.

Keywords. EHR, ICD-10, SNOMED, graph database.

1. Introduction

Within the expansive realm of medicine, the abundance and heterogeneity of data pose formidable challenges. While the adoption of Electronic Health Record (EHR) systems has ameliorated issues related to data storage adherence to specified standards [1], a persistent challenge lies in the incomplete compatibility of these standards [2]. Standards, including HL7 (v2, v3, FHIR), openEHR, and ISO 13606, exhibit distinct characteristics and requirements, contributing to difficulties in selection for specific projects or applications. This diversity impedes comparisons and integration efforts, potentially leading to information loss, data distortion, or heightened system loads when exchanging information across disparate standards.

Machine learning methods have emerged as highly effective tools for such modeling, offering the capacity to utilize implicit datasets and discern underlying relationships. In the realm of medical research, these methods facilitate the simulation of disease progression based on available patient data and conditions [3].

Currently, several studies focus on ensuring compatibility in disease classification. One study explores 4 methods: ICD-10-CM → SNOMED CT → ICD-11, ICD-10-CM → ICD-10 → ICD-11, and additionally explores the intersection and unification of

[1] Corresponding Author: Georgy Kopanitsa, Almazov National Medical Research Center, Saint-Petersburg, Russia, georgy.kopanitsa@gmail.com

methods 1 and 2 [4]. For instance, in method 1, errors occur during translation from ICD-10-CM to SNOMED, while in method 2, issues arise from incomplete coverage and accuracy of the WHO map from ICD-10-CM to ICD-11. Additionally, studies focus on translating Read codes [5]. The examination revealed that achieving a lossless correspondence between Read codes and the ICD10-CM disease coding system is unattainable due to disparities in ontologies and insurmountable ambiguity in the utilization of the Read code system by medical professionals [5].

Thus, it makes sense not to make a direct comparison of one code to another, for example, by compiling a reference book, but to consider the option of predicting the code of the requested classification based on a doctor's opinion, observations, complaints, symptoms.

The goal of this paper is to develop a system to automatically match unstructured medical texts to the ICD-10 codes.

2. Methods

2.1. System's architecture

The proposed methodology features a unique architectural design, emphasizing semantic interoperability in medical data to enable effective information exchange within healthcare systems. It comprises key components such as a disease classification graph for data collection, a training layer for experience accumulation, and model outputs for subsequent classifications.

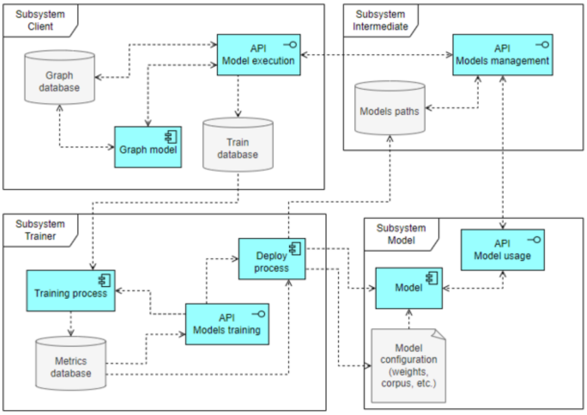

Figure 1. Architecture diagram of the system.

The resulting architecture consists of four main subsystems: client, intermediate, training, and model. Each subsystem is specialized for distinct functions. The "Client" subsystem manages input data reception, processing, and accumulation. The "Intermediate" subsystem facilitates communication between the client layer and the models.

2.2. Training layer

The training layer offers automatic model training and implementation, utilizing training data from the client layer. The "Trainer" subsystem automates model generation and training using accumulated client data. The learning process involves creating, training, and recording metrics and model configurations into the Metrics database, enabling future model selection and configuration loading without retraining. The "Deploy process" component retrieves saved model configurations to create the final model with its API service. To ensure usability, the model path must be added or updated in the directory containing model paths in the intermediate layer, ensuring operational readiness upon launch.

2.3. Knowledge base component

This component accumulates system data, performs two tasks: builds a knowledge base graph from initial component data, and predicts diagnoses and symptoms. For clarity, let's illustrate its function with an example. During the initial launch of the subsystem, each diagnosis is saved from the received set. If the diagnosis already exists in the knowledge base graph, it's not added again; otherwise, the "diagnosis" node is added. Similarly, each symptom from the received set is saved. If it already exists in the graph, it's not added again; otherwise, the "symptom" node is added. Then, each symptom is connected with every "diagnosis" node from the incoming set of diagnoses.

Based on analysis results, an associative graph is created where vertices represent symptoms and diseases, and edges signify associations between them. Graph theory algorithms and visualization methods are employed for presenting the results conveniently.

2.4. Modeling experiments

The training layer, a pivotal subsystem in the machine learning model architecture, is integral to the learning process using data from the Almazov National Medical Research Centre. The dataset comprises 83,000 medical records, including doctors' conclusions and patient complaints. This layer serves as the foundation for model adaptation to the provided data and influences its ability to generalize to new input data. At the moment, only accuracy (1), precision (2), and recall (3) metrics are measured within the process for each trained model.

$$accuracy = \frac{true\ negative + true\ positive}{true\ positive + true\ negative + false\ positive + false\ negative} \tag{1}$$

$$precision = \frac{true\ positive}{true\ positive + true\ negative + false\ positive + false\ negative} \tag{2}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{3}$$

3. Results

3.1. *Modeling results*

The Random Forest model achieved optimal accuracy, as observed in the test data (Table 1). However, due to limited computational resources and the need for prompt responses in our task, the Logistic Regression model was selected. Logistic Regression demonstrated favorable throughput characteristics, supported by the acquired metrics, making it a suitable alternative. Comparative throughput results for these two options are detailed in the subsequent subsection on load testing outcomes. The resulting metrics are based on data from the Almazov National Medical Research Centre.

Table 1. Summary obtained ICD-10 models metrics on test dataset

Options	Accuracy	Precision	Recal
Decision Tree	0,8569	0,8394	0,8367
Random Forest	0,8728	0,8566	0,8558
Logistic Regression	0,8107	0,7934	0,79
SVM	0,8607	0,8435	0,8436

3.2. *Knowledge base component*

The resulting associative graph displays the complex relationships between symptoms and diseases identified from the analysis of medical data. In the graph, you can identify clusters of symptoms associated with certain groups of diseases, as well as identify the relationship between individual symptoms and specific pathologies. During the testing procedure, outcomes for three distinct components were garnered (figure 2).

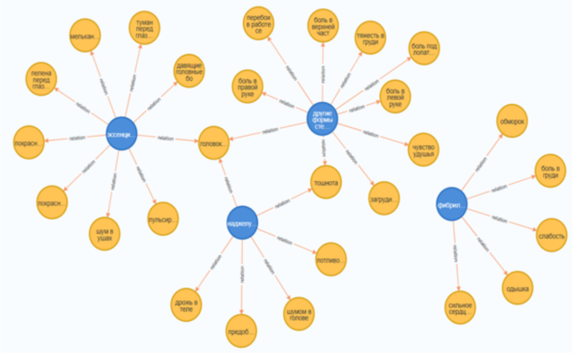

Figure 2. Generated graph derived from test samples of medi8cal data (orange nodes are symptoms, blue nodes are diseases)

3.3. *Highload results*

The proposed architecture comprises four subsystems, operating within a microservice architecture where communication occurs through APIs. It's crucial to consider the system load. Load testing results indicate that the error rate during execution of the disease prediction component with Logistic Regression is 17.96%.

Table 2. Results of highload testing models API services

Service name	Request count	Failure count	Percent of failures
Disease model ICD-10 (Random Forest)	15 000	10 067	67,11%
Disease model ICD-10 (Logistic Regression)	15 000	2 693	17,96%
Disease model ICD-10 (Decision Forest)	15 000	1 695	11,30%
Disease model ICD-10 (SVM)	15 000	1 480	**9,87**

4. Discussion

The developed system addresses challenges in disease classification compatibility, opting for an automatic matching approach to minimize errors compared to manual mapping methods. The micro-service architecture, with distinct subsystems, ensures scalability and reliability. The training layer and knowledge base component contribute to successful model adaptation and a comprehensive understanding of medical data relationships. The modeling experiments demonstrate the system's efficacy in disease classification prediction, with the Logistic Regression model chosen for its favorable throughput characteristics. Highload testing results indicate the system's robustness, with acceptable error rates in disease prediction and symptom detection.

5. Conclusion

The developed system presents a promising solution for disease classification compatibility, focusing on automatic code prediction in medical texts. While the Logistic Regression model is chosen for optimal throughput, ongoing refinement is necessary to address computational limitations. The system's potential lies in its contribution to improved healthcare informatics by enhancing disease classification compatibility and automating code prediction in unstructured medical texts.

References

[1] H. Veseli, G. Kopanitsa, and H. Demski, Standardized EHR Interoperability – Preliminary Results of a German Pilot Project using the Archetype Methodology, Ebooks.Iospress.Nl. (2012). doi:10.3233/978-1-61499-101-4-646.

[2] S.V. Kovalchuk, G. Kopanitsa, I.V. Derevitskii, G.A. Matveev, and D.A. Savitskaya, Three-stage intelligent support of clinical decision making for higher trust, validity, and explainability, Journal of Biomedical Informatics (Print). 127 (2022) 104013. doi:10.1016/j.jbi.2022.104013.

[3] M. Kashina, I. Lenivtceva, and G. Kopanitsa, Preprocessing of unstructured medical data: the impact of each preprocessing stage on classification, Procedia Computer Science. 178 (2020) 284–290. doi:10.1016/j.procs.2020.11.030.

[4] J.-M. Rodrigues, S. Schulz, A. Rector, K. Spackman, J. Millar, J. Campbell, B. StüN, C.G. Chute, H. Solbrig, V. Della Mea, and K.B. Persson, ICD-11 and SNOMED CT Common Ontology: circulatory system, Ebooks.Iospress.Nl. (2014). doi:10.3233/978-1-61499-432-9-1043.

[5] O.V. Stroganov, A. Fedarovich, E.H.M. Wong, Y. Skovpen, E.A. Пахомова, I. Grishagin, D. Fedarovich, T. Khasanova, D. Merberg, S. Szalma, and J. Bryant, Mapping of UK Biobank clinical codes: Challenges and possible solutions, PloS One. 17 (2022) e0275816. doi:10.1371/journal.pone.0275816.