Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Získavanie štruktúrovaných dát o pacientoch s ochorením COVID-19 z prepúšťacích správ a krvných výsledkov

Bakalárska práca

2023
Marián Kravec

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Získavanie štruktúrovaných dát o pacientoch s ochorením COVID-19 z prepúšťacích správ a krvných výsledkov

Bakalárska práca

Študijný program:	Dátová veda
Študijný odbor:	Informatika a Matematika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	Mgr. Vladimír Boža, PhD.

Bratislava, 2023
Marián Kravec

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Marián Kravec
Študijný program: dátová veda (Medziodborové štúdium, bakalársky I. st., denná forma)
Študijné odbory: informatika
matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Získavanie štruktúrovaných dát o pacientoch s ochorením COVID-19 z prepúšťacích správ a krvných výsledkov
Extracting structured data about patients with COVID-19 from discharge reports and blood results

Anotácia: Nemocničný informačný systém v Univerzitnej nemocnici v Bratislava obsahuje iba neštrukturované textové dáta o pacientoch (či už prepúšťaciu správu alebo výsledky krvných testov).

Takýto formát dát neumožňuje efektívnu analýzu dát a hľadanie faktorov, ktoré ovplyvňujú prognózu liečby Covid-19.

Cieľom práce je vytvoriť softvér, ktorý tieto neštruktúrované dáta premení do tabuľkovej podoby, ktoré sa dajú následne jednoducho analyzovať.

Vedúci: Mgr. Vladimír Boža, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 20.10.2022

Dátum schválenia: 22.10.2022

doc. Mgr. Tomáš Vinař, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Touto cestou by som rád poďakoval svojmu školiteľovi Mgr. Vladimírovi Božovi, Phd. za ochotu a rady pri písaní práce a kódu. Zároveň by som rád poďakoval pánovi doc. MUDr. Petrovi Sabakovi, PhD. za umožnenie použitia kódu vyvinutého počas projektu na účely tejto práce.

Abstrakt

Nemocničný informačný systém v Univerzitnej nemocnici v Bratislava obsahuje iba neštrukturované textové dáta o pacientoch (či už prepúšťaciu správu alebo výsledky krvných testov). Takýto formát dát neumožňuje efektívnu analýzu dát a hľadanie faktorov, ktoré ovplyvňujú prognózu liečby COVID-19. Cieľom práce je vytvoriť softvér, ktorý tieto neštruktúrované dáta premení do tabuľkovej podoby, ktoré sa dajú následne jednoducho analyzovať.

Kľúčové slová: prepúšťacia správa, regulárny výraz, tretie

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords: dismissal report, regular expresion

Obsah

Zoznam obrázkov

Úvod

V nemocničnom informačnom systéme Univerzitnej nemocnice v Bratislave nie je možnosť jednoducho získať tabuľku obsahujúcu dáta jedného pacienta respektíve skupiny pacientov, tieto dát sa získavali ručne buď postupným kopírovaním jednotlivých dát nachádzajúcich sa v rôznych častiach informačného systému alebo ich hľadaním a následným prepisovaním z prepúšťacej správy.

Tento proces bolo potrebné opakovať pre každého jedného pacienta čo bolo časovo náročné a vyžadovalo si nezanedbateľné množstvo ľudskej práce.

Hlavným účelom tejto práce je vytvorenie softvéru ktorý pomôže výrazne zrýchliť a zjednodušiť získavanie týchto dát.

Tento softvér na svojom vstupe dostane prepúšťaciu správu pacienta a jeho krvné výsledky v podobe klasického textu a následne z týchto ne-štrukturalizovaných dát získava jednotlivé požadované dáta pričom v prípade, že nejakú informáciu nenájde alebo zistí, že zistená hodnota nie je v očakávaných limitoch alebo, že sa v správe nachádza viacero hodnôt pre jednu informáciu tak užívateľovi oznámi o akú informáciu ide a aký je s ňou problém.

Práca je rozdelená do štyroch kapitol. Prvá kapitola sa zameriava na štruktúru prepúšťacej správy a dáta ktoré sa z nej snažíme získať.

Druhá kapitola obsahuje problémy ktoré sa objavili pre jednotlivé získavané informácie a aké spôsoby riešenia sme sa rozhodli použiť.

V tretej sú riešené časti kódu ktoré priamo nesúvisia so získavaním dát ako je zápis dát do výslednej tabuľky všetkých pacientov, zapisovanie problematických dát do logovacieho súboru alebo grafické prostredie pre jednoduchšiu prácu so softvérom.

Posledná kapitola je... nejaká asi napríklad iné možnosti prístupu k problému abo také niečo... asi nejaké analýzy chybovosti.

Kapitola 1

Podobné práce

Aj napriek moderným nemocničným informačným systémom je stále veľké množstvo nemocničných záznamov v podobe čistého alebo čiastočne štrukturalizovaného textu z ktorého je ručné získavanie dát časovo náročné. Preto sa na to využívajú automatizované systémy ktoré zväčša fungujú na jednom z dvoch princípov respektíve na kombinácii oboch. Tými prístupmi sú regulárne výrazy a metódy strojového učenia určené na spracovanie prirodzeného jazyka.

Obe tieto prístupy majú svoje výhody a nevýhody [4]. Výhodou regulárnych výrazov je ich presnosť a transparentnosť čiže možnosť vidieť a upravovať vnútorné fungovanie programu čiže jednotlivé výrazy hľadajúce konkrétne informácie ale ich nevýhodou je, že hľadané výrazy treba ručne vytvárať a vylepšovať čo je často náročné, špecifické pre určitú oblasť a v prípade komplikovaných výrazov je aj komplikovaná údržba. Na druhej strane v prípade použitia niektorej z metód strojového učenia môže často stačiť použiť už existujúcu metódu spracúvajúcu prirodzený jazyk mierne ju modifikovať pre konkrétne použitie a natrénovať model na predpripravených dátach avšak tieto modely často nie sú až tak presné ako regulárne výrazy a zároveň v prípade nájdenia častej chyby alebo nutnosti modifikácie hľadaných dát (pridanie alebo odobranie získavanej informácie) je nutné model upraviť a celý nanovo pretrénovať a validovať aj už skôr validované časti.

Medzi už existujúce systémy využívajúce regulárne výrazy patrí napríklad systém HEDEA [1] čiže Healthcare Data Extraction and Analysis ktorého autormi sú Anshul Aggarwal, Sunita Garhwal a Ajay Kumar a bol vyvinutý na získavanie Indických medicínskych dát. Systém využitím regulárnych výrazov nájde požadovanú informáciu v texte následne ju spracuje do podoby dvojica získavaný údaj, výsledok (napríklad (diabetes, áno) alebo (výška, 170cm)) a tento údaj je následne zapísaný do databázy k konkrétnemu pacientovi na základe jeho identifikačného čísla ktorý sa na každom spracovávanom texte nachádzať. Podobne ako v našom prípade je hlavnou úlohou tohto softvéru získavať medicínske dáta z čiastočne štrukturalizovaných vstupných dát čiže

lekárskych správ a výsledkov testov. Hlavné rozdiely oproti nášmu systému sú, že systém HEDEA sa snaží získavať základné dáta o pacientovi ako sú osobné údaje, výška, váha, tlak, základné krvné výsledky a tak ďalej zatiaľ čo my získavame okrem týchto dát aj dáta špecifické pre pacientov s ochorením COVID-19 ako napríklad typ oxygenoterapie alebo výsledky testov na protilátky proti vírusu SARS-CoV-2, zároveň je tento model určený na spracovávanie Indických dát takže je vytvorený pre dáta písané v oficiálnom jazyku Indie v tomto prípade angličtinu zatiaľ čo náš model je vytvorený pre dáta v slovenčine.

Systémom využívajúcim metódu strojového učenia na spracovávanie prirodzeného jazyka je napríklad systém ktorý vytvorili Fette a spol. na Univerzite vo Würzburgu [3] využívajúci metódu učenia s učiteľom s názvom Conditional random field [5] ktorej úlohou je označiť jednotlivé slová respektíve viacslovné pomenovania a následne pomocou metódy Keyword Matching with Terminology based disambiguation prepojiť nájdené slová a viacslovné pomenovania s databázou odborných pojmov tak, že ak je prepojenie jednoznačné použije ho a ak je nejednoznačné hľadá v okolí slova slovo s jednoznačným prepojením ktoré bližšie určí prepojenie nejednoznačného slova. Okrem samotného spôsobu získavanie dát je v porovnaní s našim systémom rozdiel aj v prioritách pri získavaní dát keďže náš systém je vytvorený na čo najväčšiu presnosť pri získavaní dát špecificky z prepúšťacích správ zatiaľ čo ich systém je vytvorený tak aby mohol byť natrénovaný na získavanie dát z rôznych typov medicínskych dokumentov či už lekárskych správ, výsledkov testov alebo klinických štúdii a takisto platí, že celý systém je vytvorený pre iný jazyk ako náš systém v tomto prípade ide o nemčinu.

Prístup ktorý kombinuje metódu strojového učenia s regulárnymi výrazmi využili v svojom systéme Cui a kol. [2] ktorý využili metódu s názvom Constructive heuristic ktorej úlohou nebolo priamo hľadanie získavaných informácii v texte ale generovanie čo najlepších regulárnych výrazov na to určených. Tento algoritmus začína s prázdnou množinou regulárnych výrazov a následne iteratívne túto množinu rozširuje a upravuje kým nie je splnená ukončovacia podmienka. Výhodou tohto prístupu oproti bežným metódam strojového učenia je, že na konci trénovania má užívateľ množinu regulárnych výrazov ktoré môže ďalej upravovať a nie "čiernu skrinku"ktorej vnútornému fungovaniu nerozumie, oproti len použitiu regulárnych výrazov má výhodu, že nie je nutné ich vymýšľať od začiatku ale stačí iba výstup mierne upraviť. Hlavným rozdielom oproti nášmu systému je to, že hlavnou úlohou ich systém nie je priame získavanie dát z medicínskej dokumentácie ale generovanie regulárnych výrazov ktoré je po na takýto problém možné použiť.

Kapitola 2

Štruktúra prepúšťacej správy

Táto kapitola sa zameriava na pochopenie vstupných dát a približnú lokalizáciu hľadaných informácii v nich.

2.1 Výzor vstupných dát

Dáta z ktorých sa snažíme získať informácie o pacientovi softvér dostáva v tvare textu obsahujúceho prepúšťaciu správu a krvné výsledky. Väčšina textu v prepúšťacej správe nie je generovaná automaticky nemocničným informačným systémom ale je písaná lekárom čo spôsobuje, že každá správa je do určitej miery originálna.

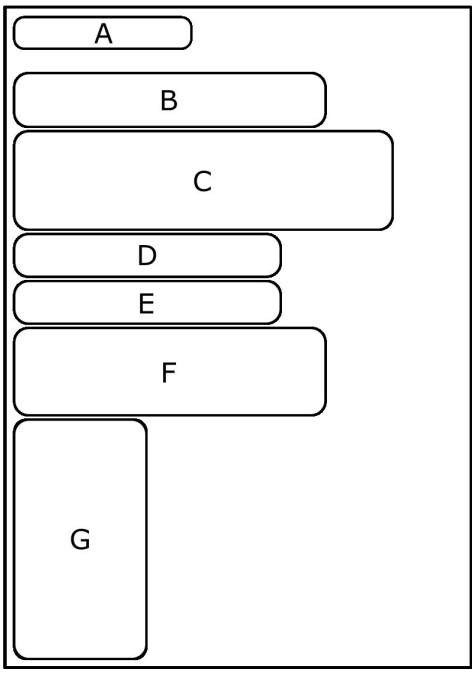
Napriek tomu existuje základná štruktúra ktorú majú spoločnú takmer všetky prepúšťacie správy vďaka ktorej je možné túto správu rozdeliť do blokov (viď obrázok 2.1) ktoré obsahujú určitý informácie. Teraz si prejdeme, čo obsahujú jednotlivé bloky a čo z nich sa mi snažíme získať.

2.1.1 Blok A - Osobné údaje a obdobie hospitalizácie

Bloku A je dvojriadková hlavička v ktorej sa nachádzajú osobné údaje pacienta, čiže jeho celé meno a rodné číslo, a zároveň sa tam nachádza dátum priatia a dátum prepustenia daného pacienta. Všetky tieto informácie sa snažíme získať.

2.1.2 Blok B - Anamnéza

Tento blok obsahuje informácie o anamnéze a stave pacienta pri prijatí do nemocnice, z tohto bloku sa snažíme získavať informácie ako sú výška, váha a saturácia krvi kyslíkom pri prijatí, a informácia o dlhodobých alebo v minulosti prekonaných chorobách a problémoch ako sú cukrovka, astma, demencia, infarkt myokardu, artériová hypertenzia, fibrilácia predsiení, srdcové zlyhanie a ďalšie.



Obr. 2.1: Rozdelenie správy do špecifických blokov podľa ich obsahu

2.1.3 Blok C - Vyšetrenia

Blok C obsahuje informácie o vykonaných vyšetreniach, pre nás sú podstatné výsledky testov na protilátky proti vírusu SARS-CoV-2 typu IgG a IgM pri prijatí, prípadne výsledok testu na ochorenie CDI (Infekcia spôsobená Clostridium difficile).

Zároveň sa tu nachádzajú aj výsledky krvných testov avšak tie sa tu nemusia nachádzať úplné. Preto softvér ktorým to spracovávame ich považuje za kontrolné a samotné informácie o krvných výsledkoch sa získavajú z posledného bloku (2.1.7).

2.1.4 Blok D - Terapia

V tomto bloku sú informácie o terapii odtiaľto získavame informáciu o liekoch ktoré boli pacientovi podané počas hospitalizácie a o tom či pacient potreboval aj oxygenoterapia a v prípade, že áno aj o aký typ oxygenoterapie išlo.

2.1.5 Blok E - Epikríza

Tento blok obsahuje časť správy s názvom epikríza čiže záverečná, súhrnná správa o pacientovi, priebehu jeho choroby a hospitalizácie. Jedinou získavanou informáciou je informácie o smrti pacienta. Táto časť sa však dá zároveň využiť aj na prípadnú kontrolu iných získavaných informácii keďže môže obsahovať informáciu o iných ochoreniach pacienta, jeho liečbe, prípadne o prítomnosti protilátok proti vírusu SARS-CoV-2.

2.1.6 Blok F - Záver, odporúčania, špecifické nálezy

Blok F obsahuje zvyšné časti správy ako sú záver, odporúčania a špecifické nálezy. Avšak obsah tohto bloku je výrazne nekonzistentný a jeho jednotlivé časti nemusia byť vôbec prítomné v správe. Našťastie tento blok by nemal neobsahovať žiadne konkrétne získavané informácie.

2.1.7 Blok G - Krvné výsledky

Záverečný blok už nie je priamo prepúšťacia správa ale ide o krvné výsledky pacienta ktoré narozdiel od výsledkov v bloku C (2.1.3) sa tu nachádzajú úplné a zároveň vďaka tomu, že nie sú napísané vedľa seba ako v bloku C ale pod sebou (každý výsledok na samostatnom riadku) je práca s nimi výrazne jednoduchšia.

2.2 Problémy pri rozdelení dát na bloky

Pri snahe o implementácia tohto delenia sme zistili, že aj napriek tomu, že sa pomerne veľký počet správ dal do takýchto blokov rozdeliť, tak sa objavilo niekoľko problémov či už pri samotnom delení správy ako aj pri informačnom obsahu jednotlivých častí kvôli ktorým sa ukazuje vhodnejšie takéto riešenie vôbec nepoužiť alebo použiť nejakú robustnejšiu verziu tohto delenie. Niektoré z nájdených problémov si teraz opíšeme.

2.2.1 Problém nenájdených blokov

Pri kontrole správ rozdelených do blokov sme zistili, že sa občas stávalo, že náš softvér nebol schopný nájsť niektorý z blokov, väčšinu z týchto problémov vieme rozdeliť do troch skupín: chýbajúci alebo nesprávne ohraničený blok F, blok E skrytý v bloku F, chýbajúci alebo nesprávne napísaný začiatok bloku.

Prvý problém pre nás nepredstavuje veľký problém keďže v bloku F by sa nemali nenachádzajú hľadané informácie.

Druhý problém je o niečo horší keďže blok E je pre nás dôležitý ale tento problém bolo jednoduché opraviť tým, že ak softvér nenájde blok E pri prvotnom delení správy ešte skontroluje či sa v bloku F náhodnou nenachádza.

Tretí problém sa ukazuje ako najproblematickejší sa ukázalo ako pomerne komplikované určiť začiatok a koniec bloku ak softvér nenájde kľúčové slová ktorými sa vo väčšine prípadov jednotlivé bloky začínajú a preto sa stávalo, že softvér niektoré bloky nenašiel a lebo ich pripojil k iným blokom. Tento problém sa stal jedným z hlavných zmeny prístupu k dátam.

2.2.2 Problém nesprávne umiestnených dát

Ako ďalší veľký problém sa ukazuje to, že niektoré informácie sa nenachádzajú sa očakávanom mieste. Zväčša išlo o informácie o anamnéze pacienta a výsledkoch jeho vyšetrení ktoré sme predpokladali, že nájdeme v blokoch B respektíve C avšak časť týchto informácii sa nachádzala až v bloku F.

Tento problém samotný sa dá riešiť tým, že informácie ktoré hľadáme v blokoch B a C budeme nakoniec hľadať avšak tento problém spôsobuje, že problém ktorý máme s nájdením a správnym ohraničením bloku F sa stáva relevantným a treba ho riešiť, nanešťastie tento problém je podobný problému ohraničeniu ostatných blokov a preto je jeho riešenie pomerne komplikované.

2.3 Riešenia problémov s rozdelením dát do blokov

Skúšaním rôznych spôsobov riešenia sa ukázalo, že najvhodnejšie je neriešiť okrajové prípady súčasného rozdelenia správy ale prerobiť samotné rozdeľovanie. Našli sme najlepšie spôsoby...

2.3.1 Homogénny text

Jednou možnosťou je považovať celý text ako jeden homogénny text v ktorom hľadáme všetky informácie.

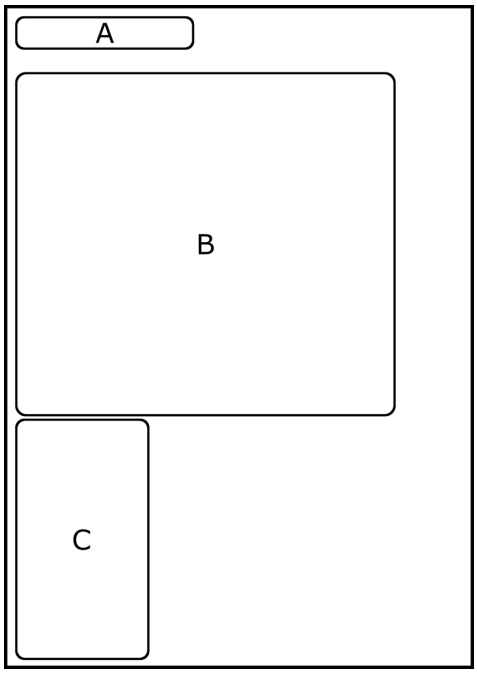
Tento prístup rieši všetky naše problémy avšak zbytočne hľadá niektoré informácie aj na miestach kde vieme, že sa nikdy nebudú nachádzať čo ho spomaľuje.

2.3.2 Menej väčších blokov

Tento prístup využíva rozdelenie do blokov ale namiesto pôvodných 7 blokov toto nové rozdelenie má iba 3 bloky, vďaka čomu nemusí hľadať všetky informácie v celom texte a zároveň pri správnom rozdelení vyriešime problémy ktoré sa v pôvodnom rozdelení objavili.

Výzor nového rozdelenia môžeme vidieť na obrázku 2.2. Vidíme, že jediná zmena ktorá nastala je taká, že sme bloky B až F spojili do jedného bloku s názvom B a pôvodný blok G sme premenovali na blok C.

Toto rozdelenie rieši všetky vyššie spomenuté problémy vďaka tomu, že o bloku A vieme, že vždy bude mať tvar dvojriadkovej hlavičky s pri skúšaní sme nenašli žiadnu výnimku, podobne pôvodný blok G čiže nový blok C nie je súčasťou samotnej prepúšťacej správy ale sú to krvné výsledky pacienta ktoré sú v našom vstupe vždy až za správou a majú tvar dvojíc testovaná veličina a výsledok testu (pozitivita alebo hod-



Obr. 2.2: Upravené rozdelenie správy do menšieho počtu väčších blokov

nota) vďaka čomu je jednoduché ich oddeliť od textu správy. Samotný text správy sa problematické pre ďalšie delenie preto ho nechávame pokope v bloku B.

Kapitola 3

Získavanie dát

V tejto kapitole si povieme niečo o jednotlivých získavaných dátach o problémoch ktorý sa pri ich získavaní vyskytli a ako sme tieto problémy riešili.

Záver

Toto má ešte čas.

Literatúra

[1] Anshul Aggarwal, Sunita Garhwal, and Ajay Kumar. Hedea: a python tool for extracting and analysing semi-structured information from medical records. *Healthcare informatics research*, 24(2):148–153, 2018.

[2] Menglin Cui, Ruibin Bai, Zheng Lu, Xiang Li, Uwe Aickelin, and Peiming Ge. Regular expression based medical text classification using constructive heuristic approach. *IEEE Access*, 7:147892–147904, 2019.

[3] Georg Fette, Maximilian Ertl, Anja Wörner, Peter Kluegl, Stefan Störk, and Frank Puppe. Information extraction from unstructured electronic health records and integration into a data warehouse. In *GI-Jahrestagung*, pages 1237–1251, 2012.

[4] Roni Romano, Lior Rokach, and Oded Maimon. Cascaded data mining methods for text understanding, with medical case study. In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pages 458–462. IEEE, 2006.

[5] Hanna M Wallach. Conditional random fields: An introduction. *Technical Reports (CIS)*, page 22, 2004.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza zdrojový kód programu a súbory s výsledkami experimentov. Zdrojový kód je zverejnený aj na stránke `http://mojadresa.com/`.

Ak uznáte za vhodné, môžete tu aj podrobnejšie rozpísať obsah tejto prílohy, prípadne poskytnúť návod na inštaláciu programu. Alternatívou je tieto informácie zahrnúť do samotnej prílohy, alebo ich uviesť na obidvoch miestach.

Príloha B: Používateľská príručka

V tejto prílohe uvádzame používateľskú príručku k nášmu softvéru. Tu by ďalej pokračoval text príručky. V práci nie je potrebné uvádzať používateľskú príručku, pokiaľ je používanie softvéru intuitívne alebo ak výsledkom práce nie je ucelený softvér určený pre používateľov.

V prílohách môžete uviesť aj ďalšie materiály, ktoré by mohli pôsobiť rušivo v hlavnom texte, ako napríklad rozsiahle tabuľky a podobne. Materiály, ktoré sú príliš dlhé na ich tlač, odovzdajte len v electronickej prílohe.