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 neumožňuje lekárovi získať štrukturalizované dáta o jednom pacientovi respektíve o množine pacientov rýchlo a jednoducho, preto sme vytvorili systém ktorého úlohou je získavania požadovaných z prepúšťacích správ a krvných výsledkov vďaka čomu sa práca lekára výrazne skrátila a zjednodušila. Systém je nastavený na získavanie dát informácii o pacientoch hospitalizovaných s ochorením COVID-19 avšak je ho možné pomerne jednoducho upraviť pre získavanie inej množiny dát.

Kľúčové slová: prepúšťacia správa, regulárny výraz, tretie

Abstract

Hospital information system in University Teaching Hospital in Bratislava does not allow the doctor to obtain structured data about one patient or a group of patients quickly and easily, therefore we have created a system whose task is to obtain the required informations from discharge reports and blood results, thanks to which the doctor's work has been significantly shortened and simplified. The system is set up for obtaining data on patients hospitalized with the disease of COVID-19, but it can be modified relatively easily to obtain another set of data.

Keywords: discharge report, regular expresion

Obsah

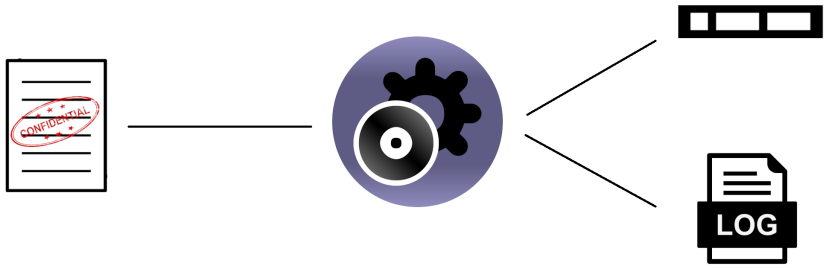
Zoznam obrázkov

Úvod

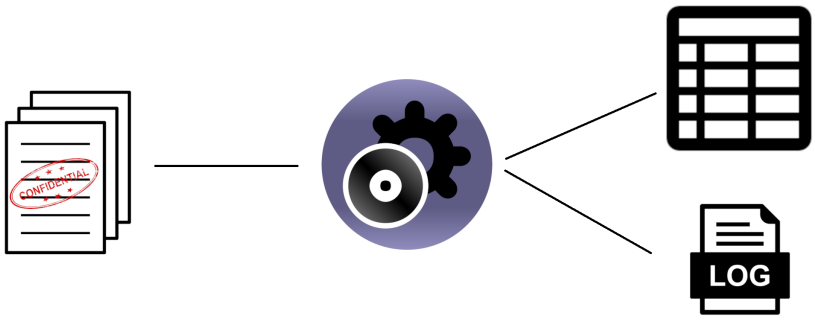
V nemocničnom informačnom systéme Univerzitnej nemocnice v Bratislave nie je možnosť jednoducho získať tabuľku obsahujúcu dáta jedného pacienta respektíve skupiny pacientov, tieto dát sa získavali ručne buď postupným kopírovaním jednotlivých dát nachádzajúcich sa v rôznych častiach informačného systému 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 textu ktorý je čiastočne generovaný nemocničným informačným systémom ale z veľkej časti je písaný lekárom a následne z týchto ne-štrukturalizovaných dát získava jednotlivé požadované dáta. Počas získavania dát si zároveň zapamätáva informáciu o dátach ktoré sa buď nájsť nepodarilo alebo sa počas ich získavania objavil nejaký problém ako napríklad viac rôznych hodnôt alebo protirečiace informácie. Výsledkom je asociatívne pole v ktorom sú ako kľúče použité názvy hľadaných informácii a text obsahujúci informáciu o nenájdených a problémových údajoch.



Obr. 1: Fungovanie systému pre jednu správu



Obr. 2: Fungovanie systému pre množinu správ

Okrem spracovania jedného pacienta je tento systém schopný spracovať aj väčšiu množinu pacientov pričom v takomto prípade už nie je výsledkom asociatívne pole ale je ním tabuľka v ktorej sú získavané údaje stĺpcami a jednotlivý pacienti riadkami. Zároveň informácia o nenájdených a problémových sa zapisuje do logovacieho súboru spolu s informáciou pričom vždy je v súbore zapísané aj o ktorého pacienta ide a ktorom hárku sa nachádza jeho prepúšťacia správa a krvné výsledky.

Náš systém je určený na získavanie základných informácii pacientovi a jeho hospitalizácii ako sú jeho osobné údaje, výška, váha, liečba, krvné výsledky a tak ďalej. Avšak okrem týchto dát náš systém získava aj dáta špecifické pre pacientov s ochorením COVID-19 ako sú oxygenoterapia alebo výsledky testov na protilátky proti SARS-CoV-2.

Práca je rozdelená do šiestich kapitol. V prvá obsahuje opis a porovnanie dvoch najbežnejších metód získavanie informácie z textu, príklady už existujúcich softvéroch využívajúcich tieto metódy na získavanie medicínskych dát a ich rozdiely oproti nášmu systému. Druhá kapitola sa zameriava na štruktúru vstupných dát, čiže prepúšťacej správy a krvných výsledkov, a na informácie ktoré sa z nej snažíme získať. Tretia kapitola obsahuje spôsoby získavania jednotlivých informácii respektíve skupín informácii, problémy ktoré sa pri tomto získavaní objavili a aké spôsoby riešenia sme sa rozhodli použiť. V štvrtej kapitole sa snažíme určiť chybovosť nášho softvéru. Posledná kapitola je o modifikáciách softvéru aby bol jednoduchšie použiteľný a modifikovateľný pre iné podobné použitia.

Kapitola 1

Metódy získavania dát a podobné systémy

Aj napriek moderným nemocničným informačným systémom je stále veľké množstvo nemocničných záznamov v podobe čistého alebo čiastočne štrukturovaného textu z ktorého je ručné získavanie dát časovo náročné. Preto sa na to využívajú automatizované systémy ktoré z týchto textov dokážu získať potrebnú informáciu.

Táto kapitola sa zameriava na zoznámenie sa s dvomi najčastejšími prístupmi na takéto získavanie dát. Zároveň táto kapitola obsahuje na príklady systémov ktoré tieto prístupy využívajú na získavanie medicínskych dát a ich rozdiely oproti nášmu systému.

1.1 Metódy získavania dát z textu

Väčšina systémov určených na získavanie dát z textu funguje na jednom z dvoch princípov, respektíve na kombinácii oboch. Týmito prístupmi sú regulárne výrazy a metódy strojového učenia určené na spracovanie prirodzeného jazyka.

Metóda regulárnych výrazov využíva na hľadanie informácii v texte špeciálne kódované reťazce znakov, takzvané regulárne výrazy, ktoré slúžia ako vzor. Systém hľadá v texte úseky ktoré sa zhodujú so vzorom a z nájdených úsekov sa následne získava konkrétna hľadaná informácia.

Metóda strojového učenia využíva algoritmy učenia umelej inteligencie s učiteľom. Systém dostane trénovacie dáta, čiže množinu textov z ktorých má získavať informácie a množiny správne získaných informácii, a pomocou nich sa snaží naučiť sa tieto informácie v textoch hľadať.

Oba tieto prístupy majú svoje výhody a nevýhody [1]. Výhodou regulárnych výrazov je ich presnosť a transparentnosť čiže možnosť vidieť a upravovať vnútorné fungovanie programu, presnejšie možnosť upravovať jednotlivé regulárne výrazy hľadajúce konkrétne informácie. Medzi nevýhody tohto prístupu patrí napríklad fakt, že všetky

regulárne výrazy ktoré softvér využíva treba ručne vytvárať a vylepšovať čo je často náročné, väčšinou sú špecifické pre určitú oblasť pre ktorú je softvér vytváraný, čo komplikuje využívanie regulárnych výrazov, ktorých správne fungovanie bolo už otestované v iných softvéroch. Ďalšou nevýhodou je, že v prípade komplikovaných výrazov je aj ich upravovanie a vylepšovanie komplikované. Na druhej strane v prípade použitia niektorej z metód strojového učenia je často možné využiť už existujúcu metódu spracúvajúcu prirodzený jazyk a modifikovať ju pre konkrétne použitie a natrénovať model na predpripravených dátach. Avšak aj tento prístup má nevýhody. Natrénovaný model často nie je až tak presný ako dobre nastavené regulárne výrazy. Pre zvýšenie presnosti je často nutné zväčšiť množstvo ručne spracovať dát určených na trénovanie modelu. 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.

Pre náš systém sme si vybrali metódu regulárnych výrazov. Hlavným dôvodom tohto výberu bolo malé množstvo dát použiteľných na trénovanie modelu strojového učenia.

1.2 Podobné systémy

Teraz sa pozrieme na 3 príklady systémov určených na získavanie medicínskych dát z textu. Pričom každý z týchto systémov funguje na inom princípe.

Príkladom systému využívajúceho regulárne výrazy je Healthcare Data Extraction and Analysis (HEDEA) [2], ktorého autormi sú Anshul Aggarwal, Sunita Garhwal a Ajay Kumar, a bol vyvinutý na získavanie Indických medicínskych dát. Systém využitím regulárnych výrazov hľadá každú jednu získavanú informáciu tak, že hľadá v texte kľúčové slovo označujúce požadovanú informáciu, následne ak by mala k danej informácií existovať aj konkrétna hodnota (či už číselná alebo slovná) tak hľadá túto hodnotu v okolí kľúčového slova a nakoniec túto informáciu aj s jej hodnotou zapíše do databázy ku konkrétnemu pacientovi na základe jeho identifikačného čísla ktoré sa na každom spracovávanom texte musí 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 iba základné dáta o pacientovi ako sú osobné údaje, výška, váha, krvný tlak, základné krvné výsledky a prekonané ochorenia inými slovami jeho účelom je vytvoriť databázu obsahujúcu anamnézy jednotlivých pacientov ktorú môže využiť lekár pri diagnostike daného pacienta 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 a našou snahou je vytvoriť tabuľku s dátami o ochorení COVID-19 ktorá sa dá využiť na analýzu rizikových faktorov a liečby tohto ochorenia, zároveň systém HEDEA 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 kol. na Univerzite vo Würzburgu [3] ktorý využívajúci metódu učenia s učiteľom s názvom Conditional random field ktorej úlohou je označiť jednotlivé slová respektíve viacslovné pomenovania [4] a následne pomocou metódy Keyword Matching with Terminology based disambiguation prepojiť nájdené slová a viacslovné pomenovania s databázou odborných pojmov tak, že v prípade jednoznačného prepojenie považuje danú informáciu za klasifikovanú a v prípade nejednoznačného prepojenia (dané slovo môže byť časťou rôznych informácií napríklad v prípade číselnej hodnoty nevieme bez ďalšej informácie určiť k čomu patrí) hľadá v okolí označeného slova iné označené 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 ho bolo možné byť natrénovaný na získavanie dát z rôznych typov medicínskych dokumentov či už lekárskych správ, výsledkov testov alebo klinických štúdii pričom jedinou podmienkou je aby sa všetky získavané údaje nachádzali v databáze odborných pojmov 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. [5] 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 [6]. 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" ako v prípade bežnej metódy strojového učenia, ktorej vnútornému fungovaniu je pre človeka nepochopiteľné. Oproti len použitiu regulárnych výrazov má výhodu, že nie je nutné regulárne výrazy vymýšľať od začiatku ale stačí iba výstup mierne upraviť. Nevýhodou je, že pochopenie a upravenie regulárnych výrazov je síce možné ale môže to byť pomerne náročné keďže ide o počítačom generované regulárne výrazy ktoré aj napriek tomu, že fungujú rovnako dobre môžu sa výrazne líšiť od toho čo by napísal človek. 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

Vstupné dáta a ich štruktúra

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

2.1 Zdroj dát

Prepúšťacie správy a krvné výsledky ktoré náš systém spracováva sú správami pacientov ktorí boli v období od marca roku 2020 do decembra roku 2021 hospitalizovaní na klinike infektológie a geografickej medicíny univerzitnej nemocnice v Bratislave.

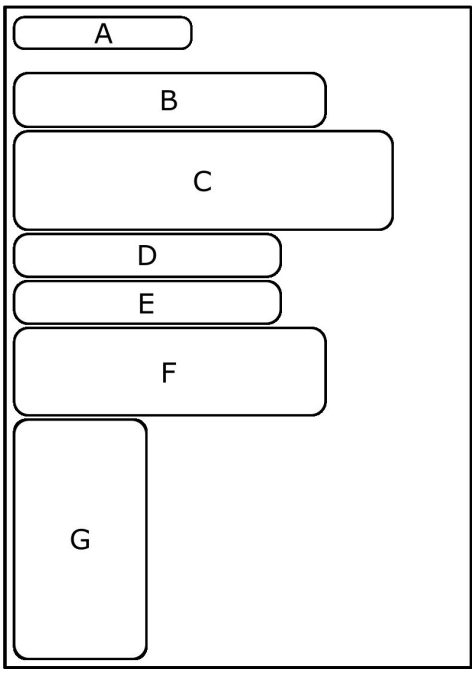
2.2 Výzor vstupných dát

Vstupné dáta z ktorých sa náš systém snažíme získať informácie o pacientovi sú 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 my snažíme získať.

2.2.1 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.



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

2.2.2 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.

2.2.3 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 (2.2.7).

2.2.4 Blok D - Terapia

V tomto bloku sú informácie o terapii, čiže o liečbe ktorá bola pacientovi počas hospitalizácie podaná, odtiaľto získavame informáciu o liekoch ktoré pacient počas hospitalizácie užíval a o tom či bola pacientovi podaná oxygenoterapia a v prípade, že áno aj o aký typ oxygenoterapie išlo.

2.2.5 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.

2.2.6 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.

2.2.7 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 (2.2.3) 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.

2.3 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.

2.3.1 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.

2.3.2 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.

2.4 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...

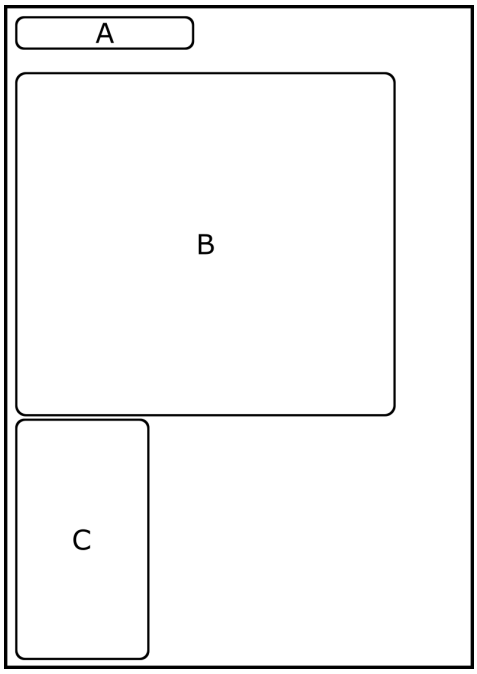
2.4.1 Homogénny text

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.

2.4.2 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.



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

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 vytváraní a testovaní systému 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 hodnota) vďaka čomu je jednoduché ich oddeliť od textu správy. V prípade samotného textu správy sa ukazuje, že jeho delenie vytvára vyššie spomenuté problémy preto ho nechávame pokope v bloku B.

2.5 Predspracované dáta

Na účely kontroly či náš systém funguje správne sme mali v dispozícii aj niekoľko desiatok predspracovaných pacientov, čiže pacientov u ktorých sme mali ich prepúšťaciu správu a krvné výsledky a zároveň sme poznali získavané informácie.

Problémom týchto dát však bolo, že často boli pre konkrétneho pacienta buď nekompletné alebo obsahovali informáciu ktorá sa nenachádzala ani v správe ani v krvných výsledkoch alebo boli nesprávne čo bolo spôsobené tým, že tieto dáta boli získavané iným spôsobom konkrétne tak, že poverená osoba dostala prístup do nemocničného informačného systému z ktorého následne jednotlivé dáta získavala. Konkrétne prob-

lém chýbajúcich dát boli často spôsobený tým, že v čase spracovávania pacienta daný pacient ešte nebol prepustený z nemocnice, v prípade problému dát nenachádzajúcich sa v prepúšťacej správe a krvných výsledkov išlo o prípady kedy lekár túto informáciu do prepúšťacej správy nedal a v prípade nesprávnych údajov išlo s zväčša o ľudskú chybu keďže tieto dáta boli získavané ručne.

Kapitola 3

Získavanie dát

V tejto kapitole si povieme niečo o tom ako sme postupovali pri získavaní dát všeobecne a následne si prejdeme k jednotlivým získavaným dátam ich špecifiká a problémy ktoré sa pri ich získavaní vyskytli a ako sme tieto problémy riešili. Keďže však viaceré dáta majú veľmi podobný až totožný spôsob získavania rozhodli sme sa ich utriediť do skupín dát ktoré majú spolu súvis a majú rovnaký spôsob získavania.

3.1 Všeobecný postup

Vo všeobecnosti bol postup pri získavaní jednotlivých údajov následovný:

1. Výber "trénovacej" množiny dát
2. Zistenie najbežnejších spôsobov zápisu daného údaju
3. Vytvorenie jednoduchých regulárnych výrazov schopných nájsť požadovanú informáciu
4. Pokus o získavanie informácie z dát
5. Kontrola výsledkov, identifikácia problémových miest
6. Modifikácia regulárnych výrazov, eliminácia chybových výsledkov
7. Opakovanie krokov 4 až 6 kým sa v "trénovacej" množine dát vyskytujú chyby
8. Výber "kontrolnej" množiny dát
9. Pokus o získavanie informácie z dát
10. Kontrola výsledkov, identifikácia problémových miest
11. V prípade, že sa chyby nevyskytnú považujeme, naše regulárne za správne, v opačnom prípade považujeme našu "kontrolnú" množina dát za trénovaciu a opakujem potup od kroku 4

Meno	RČ	Dátum prijatia	Dátum prepustenia
Priezvisko Meno	XXXXXX/XXXX	DD.MM.YYYY	DD.MM.YY

Obr. 3.1: Výzor hlavička pacienta ktorý nebol preložený na JIS

Meno	RČ	Dátum prijatia	Dátum prepustenia	
Priezvisko Meno	XXXXXX/XXXX	DD.MM.YYYY	DD.MM.YY preklad	DD.MM.YY

Obr. 3.2: Výzor hlavička pacienta ktorý bol preložený na JIS

3.2 Vstupný súbor a jeho čítanie

Vstupné dáta dostávame ako jeden súbor vo formáte XLSX v ktorom sa každý pacient nachádza v samostatnom hárku.

V hárku boli údaje umiestnené približne tak ako je zobrazené na obrázku 2.1. Náš systém vždy prečíta jeden hárok spracuje pacienta ktorého údaje sú v hárku zapísané a až následne číta ďalší hárok. Systému vieme povedať či už zoznam alebo interval hárkov ktoré má spracovať.

Systém číta hárok postupne a text delí do skupín definovaných v časti 2.4.2. Spôsob akým sa systém rozhodne kde končí jedna časť správy a začína ďalšia je následovný, "Blok A" hlavička je vždy prvá časť a tvorí vždy presne prvé dva riadky tabuľky, následne začína "Blok B" čo je samotný text prepúšťacej správy, za riadok ktorý oddeľuje "Blok B" od "Bloku C" sa považuje riadok obsahujúci iba text "Imunologické vyš.:" čo je riadok ktorý sa samostatne v správe nenachádza ale je vždy prítomný na začiatku časti s výsledkami krvných testov. Následne systém z jednotlivých častí získava požadovanú informáciu.

3.3 Osobné údaje pacienta a obdobie hospitalizácie

Ako prvé boli získavané dáta nachádzajúce sa v hlavičke dokumentu konkrétne išlo o meno pacienta, jeho rodné číslo dátumy prijatia a prepustenia. Zároveň sa v tejto časti dopočítavali 2 hodnoty a to vek pacienta a dĺžka hospitalizácie.

Hlavička má výzor tabuľky s dvomi riadkami a štyrmi alebo piatimi stĺpcami v závislosti od toho či bol pacient počas hospitalizácie preložený z infekčného oddelenia na jednotku intenzívnej starostlivosti (JIS) respektíve na iného oddelenie alebo nie. Konkrétne v prípade ak pacient nebol preložený na iné oddelenie počas hospitalizácie hlavička vyzerá ako na obrázku 3.1 a v prípade, že preložený bol tak vyzerá takto ako na obrázku 3.2.

Softvér z tejto tabuľky vyberie hodnoty z druhého riadka pričom v prípade, že v

štvrtom stĺpci nájde informáciu o preklade na iné oddelenie tak dátum v tomto stĺpci ignoruje a za dátum prepustenia považuje dátum v piatom stĺpci.

Dátumy prijatia a prepustenia sú následne pre-typované do podoby časovej značky čím sa zároveň skontroluje platnosť dátumu (či taký dátum môže existovať) v prípade, že pri niektorom z dátumov vyskytne problém je táto informácia zapísaná do logovacieho súboru. V prípade, že sú obe dátumy v poriadku je následne spravený ich rozdiel čím sa vypočíta dĺžka hospitalizácie, čiže počet dní medzi prijatím a prepustením pričom tento výsledok nám dáva ďalšiu kontrolu dátumov keďže očakávame, že dĺžka hospitalizácie je kladné číslo avšak nepredpokladáme, že to číslo bude veľmi veľké, konkrétne pri testovaní sa ukázalo, že priemerná dĺžka hospitalizácie je približne 12 dní pričom štandardná odchýlka je približne 9 dní avšak občas sa objavia aj prípady ktorých dĺžka hospitalizácie je cez 50 dní pričom nenastal žiaden preklep pri zapisovaní dátumom preto sme sa rozhodli použiť ako hornú hranicu hodnotu 100 ktorá pokrývala všetky naše doterajšie prípady.

Nakoniec v prípade, že dátum prijatia je v poriadku tak softvér určí z rodného čísla dátum narodenia a pomocou dátumu narodenia a dátumu prijatia vypočíta vek pacienta pri prijatí. Tento vek následne kontroluje či je v intervale 0 až 120, v prípade, že do tohto intervalu nepatrí indikuje to chyba buď v dátume prijatia alebo v rodnom čísle pacienta.

3.4 JIS a smrť

Keďže naše dáta sú prepúšťacie správy iba z infekčného oddelenia tak nevieme čo sa s pacientom dialo po opustení oddelenia avšak to čo nás zaujíma je, že dôvod opustenia oddelenia konkrétne to čo nás zaujíma je, že či sa mu stav zhoršil na toľko, že musel byť preložený na jednotku intenzívnej starostlivosti respektíve či dôvodom na opustenie oddelenia nebolo úmrtie pacienta.

V prípade preloženia na JIS je táto informácia v časti "Epikríza" (2.2.5) pričom je väčšinou zapísaná ako "prekladáme na JIS" respektíve "prekladnáme na jednotku intenzívnej starostlivosti" pričom avšak keďže sa táto informácia do epikrízi píše iba v prípade, že preklad uskutočnil tak sa ukázalo, že je vhodnejšie v nej hľadať iba informáciu "JIS" respektíve "jednotka intenzívnej starostlivosti" keďže pri hľadaní aj informácie "prekladáme na" sa vyskytlo viac prípadov keď mal systém problém túto informáciu odhaliť z dôvodu rôzneho skloňovania a gramatických chýb.

V prípade smrti pacienta sa na koniec epikrízi napíše informácia "exitus lethalis". Avšak ukázalo sa, že nie vždy sa táto informácia v správe nachádza aj napriek tomu, že smrť nastal čo je spôsobené tým, že smrť nastala na jednotke intenzívnej starostlivosti alebo sa na ňu z nejakého dôvodu zabudlo preto sa po komunikácii s pánom doktorom

Sabakom rozhodlo, že v prípade, že bol pacient preložený na jednotku intenzívnej starostlivosti alebo bol pripojený na umelú pľúcnu ventiláciu a v správe sa nenachádza informácia o smrti tak sa táto situácia zaznamená do logovacieho súboru aby to mohol užívateľ správnosti informácie skontrolovať.

3.5 Výška a váha

Ďalšie informácie ktoré sa o pacientovi získavali boli jeho výška a váha, pričom následne bola z týchto hodnôt ešte počítaná hodnota BMI. Ukazuje sa, že existuje viacero spôsobov akými lekári zapisujú túto informáciu do správy pričom našou snahou je aby náš systém zvládal nachádzať túto informáciu pri všetkých možnostiach zápisu. Tieto spôsoby sú následovné:

- [výška/váha/hmotnosť] [hodnota][cm/kg] (názov a hodnota)
- [hodnota][cm/kg],[hodnota][cm/kg] (hodnoty oddelene čiarkou)
- [hodnota][cm/kg]/[hodnota][cm/kg] (hodnoty oddelene medzerou)
- [hodnota][cm/kg] [hodnota][cm/kg] (hodnoty oddelene lomkou)

Systém kontroluje prítomnosť každej z týchto možností v texte správy a v prípade nájdenie zhody túto informáciu následne čistí rozdelením v prípade ak sa jedná o dve hodnoty oddelené nejakým deličom a následným odstránením všetkých nepodstatných informácii tak aby nakoniec ostali iba číselné informácie pričom však systém si stále pamätá ktorá hodnota prislúcha ktorej informácii (či ide o výšku alebo váhu). Následne je táto číselná hodnota kontrolovaná či je v "rozumnom" intervale pričom pre výšku je využívaný interval 20-250 cm a pre váhu je interval 10-300 kg ak sa nájde hodnota ktorá nepatrí do týchto intervalov je to považované za možnú chybu a informácia o tom je zapísaná do logovacieho súboru, takisto za chybu sa považuje aj prípad ak systém nedokázal nájsť niektorú z týchto informácii v texte.

V prípade nájdenia oboch informácii (aj výšky aj váhy) je z týchto hodnôt počítaná hodnota BMI pričom tak ako predchádzajúce hodnoty aj táto je kontrolovaná či sa nachádza v "rozumnom" intervale 10-60, vďaka tejto kontrole vieme zachytiť aj chyby s výškou a váhou ktoré nie su evidentné z jednotlivých informácii, medzi takéto prípady patria napríklad chyby ako vymenenie hodnôt výšky a váhy lekárom (príklad z textu správy "výška/váha: 75cm/150kg") alebo možná chyba v niektorom údajov (príklad z textu správy "výška/váha: 103cm/90kg"), samozrejme ani v jednom prípade si nemôžeme byť istý či chyba naozaj nastala a aká konkrétne je a preto v takýchto prípadoch údaje považujeme za priebežne správne a do logovacieho súboru dávame informáciu užívateľovi aby správnosť údajov skontroloval.

3.6 Saturácia krvi kyslíkom pri prijatí

Jednou z ďalších informácii ktorá nás zaujímala bola saturácia krvi kyslíkom. Táto informácia bola pre nás zaujímavá keďže sa jedná o pacientom s ochorením COVID-19 ktoré sa zväčša prejavuje ako respiračné ochorenie a saturácia krvi kyslíkom je zaujímavý ukazovateľ stavu pacienta. Konkrétne nás zaujímala táto hodnota pri prijatí aby sme mali predstavu v akom stave bol pacient hospitalizovaný.

Pri hľadaní sme postupovali tak, že sme hľadali kľučové slovo respektíve skratku (konkrétne sa jednalo o slová "SpO2" alebo "SatO2" a ich varianty) nasledované číselnou hodnotou pričom občas sa stávalo, že medzi hľadaným slovom a hodnotu sa nachádzala ešte informácia "bez kyslíka" označujúca, že bola meraná pred podaním oxygenoterapie pacientovi.

V prípade, že systém našiel v texte viackrát zapísanú informáciu o saturácii krvi kyslíkom tak za správnu považoval tú ktorá sa nachádza v texte najskôr keďže sa informácia pri prijatí zväčša píše do časti "Anamnéza" (2.2.2) ktorá sa nachádza na začiatku správy.

Občas sa objavila situácia, že hodnota nebolo jedno číslo ale bol to interval (príklad z textu správy "SatO2: 90-93%"). V takomto prípade sme sa rozhodli považovať za skutočnú hodnotu dolnú hranicu tohto intervalu.

V prípade nenájdenia žiadnej hodnoty je informácia o tom zapísaná do logovacieho súboru.

3.7 Protilátky proti vírusu SARS-CoV-2 pri prijatí

Keďže náš systém analyzuje prepúšťacie správy pacientov s ochorením COVID-19 tak jednou z pre nás veľmi zaujímavých informácii je výsledok testu na protilátky proti tomuto ochoreniu. Pri hľadania výsledkov testov na protilátky sa ukázalo, že každý lekár ich zapisuje iným spôsobom pričom aj v správach jedného lekára sa nachádzajú rozdiel medzi jednotlivými zápismi. Niekoľko konkrétnych ukážok je na obrázku 3.3.

Pri hľadaní a následnom získavaní informácie sme využili fakt, že všetky spôsoby zápisu týchto testov obsahujú rovnaké respektíve podobné kľúčové slová a pre nás dôležitú informáciu v rovnakom poradí konkrétne vždy prvým kľúčovým slovom je názov vírusu alebo názov choroby v našom prípade je to vírus SARS-CoV-2 respektíve v niektorých prípadoch ochorenie COVID-19, následne ide kombinácia typov protilátok, konkrétne IgG a IgM, a samotných výsledkov testov. Ukážka zvýraznenia hľadaných slov je na obrázku 3.4.

Systém najskôr nájde celé úryvky textu obsahujúce testy následne z týchto úryvkov odstráni všetky slová a znaky ktoré sú pre samotný výsledok nepodstatné ako napríklad slová "slabo", "silno" alebo "rýchlotest". Keďže systém nájde iba špecificky testy na pro-

```
rýchlotest SARS-CoV-2: protilátky IgM:  POZIT,  IgG:  POZIT

S-anti-SARS-CoV-2 IgM (ELISA)/1.12.2020/          55,5          NTU  pozitívny
S-anti-SARS-CoV-2 IgG (ELISA)/1.12.2020/          34,7          NTU  pozitívny

09.12.2020- PROTILÁTKY  SARS - CoV - 2:  IgM:  POZIT IgG:  POZIT

Rýchlotest na vyšetrenie protilátok proti SARS2-COV: IgM silno pozit, IgG slabo pozit

16.3.2021 Protilátky SARS-CoV-2: IgM:pozit., IgG: pozit

Rýchlotest na protilátky SARS CoV-12: IgM slabo pozit.       IgG negat.

Protilátky COVID 19 rýchlotest z kapilárnej krvi: IgG: POZIT    IgM: NEGAT

Rýchlotest na protilátky SARS-CoV2 (z kapilárnej krvi): IgM pozitívne, IgG pozitívne

PROTILÁTKY  SARS - CoV - 2:/5.1.21/.:IgM: POZIT,IgG: POZIT

Protilátky na SARS-Cov-2: 29.12. IgM, IgG negat., 5.1.21 IgM,IgG pozit.
```

Obr. 3.3: Rôzne spôsoby zápisu testov na protilátky v prepúšťacích správach

```
rýchlotest SARS-CoV-2: protilátky IgM:  POZIT,  IgG:  POZIT

S-anti-SARS-CoV-2 IgM (ELISA)/1.12.2020/          55,5          NTU  pozitívny
S-anti-SARS-CoV-2 IgG (ELISA)/1.12.2020/          34,7          NTU  pozitívny

09.12.2020- PROTILÁTKY  SARS - CoV - 2:  IgM:  POZIT IgG:  POZIT

Rýchlotest na vyšetrenie protilátok proti SARS2-COV: IgM silno pozit, IgG slabo pozit

16.3.2021 Protilátky SARS-CoV-2: IgM:pozit., IgG: pozit

Rýchlotest na protilátky SARS CoV-12: IgM slabo pozit.       IgG negat.

Protilátky COVID 19 rýchlotest z kapilárnej krvi: IgG: POZIT    IgM: NEGAT

Rýchlotest na protilátky SARS-CoV2 (z kapilárnej krvi): IgM pozitívne, IgG pozitívne

PROTILÁTKY  SARS - CoV - 2:/5.1.21/.:IgM: POZIT,IgG: POZIT

Protilátky na SARS-Cov-2: 29.12. IgM, IgG negat., 5.1.21 IgM,IgG pozit.
```

Obr. 3.4: Zvýraznené hľadaná informácie v rôznych spôsoboch zápisu testov na protilátky

Tabuľka 3.1: Rozhodnutia systém pri rôznych poradiach názvov testov a ich výsledkov

Poradie častí	**Rozhodnotie**
[protilátka] [výsledok] [protilátka] [výsledok]	názov protilátky pred výsledkom
[protilátka] [protilátka] [výsledok]	názov protilátky pred výsledkom
[výsledok] [protilátka] [výsledok] [protilátka]	názov protilátky za výsledkom
[výsledok] [protilátka] [protilátka]	názov protilátky za výsledkom
[protilátka] [výsledok] [protilátka]	problém (systém považuje daný test nejasne zapísaný)
[výsledok] [protilátka] [výsledok]	problém (systém považuje daný test nejasne zapísaný)

tilátky proti SARS-CoV-2 je pre nás v nájdenom úryvku nepodstatná táto informácia. Nakoniec nám ostanú iba názvy protilátok a výsledky pričom sa ukazuje, že vyskytujú v dvoch formách a to dvojice protilátka a výsledok alebo ako jeden spoločný výsledok pre obe protilátky. V testovaných prípadoch sa vždy vyskytol názov protilátky pred výsledkom testu avšak nakoniec sme sa rozhodli spraviť systém všeobecnejší tým, že prípadov keď je výsledok za názvom testu je schopný správne rozlíšiť a spracovať aj prípady keď je poradie opačné. Konkrétne v prípade nájde iba jeden názov protilátky (obrázok 3.4 druhý príklad) a iba jeden výsledok poradie je jasne určené, v ostatných prípadoch sa rozhoduje podľa tabuľky 3.1.

V prípade, že sa v správe nachádza viacero testov na protilátky proti vírusu SARS-CoV-2 zaujíma nás ten najstarší. Pôvodne bola snaha využiť dátumy avšak ukázalo sa, že veľkom počte prípadov informácia o dátume testovania chýba preto sme sa rozhodli tento postup nevyužiť. Ďalším pokusom bola heuristická úvaha, že čím skôr je test v správe tým je starší, táto heuristika však vykazovala príliš vysokú chybovosť preto sme ani ju nevyužili. Nakoniec sme sa po debate s pánom doktorom Sabakom rozhodli využiť heuristickú úvahu, ktorá hovorí, že ak existuje test na určitú protilátky s negatívnym výsledkom tak aj najstarší test bude mať negatívny výsledok a naopak ak test s negatívnym výsledkom neexistuje (a existujú s pozitívnym výsledkom) tak aj najstarší je má pozitívny výsledok. Táto heuristika sa opiera o výsledky štúdie ktorá ukázala, že test na protilátky typu IgM je pozitívny aj po viac ako 60 dňoch od nástupu príznakov a v prípade protilátok typu IgG je to až 90 dní [7]. Keďže bola väčšina pacientov hospitalizovaná výrazne kratšie obdobie ako 60 dní tak nepredpokladáme, že by počas hospitalizácie nastala situácia, že pacientovi vyjde negatívny výsledok testu aj napriek pozitivite predchádzajúceho. Zároveň sa ukazuje, že ak pacientovi vyjde pozitívny test na obe protilátky už ďalší test na protilátky nie je vykonaný keďže ani lekár zväčša nepredpokladá, že by množstvo protilátok v krvi počas hospitalizácie výrazne kleslo.

3.8 Oxygenoterapia

V prípade oxygenoterapie sme zo správy získavali dve informácie a to či bola pacientovi podaná oxygenoterapia a ak bola určiť o aký typ oxygenoterapie išlo. Konkrétne rozlišujeme tri základné typy a low-flow, HFNO (high-flow nasal oxygenotherapy) a UPV (umelú pľúcnu ventiláciu).

V prípade, že systém nenašiel žiadnu informáciu (žiadne kľučové slová) o podaní oxygenoterapie zapísal si k pacientovi informáciu "bez oxygenoterapie" v opačnom prípade sa typ určí na základe toho aké kľúčové slová boli v texte nájdené. Konkrétne systém nájdené slová vyhodnocuje následovne:

- "low flow", "low-flow" a podobne - low-flow
- "high flow", high-flow", "HFNO" a podobne - HFNO
- "umelá pľúcna ventilácia", "UPV" a podobne - UPV

V prípade, ak sa v texte nachádza informácie aj o low-flow aj o HFNO alebo UPV tak je informácia o low-flow zanedbaná keďže ide o menej "extrémny" typ. Avšak v prípad ak sa v správe nachádza informácia aj o HFNO aj o UPV sú do výslednej informácie o pacientovi zapísané obe a to v tvare "HFNO/UPV". Ak v správe nebol typ oxygenoterapie presne určený ale nachádzala sa tam iba informácie o spôsobe podania ako "maskaälebo "nosová kanyla"je to považované za low-flow oxygenoterapiu pričom do výslednej informácie o pacientovi je to zapísané ako "low-flow (neupresnené)" aby bolo jasne označené, že konkrétna informácii o typu nie je v správe prítomná.

3.9 Lieky

Získavanie informácie sa ukázalo ako veľmi priamočiare keďže našou snahou nebolo získať množinu liekov ktorú pacient dostával ale mali sme množinu liekov a zisťovali sme ktoré z nich pacient dostával. Dôvodom pre tento prístup bol fakt, že pacienti často dostávali lieky a vitamíny určené na liečby iných problémov ktorými títo pacienti trpeli a pre nás bola dôležitá len množina liekov ktoré mohli byť použité na liečbu ochorenia COVID-19. Konkrétne išlo o tieto lieky: Dexametazon, Remdesivir, Olumiant, Favipiravir, Ivermectin a Colchicin.

Konkrétne hľadanie týchto liekov bolo zisťovanie či sa názov lieku nachádza v správe, zo začiatku sme skúšali hľadanie iba v časti "Terapia" (2.2.4) avšak nakoniec sme sa rozhodli hľadať v celej správe aby sme predišli prípadom keď v správe je informácia o danej liečbe ale táto informácia sa už nedostala do terapie a takisto sme chceli zachytiť prípady kedy išlo o samoliečbu pacienta respektíve lieky predpísané

obvodným lekárom ktoré už ale po hospitalizácii neužíval (hlavne v prípade lieku Ivermectin). Keďže sa nenašli prípady kedy by lekár do správy zapísal informáciu o lieku ktorý pacient neužíval tak sa tento spôsob hľadania informácie o týchto liekoch ukázal ako najspoľahlivejší.

Boli aj snahy zistiť okrem samotných liekov aj informáciu o počte dní počas ktorých boli tieto lieky pacient užíval a dávku lieku ktorá mu bola predpísaná avšak sa ukázalo, že vo výraznej väčšine správ buď táto informácia úplne chýba alebo je neúplná a tieto informácie z nej nie je možné získať preto nakoniec padlo rozhodnutie nezískavať tieto informácie.

3.10 Choroby pacienta

Keďže pri ochorení COVID-19 tak ako aj pri iných ochoreniach sa niektoré iné diagnózy považujú za rizikové faktory. Preto aj našou snahou bolo sa pokúsiť získať informáciu či pacient niektorou z diagnóz považovaných za možné rizikové faktory trpí respektíve ju prekonal. Našťastie pre nás sa táto informácia často nachádza v časti "Anamnéza" (2.2.2) avšak môže sa objaviť aj iných častiach správy keďže k danému problému môže dôjsť aj počas hospitalizácie.

Konkrétne diagnózy ktorých prítomnosť alebo prekonanie u pacienta sa snažíme určiť sú následovné: cukrovka, artériová hypertenzia, srdcové zlyhanie, infarkt myokardu, fibrilácia predsiení, periférne artériové ochorenie dolných končatín, chronická obštrukčná choroba pľúc, astma, cievna mozgová príhoda, demencia, sepsa a kolitída. Pričom v prípade ochorení sepsa a kolitída nás nezaujíma či ich pacient v minulosti prekonal ale či sa vyskytli pri prijatí alebo počas hospitalizácie.

V prípade väčšiny týchto chorôb je ich získavanie priamočiare, konkrétne sa jedná o zisťovanie či sa v texte správy nachádza informácia o tejto chorobe u pacienta či už vo forme celého názvu (pri niektorých chorobách niektorého z viacerých používaných názvov napríklad cukrovka a diabetes alebo periférne artériové ochorenie dolných končatín a ateroskleróza) alebo vo forme skratky (napríklad CHOCHP pre chronická obštrukčná choroba pľúc alebo IM pre infarkt myokardu). Tento postup sa ukázal ako vhodný aj pre sepsu keďže prekonanie tejto choroby sa nezapisuje do anamnézy a v anamnéze respektíve v správe sa vyskytuje iba v prípade ak ňou pacient trpí pri alebo počas hospitalizácie.

Špecifické bolo určovanie kolitídy pri alebo počas hospitalizácie. V tomto prípade sa ukázal problém, že veľmi často informácia, že sa v správe nachádza názov tohto ochorenia neznamená automaticky prítomnosť tohto ochorenia u pacienta ale môže isť o negatívny výsledok testu na toto ochorenie alebo prípad, že toto ochorenia bolo pacientom prekonané v minulosti. Konkrétny prístup ktorý sme sa rozhodli použiť pre

čo najlepšie určenia prítomnosti tohto ochorenia u pacienta je tvorený dvomi časťami. Najskôr prebehne kontrola či sa v správe nachádza hocijaká informácie o tomto ochorení čiže či sa tam nachádza názov (kolitída alebo clostridium difficile) alebo skratka (CDI) tejto choroby, v prípade ak sa v správe nenachádza žiadna informácia tak to systém považuje za prípad keď pacient na kolitídu počas hospitalizácie netrpel. Ak sa nejaká informácia o tomto ochorení v správe nachádza nasleduje druhá časť hľadanie kedy hľadáme informáciu o teste na toto ochorenie, konkrétny spôsob hľadania a vyhodnocovania testov na toto ochorenie je podobný ako zisťovanie výsledkov testov na protilátky proti vírusu SARS-CoV-2 (3.7) s rozdielom, že namiesto SARS-CoV-2 hľadáme clostridium difficile a keďže ide o test na prítomnosť protilátok ale baktérie nás netrápi typ (pri SARS-CoV-2 sme rozlišovali IgG a IgM protilátky). V prípade ak sa nájde pozitívny test tak tvrdíme, že pacient počas hospitalizácie na toto ochorenie trpel, naopak v prípade, že všetky nájdené testy sú negatívne tak tvrdíme, že pacient netrpel na toto ochorenie počas hospitalizácie, špeciálny prípad je ak systém nenájde žiaden test na toto ochorenie v takom prípade systém nevie určiť správny výsledok a preto zapíše informáciu o tomto probléme do logovacieho súboru aby to mohol užívateľ skontrolovať.

3.11 Krvné výsledky

Záverečná ale najväčšia skupina informácii ktoré sme získavali boli niektoré výsledky krvných testov. Nie su pre nás dôležité všetky výsledky ale iba určitá vybraná podmnožina konkrétne ide o výsledky ktoré sa v laboratórnych výsledkov označujú nasledujúcimi skratkami: CRP, PCT, S_D3, GLU, KREA, ALT, GMT, FERR, TnT, NEU#, LYM#, EO#, PLT, CD4A, CD8A, INR, FBG, DD.

Hľadanie týchto výsledkov by sa dalo rozdeliť na dve časti a to hlavné hľadanie a doplnkové hľadanie. V prípade hlavného hľadania sú tieto údaje hľadané časti správy "Krvné výsledky" (2.2.7). Konkrétne hľadá informácie v tvare "[názov]: [hodnota]" pričom následne z toho získame samotnú číselnú hodnotu. Priebežne si zapamätáva aj informáciu ktoré údaje nenašiel, v tomto prípade chýbajúci údaj nie je nutne chybou keďže nie každému pacientovi boli robené rovnako komplexné krvné testy. Po pokuse o získania všetkých údajov nastáva druhá časť ktorou je doplnkové hľadanie v tejto časti sa údaje ktoré sa nepodarilo nájsť v časti "Krvné výsledky" hľadajú v texte samotnej prepúšťacej správy. Hlavným dôvodom pre toto hľadanie boli prípady kedy či už z nejakého dôvodu sme nedostali úplné krvné výsledky alebo bolo počas hospitalizácie robených viacero nezávislých krvných testov pričom my sme dostali iba niektoré ale výsledky týchto testov boli zapísané do správy. Tieto dáta sa nachádzajú v správe v časti "Vyšetrenia" (2.2.3) avšak ako bolo aj pri popise výzoru vstupných dát vysvetlené

v tejto časti sa často nachádza iba nejaký výber výsledkov ktorý nemusí obsahovať tie ktoré my potrebujeme. Problém ktorý sa objavuje v tejto časti je fakt, že pre určitý údaj sa v správe môže nachádzať viacero hodnôt keďže ten istý test bol počas hospitalizácie spravený pacientovi niekoľkokrát v takom prípade je informácia zapísaná v tvare "[názov]: [hodnota 1]; [hodnota 2]; ... [hodnota n]," pričom v takom prípade my berieme prvú hodnotu keďže tieto hodnoty sú usporiadané v poradí v akom boli testy robené (časovo) a nás zaujíma hodnota najbližšie k začiatku hospitalizácie keďže chceme vedieť v akom stave bol pacient na začiatku nemocničnej liečby.

V prípade, že ani hlavné, ani doplnkové hľadanie nenájde nejaké výsledky testov je informácie o ich nenájdený zapísaná do logovacieho súboru avšak keďže táto informácia sa nemusí nutne v správe nachádzať nie je táto informácia zapísaná medzi chyby ktoré sa pri získavaní informácii objavili ale je samostatne označená ako nenájdené výsledky testov. V prípade, že sa pri doplnkovom hľadaní zistí, že sa výsledok testu objavuje v správe viackrát pričom sú pri ňom odlišné hodnoty je to považované za chybu a informácia o tejto chybe je zapísaná do logovacieho súboru.

3.12 Chýbajúce časti vstupu

Vo väčšine prípadov je výzor našich presne taký ako je popísaný v kapitole "Štruktúra prepúšťacej správy" (2) avšak občas sa objavil problém neúplnosti dát. Konkrétne išlo o prípady kedy sa vo vstupnom texte nenachádzali výsledky krvných testov, v takom prípade jediná zmena ktorú systém spraví je, že pri získavaní údajov z krvných výsledkoch preskočí hlavné hľadanie (keďže nemá na to potrebný text) a údaje získava iba pomocou doplnkového hľadania v texte prepúšťacej správy. Občas sa objavili ešte extrémnejší prípad kedy sme pre dané pacienta dostali iba hlavička (2.2.1) a informácia "chýba správa" v takomto prípade sa zo získavania dát vykoná iba získavanie osobných údajov pacient a obdobia jeho hospitalizácie (3.3) aby bola zachovaná existencia tohto pacienta a logovacieho súboru sa zapisuje informácia o chýbajúcej správe.

3.13 Výstupné súbory

Výsledkom celého získavania informácie zo vstupných dát je dvojica súborov, ide o súbor vo formáte XLSX ktorý je možné otvoriť v programe Microsoft Excel a obsahuje tabuľku v ktorej jednotlivé riadky sú pacienti a stĺpce sú jednotlivé získavané údaje, druhým súborom je textový logovací súbor obsahujúci pre každého spracovávaného pacienta v tvare zobrazenom na obrázku 3.5.

```
Pacient z hárku [číslo hárku v ktorom sa nachádza daná správa]:
Meno pacienta: [priezvisko a meno pacienta]
Problemove udaje: [názvy údajov ktoré boli počas získavania označené za problémové]
Nenajdene vysledky: [názvy krvných výsledkov ktoré sa v správe nepodarilo nájsť]
```

Obr. 3.5: Výzor zápisu informácie o chýbajúcich a problémových údajoch v logovacom súbore pre jedného pacienta

Kapitola 4

Výsledky systému

Táto kapitola je zameraná na zhodnotenie výsledkov nášho systému. Konkrétne je zameraná na porovnanie rýchlosti systému oproti ručnému spracovávaniu, určenie približnej chybovosti a na využitie v praxi.

4.1 Časová náročnosť

Jedným z hlavných účelov nášho systému je šetrenie času oproti ručnému spracovávaniu. Na určenie či náš systém je naozaj schopný signifikantne skrátiť čas spracovania sme sa rozhodli odmerať časy oboch spôsobov spracovania na rovnakej množine pacientov a tieto časy následne porovnať.

Na tento účel sme vybrali množinu 100 pacientov ktorý dovtedy neboli spracovávaný a vykonali sme obe spôsoby spracovávanie. V prípade ručného spracovania trvalo spracovanie jedného pacienta v priemere 5 minút, čiže spracovanie 100 pacientov trvalo okolo 500 minúť čiže viac ako 8 hodín (tento čas je však ovplyvnený našou neskúsenosťou s ručným získavaním takýchto dát). Náš systém dokázal spracovať túto množinu pacientov za menej ako minútu, konkrétne priemerný čas ktorý na to potreboval bol 40 sekúnd (systém sme spustili niekoľko krát, keďže si systém neuchováva informáciu o predchádzajúcich spustenia boli z pohľadu času jednotlivé spustenia nezávislé), následné ručné vylepšenie pomocou logovacieho súboru trvalo približne 30 minút. Celkový čas ktorý je potrebný na spracovanie 100 pacientov s použitím nášho systému je po zaokrúhlení 31 minút čo približne 16-krát menej ako čas potrebný na ručné spracovávanie. Aj napriek našej neskúsenosti ktorá mala s najväčšou pravdepodobnosťou negatívny vplyv na výsledný čas ručného získavania nepredpokladáme, že by bolo zlepšenie výrazné. Na základe týchto výsledkov môžeme tvrdiť, že čas potrebný na spracovanie určitého počtu pacientov našim systémom je výrazne kratší ako čas potrebný na ručné spracovanie toho istého počtu pacientov.

4.2 Chybovosť systému

V tejto časti sa pokúsime odhadnúť chybovosť nášho systému.

4.2.1 Validačné dáta

Z dôvodu malého množstva predspracovaných dát a problémoch s nimi popísaných v časti 2.5 sme sa rozhodli, že na účely validácia výsledkov nášho systému ručne pracujeme dáta o 100 pacientoch ktorý neboli použitý pri vytváraní regulárnych výrazov.

Konkrétny postup ktorý sme zvolili bol následovný:

1. Príprava súboru obsahujúceho vstupné informácie o 100 pacientoch
2. Ručné spracovanie pacientov
3. Spracovanie pacientov využitím nášho systému
4. Vytvorenie kópie systémom spracovaných pacientov
5. Vylepšenie systémom spracovaných pacientov pomocou logovacieho súboru
6. Porovnanie ručne získaných dát s dátami získanými pomocou nášho systému pôvodnými aj vylepšenými
7. Určenie chybovosti nášho systému

Keďže naše "validačné" dáta boli získavané ručne nemáme istotu, že sú bezchybné, preto sme sa pre účely určenia chybovosti systému rozhodli každý údaj ktorý je totožný s údajom získaným našim systémom považovať za správny a každý údaj ktorý sa líši skontrolujeme aby sme určili či je to chybou nášho systému alebo či ide o ľudskú chybu.

4.2.2 Chybovosť jednotlivých skupín získavaných

Z dôvodu, že náročnosť získavania nie je rovnaká pre všetky získavané dáta ani chybovosť nie je rovnaká preto si najskôr prejdeme chybovosť nášho systému pre jednotlivé skupiny získavaných dát (skupiny sú rovnaké ako v kapitole 3).

Osobné údaje pacienta a obdobie hospitalizácie

V prípade údajov ako je meno a priezvisko pacienta jeho rodné číslo a dátumy prijatia a prepustenia sa neobjavili žiadne chyb. V prípade dĺžky hospitalizácie sa takisto neobjavili žiadne chyby. Pri informácii o veku pacienta sa objavilo pomerne veľké množstvo nezrovnalosti konkrétne 22 pričom toto bolo spôsobený tým, že pri ručnom získavaní bola využitá hodnota veku ktorú lekár napísal do správy a v prípade nášho systém bol

vek vypočítaný na základe rodného čísla a dátumu prepustenia. Presnejšie išlo o to, že lekár s najväčšou pravdepodobnosťou počítal vek iba na základe roku narodenia a z tohto dôvodu do správy napísal vyšší vek aj napriek tomu, že pacient ešte v danom roku nemal narodeniny.

Kontrola s logovacím súborom nepriniesla žiadne zlepšenie. Pri získavaní týchto informácii vykazuje náš systém nulovú chybovosť.

JIS a smrť

V prípade určenia či bol pacient hospitalizovaný sa jednotke intenzívnej starostlivosti a či počas hospitalizácie umrel sa nevyskytol žiaden prípad kde by sa systém nezhodol s ručne získanou informáciou. Takže v tomto prípade to podľa výsledkov vyzerá takisto na nulovú chybovosť, to zároveň znamená, že ani v tomto prípade úprava použitím logovacieho súboru nijak nezlepšila výsledok.

Výška a váha

Pri zisťovaní výšky a váhy sa objavilo dohromady 5 chýbajúcich údajov pri troch pacientoch, trikrát išlo o váhu a dvakrát o výšku v jednom prípade išlo o problém keď systém pri váhe prekvapila informácia "cca" na ktorú nevedel reagovať, v ďalšom prípade nastal problém, keď bolo príliš veľa medzier medzi hodnotami výšky a váha, čo spôsobilo, že systém to nebol schopný nájsť keďže z bezpečnostných dôvodov (aby nepovažoval informáciu nachádzajúcu sa ďalej za hľadanú). Posledným problémom bola situácia keď hľadané informácie boli v špecifickom tvare konkrétne "cm[hodnota]/[hodnota]kg" čo náš systém nebol schopný nájsť.

Všetky tieto problémy boli zapísané v logovacom súbore vďaka čomu pri úpravy výsledkov s jeho použitím boli eliminované všetky chyby.

Takže náš systém spravil chyby pri troch pacientoch z čoho vyplýva chybovosť 3%, ale ak sa pozrieme na počet chybných údajov ide o 5 chýb pri 200 hodnotách (2 údaje, 100 pacientov) z čoho vyplýva chybovosť približne 2,5%, avšak pri použití logovacieho súboru je táto chybovosť nulová.

Saturácia krvi kyslíkom pri prijatí

Pri hľadaní informácie o saturácie krvi kyslíkom pri prijatí náš systém spravil 3 chyby, v dvoch prípadoch za správnu informáciu považoval tú ktorá hovorili o hodnote saturácie po pripojení na oxygenoterapiu keďže bola v správe zapísaná prioritne a informácia o saturáciu bez prídavného kyslíka bola napísaná až neskôr v správe. Jedna chyba bola zapríčinené neočakávaným slovom "RZP" nachádzajúcim sa pred hodnotou saturácie.

Pri vylepšení pomocou logovacieho súboru sa podarilo odstrániť chybu zapríčinenú slovom "RZP", avšak hodnoty ovplyvnené oxygenoterapiou sa opraviť nepodarilo. Práve

kvôli takýmto prípadom boli úvahy o zmene prístupu pri hľadaní tejto informácie a nevyužívať hodnotu ktorá bola zapísaná ako tá hlavná pri prijatí pacienta ale využiť najnižšiu nájdenú hodnoty, tento spôsob vychádzala z úvahy, že po prijatí by vďaka liečbe nemala saturácia klesnúť (oxygenoterapia by mala byť nastavovaná tak aby takáto situácia nenastala), avšak pri vytváraní systém sa táto úvaha ukázala ako nesprávna a často vykazovala nesprávne hodnoty preto nebola nakoniec využitá.

Výsledná chybovosť systému bola teda pri saturácii 3% a pri použití logovacieho súboru klesla na 2%.

Protilátky proti vírusu SARS-CoV-2 pri prijatí

Získavanie výsledkov testov na protilátky proti vírusu SARS-CoV-2 bolo nesprávne pri štyroch pacientoch vo všetkých štyroch prípadoch išlo o problém keď výsledky testov boli zapísané v časti "Epikríza" (2.2.5) pričom v tejto časti správy lekári často nenapísali k testu názov vírusu (daný názov iba vyplýval z kontextu) čo spôsobilo, že systém to nedokázal nájsť. V troch prípadoch to bol jediný test v celej správe čo spôsobilo, že systém nenašiel žiadne výsledky vďaka čomu to ich systém označil za problém a pri oprave pomocou logovacieho súboru sa tieto chyby odstránili. V jednom prípade správa obsahovala niekoľko testov na protilátky ktoré boli pozitívne v oboch protilátkach avšak v epikríze bola informácie o teste s negatívnym výsledok pri protilátky typu IgG čo spôsobilo chybu ktorá nebola do logovacieho súboru zapísaná.

Takže chybovosť systému samotného bola 4% a chybovosť po vylepšení logovacím súborom klesla na 1%.

Oxygenoterapia

Počet chýb pri určovaní typu oxygenoterapie bol nečakane vysoký a konkrétne išlo o 6 chýb z ktorých iba jednu sa podarilo opraviť pomocou logovacieho súboru. Z nezachytených chýb boli 4 prípady kedy bolo rôznymi spôsobmi zapísaná informácia o tom, že oxygenoterapia pacientovi nebola podaná konkrétne išlo o takéto spôsoby zapísania: "bez potreby oxygenoterapie", "bez nutnosti oxygenoterapie", "nevyžaduje oxygenoterapiu" a "nevyžaduje podpornú oxygenoterapiu". Počas vytvárania softvéru sa takéto situácie nestávali a preto systém nebol na takéto situácie schopný správne reagovať, o príčine prečo sa to počas vytvárania môžeme iba špekulovať. Jedným z možných dôvodov môže byť obdobie z ktorého sú správy keďže systém bol vytváraný hlavne na základe pacientov z prvého roka pandémie (v tej dobe existovali iba tieto dáta) zatiaľ čo na toto testovanie boli použitý aj pacienti ktorý boli hospitalizovaný neskôr kde vďaka očkovania mohlo dôjsť k zníženiu počtu pacientov ktorý si vyžadovali oxygenoterapiu. Posledná chyba je podobná ako predchádzajúce s rozdielom, že išlo o určenie konkrétneho typu oxygenoterapie konkrétne nastala situácia, ktorá mala byť vyhodnotená ako

"HFNO" avšak bola vyhodnotená ako "HFNO/UPV" z dôvodu, že v správe bola takáto informácia: "UPV neindikujeme". Ani s touto situáciou sa sme sa pri vytváraní softvéru nestretli.

Z toho vyplýva, že chybovosť nášho systému samotného je pri oxygenoterapii 6% a po oprave pomocou logovacieho súboru klesla na 5%.

Lieky

Pri určovaní či pacient užíva vybrané lieky sa objavili 3 chyby. V jednom prípade lekár spravil chybu a napísal "Dexamentazon" namiesto "Dexametazon" čo bola chyba ktorú systém neočakával. Ďalšie dve chyby vznikli z dôvodu, že lekár do časti "Anamnéza" (2.2.2) informácia "Ivermektín neužíva", takáto situácia sa počas vytvárania softvéru neobjavila a preto ju systém nebol schopný správne vyhodnotiť.

V tomto prípade logovací súbor nijak nepomohol s opravou chýb preto je chybovosť z pohľadu počtu pacientov s chybou 3% (3 zo 100) avšak chybovosť z pohľadu počtu nesprávne určených hodnôt je iba 0.5% (3 zo 600 (6 liekov, 100 pacientov)).

Choroby pacienta

Pri určovaní či pacient trpí niektorými s zo vybraných chorôb (konkrétny zoznam v časti 3.10) sa vyskytlo 5 chýb. Každá chyba bola jedinečná. Objavil sa jeden prípad kedy si systém nebol istý či pacient počas hospitalizácie trpel kolitídou čo bolo však v tomto prípade žiadúce správanie a informácia o tejto neistote bola zapísaná do logovacieho súboru a pri kontrole opravená. Jedna chyba nastala aj v prípade určovania ateroskleróz dolných končatín kde lekár namiesto skratky "PAOO DK" zapísal do správy skratku "PADO DK" kvôli čomu nebol schopný nájsť informáciu o tomto ochorení. Ďalšia chyba nastala keď lekár do správy ku konkrétnemu lieku napísal informáciu "počas sepsy prerušiť liečbu" čo systém nesprávne interpretoval ako informáciu, že pacient počas liečby sepsou trpel. V jednom prípade systém určil, že pacient trpí artériovou hypertenziou na základe toho, že v správe našiel informáciu o tomto ochorení, avšak ako sa pri kontrole ukázalo tento záver systému bol nesprávny keďže išlo o prípad keď lekár do správy zapísal aj diagnosticky významné ochorenia blízkych rodinných príslušníkov pacienta a konkrétne v tomto prípade išlo o ochorenie ktorým trpí matka nami skúmaného pacienta. V prípade poslednej chyby je otázne či ide o chybu systému alebo lekára, alebo o nedostatok informácie v správe, ide o problém pri určovaní kolitídy kedy lekár do záveru správy napísal, že pacient kolitídou trpí aj napriek tomu, že podľa správy boli vykonané testy na toto ochorenie s negatívnym výsledkom. V prípade týchto štyroch chýb oprava pomocou logovacieho súboru nepriniesla ich odstránenie.

Takže z pohľadu chybovosti percenta pacientov s chybou je chybovosť 5% pred kontrolou a 4% po kontrole s logovacím súborom a z pohľadu percenta chybne určených

chorôb je chybovosť približne 0,4% (5 z 1300 (13 chorôb, 100 pacientov)) pred kontrolou a 0,3% (4 z 1300) po kontrole s logovacím súborom

Krvné výsledky

Keďže krvné výsledky sú generované počítačom a nie písané lekárom (pričom to platí aj pre samostatné krvné výsledky aj výsledky zapísané v prepúšťacej správe) tak ich náš systém vie nachádzať s veľkou presnosťou. Konkrétne sa objavilo 5 prípadov kedy náš systém sa nezhodol s ručne získanými dátami (v skutočnosti tých prípadov bolo výrazne viac, o tom v časti 4.2.4). Z týchto chýb 3 nastali z dôvodu, že hľadaný výsledok bol v správe zapísaný viackrát v rôznych častiach správy a náš systém si z možných hodnôt nevybral tú ktorú sme považovali za správnu avšak vo všetkých troch prípadoch bol daný výsledok zapísaný v logovacom súbore ako problémový vďaka čomu bolo možné túto chybu opraviť. Zvyšné dve chyby sa objavili v jednej správe a boli spôsobené tým, že v prvom teste nastala hemolýza kvôli ktorej sa výsledky nepovažujú za dôveryhodné a namiesto toho aby boli tieto výsledky zo správy vynechané bol na ich mieste vynechaný priestor ktorý poplietol náš systém a preto sa mu ich nepodarilo nájsť. Avšak keďže sa mu výsledky nepodarilo nájsť vôbec znamenal, že boli zapísané do logovacieho súboru ako nenájdené dáta a pri kontrole sa ich podarilo nájsť.

Ak by sme za chybovosť brali počet pacientov pri ktorých nastala chyba bola by chybovosť nášho systému pri hľadaní výsledkov testov 4% avšak ak vezmeme do úvahy, že systém hľadal 19 rôznych výsledok je chybovosť iba približne 0,21% pričom platí, že po kontrole s logovacím súborom je chybovosť nulová.

4.2.3 Celková chybovosť

Celkovo sa dohromady objavilo 31 chýb, pričom tieto chyby sa vyskytli pri 23 pacientoch. Časť s týchto chýb sa podarilo opraviť pomocou logovacieho súboru a nakoniec ostalo neopravených 19 chýb a s chybou ostalo 15 pacientov.

Z toho vyplývajú chybovosti z pohľadu percenta pacientov s aspoň jednou chybou na 23% pre vylepšením a 15% po vylepšení a z pohľadu zle získaných hodnôt (pre každého pacienta sme získavali 54 údajov) na 0,57% pre vylepšením a 0,35% po vylepšení.

4.2.4 Ľudská chyba

Pri porovnávaní ručne získaných informácii a tých získaných naším systémom sa ukázalo, že chyby nerobí iba systém ale nezanedbateľný počet chýb sme spravili aj my. Konkrétne sa pri ručnom získavaní objavili chyby pri zapisovaní krvných výsledok kedy sme namiesto hodnoty pre "ALT" opísali hodnotu pre "ALP" alebo namiesto hodnoty

”8,6” sme opísali hodnotu ”6,8”. Takisto pri chorobách a liekoch sa vyskytli situácie keď sme si danú chorobu respektíve liek v správe nevšimli. Dokonca v týchto troch skupinách (výsledky krvných testov, lieky, choroby) bola chybovosť ručne získaných vyššia ako chybovosť softvéru, konkrétne sme pri ručnom spracovaní spravili v týchto skupinách dohromady 26 chýb zatiaľ čo systém spravil iba 13 chýb z ktorých sa 6 podarilo opraviť pomocou logovacieho súboru vďaka čomu konečný počet chýb bol iba 7. V ostatných skupinách dát bola chybovosť systému rovnaká alebo mierne vyššia ako pri ručnom spracovávaní.

Celkový počet chýb bol pri ručnom spracovaní bolo 34 čo je výrazne viac ako 19 ktoré spravil náš systém s použitím logovacieho súboru.

Samozrejme vyššia počet chýb pri ručnom získavaní dát je do istej miery spôsobená našou neskúsenosťou avšak väčšina chýb bola typu ktorý by sa mohol prihodiť ale oveľa skúsenejšiemu človeku, konkrétne išlo o chyby typu prehliadnutý názov lieku alebo choroby, opísaná vedľajšia hodnota z testu, zle opísaná hodnota.

4.2.5 Zhodnotenie výsledkov

Celkovú chybovosť nášho systému môžeme považovať za dobrú keďže je nižšia ako chybovosť nami ručne získanými dátami pričom toto získavanie trvalo systému zlomok času ručného získavania a to aj v prípade započítania času kontroly výsledkov (4.1).

Zároveň môžeme náš systém porovnať zo systémom na extrakciu zo slabo štruktúrovaného textu ktorý vytvoril Matej Minárik [8]. Ktorý je určený na získavanie názvov a množstiev aktívnych látok v liekoch z ich príbalových letákov vykazoval chybovosť vyše 14% (presnosť takmer 86%) a to aj pri zanedbaní duplicít a ignorovaní informácii ktoré systém extrahoval navyše. V porovnaní s naším systémom vidíme, že chybovosť z pohľadu celkovej početnosti chýb je v našom systéme výrazne nižšia.

4.3 Využitie v praxi

Tento systém bol už využitý aj v praxi, konkrétne dáta o hospitalizovaných pacientoch s ochorením COVID-19 ktoré boli pomocou tohto systému získané boli využité ako jeden zo zdrojov dát článku publikovanom v časopise Infectious Disease Reports ktorého autormi sú Peter Sabaka a kol. [9].

Táto štúdia skúmal vplyv prítomnosti protilátok proti vírusu SARS-CoV-2 pri prijatí pacienta do nemocnice na úmrtnosť na ochorenie COVID-19 počas a po skončení hospitalizácie. Výsledky štúdie potvrdzujú predpoklad, že oneskorená tvorba protilátok je spojená s vyšším rizikom úmrtia počas hospitalizácie a aj po jej skončení.

4.4 Možné zlepšenia

Vidíme, že systém síce v malom množstve ale stále produkuje chyby čo znamená, že je stále možnosť zlepšenia. Riešenia niektorých chýb už boli do systému implementované ako napríklad riešenie pre... Pri niektorých problémoch stále prebieha diskusia a ich možnom riešení. Po implementovaní riešení pre čo najviac problém by bola vhodná ďalšia analýza chybovosti avšak ani tieto riešenia ani táto ďalšia analýza nie sú súčasťou tejto práce.

Kapitola 5

Zovšeobecnenie sofvéru

Obsahom tejto kapitole sú úpravy a vylepšenia softvéru tak aby bol jednoduchší na používanie pre používateľa a aby ho bolo možné jednoducho modifikovať pre získavanie dát zo správ pacientov s iným ochorením ako je COVID-19 respektíve pre iných informácii nachádzajúcich sa každej prepúšťacej správe.

5.1 Oddelenie hľadaných informácii od funkcii na hľadanie

Náš systém mal pôvodne výzor skriptu napísaného v programovacom jazyku Python ktorý obsahoval všetky informácie o tom aká informácie hľadáme a pre nich špecifické regulárne výrazy a aj funkcie ktoré ich dáta pomocou nich a všeobecnejších regulárnych výrazov získavali. Tento prístup je však neprakticky z pohľadu užívateľa keďže ak by chcel upraviť aké dáta sa získavajú musí pracovať priamo so skriptom. Preto sme sa rozhodli informácia ktoré dáta chceme extrahovať a ako by mali byť zapísané v správe do samostatného súbor. Pre čo najlepšiu čitateľnosť súboru používateľom sme si pre tento účel vybrali formát YAML. Ide o človekom ľahko čitateľný súborový formát určený na serializáciu dát [10], vďaka čomu je perfektný pre naše použitie.

5.2 Obsah YAML súboru

Obsah súboru "informations.yml" môžeme rozdeliť do niekoľko častí. Prvou časťou je časť "vyberove" v tejto časti je niekoľko riadkov obsahujúcich názov informácia a boolean (pravda/nepravda) hodnotu ktorými hovoríme nášmu systému či má danú informáciu zo správy získavať konkrétne ide o informácie z hlavičky správy, veku, doby hospitalizácie, výške a váhe, saturácii kyslíka v krvi, oxygenoterapii, hospitalizácii na jednotke intenzívnej starostlivosti a smrti pacienta. Takmer všetky tieto hodnoty sú vzájomne nezávislé avšak výnimkou sú vek pacienta a doba hospitalizácie ktoré je

možné získavať iba ak je získavaná informácia z hlavičky preto v prípade hodnoty false pre hlavičku sa tieto informácie nebudú získavať bez ohľadu na ich hodnotu. Druhá časť s názvom "lieky" obsahuje lieky o ktorých chceme vedieť či ich pacient užíval. Jednotlivé lieky sú v zaznamenávané následovne, ako prvé jeden názov stĺpca do ktorého sa bude zapisovať informácia či pacient užíval daný liek (väčšinou je to názov samotného lieku) následne je napísaná dvojbodka za ktorú sa nachádza zoznam najbežnejších názvov lieku (zoznam je ohraničený hranatými zátvorkami, jednotlivé názvy sú v apostrofoch a sú oddelené čiarkami) pričom všetky písmená v týchto názvoch musia byť malé keďže náš systém získava túto informáciu z textu ktorý si upravuje tak aby aby boli všetky písmená malé. Na obrázku 5.1 vidíme ukážku pre liek kolchicín. V tretej časti s názvom "choroby" sú choroby pri ktorých chceme zistiť či nimi náš pacient netrpí respektíve či ich neprekonal, z pohľadu spôsobu zápisu je táto časť totožná s časťou "lieky", malým rozdielom je, že sa tu pre danú chorobu zväčša nachádza viac spôsobov zápisu keďže jedna choroba môže mať viacero označení a skratiek zároveň sú tieto názvy často aj viacslovné a môžu byť rôzne vyskloňované preto si občas vyžadujú mierne komplikovanejšie regulárne výrazy na ich hľadanie. Takisto aj v prípade skratiek treba zaručiť aby nájdená skratka nebola iba podskraktou inej skratky respektíve náhodnou súčasťou textu napísaného výlučne veľkými písmenami. Na obrázku 5.2 vidíme ukážku zápisu pre srdcové zlyhávanie. Vo štvrtej časti nazvanej "vysledky" sa nachádzajú informácie ktorých hodnoty chceme získať z krvných testov ktoré boli pacientovi vykonané. Štruktúra zápise je zase totožne s častou "lieky" bez významných špecifík. Posledná a asi najkomplikovanejšia časť má názov "protilatky" táto časť je určená na špecifikáciu choroby respektíve patogénu a názvov protilátok pre získavanie výsledkov testov na tieto protilátky. Každý takto získavaný výsledok sa do súboru zapisuje následovne, do prvého riadku zapíšeme názov, tento názov je pre náš systém irelevantný a slúži iba ako označenie aby používateľ vedel určiť aký test je hľadaný, nasledujú tri riadky ktoré sú odsadené tabulátorom a majú pevne stanovené názov a štruktúru. Prvý z týchto riadkov s názvom "nazvy_stlpcov" obsahuje zoznam názvov stĺpcov pre jednotlivé získavané protilátky, druhý sú "nazvy_ochorenia" to je zoznam názvov pre ochorenie respektíve patogén na ktorého skúmame prítomnosť protilátok u pacienta a v treťom riadku s názvom "nazvy_protilatok" sa nachádza zoznam názvov protilátok ktoré skúmame, pre druhý a tretí riadok platí, že názvy musia obsahovať iba malé písmená. V prípade, že zoznam "nazvy_protilatok" je neprázdny jeho dĺžka (počet záznamov v ňom) musí byť totožná s dĺžkou zoznamu "nazvy_stlpcov". Špeciálny prípad je ak zoznam "nazvy_protilatok" je prázdny v takomto prípade systém nebude pre dané ochorenie hľadať výsledky testov na protilátky proti danému patogénu ale bude hľadať výsledky testov na prítomnosť daného patogénu respektíve testov na dané ochorenie. Konkrétnu ukážku zápisu pre vírus SARS-CoV-2 vidíme na obrázku 5.3.

```
colchicin: ['colchicin', 'kolchicin']
```

Obr. 5.1: Ukážka zápisu skúmaného lieku do YAML súboru

```
SZ: ['[\s,\n]SZ[\s,\n]', 'CHSZ', 'ChSZ', 'Srdc.{3,6}zlyh', 'srdc.{3,6}zlyh', 'Srdc.{3,6}Zlyh']
```

Obr. 5.2: Ukážka zápisu skúmaného choroby do YAML súboru

5.3 Spôsob modifikácie používateľom

V prípade, že sa používateľ rozhodne zmeniť množinu získavaných údajov je postup následovný:

1. Otvorenie konfiguračného súboru ”informations.yml”
2. Zmena boolean hodnôt (true na false alebo naopak) v časti ”vyberove” pre pridanie alebo odobratia získavania danej informácie
3. Odstránenie riadkov určených na získavanie informácie ktorú získavať už nechceme z častí ”lieky”, ”choroby” a ”protilátky”
4. Pridanie nových riadkov do týchto častí vo formáte definovanom v podkapitole 5.2
5. Uloženie upraveného konfiguračného súboru
6. Spustenie systému na získavanie dát

5.4 Možnosti ďalšieho vylepšenia

Aj napriek výraznému zlepšeniu možnosti modifikácie získavaných údajov oproti pôvodnému systému ani tento spôsob nie je dokonalý a je stále pomerne komplikovaný pre používateľa bez znalosti regulárnych výrazov a prácou s súbormi vo formáte YAML. Preto sa v tejto časti skúsime zamerať na možnosti riešenia týchto problémov. Táto časť práce je však rýdzo teoretická a samotná implementácia jednotlivých vylepšení nie je súčasťou tejto práce.

```
sars:
  nazvy_stlpcov: ['IgG pri prijatí', 'IgM pri prijatí']
  nazvy_ochorenia: ['sars.{0,3}cov.{0,3}2?', 'covid.{0,3}19']
  nazvy_protilatok: ['igg', 'igm']
```

Obr. 5.3: Ukážka zápisu hľadaného testu na protilátky do YAML súboru

5.4.1 Určenie problémov

Ako prvé si musíme určiť aké problémy môže mať používateľ pri práci s naším systémom. Inými slovami, čo komplikuje využívanie a modifikovanie systému. Ako bolo už v úvode tejto časti spomenuté tieto problémy sú dva a to:

- používateľ musí poznať všetky časté spôsoby zápisu danej informácie a často potrebuje byť schopný písať aspoň jednoduché regulárne výrazy
- používateľ musí vedieť pracovať so súborom typu YAML

5.4.2 Problém znalosti spôsobu zápisu informácie a regulárnych výrazov

Podstatou tohto problému je, že vyžadujeme of používateľa hneď niekoľko znalostí ktoré nemusí mať. Konkrétne vyžadujeme aby poznal všetky časté spôsoby zápisu požadovanej informácie, napríklad chceme aby vedel že v prípade fibrilácie predsiení sa využíva aj skratka FP aj skratka FiP. Zároveň vyžadujeme aby vedel využívať syntax regulárnych výrazov, napríklad okolo skratky IM (infarkt myokardu) sa nesmú nachádzať iné písmená inak môže nastať situácia, že systém určí, že pacient na toto ochorení trpí z dôvodu, že v texte nájde meno IMRICH zapísané veľkými písmenami.

Ďalším problémom ktorý z tohto vyplýva je nemožnosť zaručenia správnosti výsledkov keďže dané regulárne výrazy písal používateľ a preto overenie ich správneho fungovanie je na ňom.

Ako jediné rozumné riešenie tohto problému sa javí vytvorenie databázy do ktorej by sme ručne vložili potrebné regulárne výrazy pre čo najväčšie množstvo rôznych informácii ktoré by mohol používateľ požadovať a následne dať používateľovi možnosť si vybrať ktoré z týchto informácii chce získať a dovoliť mu písať vlastné regulárne výrazy iba pre informácie ktoré nie sú obsiahnuté v databáze.

5.4.3 Problém znalosti YAML formátu

Tento problém nie je až tak závažný ako ten predchádzajúci, keďže formát YAML bol vyvíjaný tak aby bol čo najčitateľnejší pre človeka. Zároveň my nežiadame od používateľa vytváranie komplikovaných objektov ale využívanie reťazcov, zoznamov a asociatívnych polí. Avšak faktom je, že od používateľa žiadame aby modifikoval textový konfiguračný súbor čo sa dá považovať za komplikáciu.

Najrozumnejším riešením tohto problému je vytvorenie grafického používateľského rozhrania (GUI) v ktorom by používateľ mohol modifikovať množinu získavaných informácii.

Záver

V tejto práci sme sa venovali problematike získavania medicínskych dát z prepúšťacej správy a krvných výsledkov. Našou snahou bolo vytvorenie systému ktorý by takéto získavanie dát zautomatizoval keďže nemocničný informačný systém Univerzitnej nemocnice v Bratislave (s ktorou sme na tomto projekte spolupracovali) neumožňoval jednoduché a rýchle získania štrukturovaných dát.

Prvá kapitola je venovaná definovaniu metód využívaných na získavanie informácie z textu. Táto kapitola zároveň hovorí o troch už existujúcich systémov určených na problém získavania medicínskych dát a o rozdieloch medzi týmito systémami a naším systémom kvôli ktorým nebolo možné žiaden z týchto systémov priamo využiť pre naše dáta.

Druhá kapitola sa zaoberá vstupnými dátami, čiže textom z ktoré sa informácie snažíme získať. Konkrétna sa venuje výzoru týchto dát, ich obsahu čiže o tomu aké informácie tento text obsahuje a ktoré z nich sa mi snažíme získať. Taktiež sa táto kapitola venuje ak možným rozdeleniam tohto textu na menšie časti.

Tretia kapitola obsahuje jadro celej práce. Hovorí o postupoch využitých pre jednotlivé získavané informácie respektíve pre skupiny informácii. Zároveň hovorí o problémoch pri hľadaní týchto informácii ktoré sa počas vytvárania a nastavovania softvéru vyskytli a aké riešenia boli použité na ich odstránenie. V závere tejto kapitole sa nachádza informácia o výzore výstupných súborov ktorými sú tabuľka obsahujúca získané a logovací súbor obsahujúci informácie o chýbajúcich a problematických dátach.

Obsahom štvrtej kapitoly je určenie chybovosti nášho systému inými slovami určenie ako často náš systém robí chyby. Túto chybovosť určujeme pre celý systém ako aj pre jednotlivé skupiny získavaných dát. Okrem chybovosti ktorú má náš systém sám o sebe určuje aj chybovosť po ľudskej kontrole a oprave výsledkov pomocou logovacieho súboru do ktorého si systém zaznamenáva nenájdené údaje a údaje pri ktorých získavaní nastal neočakávaný problém.

Piata kapitola je zameraná na vylepšenie systému tak aby bolo pre používateľa jednoduchšie upravovať množinu získavaných údajov. Snahou bolo aby používateľ potreboval na používanie a modifikáciu nášho systému čo najmenej vedomostí z oblasti informatiky konkrétne programovacieho jazyka Python a regulárnych výrazov. Tento cieľ sa doposiaľ podarilo spĺniť iba čiastočne pomocou oddelenia konfigurácie (množiny

získavanej informácie) do samostatného súboru ktorý je písaný v človek ľahko čitateľnom formáte. Avšak stále je potrená znalosť aspoň základov o regulárnych výrazoch.

Literatúra

[1] 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.

[2] 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.

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

[5] 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.

[6] Constructive heuristic. in wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Constructive%20heuristic&oldid=1032227019`, 2022. [Online; accessed 07-December-2022].

[7] Rachel West, Amanda Kobokovich, Nancy Connell, and Gigi Kwik Gronvall. Covid-19 antibody tests: a valuable public health tool with limited relevance to individuals. *Trends in microbiology*, 29(3):214–223, 2021.

[8] Bc Matej Minárik (2016). Extrakcia informácií zo slabo štruktúrovaného textu. *Diplomová práca, Vysoké učení technické v Brně.*

[9] Ján Jurenka, Anna Nagyová, Mohammad Dababseh, Peter Mihalov, Igor Stankovič, Vladimír Boža, Marián Kravec, Michal Palkovič, Martin Čaprnda, and Peter Sabaka. Anti-sars-cov-2 antibody status at the time of hospital admission and the prognosis of patients with covid-19: A prospective observational study. *Infectious Disease Reports*, 14(6):1004–1016, 2022.

[10] Wikipedia contributors. Yaml — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=YAML&oldid=1144993083, 2023. [Online; accessed 26-March-2023].

Príloha A: obsah elektronickej prílohy

Príloha B: Používateľská príručka