VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

SÉMANTICKÁ ANALÝZA MATRIČNÍHO ZÁZNAMU

SEMANTIC ANALYSIS OF PARISH RECORD

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ADAM KAŇKOVSKÝ**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAROSLAV ROZMAN, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

146893

Ústav: Ústav inteligentních systémů (UITS)
Student: **Kaňkovský Adam**
Program: Informační technologie
Specializace: Informační technologie

Název: **Sémantická analýza matričního záznamu**
Kategorie: Umělá inteligence
Akademický rok: 2022/23

Zadání:

1. Nastudujte matriční knihy a různé typy zápisů do nich. Nastudujte metody a knihovny pro sémantickou analýzu textu. Zaměřte se na rozpoznávání jmen, příjmení, povolání, obcí a příčin úmrtí. Nastudujte OCR program PERO z FIT VUT.
2. Navrhněte program, který bude mít na vstupu text získaný z OCR pro ručně psaný text a výstupem budou křestní jména, příjmení, povolání, atd. která se budou ukládat do příslušných míst v databázi.
3. Navržený program implementujte.
4. Otestujte úspěšnost rozpoznávání jednotlivých kategorií slov a navrhněte případná vylepšení.

Literatura:

- Straka, M. a Straková, J., 2014, NameTag, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
- Kišš, M., Hradiš, M. a Kodym, O. Brno Mobile OCR Dataset. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. September 2019, p. 1352–1357. ISSN: 1520-5363.

Při obhajobě semestrální části projektu je požadováno:
První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 3.11.2022

Abstrakt

Cílem této práce je navrhnout a implementovat aplikaci pro sémantickou analýzu matričních záznamů, která bude mít na vstupu text získaný ze scanu matriky. Tyto záznamy zpracuje a výsledky vyexportuje do příslušných polí tabulky.

Abstract

The aim of this work is to design and implement an application for semantic analysis of matrix records, which will take as input text obtained from a matrix scan. The extracted information will then be entered into the appropriate fields of the table.

Klíčová slova

Genealogie, Sémantická analýza, Zpracování přirozeného jazyka, NLP, NER, CNEC, Matrika, Python, Spacy, PERO OCR

Keywords

Genealogy, Semantic Analysis, Natural Language Processing, NLP, NER, CNEC, Matrix, Records, Register, Python, Spacy, PERO OCR

Citace

KAŇKOVSKÝ, Adam. *Sémantická analýza matričního záznamu*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Sémantická analýza matričního záznamu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Adam Kaňkovský
7. května 2023

Poděkování

Děkuji svému vedoucímu práce za jeho ochotu a čas který mi při vytváření práce věnoval. Dále bych chtěl poděkovat své sestře Marii, která mi práci gramaticky opravovala a celé rodině, za podporu při studiu.

Obsah

Kapitola 1

Úvod

Genealogie a všechny činnosti s ní spjaté jsou mezi lidmi čím dál více populární. Někteří chtějí najít jen zapomenuté příbuzenstvo, jiní vědět kam až sahají kořeny jejich rodiny, nebo si sestavit rodokmen jako určitý způsob dekorace. Ať už jsou jejich důvody zkoumání svých rodinných kořenů jakékoliv, záhy zjistí, že studium matričních záznamů není nijak jednoduchá věda a vytvoření takového rodokmenu, pokud nejsme v genealogii nijak zběhlí, není vůbec jednoduché.

I když už nyní můžeme ke všem neživím matrikám přistupovat online, jsou bohužel přístupné pouze ve formátu scanu a není tak možné v nich nijak automaticky vyhledávat a nezbývá nám potom jiný způsob než matriční knihy přečíst. Mnohdy nám práci ještě ztěžuje skutečnost, že matriky se v průběhu staletí vyvíjely, jak jazykově, formátově, tak třeba stylem psaní v daném časovém období. I toto nám hledání velmi znepříjemňuje.

Na digitalizaci a restrukturalizaci se proto v poslední době klade taky velký důraz a dostávají se nám tedy nástroje kterými můžeme převést text z obrázku do počítačové podoby. OCR (typ programu který se touto funkcí zabývá) nám může být při hledání v matrikách velkým pomocníkem. Kdybychom měli záznamy po jednom uložené v databázi dalo by se tak v záznamech efektivněji vyhledávat, nebo záznamy filtrovat. Bylo by proto dobré vytvořit nějaké programové řešení, právě pro výběr důležitých informací z textu.

Práce se proto bude věnovat sémantické analýze, která se problematikou porozumění a extrakce textu zabývá. Nejdůležitější prvotní fáze srozumění se s problematikou se však samotné sémantické analýzy zas tolik nedotýká, ale její pochopení nám pomáhá vybrat správné technologie pro práci z textem.

Vývoji matrik je proto věnována velká část kapitoly studium problematiky. Další důležitá část této kapitoly se týká převodu obrazu na text (OCR). Zbytek druhé kapitoly už se sémantické analýze věnuje a to hlavně ve vztahu k českému jazyku, ve kterém bude nejčastěji používána a technologiím, které nám poskytnou potřebné nástroje k jejímu provedení.

Třetí kapitola už se zabývá samotným návrhem systému, který bude celé zpracování textu vykonávat. Důraz je kladen na otázky výběru jazyka a důvodu výběru předem popsaných technologií.

Další kapitoly se věnují už implementaci dané aplikace a jejímu testování, ať už dílčích částí, tak i aplikace jako celku. Hlavní informací pro nás potom je úspěšnost rozpoznávání jednotlivých kategorií slov.

Kapitola 2

Studium problematiky

2.1 Matriky

Matrika je základním archivním pramenem pro genealogicky zaměřené badatele [3]. Název Matrika vychází z latinského slova *Matrix* v překladu matka [8]. Podle aktuálního znění zákona č. 301/2000 se za matriku považuje státní evidence narození, uzavření manželství, vzniku registrovaného partnerství a úmrtí fyzických osob na území České republiky [23].

2.1.1 Vývoj Matrik

V následující části si projdeme vývoj matrik od počátku jejich zaznamenávání u nás až po přítomnost, Jaký byl jejich důvod jejich zaznamenávání a jaké údaje popisovali.

Předchůdci matrik

Ačkoliv matriky jak je známe teď vznikaly až v pozdějších dobách, jejich předchůdce v různých formách můžeme nalézt už ve starověkých kulturách. Už Egypťané, Židé nebo staří Římané měli jakési spisy obyvatel, ty však nevyužívali k vojenským nebo ekonomickým účelům, ale převážně k zajištění přesného počtu členů království božího na zemi. O samotné šíři těchto záznamů však víme z dějin velmi málo [2].

Vznik matrik

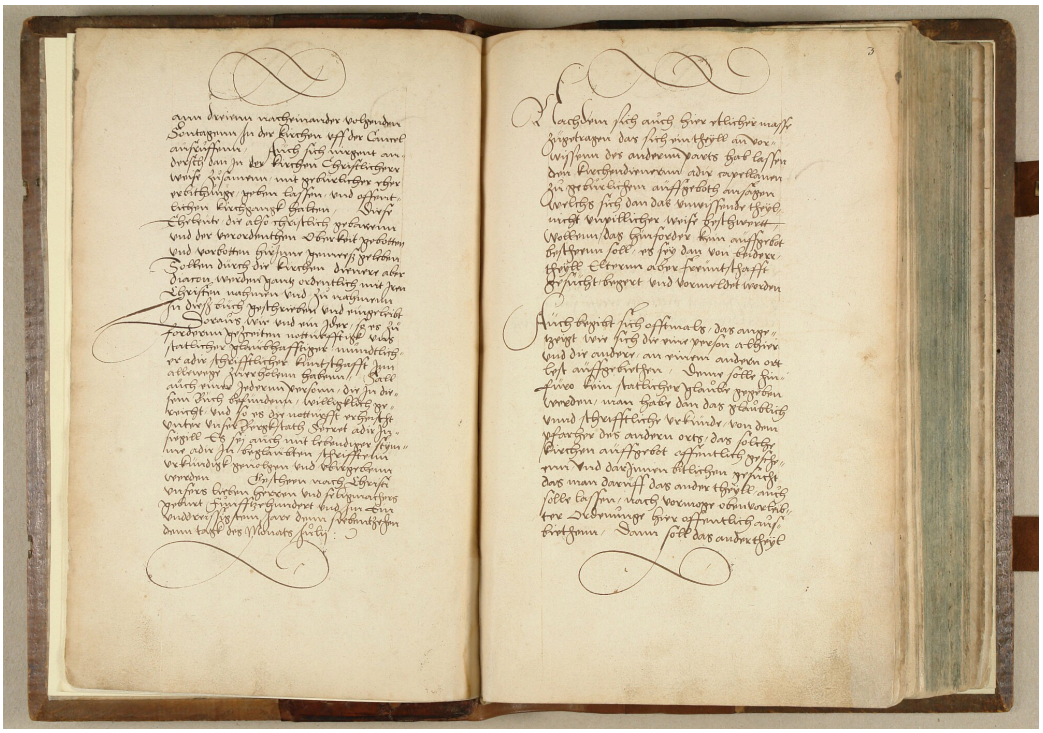
Kdy vznikly matriky nelze přesně určit, ale patrně mohou být spjaty se vznikem far ve 3. století. V českých zemích vznikaly farní organizace až ve 12. století. Na lateránském koncilu v roce 1137[1] je už s vedením matrik počítáno. Johann Jacob Speidel uvádí, že pařížský koncil roku 1212 stanovil vedení matrik jako povinnost. Obsahem měl být datový údaj o narození, křtu a úmrtí [4, s. 42]. První dochované evropské matriky pocházejí z románských zemí: Itálie, Francie, Belgie a Španělska z 14.-15. století [12, s. 41]. Základní církevní směrnice byly však specifikovány až na tridentském koncilu[2] v rámci reformy manželských otázek. Tridentský koncil také odstartoval vývoj matrik v Německu [12, s. 41]. Od té doby měl každý kněz vést knihu do které zapíše jména snoubenců a svědků, den a místo sňatku. U křtu pak musel kněz ještě před obřadem znát jména kmotrů, která následně i se jmény rodičů zapsal do knihy [4, s. 42].

[1] Byl 3. lateránský koncil a 11. ekumenický koncil svolaný papežem Alexandrem III.

[2] Byl 19. ekumenický koncil uznaný katolickou církví svolaný papežem Pavlem III. roku 1545

Začátek vedení matrik u nás

Zakládání prvních matrik u nás úzce souvisí právě s německou iniciativou. I proto se nám prvním matriky dochovaly ze severozápadního pohraničí a to z Jáchymova[3], po kterém nasledovaly další obce západních a severních Čech jako jsou Horní Blatná 1541, Abertamy 1544. První matriky z jiné části země byly matriky středočeské (Slaný 1556, Kutná Hora 1571, Praha 1584) [12, s. 41]. Vedení matrik u nás bylo přímo nařízeno až olomouckou synodou 1591 a pražskou synodou konanou v roce 1605, které přijaly ustanovení Tridentského koncilu. Do formy zápisu matrik pak v přímo zasahoval Rituale Romanum z roku 1614[4], díky kterému byly stávající dvě knihy rozšířeny o knihu zemřelých. Tyto knihy měl vést každý kněz pro svůj obvod působení. Je zde také definováno pořadí: svatební, křestní, úmrtní [4, s. 42].



Obrázek 2.1: První dochovaná matrika oddací z Jáchymova z roku 1531[5]

Tyto dvě směrnice na dlouhou dobu definovaly způsob vedení matrik. Pozdější výzkum však zjistil, jak byly matriky vedeny, protože samotné formuláře, podle kterých se matriky měly psát, byly stručnější než definovaly předpisy.

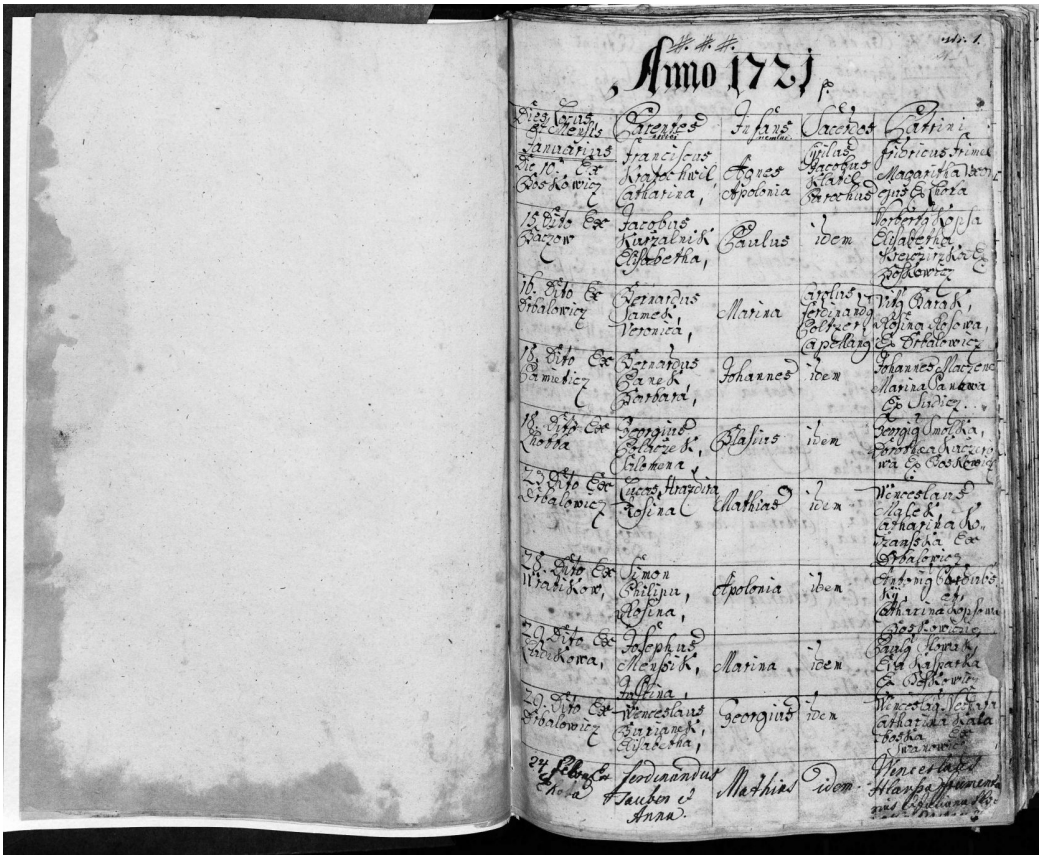
Nejdříve byly tyto matriky psány *per extensum*[6] po celé straně, později se však začaly vypouštět opakovaná slova díky zavedení rubrik. I to zapříčinilo, že byly matriky u nás nejednotné. Psaní matrik se v 17. století ujímali kromě kněží taky učitelé. Změna zápisu nastala v 18. století [4, s. 43].

[3] Oddací matrika z roku 1531

[4] Římský rituál - Liturgická kniha obsahující bohoslužebné předpisy a texty římskokatolické vydaná papežem roku 1614.

[5] Zdroj: `http://www.portafontium.eu/iipimage/30063190/`

[6] Latinsky - v plném rozsahu [8]



Obrázek 2.2: Případ použití rubrik z matriky narození Boskovice 1721[7]

Vývoj matrik za vlády Habsburků

Od konce 18. století mají matriky u nás zachovaný vesměs stejný ráz po celém území bývalého Rakouska, kde takzvaný „osvícenský stát“ tyto jednotné matriky potřeboval k vojenským důvodům [4, s. 44].

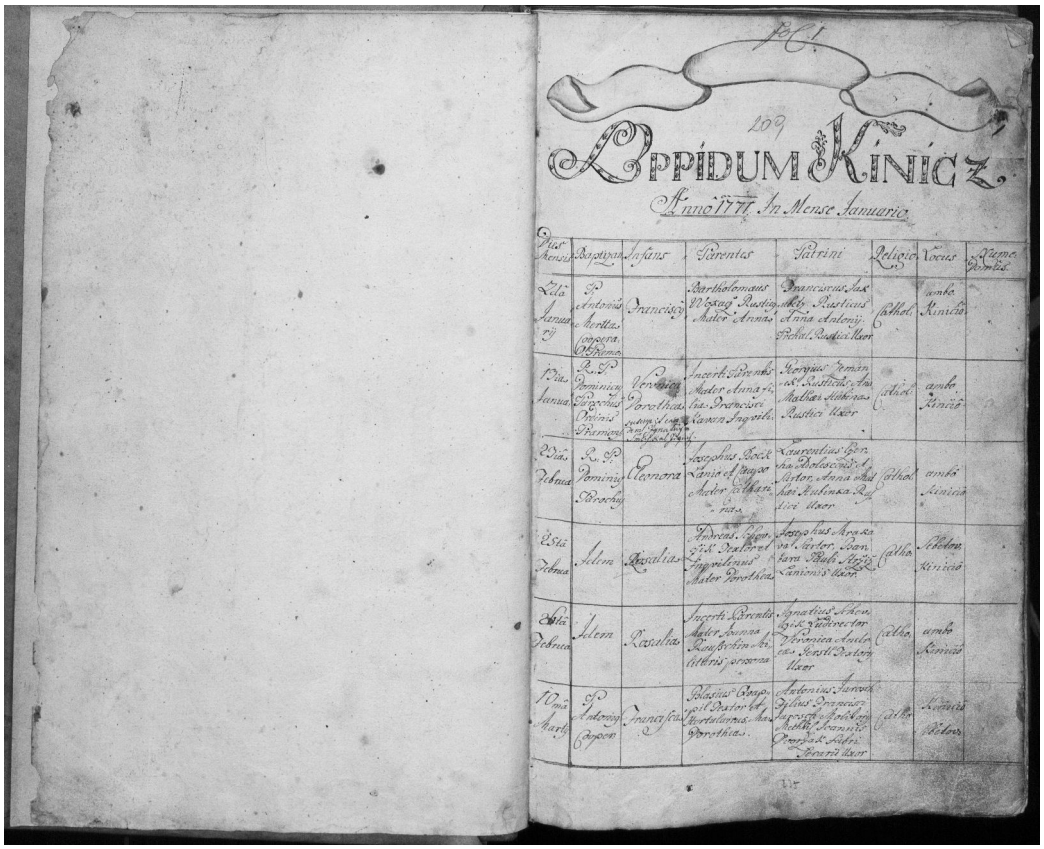
První nařízení které tomuto sjednocení velmi pomohlo bylo nařízení pražské konzistoře z roku 1760 a to zavedení jednotné latinské formule po vzoru Rituale Romanum [12, s. 46].

Úpravy matrik z důvodu válek

Špatné zkušenosti Marie Terezie z války o dědictví a ze sedmileté války ukázaly, že největší váhu v samotných válkách má velikost vojska. Pro správné odvody bylo proto potřeba mít přesný pojem o počtu obyvatel. Zatímco v Praze prováděla roku 1770 komise sčítání dům od domu podle domovního čísla, na venkově měli tuto evidenci o počtu vést právě matriky, ze kterých měli správci po ukončení expirace podávat krajskému úřadu čtvrtletní zprávu ze všech třech matričních knih. Toto byl velký zásah státu do vedení matrik ze strany církevní obce [4, s. 44]. Navíc byly 6. října 1770 dvorským dekretem nařízeny jednotné formuláře pro psaní matrik, které nahradily větný zápis, zápisem do rubrik. Díky tomu se matriky na farních úřadech stávaly stále více jednotné [12, s. 47]. Úpravu vojenských a židovských

[7] Zdroj: `https://www.mza.cz/actapublica/matrika/detail/2903?image=216000010-000253-003368-000000-000170-000000-00-B01781-00010.jp2`

matrik také ovlivnil návrh rubrik stejně, jak to mu bylo ve farních matrikách z roku 1770, když se zde dekret konečně po 14 letech uskutečnil.



Obrázek 2.3: Případ použití rubrik předepsaných dvorským dekretem v matrice narození Knínice u Boskovic 1771[9]

Změna zápisu jmen

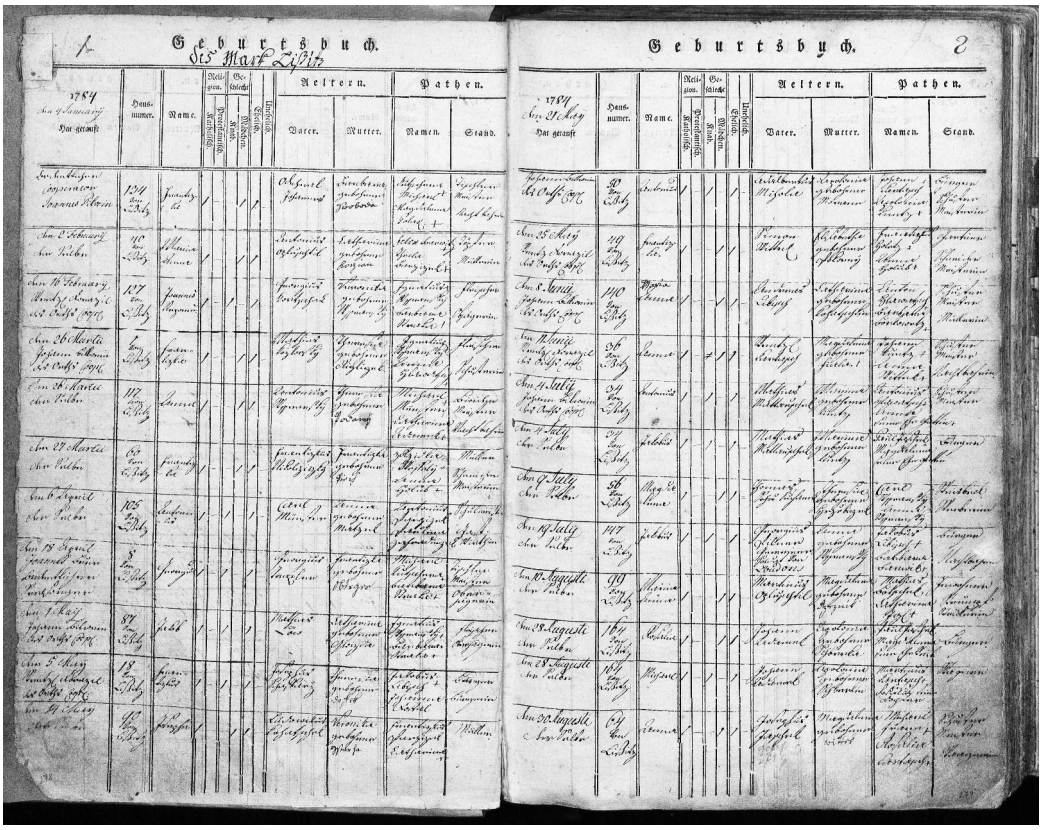
Další nařízení, které se citelně dotklo způsobu psaní matrik, bylo guberniální nařízení z 20. října 1779. To definovalo, aby u lidí se dvěma jmény byly uváděna obě. Toto roku bylo také nařízeno, že si lidé tyto jména nemohou měnit, ovšem až dekretem dvorského kancléře z roku 1826 byly tyto změny příjmení opravdu zakázány. Roku 1866 byla pravomoc k povolování změny příjmení přesunuta na zemské politické úřady [4, s. 44].

Významné nařízení Josefa II.

Zákon o psaní matrik v celém státě je připisován patentu Josefa II. z 20. února 1784, který stanovil nový, mnohem důkladnější formulář, sloužící přes menší úpravy až do roku 1949 [12, s. 47]. Roku 1784 také přestalo užívání latiny při psaní matrik. Některé matriky se proto ihned začali psát v češtině, některé byly však po krátkou dobu psané německy [12, s. 48]. Obzvláště v pohraničí se však vyskytovaly matriky, kde se německy psalo ještě v polovině 19. století. V tomto roce bylo také dekretem stanoveno, že má být pro každou obec vedená

[9]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/2903?image=216000010-000253-003368-000000-000170-000000-00-B01781-00010.jp2`

zvláštní matrika nebo alespoň oddíl [12, s. 49]. Dvorským dekretem roku 1790 bylo nařízeno, aby všechny tři druhy matričních knih byly vybavené abecedními indexy, které usnadnily hledání v matrikách [12, s. 50]. Od konce 18. století jsou už všude zavedené matriky. Od roku 1784 se vydávají první přehledy, které jsou základem statistky obyvatel [4, s. 48].



Obrázek 2.4: Použití předepsaných tištěných rubrik Lysice 1784[10]

Další změny ve vedení matrik

Dvorským dekretem 4. července 1801 bylo definováno, že i při dlouhé nepřítomnosti otce u rodiny má být děti zapsány jako manželské (pater est, quem justae nuptiae demonstrant[11]). Zásada byla zapsána i v občanském zákoníku, dávala však otci možnost podat žádost na od-uznání otcovství. Dekret dvorské kanceláře 2. července 1825 nařizoval zapisovat do křestní matriky jména porodní báby jako svědecký prvek [4, s. 48].

Vedením matrik nekatolických církví

. Dvorský dekret z 22. února 1782 definoval vedení evangelických matrik katolickými faráři. Roku 1784 bylo dokonce nařízeno, že i lidé s nekatolickým vyznáním se musí dostavit na katolické fary a nařízení z 30. dubna 1789 dokonce výslovně stanovuje, že katoličtí faráři mají na starost psát do matrik katolíky i nekatolíky. Až 26. listopadu 1829 dekret kancléře

[10] Zdroj: `https://www.mza.cz/actapublica/matrika/detail/3488?image=216000010-000253-003368-000000-000625-000000-00-B00117-00010.jp2`

[11] Otcem dítěte manželčina jest její manžel [14].

udával právo nekatolickým duchovním správcům vést také všechny tři druhy matrik. Tento dekret také udával, že nekatoličtí správci musí vést duplikát jejich matrik, který následně pošlou katolickému faráři, který ho připojí ke své matrice. Ministerský výnos 30. ledna 1849 specifikoval, že za nekatolíky se považují evangelíci augsburského nebo helvetského vyznání [4, s. 48].

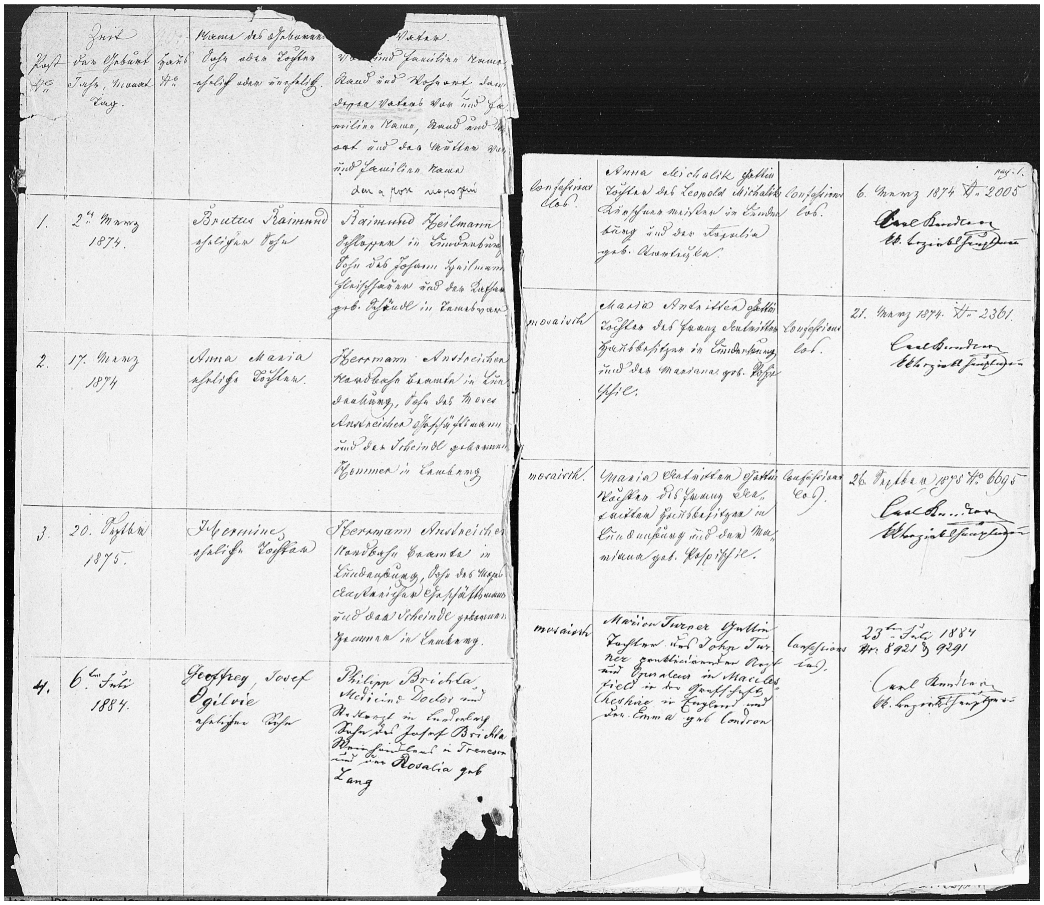
Vedení židovských matrik

Dne 27. ledna 1766 nařídila Marie Terezie vést matriky obřezaných mužů v synagogách. Pro dívky byly matriky zavedeny až roku 1783. V matričním patentu roku 1784 bylo nařízeno rabínům stejně jako katolickým farářům vést matriční záznamy pro porody sňatky a pohřby. Roku 1787 bylo Židům přikázáno, aby si zvolili stálé nezměnitelné příjmení. Až do roku 1869 zpracovávali židovské matriky katoličtí faráři, tuto praxi změnily až zákony z let 1868-1869, kdy byla židovským matrikám přiznána veřejná platnost [12, s. 52].

Vedení vojenských matrik

Nejstarší vojenská matrika byla dochována z let 1621-1622, přičemž první předpisy pro jejich vedení byly stanoveny za vlády Marie Terezie. Roku 1768 byla jejich správa nařízena vojenským duchovním. Roku 1816 pro ně byl vytvořen zvláštní formulář [12, s. 53].

Státní směrnice z 18. století, doplněné četnými předpisy v 19. století byly dlouho dobu hlavním smyslem matričních zákonů. Byly zde malé rozdíly, ale ty spočívaly spíše ve správních změnách. Velmi brzy po zavedení formulářů roku 1784 se začaly záznamy v matrikách průběžně číslovat i když to samotný patent nepředepisoval, avšak třeba v guberniálním nařízení z 1. června 1811 se s tímto faktem už předpokládalo. Definitivně upravily číslování až ministerské výnosy z let 1881-1882 [12, s. 51]. Zemský zákon pro Čechy 25. března 1850 stanovil, aby se místo dosavadního panství do matrik uváděl kraj, krajský úřad a okresní soud. Občanský zákoník z roku 1811 definuje právo manželské dědické a právo mezi rodiči a dětmi. Mezi lety 1855 až 1867 byly manželské záležitosti katolíků opět součástí duchovní jurisdikce. Až Říšský zákon číslo 47 z 25. května 1868 obnovil stav z let 1811-1855 občanského zákoníka a jeho vztahu i na katolíky. Církevní manželství se stalo nepovinné a věci manželské patřily opět před světské úřady. Roku 1870 bylo nařízeno okresním hejtmanstvím a obcím s vlastním statusem vést rejstříky lidí nepatřící do žádné státem uznávané církve [4, s. 49].



Obrázek 2.5: Ukázka civilní matriky Hodonín 1874[12]

Koncem 19. století na Pražském koncilu roku 1860 a pražské synodě roku 1873 byly vydány podrobné předpisy o způsobu vedení matrik. Definitivní sjednocení matričních formulářů bylo potom provedeno na konferenci zástupců pražské, litoměřické a královéhradecké konzistoře roku 1890 [4, s. 49].

Vedení matrik rukou církve pokračovalo i po rozpadu Rakouska-Uherska. Po 2. světové válce byl nejdříve vrácen původní stav vedení matrik církví a teprve 7. prosince 1949 byly s úpravou rodinného práva matriky převedené pod stát. Definitivně byl tento celorepublikový jednotný systém státních matrik zaveden 1. ledna 1950. Všechny církevní matriky se tak staly majetkem státu [12, s. 54]. Tyto matriky však byly i nadále uloženy na farních úřadech, ale výpisy z nich mohly poskytovat pouze příslušné matriční úřady. Teprve roku 1951 bylo vydáno nařízení, aby okresní matrikáři převzali do 15. dubna 1952 matriky od všech farních úřadů. Matriky do roku 1870 byly nadále uloženy u příslušného matričního úřadu, starší byly předány krajskému (později státnímu) archivu [13, s. 27]. Přijetím zákona č. 301/2000 Sb. se převádí zpráva matrik na matriční úřady. Tento zákon také určuje kdo může správcem matrik (matrikářem) být. Přijetím zákona o registrovaném partnerství č. 115/2006 Sb. byl změněn zákon č. 301/2000 Sb. a byla mezi matriční knihy přidána i kniha o registrovaném partnerství [23].

[12]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/10085?image=216000010-000253-003372-000000-006139-000000-FM-B10114-00020.jp2`

2.1.2 Živé a mrtvé matriky

To jestli je matrika považována za živou nebo mrtvou se odvíjí od doby posledního zápisu do matriky. V aktuálně platné zákoně je přesně stanovena jejich doba uchování na příslušném matričním úřadě.
Doba uchování od posledního zápisu je stanovena následovně:

Kniha narození po dobu 100 let.

Kniha manželství po dobu 75 let.

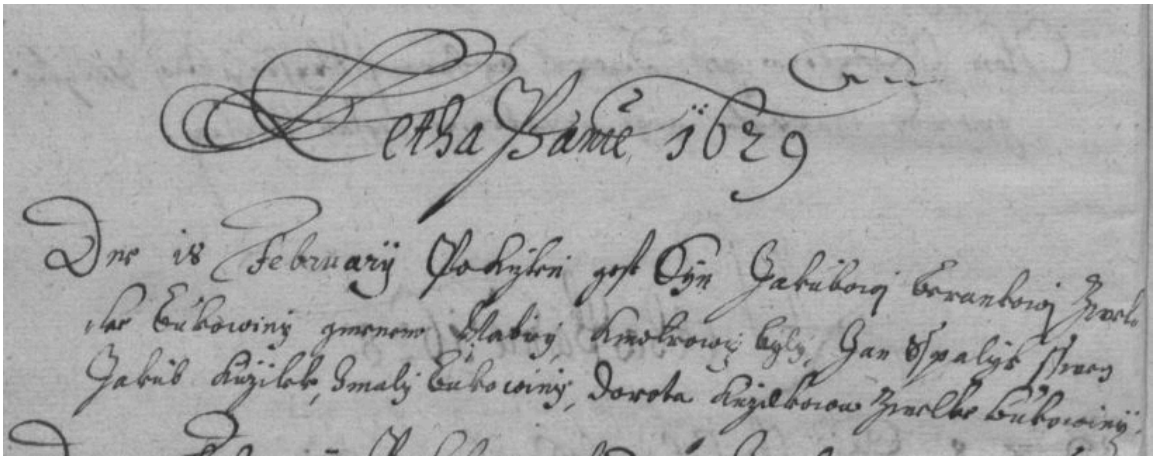
Kniha partnerství po dobu 75 let.

Kniha úmrtí po dobu 75 let.

Po uplynutí stanovené doby se matriční knihy předají k archivaci příslušnému státnímu oblastnímu archivu, není-li dále stanoveno jinak [23]. Během krátké doby je tato matrika přístupná online na stránce příslušného zemského archivu. Pokud se matrika stále považuje za živou, může matrikář podávat údaje o osobách, kterých se informace týká, jen jim samotným, jejich rodině nebo jejím zmocněncům. Dále pak může podávat informace pouze úřadům, které je potřebují pro uplatnění práv [23].

2.1.3 Kniha narození

Kniha narození (dříve křestní), byla jedna z prvních matrik, které byly založené se zakládáním far a sloužila především pro faráře k evidenci pokřtěných. Přestože první nařízení o vedení těchto matrik bylo už z let 1591 a 1605, první přesně předepsaná formule byla stanovena až pomocí Rituale Romanum[13]. Tento předpis uváděl, že by v křestních matrikách mělo být uvedeno: **datum narození a křtu, jméno kněze, jméno dítěte, jméno a farnost rodičů a kmotrů**.



Obrázek 2.6: V únoru roku 1629 byl Jakubovi Beránkovi pokřtěn syn Matěj v Bukovině[14]

Zápisy byly psány převážně latinsky a české znění znělo následovně:

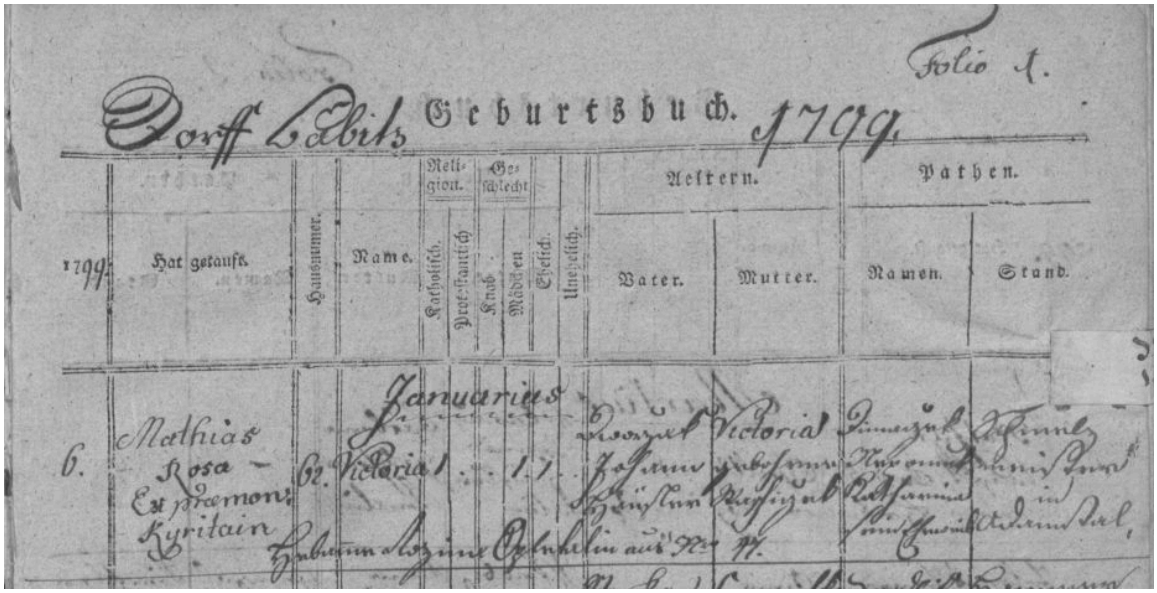
Léta Páně ... dne ... měsíce... já A., farář tohoto kostela sv. B. města (nebo obce) C., křtil jsem dítě, narozené dne ... D. a E. manželů této farnosti (nebo farnosti sv. F. a z takové rodiny), jemuž dáno jméno ... Kmotři byli G., syn H. z farnosti (nebo místa) J., a K. manželka L., dcera M. z farnosti (nebo obce) N.

I když byla tato forma zápisu předepsána, vedení matrik hodně ovlivňoval jedinec, který tuto matriku spravoval. Většinou tak obsahem křestních matrik bylo pouze: **datum křtu (ne však narození), jméno křtěnce, jméno a bydliště rodičů a kmotrů**. Až roku 1760 byla přímo zavedena latinská formule vycházející z Rituale Romanum. Tento způsob zápisu však netrval dlouho, protože už 6. října 1770 byl dvorským dekretem vyhlášen nový způsob zápisu do matričních formulářů. Tento zápis nahradil dřívější větný zápis do rubrik. Roku 1771 také začala arcibiskupská tiskárna vydávat tištěné formuláře, které začaly postupně nahrazovat ty ručně psané. Matriky narození se rozdělovaly na tyto rubriky: **den narození a křtu, kněz, dítě, rodiče a kmotrové, náboženství, místo a číslo domu.**

Patent Josefa II. ze dne 20. února 1784 vydal nový důkladnější patent jehož rubriky obsahovaly tyto informace: **datum narození, domovní číslo, křestní jméno dítěte, náboženství, pohlaví, původ, křestní jméno a příjmení rodičů a kmotrů.** Kmotr se musel ještě vlastnoručně podepsat (nebo ověřit znamením podpis provedený cizí rukou). S tímto patentem skončil také latinský zápis do matrik. Někteří kněží tak přešli rovnou do zápisu českého a někteří (nejčastěji matrikáři z německého pohraničí) zapisovali matriky německy.

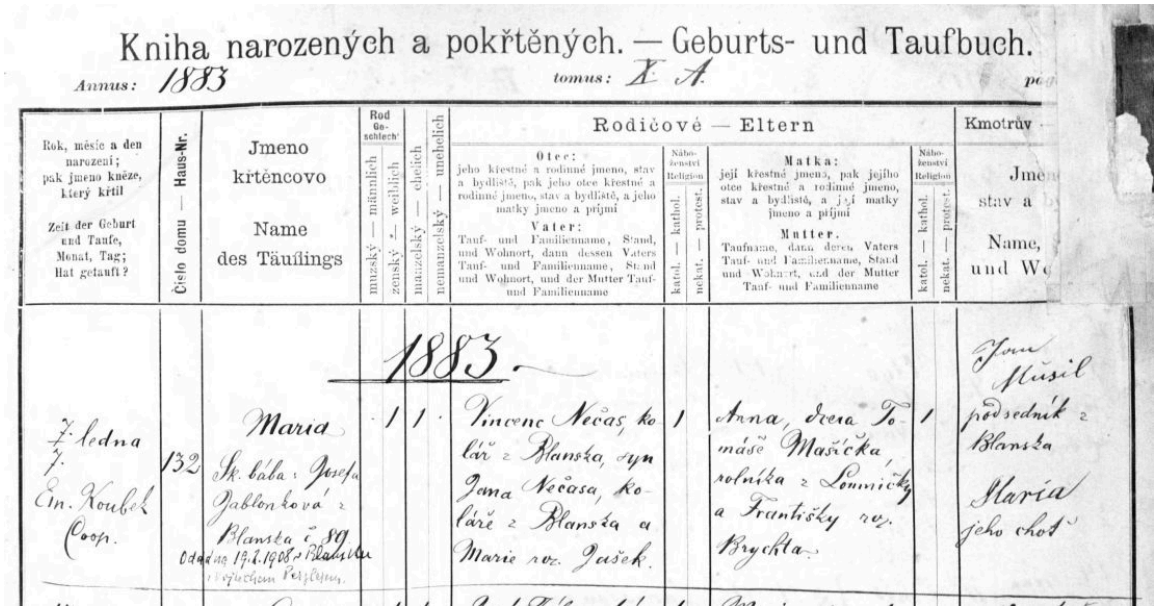
[13] Římský rituál - Liturgická kniha obsahující bohoslužebné předpisy a texty římskokatolické vydaná papežem roku 1614.

[14] Zdroj: `https://www.mza.cz/actapublica/matrika/detail/3282?image=216000010-000253-003368-000000-000414-000000-00-B02134-00090.jp2`



Obrázek 2.7: Dne 6. ledna 1799 byla v Babicích nad Svitavou pokřtěna Victoria a její otec je Johan Dvořák (Dworžak)[16]

Na přelomu 18. a 19. století ze začali do křestní matrik zapisovat i prarodiče. Dvorský dekret roku 1825 poté dodal do oddacích matrik ještě informaci o **jméně porodní báby.**



Obrázek 2.8: Údaj z matriky narození Blansko 1883 s rubrikami v češtině[17]

V roce 1949 byly zákonem č. 268 navržený nový formulář který obsahoval:

[16]Zdroj: https://www.mza.cz/actapublica/matrika/detail/3758?image=216000010-000253-003369-000000-000998-000000-00-B08794-00010.jp2

[17]Zdroj: https://www.mza.cz/actapublica/matrika/detail/15?image=216000010-000253-003368-000000-000055-000000-00-B00829-00010.jp2

(a) jméno, příjmení a pohlaví dítěte,

(b) den, měsíc, rok a místo narození dítěte, jakož i časové pořadí narození, jde-li o dítě narozené spolu s druhým (dvojčata) nebo s více dětmi,

(c) státní občanství dítěte,

(d) jméno a příjmení, den, měsíc, rok, místo narození, povolání a bydliště rodičů, jakož i jména a příjmení prarodičů,

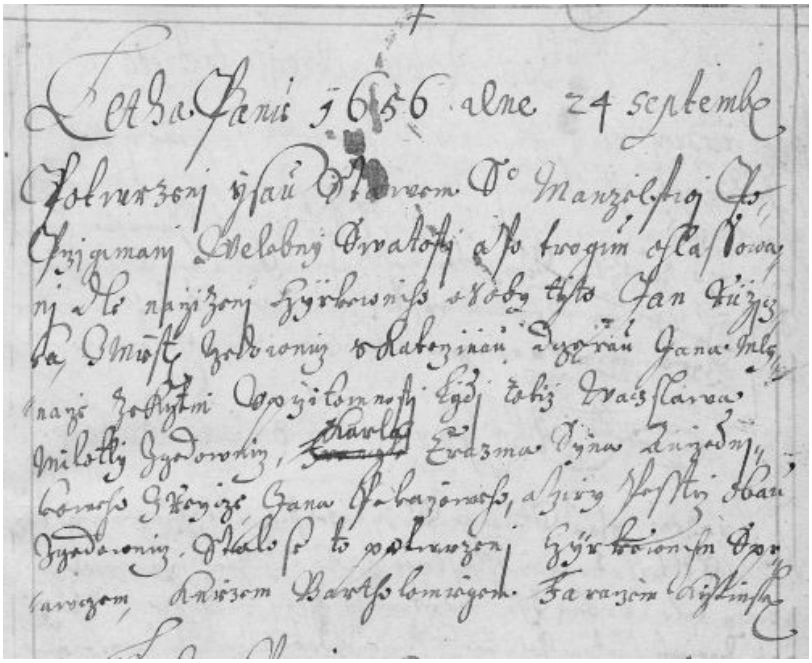
(e) dohoda rodičů o příjmení jejich dítěte, mají-li rodiče různá příjmení,

(f) datum zápisu.

V roce 2000 byl vydán zatím poslední zákon upravující matriční agendu. Oproti zákonu z roku 1949 doplnil tyto informace: **rodné číslo dítěte, místo narození dítěte i rodičů a podpis matrikáře**

2.1.4 Kniha manželství

Kniha manželství (dříve oddací), byla stejně jako křestní matrika jedna z nejdříve zakládaných matrik. Na našem území se dokonce jako vůbec první zachovaná matrika, dochovala oddací matrika z Jáchymovska založená roku 1531. I když bylo vedení matrik v našich zemích nařízeno až olomouckou a pražskou synodou v letech 1591 a 1605 první nařízená forma zápisu byla zavedena až později a to přesnými formulemi Rituale Romanum[18]. Podle toho mělo být v oddacích matrikách uvedeno: **datum ohlášek, existence překážek, datum sňatku, jméno oddávajícího, jméno, bydliště a farnost snoubenců (u vdov jméno bývalého manžela) a svědků**.



Obrázek 2.9: Dne 24. září 1656 se oddali Jan Růžička s Kateřinou dcerou Jana Mlynáře ze Křtin[20]

Zápis měl probíhat v latině, ale někde se matriky psaly i Německy nebo česky. České znění znělo následovně:

Roku ... dne ... měsíce, po třech předchozích ohláškách, o třech následných svátcích, z nichž prvá byla dne.., druhá dne.., třetí dne.., při svátečních bohoslužbách, když se nevyskytla žádná zákonitá překážka, já A., správce tohoto farního chrámu B. města (nebo obce) C., mládence D. z farnosti sv. E., a panny F. (nebo vdovy po kdysi G., když byla vdovou) z této (nebo sv. H.) farnosti, v kostele B. jsem se otázal a maje jejich vzájemný souhlas, slavnostně slovy v přítomném manželství jsem spojil; v přítomnosti svědků J., syna K., jenž bydlí ve farnosti sv. H., a L. syna M. etc. Posléze jsem jim podle obřadu sv. Matky církve (pouze když sňatku požehnal) při bohoslužbě požehnal.

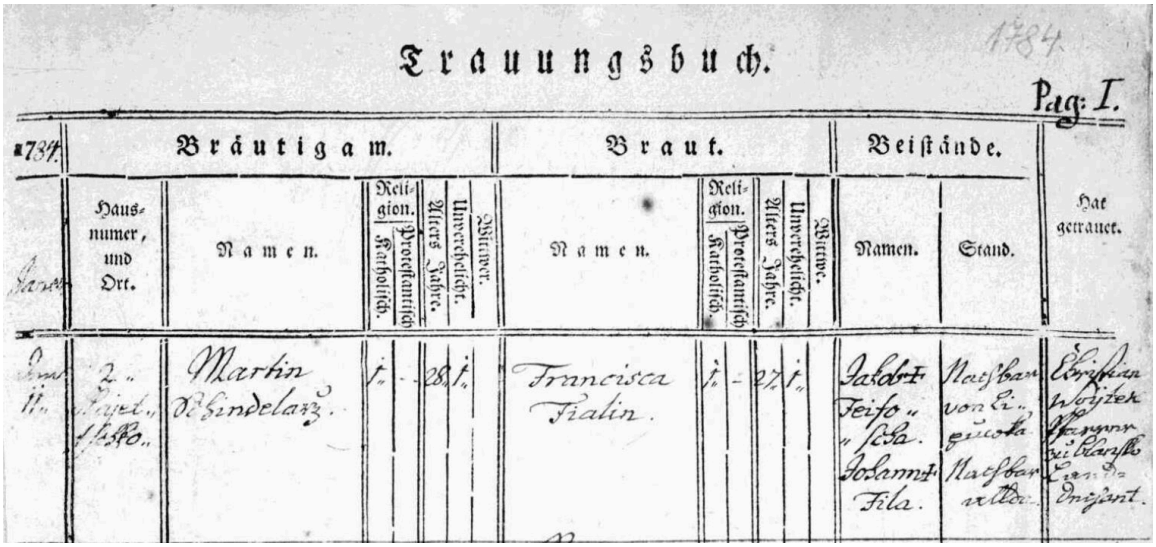
Podle této formy se bohužel neřídilo všude a od mnohých údajů se upouštělo. V křestních matrikách se proto většinou uvádělo jen: **jméno a bydliště ženicha,** u mladších

[18]Římský rituál - Liturgická kniha obsahující bohoslužebné předpisy a texty římskokatolické vydaná papežem roku 1614.

[20]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/3290?image=216000010-000253-003368-000000-000429-000000-00-B08672-00010.jp2`

jméno otce, u starších **zaměstnání, dále jméno bydliště a rodinný stav nevěsty, jméno otce nebo zemřelého manžela, jméno a bydliště svědků**. Mnohdy však byl ženich nebo nevěsta uvedeny jen křestním jménem, nebo nebyl jeden z nich uveden vůbec. Poměrně zřídka se také v 17. století začíná v matrikách objevovat záznam o vrchnostenském povolení ke sňatku. Znovu a závazně přikázala formu zápisu podle Rituale Romanum až pražská konzistoř roku 1760. Už 6. října 1770 byl však vyhlášen zápis do formulářů podle rubrik místo dřívějšího větného. Nejdříve využívali kněží ručně psané formuláře, ale již roku 1771 začala vydávat arcibiskupská tiskárna formuláře tištěné. U oddacích matrik byly tyto rubriky: **den, oddávající, snoubenci, svědkové, náboženství, místo a číslo domu.**

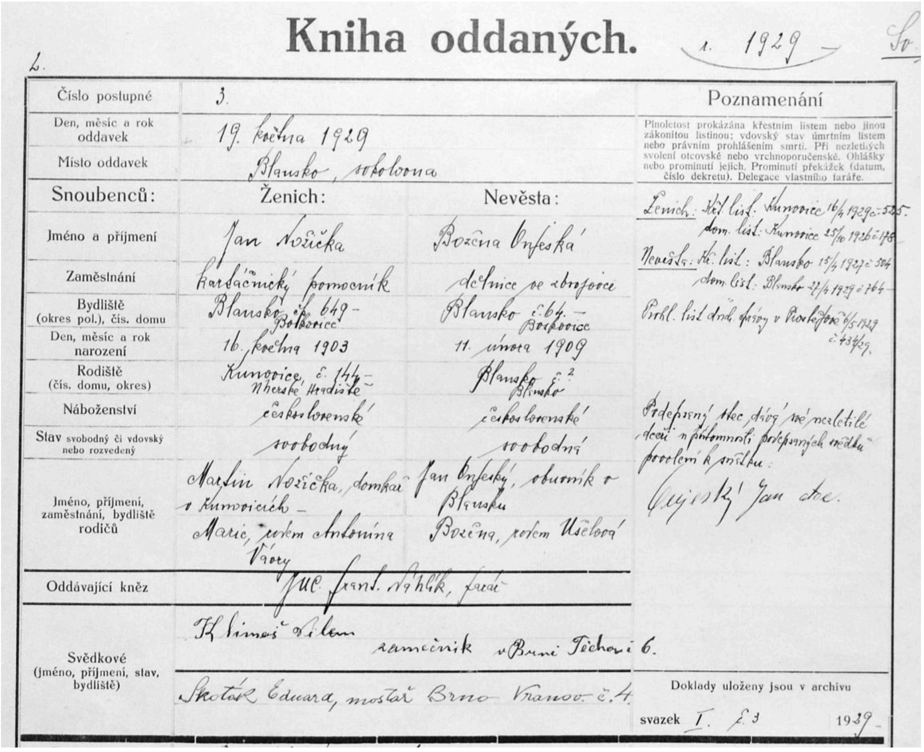
Dne 20. února 1784 vydal Josef II patent, který vytvořil mnohem důkladnější formulář který sloužil s menšími úpravami až do roku 1949. Obsahoval tyto informace: **datum, číslo domu, křestní jméno, příjmení ženicha, jeho náboženství a stáří, rodinný stav, to samé nevěsta, křestní jméno, příjmení a stav svědků.** Svědkové museli připojit svůj podpis. S patentem Josefa II. skončil také latinský zápis. Někteří kněží přešli rovnou na zápis český. Jiní (nejčastěji matrikáři z německého pohraničí) zapisovali matriky německy.



Obrázek 2.10: Dne 11. ledna byl oddán Martin Šindelář s Francisca (Františka) Fialin (Fialová) v Blansku[22]

Na přelomu 18. a 19. století se do matrik oddacích začali zapisovat i prarodiče.

[22]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/2813?image=216000010-000253-003368-000000-000077-000000-AP-B11561-00020.jp2`



Obrázek 2.11: Údaj z matriky oddací Blansko 1929[23]

Zákon z roku 1949 definoval formulář takto:

(a) jméno a příjmení, den, měsíc, rok a místo narození, státní občanství, stav, povolání a bydliště snoubenců,

(b) den, měsíc a rok uzavření manželství,

(c) jméno a příjmení, den, měsíc, rok, místo narození a povolání rodičů snoubenců,

(d) jméno a příjmení, povolání a bydliště svědků,

(e) příjmení, o němž se snoubenci dohodli, že ho budou užívat,

(f) příjmení, o němž se snoubenci, kteří si ponechali svá dosavadní příjmení, dohodli pro děti, které z manželství vzejdou,

(g) datum zápisu.

V roce 2000 byl vydán zatím poslední zákon upravující matriční agendu. Oproti zákonu z roku 1949 doplnil tyto informace: **místo narození a rodné číslo manželů, místo uzavření sňatku, místo narození rodičů manželů, rodná čísla svědků, podpis matrikáře**

[23]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/11364?image=216000010-000253-003368-000000-000989-000000-00-B02189-00020.jp2`

2.1.5 Kniha registrovaného partnerství

Matriční kniha registrovaného partnerství vznikla v roce 2006 společně se zákonem č. 115/2006, který jeho zněním upravoval zákon o matrikách č. 301/2000. V knize partnerství (takto se kniha pro zaznamenávání vstupu do registrovaného partnerství nazývá) se vyskytují tyto informace:

(a) jména a příjmení, popřípadě rodná příjmení, den, měsíc, rok a místo narození, rodná čísla, osobní stav a státní občanství partnerů,

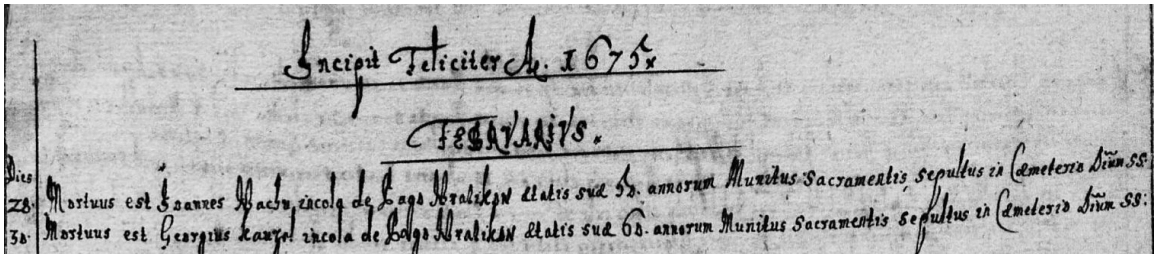
(b) den, měsíc, rok a místo prohlášení o vstupu do partnerství,

(c) jména a příjmení, popřípadě rodná příjmení, den, měsíc, rok a místo narození rodičů partnerů,

(d) datum zápisu a podpis matrikáře.

2.1.6 Kniha úmrtí

Kniha úmrtí (dříve zemřelých) byla u nás na rozdíl od matrik křestních a oddacích nařízena až samotným Rituále Romanum roku 1614. Jejím obsahem měl být: **den úmrtí, jméno zemřelého a jeho otce, bydliště, věk, den pohřbu, pohřební místo, jméno kněze, datum zpovědi, přijetí nejsvětější svátosti a poslední pomazání**.



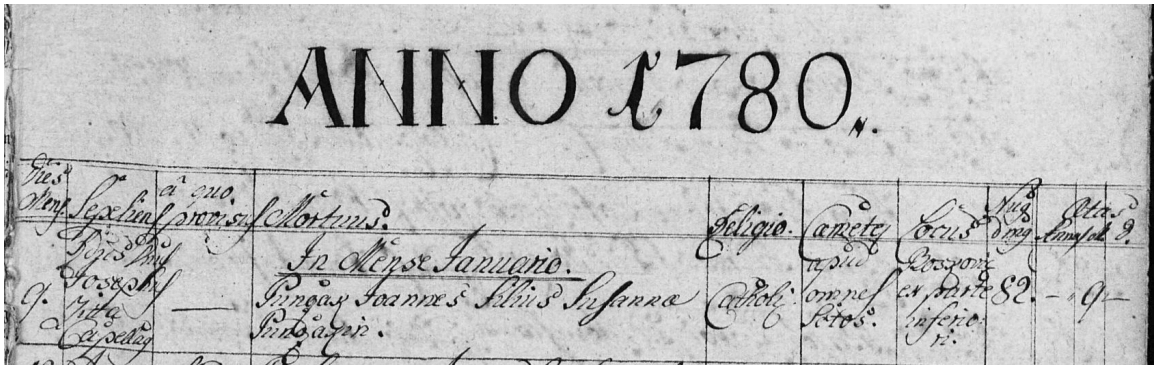
Obrázek 2.12: Údaj z matriky zemřelých Boskovice 1675[24]

Forma byla předepsaná v latině, ale matriky se psaly i v němčině nebo češtině. Její český předpis zněl takto:

Roku ... dne ... měsíce ... A., syn (či dcera) B., z obce C., věku... (možno-li to zvědět) v domě ... ve společenství sv. Matky církve duši Bohu odevzdal; jehož tělo bylo pohřbeno dne... v kostele sv. D., ode mne E. (nebo F. zpovědníka) vyzpovídán dne..., svatými svátostmi zaopatřen dne.., pomazáním svatého oleje posílen mnou dne..

Tato forma však nebyla všude dodržována a dost informací se vůbec nezapisovalo. Úmrtní matriky byly ze všech ostatních matrik nejvíce zanedbávané. Nejčastěji zde bylo uvedeno jen: **den pohřbu a jméno a bydliště zemřelého**. Během druhé poloviny 17. století se začal do matriky uvádět i úmrtní věk. Už v 17. století se v některých matrikách evidují i příčiny úmrtí. Většinou se tato informace zapisovala při velkých epidemiích. Jinak byl tento zápis spíše výjimkou. Důležitý příkaz o zavedení formulí podle Rituale Romanum byl vyhlášen roku 1760. 6. října 1770 však byl vyhlášen nový způsob zápisu, předepsaný formuláři s využitím rubrik. Od roku 1771 taky postupně nahrazovaly tištěné formuláře ty ručně psané. V úmrtních matrikách nacházíme tyto rubriky: **datum (většinou zlomek den úmrtí/pohřeb), pohřbívající, zaopatření, zemřelý, náboženství, hřbitov, místo, číslo domu a věk.** Josef II. vydal dne 20. února 1784 patent který ještě více specifikoval formulář matrik a tento formulář vydržel s menšími úpravami až do roku 1949. Jeho obsahem byly tyto informace: **datum úmrtí, číslo domu, jméno, náboženství, stáří zemřelého a příčina úmrtí.** Tímto patentem skončil také latinský zápis do matrik. Dále se do nich zapisovalo už pouze česky, nebo na některých místech (většinou u pohraničí) se zapisovaly matriky německy.

[24]Zdroj: `https://www.mza.cz/actapublica/matrika/detail/2973?image=216000010-000253-003368-000000-000191-000000-00-B01820-00010.jp2`



Obrázek 2.13: Údaj z matriky zemřelých Boskovice 1780 s ručně psanými rubrikami[25]

Obrázek 2.14: Údaj z matriky zemřelých Boskovice 1940 psaný česky[26]

V novém formuláři z roku 1949, který byl definován zákonem č. 268, byly tyto informace:

(a) jméno a příjmení, povolání, stav, bydliště, den, měsíc, rok a místo narození zemřelého,

(b) hodina, den, měsíc, rok a místo úmrtí,

(c) jméno a příjmení manžela,

(d) jméno a příjmení rodičů zemřelého,

(e) příčina smrti,

(f) datum a místo pohřbení,

[25] Zdroj: https://www.mza.cz/actapublica/matrika/detail/2974?image=216000010-000253-003368-000000-000192-000000-00-B00064-00020.jp2

[26] Zdroj: https://www.mza.cz/actapublica/matrika/detail/11421?image=216000010-000253-003368-000000-000196-000000-VR-B08267-02460.jp2

(g) datum zápisu.

V roce 2000 byl vydán zatím poslední zákon upravující matriční agendu. Oproti zákonu z roku 1949 doplnil tyto informace: **rodné číslo dítěte, místo narození dítěte i rodičů a podpis matrikáře**

2.2 OCR

OCR - optical character recognition (optické rozpoznávání znaků) je zkratka pro metodu počítačového rozpoznávání písmen z obrazové předlohy a také pro program, který tuto činnost provádí [7, s. 168]. Výzkumu a vývoji OCR se v poslední době věnuje velký zřetel, z důvodu digitalizace ve většině odvětvích.

2.2.1 Využití OCR

Jak už bylo dříve řečeno, OCR se využívá k digitalizaci starých tištěných nebo ručně psaných dokumentů jako jsou například: faktury, formuláře, právní dokumenty, tištěné smlouvy atd. Výhoda OCR oproti jiným způsobům digitalizace dokumentů (např. skenování) je, že výstupem programu OCR je textový soubor, který můžeme dále editovat.

2.3 Pero OCR

Pero OCR je balíček, který poskytuje komplexní OCR pipeline[27] obsahující:

- detekce odstavců
- detekce řádků
- přepis textu
- upřesnění textu podle jazykového modelu

Balíček je vyvinutý v jazyce Python a lze využít buď jako aplikaci příkazového řádku, nebo jako Python knihovnu.

Jako knihovna obsahuje balíček dvě třídy. Jedna z nich je třída pro zpracování dokumentu a druhá slouží k procházení obsahu stránky dokumentu.

2.3.1 Analýza struktury textu

Základem je CNN model[28] pro společnou detekci: základních linií textu, polygonů textových čar a textových bloků z široké škály tištěných a ručně psaných dokumentů. Systém také zpracovává dokumenty s libovolnými směry textu s kombinací detekcí rozvržení ve více orientacích pomocí specializovaného modelu odhadu husté orientace textu [10].

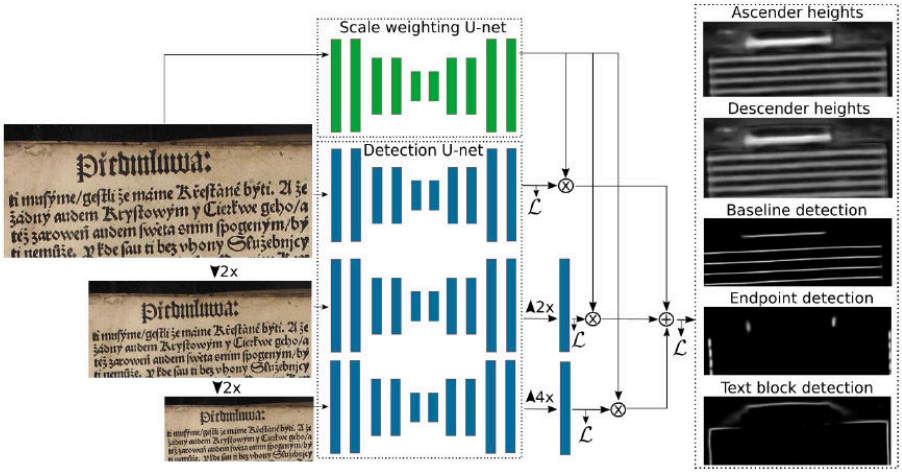
Detekce řádků a bloků textu

Metoda detekce textových řádů a bloků se skládá z plně konvexní neuronové sítě (ParseNet), doplněné sadou kroků následného zpracování, která určí jednotlivé linie textu a ohraničující polygony textových řádků přímo z výstupu sítě. Následně se polygony řádků prochází zdola nahoru a shlukují se do textových bloků pomocí pravděpodobnosti ohraničení bloku. [10] Navrhnuta síť ParseNet je inspirována sítí ARU.[29]

[27] pipeline - výpočetní postup při které se jedna nebo více datových sad upravuje podle řady chronologických kroků [5]

[28] Convolutional neural network (Konvoluční neuronová síť) - obrázek se čte postupně po velmi malých částech a aplikuje se na něj stále stejná operace [16]

[29] ARU - ARU-Net je neuronová síť na značkování pixelů pro analýzu rozložení historických dokumentů [6]



Obrázek 2.15: Architektura navrhnutého modelu vychází především z modelu ARU-Net [6], doplněný o přímou tvorbu konečného výstupu pomocí váženého průměrování a výpočet ztrát na každém výstupu stupnice. Jak můžete vidět výstupem jsou dva kanály pro detekci výšky textu, kanály pro detekci základní linie a koncového bodu a kanál pro detekci hranice textového bloku[31]

Detekce textu s různou orientací

Jelikož by systém ParseNet tak, jak je popsaný v předchozím odstavci nedokázal správně zpracovat dokumenty obsahující jak vertikální, tak horizontální text, byl proto návrh obohacen o další samostatný model, který odhaduje lokální orientace textu a podle nich sloučí text do bloků. [10]

Obrázek 2.16: Příklady řádků textu v různých datových sadách[32]

2.3.2 Datové sady

Jelikož je PeroOCR určené jak pro tištěné, tak i pro ručně psané texty, je potřeba použít datové sady pro obě tyto odvětví.

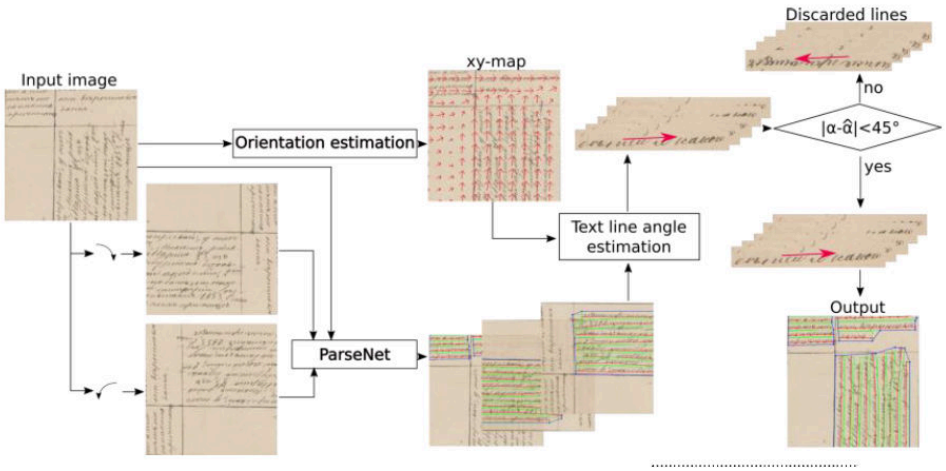
[31] Obrázek byl převzat z práce: Page Layout Analysis System for Unconstrained Historic Documents [10]
[32] Obrázek byl převzat z práce: Page Layout Analysis System for Unconstrained Historic Documents [10]

Ručně psané datové sady

Pro strojové učení OCR s ručně psanými daty byla použita datová sada ICDAR 2017 READ dataset [18], datová sada ICFHR 2014 Bentham Dataset [17] a další neanotované stránky získané z projektu Bentham[33]. Dataset READ obsahuje stránky psané v němčině, italštině a francouzštině. Dataset Bentham a neanotovaný dataset Bentham se ale skládají ze stránek psaných v angličtině na počátku 19. století. Dataset Bentham má ještě jeden unikát, na rozdíl od ostatních datasetů je psaný převážně samotným Benthamem a jeho sekretářkou. [9]

Tištěné datové sady

Pro strojové učení OCR pro účely zpracování tištěného textu byla použita datová sada IMPACT [15] jako související doménová data. V této datové sadě se nejčastěji vyskytuje španělština, angličtina a nizozemština. Jako cílová doména byly použity historické české noviny, z nichž 2000 stran bylo částečně popsáno dobrovolníky. Tyto data tedy byla použita jako anotovaná data cílové domény. [9]



Obrázek 2.17: Obrázek navrženého modelu pro extrakci více orientovaného textu. Vstupní obrázek je zpracován pomocí sítě ParseNet ve třech různých orientacích a následně jsou vyloučeny texty které se svým natočením liší o více než 45°[35]

2.3.3 Trénink OCR

Na vstup trénování OCR byla přivedena tato data:

1. data příbuzné domény (volně odpovídají stylu, jazyku nebo celkovému stavu cílové domény).
2. anotovaná data cílové domény (lidské anotace).
3. neanotovaná data cílové domény (chceme využít bez dalšího lidského vstupu).

[33] více o projektu Bentham: https://www.ucl.ac.uk/bentham-project

[35] Obrázek byl převzat z práce: Self-Training Adaptation Strategy for OCR in Domains with Limited Transcriptions [9]

Samotný proces tréningu probíhá následovně:

1. Na výchozích datech je vycvičen systém OCR.
2. Stávající systém OCR je použit pro zpracování všech neanotovaných řádků cílové domény.
3. Pro každý z takto zpracovaných řádků je vypočítáno skóre důvěryhodnosti, pomocí vhodné míry důvěryhodnosti.
4. Data označené za důvěryhodné přidáme ke strojově anotovaným datům (MA).
5. Data MA jsou sloučena se vstupními daty a je na nich trénováno nové OCR.

Tyto kroky se poté mohou pro předem stanovenou dobu opakovat. [9]

2.4 Sémantická analýza

Sémantická analýza patří mezi základní lingvistické metody. Slouží pro rozbor názvosloví a obsahové struktury textu. Sémantická analýza také patří mezi základní dílčí úkoly zpracování přirozeného jazyka (NLP - Natural Language Processing), který se hojně využívá třeba pro chat-boty a nebo vyhledávače [21]. V této práci bude sémantická analýza používána pro analýzu textu a získání potřebných informací z něj.

2.4.1 Části sémantické analýzy

Sémantická analýza přirozeného jazyka může být rozdělena do dvou částí [20] :

1. **Lexikálně sémantická analýza:** zahrnuje pochopení významu každého slova v textu jednotlivě. V podstatě dává stroji instrukci jakou informaci má v textu nést.
2. **Kompoziční sémantická analýza:** zahrnuje pochopení významu vět tvořených kombinací jednotlivých slov.

2.4.2 Kritické části

Mezi kritické části sémantické analýzy patří tyto duhy slov [11] :

- **homonyma** - stejná slova se rozdílným významem (podepřít - zapřít, bábovka ž. - bábovka m.)
- **synonyma** - různá slova stejného nebo podobného významu (dopis - psaní)
- **polysémie** - slova mnohovýznamná víceznačná (koruna, hlava)
- **antonyma** - slova s opačným významem (dobro - zlo)
- **hyponyma** - slovo významově podřazené slovu nadřazenému (člověk - bytost)

Tyto kritické části je potřeba v sémantické analýze identifikovat, aby byl stroj schopen pochopit kontext jakékoliv věty nebo odstavce.

2.4.3 Techniky sémantické analýzy

Sémantická analýza má dvě základní techniky, ze kterých si můžeme vybrat v závislosti na typu informací které zkoumáme. Jedná se o **model klasifikace textu** a **model extrakce textu** [21] .

Modely klasifikace textu

- **Tématická klasifikace:** třídění na základě obsahu do předem definovaných kategorií. Například eshop si může třídit příchozí emaily podle různých kategorií (reklamace, doprava, platba atd.).
- **Analýza sentimentu:** detekce pozitivních, negativních nebo neutrálních emocí v textu. Například slouží značkám k zjištění zpětné vazby například ze sociálních sítí.
- **Klasifikace záměrů:** klasifikace textu na základě toho co chce jeho pisatel provést dále. Slouží například k zjištění zájmu z emailu a jejich následné rozdělení na „Má zájem“ a „Nemá zájem“.

Modely extrakce textu

- **Extrakce klíčových slov:** vyhledání relevantních slov a výrazů z textu. Slouží například k vytažení nejčastěji používaných slov u recenzí, které byly zařazené do kategorie „Negativní“.
- **Extrakce entit:** identifikace pojmenovaných entit v textu, jako jsou jména osob firem, míst atd.

2.5 Detekce jmenných entit - NER

Detekce jmenných entit je jedna z dříve jmenovaných technik sémantické analýzy textu. Tato technika se zabývá rozeznáním jmenných entit v psaném textu jako jsou třeba: jména, místa nebo časové záznamy. Strojové učení takto zaměřeného modelu probíhá pomocí korpusů.

2.5.1 Jazykový korpus

Jazykový korpus (dále už jen korpus) je vnitřně strukturovaný, unifikovaný a obvykle i oindexovaný a ucelený rozsáhlý soubor elektronicky uložených a zpracovaných jazykových dat většinou v textové podobě, organizovaný se zřetelem k využití pro určitý cíl, vůči němuž je pak také považován za reprezentativní. Každý soubor textů v počítači se však nepovažuje hned za korpus. Mezi korpusy nepatří: volné kolekce textů, elektronické knihovny, nebo souhrnné elektronické archivy. Nejčastěji se vybírá určité odvětví, na které je poté daný korpus specializovaný. [1, s. 18]

2.5.2 Czech Named Entity Corpus

Czech Named Entity Corpus (dále CNEC) je korpus zabývající se identifikací vlastních jmen, které jsou řazeny do předem definovaných kategorií, jako jsou jména osob, zeměpisná jména, názvy organizací atd. Korpus CNEC je přímo motivovaný potřebami aplikací pro zpracovaní přirozeného jazyka (NLP), proto je jako v ostatních úlohách NLP potřeba při

vývoji takto cíleného modelu používat anotovaná data. CNEC 1.0 byl také prvním českým veřejně vydaným korpusem vůbec a doteď není přístupný žádný konkurenční produkt, kromě vylepšených verzí sebe samého. Nejnovější verze korpusu je verze 2.0.

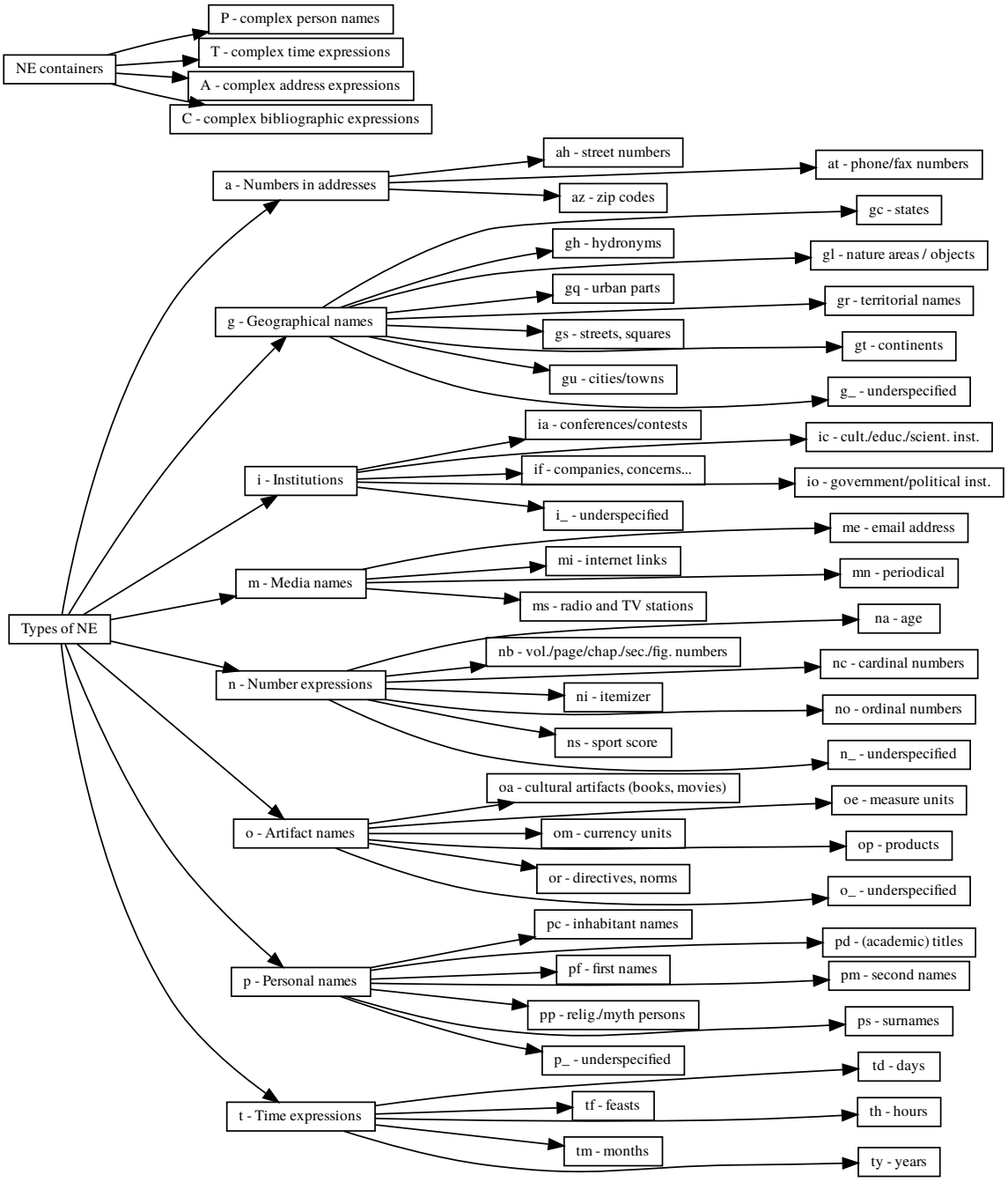
Datové formáty korpusu

Korpus je uložený v následujících formátech:

- **prostý text**
- **simple xml**
- **treex** - formát xml z Treexu s morfologickou analýzou
- **html** - html se zvýrazněnými pojmenovanými entitami.

Klasifikace textu

Klasifikace vlastních jmen v češtině se provádělo pomocí dvoustupňové hierarchie [19].



Obrázek 2.18: Obrázek výsledné hierarchie klasifikace upravené verzí 2.0[36]

[36]Zdroj: `https://ufal.mff.cuni.cz/~strakova/cnec2.0/ne-type-hierarchy.pdf`

2.6 SpaCy

SpaCy je bezplatná open-source k knihovna pro pokročilé zpracování přirozeného jazyka (NLP) v jazyce Python. Pomáhá vytvářet aplikace, které zpracovávají a „rozumí“ velkým objemům textu. Lze ji použít jak k vytváření systémů pro extrakci informací, tak pro porozumění přirozenému jazyku. Díky spoustě funkcí a hlavně vysoké rychlosti vytvořených modelů patří knihovna SpaCy mezi jedny z nejlepších volně dostupných knihoven pro NLP.

2.6.1 Funkce

Tokenizace: Segmentace textu na slova interpunkční znaménka atd.

Tagování (Part-of-Spech, POS): Přiřazení slovních slovních druhů k tokenům (podstatná jména, slovesa atd.).

Analýza závislostí: Přiřazení syntaktických závislostních značek (podmět, přísudek atd.).

Lemmatizace: Hledá základní tvar slova(lemma, myši - myš, dělání - dělat).

Detekce hranic věty - SBD

Rozpoznávání pojmenovaných entit - NER

Propojování entit: Propojování entit na sebe závislé v rámci textu (Paříž je krásné město. Sousední zemí Belgie je Francie. Paříž - hlavní město Francie).

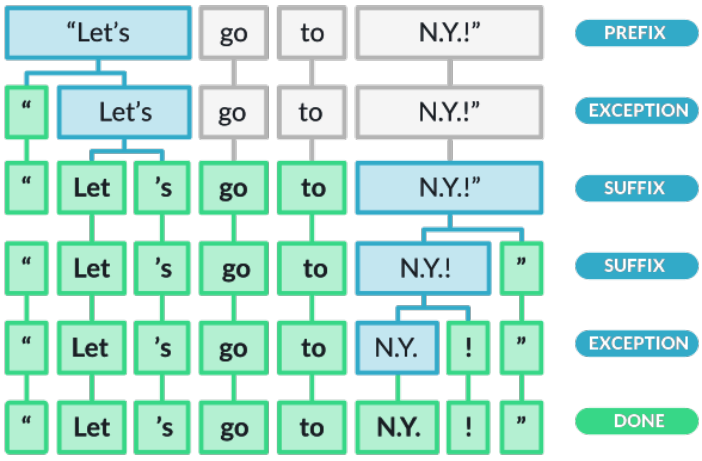
Podobnost: Porovnávání různých částí textu a sledování jejich podobností.

Klasifikace textu: Přiřazení kategorií celému textu nebo jeho částem.

Shoda založena na pravidlech: Hledání stejného textu, nebo jeho části v závislosti na zadaná pravidla.

Tréning a vylepšování stávajícího statistického modelu

Serializace: Ukládání objektů do souborů nebo bitového zápisu.



Obrázek 2.19: Ukázka příkladu tokenizace[37]

[37] Zdroj: `https://spacy.io/usage/spacy-101`

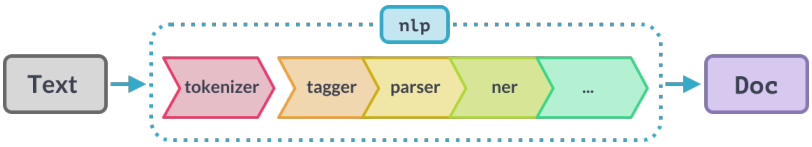
2.6.2 Statistické modely

Některé funkce SpaCy fungují samostatně, ale většina potřebuje k jejich práci načtení vytrénovaných modelů, které SpaCy poskytnou informace pro předpovídání jazykových anotací (NER, tokenizace, tagování atd.). Taková natrénovaná pipeline se může skládat z více komponent, které využívají statistický model natrénovaný na označených datech. Balíčky pipeline se mohou lišit rychlostí, využitím paměti, přesností a daty, která obsahují. Takové balíčky obvykle obsahují následující součásti:

- **Binární váhy** pro POS a NER. Předpovídá přiřazení příslušných anotací v kontextu.
- **Lexikální položky ve slovníku** to jsou slova a jejich atributy nezávisle na kontextu (pravopis).
- **Datové soubory** jako jsou pravidla lemmatizace a vyhledávání.
- **Slovní vektory**, jde o vícerozměrné významové reprezentace slov, které umožňují určit, jak jsou si slova podobná.
- **Kongifurace** a její výběr, jde o nastavení jazyka zpracování pipeline a implementace modelu, které se mají použít na stanovený text.

2.6.3 Pipeline (potrubí)

Když se zavolá NLP na nějaký text, následuje posloupnost kroků který se na tento text aplikují. Obvykle pipeline obsahuje: **tagger, lemmatizátor, parse a NER**. Každá komponenta vrací jinak zpracovaný komponent Doc.



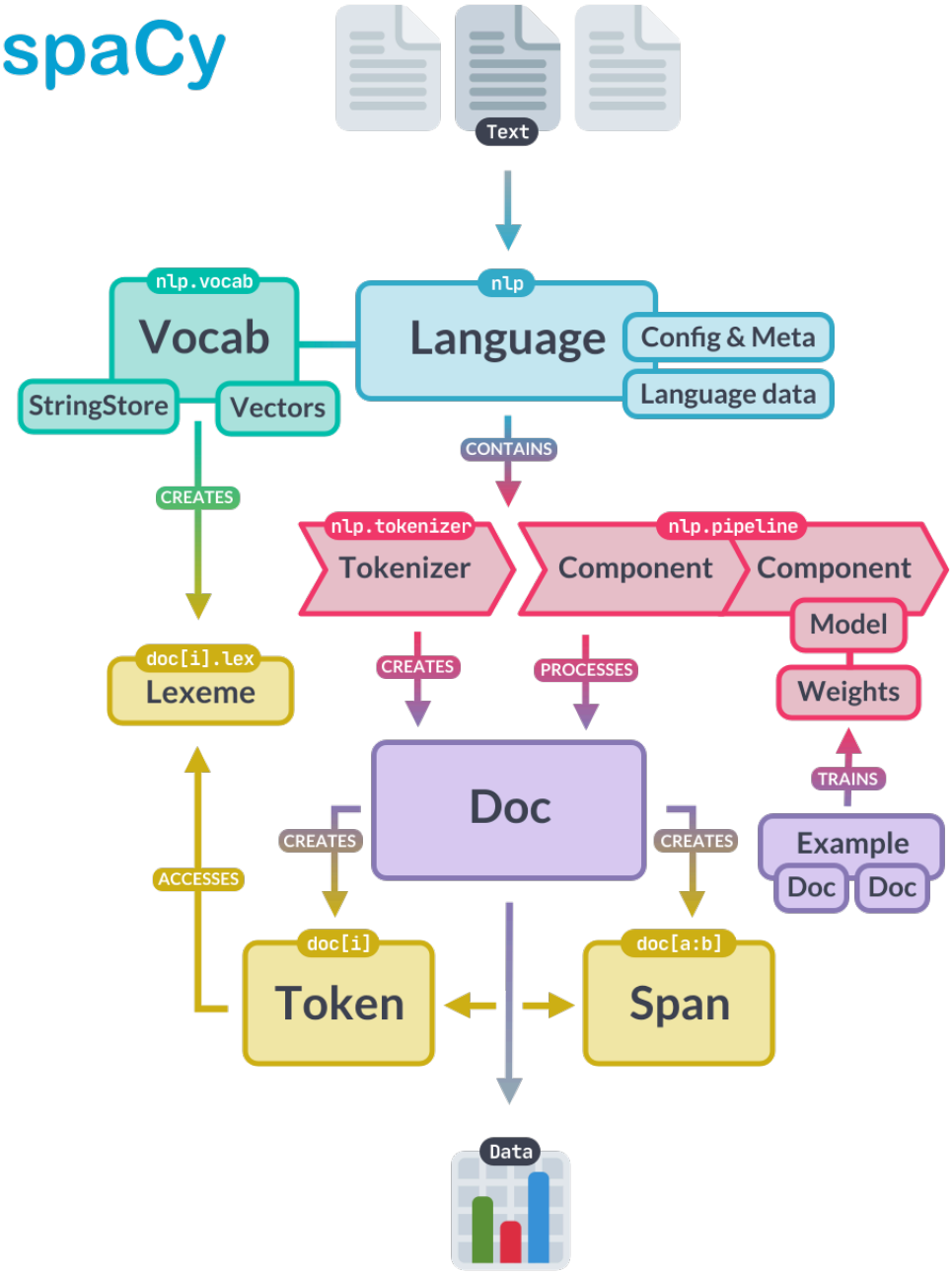
Obrázek 2.20: Pipeline pro zpracování NLP[38]

2.6.4 Architektura

Hlavními datovými strukturami ve SpaCy jsou třída Language, Vocab a objekt Doc. Třída Language je většinou uložena jako proměnná nlp a slouží ke zpracování textu na objekt Doc. Objekt doc obsahuje vlastní posloupnost tokenů a všechny jejich anotace. V objektu Vocab je zase uložena kopie dat centralizace vektorů slovních vektorů a lexikálních atributů. Textové anotace jsou navrženy tak, aby maximálně šetřily paměť: Doc obsahuje data a Span a Token ukazují na tyto data.

Doc je konstruován Tokenizérem a poté je na místě modifikován komponentami pipeline. Objekt Language se stará o korordinaci těchto komponent. Posílá surový text, posílá jej přes pipeline a vrací anotovaný dokument. Třída language také organizuje školení a serializaci.

[38] Zdroj: `https://spacy.io/usage/spacy-101`



Obrázek 2.21: Architektura pro zpracování textu[39]

2.6.5 Tréning modelu

Většina komponent SpaCy je založena na statistických modelech. Každé „rozhodnutí“, které tyto modely učiní (označí slovo za jméno osoby), je předpověď založená na aktuálních váhových hodnotách modelu. Hodnoty vah jsou odhadnuty na základě vstupních příkladů,

[39] Zdroj: `https://spacy.io/usage/spacy-101`

které byly modelu během tréningu zadány. K trénování modelu jsou tedy potřeba trénovací data jimiž jsou nejčastěji manuálně anotované texty.

Samotné trénování je iterativní proces, při kterém se předpovědi modelu porovnávají s referenčními anotacemi, aby se odhadl **gradient ztráty**. Gradient ztráty se pak použije k výpočtu gradientu vah pomocí zpětného šíření. Gradienty udávají, o kolik by se měly změnit hodnoty vah, aby se předpovědi modelu více podobaly referenčním štítkům.

Trénování modelu není o tom, aby si model pouze zapamatoval vstupní příklady, ale aby vypracoval teorii, kterou lze zobecnit na neznámá data. Důležité pro samotný tréning modelu je trénování modelu na kategorii textu podobnou té zkoumané. Model naučený na příspěvcích ze sociálních sítí, nebude moc dobře fungovat na lékařské texty.

2.6.6 F-score

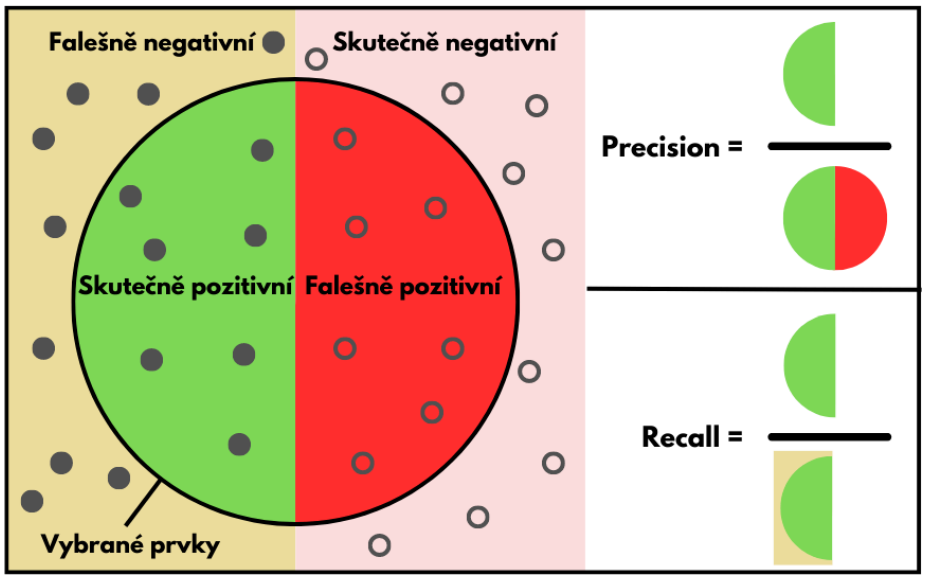
F-score - je měřítko přesnosti modelu na datové sadě. Často se nazývá F1-skóre. Nejčastěji se používá k hodnocení binárních klasifikačních systémů, které klasifikují příklady na „pozitivní“ nebo „negativní“.

Vzorec pro standardní skore F1 je harmonický průměr průměr přesnosti a opakovaného volání (precision, recall). Nejpřesnější model se tak co nejvíce přibližuje hodnotě 1, která je hodnota maximální [22].

$$F_1 = \frac{2}{\frac{1}{recall} \times \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}(fp + fn)}$$

precision - Podíl skutečně pozitivních výsledků, vydělený součtem falešně pozitivních a skutečně pozitivních výsledků.

recall - Podíl skutečně pozitivních výsledků, vydělený součtem skutečně pozitivních a falešně negativních výsledků.



Obrázek 2.22: Vizualizace F-skore Precision a Recall

Kapitola 3

Návrh a implementace

3.1 Zadání aplikace

Nežli se můžeme zabývat samotným návrhem aplikace, je nejdříve nutné si specifikovat, co se vlastně od naší aplikace požaduje. Je potřeba naprogramovat program, na jehož vstupu bude text získaný z OCR. Tento text je potřeba zpracovat pomocí nástrojů sémantické analýzy a připravit ho tak, abychom ho mohli vkládat do příslušných polí databáze. Do databáze se mají takto zaznamenávat všechny důležité informace o lidech kterých se tato matriční událost týká.

3.2 Výběr technologií

Tato část se zabývá výběrem navržených technologií a odůvodněním jejich výběru.

3.2.1 Programovací jazyk programu

Jak už jsme mohli zjistit ze zadání, v samotném programu bude třeba využít umělé inteligence pro rozpoznání vlastních jmen. Pro vývoj umělé inteligence se využívá více programovacích jazyků jako jsou Python, Java a jazyk R, nejvíce se však používá Python. A právě tento programovací jazyk byl vybrán i pro programování této práce.

Proč se vlastně Python jeví pro tuto práci jako nejlepší řešení? Důvodů je hned několik. Jak už bylo dříve řečeno, Python je v tomto odvětví nejpoužívanější a i díky tomu má velmi rozsáhlý ekosystém knihoven, které se dají při strojovém učení použít. Druhá velká výhoda je, že má Python jednu z nejlepších možností vizualizace, která se obzvláště u práce s daty hodí. Další výhoda je rychlost, které je docíleno knihovnami založenými na jazyce C++ a výběrem různých druhů interpretrů.

3.2.2 Korpus NER

Výběr jazykového korpusu pro NER byl ztížen faktem, že pro český jazyk aktuálně existuje pouze jeden korpus, který je se svými výsledky použitelný. Proto byl vybrán korpus CNEC2, který navazuje na své předchůdce CNEC1.1 a CNEC1. Ten obsahuje anotovaná data skoro všech kategorií potřebných pro matriční údaje. Další kategorie je potřeba doanotovat například pomocí nástroje knihovny SpaCy.

3.2.3 SpaCy

Jak už bylo v předchozí kapitole popsáno, SpaCy je open-source knihovna poskytující velké množství nástrojů pro zpracování textu. I díky tomu jsou modely natrénované touto knihovnou jedny z nejrychlejších v porovnání s konkurencí.

3.3 Návrh

Plánování návrhu začíná u vstupních dat programu, což je výstup z OCR. Při experimentování s OCR bylo zjištěno, že program neumí u matrik s rubrikami dobře rozpoznávat pole tabulky a výstup xml je tak členěn po slovech. Takto členěný text však není dobrým vstupem pro NLP pipeline, proto je nejdříve potřeba si tento výstup OCR upravit.

3.3.1 Definice tabulky

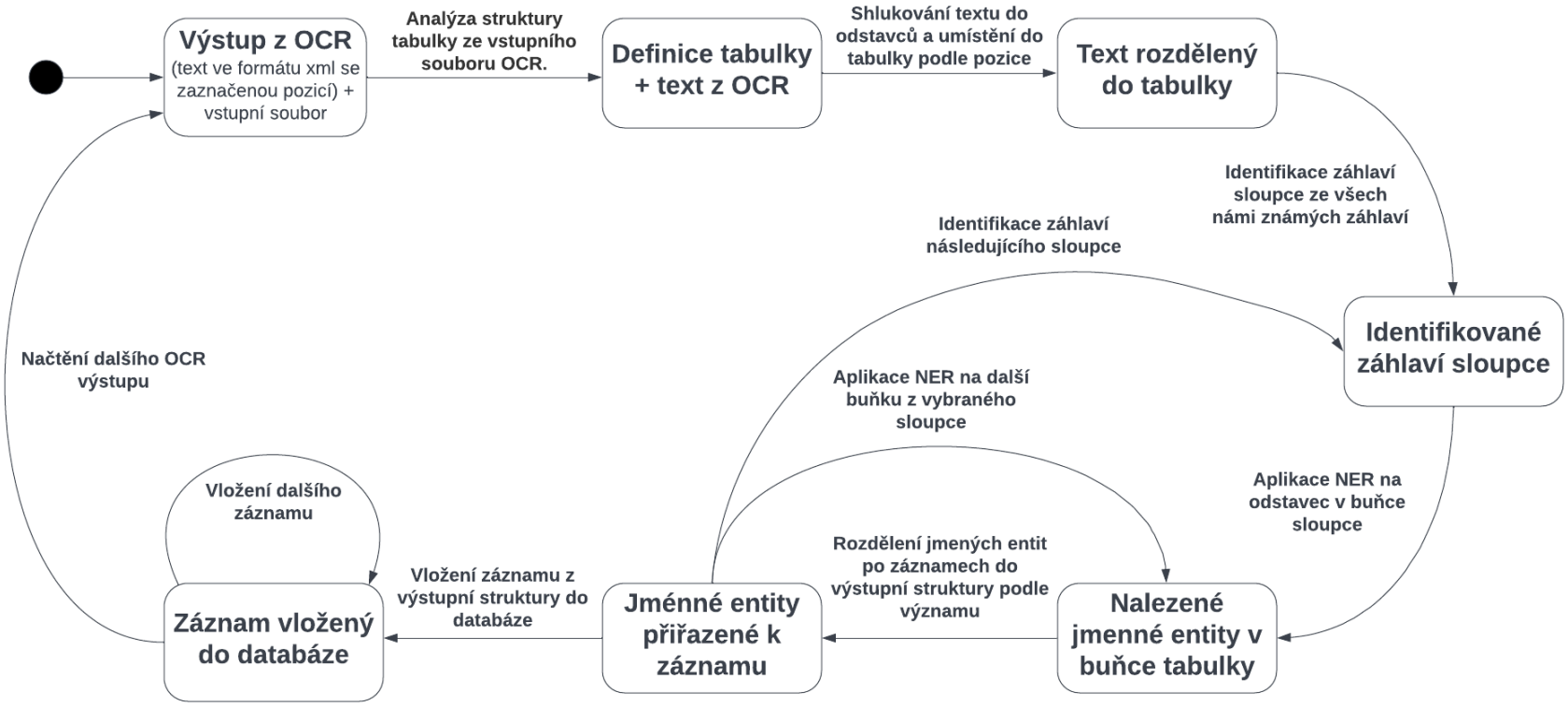
K takovéto úpravě dat musíme nejdříve zjistit rozložení tabulky z šablony matriky. Na tuto práci použijeme knihovnu cv2, která se zabývá zpracováním obrazu.

3.3.2 Shlukovaní textu

Po definici tabulky můžeme začít shlukovat slova do odstavců podle tabulky. Toto shlukování bude repetetivní činnost, která se bude provádět pro každý výstup OCR zvlášť. Pro každý výstup si určíme střed textového pole. Tento střed si poté zařadíme do jednotlivého pole tabulky. Díky znalosti rozložení tabulky známe informace, které máme v daném sloupci hledat.

3.3.3 Identifikace vlastních jmen

V následujícím kroku využijeme náš dříve popsaný model vytvořený pomocí knihovny SpaCY a korpusu CNEC Extended. Ze záhlaví je zjištěno, které informace by se měly ve sloupci nacházet. Potom se přistoupí do nezpracovaného řádku a na odstavec v něm se použije NLP model. Zjištěná vlastní jména umístíme do příslušných polí databáze.



Obrázek 3.1: Stavový automat navržené aplikace

Kapitola 4

Implementace

Tato kapitola je zaměřena na popis implementace řešení dříve specifikovaného zadání. Při implementaci byl brán zřetel na dřívější návrh, ale od tohoto návrhu se řešení může lehce lišit. I proto je možné si prohlédnout finální stavový automat, který odpovídá aplikaci.

Hlavní funkcí programu je funkce `main` v souboru **main.py**, která po zadání validních parametrů a vstupů provede sémantickou analýzu matričního textu zadaného na vstupu pomoci souboru s příponou **.xml** ve formátu **PAGE**.

4.1 Hlavní funkce programu `main`

Už dříve bylo řečeno, že funkce `main` je hlavní funkcí programu a je vlastně takovou spojkou mezi zpracováním argumentů programu prováděnou funkcí `parse_arg` ze souboru **arg_parser.py** a samotnou sémantickou analýzou, o kterou se stará funkce `semantic_analysis` implementovaná v souboru **analysis**.

Samotná funkce `main` může nabrat třech různých stavů:

1. Jsou zadány špatné argumenty ⇒ navrácena chyba, ukončený program.
2. Na vstupu je zadán jeden soubor **.xml** ⇒ zavolána funkce `semantic_analysis` pro tento soubor.
3. Na vstup je zadána složka obsahující soubory **.xml** ⇒ zavolána funkce `semantic_analysis` pro každý z nich.

Po zpracování všech vstupů je program ukončen.

4.2 Funkce pro zpracování argumentů `parse_arg`

Funkce `parse_arg` zpracovává argumenty programu. Jejich možnosti a použiti jsou popsány v části použití programu. Samotná funkce už pouze zabezpečuje jejich správné zpracování a kontroluje pravidla, která jsou pro správný běh potřeba (pouze jeden s argumentů -f -d, nebo chybějící jméno výstupního souboru).

4.3 Hlavní funkce sémantické analýzy `semantic_analysis`

Hlavní běh sémantické analýzy matričních záznamů je prováděn ve funkci `semantic_analysis` v souboru **analysis.py**. Funkce bere jako parametr **.xml** soubor ve formátu PAGE a jméno výstupního souboru.

Dříve než je možné se do samotné sémantické analýzy pustit, je nejdříve nutné inicializovat objekt pro práci NER a to *NLP*. Ten při své inicializaci načte námi vytrénovaný model a uloží ho do objektu pro další zpracování.

V tuto chvíli už přišel čas rozdělit vstupní xml soubor. K tomuto zpracování byla využitá třída `PageLayout` z aplikace PeroOCR.

Po rozdělení je nutné zpracovávanou stranu rozdělit na dvě poloviny pomocí funkce `split_page_polygons`, protože většina matrik má na jedné dvoustraně (Scanu), dvě tabulky se záznamy a je proto jednoduší tabulky zpracovávat zvlášť. Pro každou tabulku scanu poté provádíme stejné kroky zpracování.

Zpracování matrik

Pro stránku (matriku) vytvoříme objekt `Page` deklarovaný v souboru **page.py**, kterému při inicializaci předáme zpracovaný vstup (polovinu instance objektu `PageLayout`, tzn. jednu stranu matričního scanu). Třída textttPage nám v celé funkci pro sémantickou analýzu zabezpečuje zpracovávání tabulky matriky a uchovává nám důležitá data o této konkrétní straně.

První a jeden z nejdůležitějších kroků u zpracování strany matričního záznamu je identifikace hlavičky matriky. I když se může na první pohled zdát tento problém jako zbytečný, z nadpisu matriky je možné získat spoustu důležitých informací o formátu matriky, kterou zpracováváme.

V hlavičce matriky můžeme nalézt:

1. Druh matriky, kterou zpracováváme (Křestní, Oddací, Úmrtní)
2. Druh šablony matriky
3. Město jehož obyvatelích matrika uchovává informace.
4. Rok zápisu

Obrázek 4.1: Ukázka hlavičky matriky oddání Blansko

Po zpracování hlavičky matriky funkcí `page_title_analyze` následuje analýza rubrik matriky. O analýzu rubrik se stará funkce `headers_analyze`, jejímž výstupem je list obsahující kategorie rubrik a list s rozpětím rubrik.

V původním listu Regions vyčteného z xml souboru se nám v tuto chvíli nacházejí pouze textové polygony záznamů. Během let se však s šablonami matrik měnil i způsob zápisu do nich a tak v určitých letech se nepsalo ke každému záznamu do položky datum - rok

dané události, který se psal pouze při jeho změně. V následujícím kroku si proto všechny tyto záznamy projdeme a zjistíme, jak je zde rok zapisován. Pokud se tento rok ke každému záznamu nezapisuje, rozdělíme tyto záznamy pro další zpracování po letech, ve kterých se udály. Pokud jsme tak učinili, tak následující zpracování záznamů, budeme provádět pro každý rok zvlášť.

Zpracování záznamů začíná tím, že si textové polygony rozdělíme do řádků podle záznamů pomocí funkce `create_rows`. Stejně tak rozdělíme textové polygony do sloupců podle rozpětí registrů zajištěných funkcí `headers_analyze`. K samotnému rozdělení použijeme funkci `create_columns`. Ze sloupců a řádků následně vytvoříme tabulku pro další zpracování pomocí funkce `create_table`.

V tabulce vytvořenou předchozími kroky se nám sále mohou vyskytovat řádky/záznamy, které nenesou žádnou podstatnou informaci. Jediná možnost, jak tyto pro nás vadné záznamy identifikovat, je zjistit v kolika sloupcích nenesou žádnou informaci a záznamy s více než polovinou prázdnými sloupci zahazujeme. O tuto kontrolu se nám stará funkce `delete_suspicious_rows`. Všechny funkce zabezpečující vytvoření sloupců, řádků a tabulky jsou definovány v souboru **polygons.py**

Vytvořenou tabulku dále analyzuje pomocí funkce `analyze_table` a vracíme strukturované záznamy v listu. Záznamy ukládáme na Google Drive do Google Tabulek. K tomuto účelu používáme funkci `export_to_googlesheets`.

4.4 Detekce záznamů `create_rows`

Detekce záznamů byla v rámci implementace velkým oříškem. V rámci OCR bychom totiž mohli brát v potaz možné ohraničení každého tohoto záznamu, jenže na vstup tohoto programu jsou přivedeny pouze textové polygony. Úkolem tedy bylo najít funkční řešení shlukování těchto polygonů po záznamech k dalšímu zpracování.

Nejdříve jsme tedy zkoušeli tyto polygony shlukovat, pomocí různých metod k tomu určených.

Mezi tyto metody patří například:

- Voronoi
- Kmeans
- Adaptive treshold

Jak můžeme z ukázky vidět, žádná z těchto vypsaných metod nebyla moc úspěšná a proto byla nakonec použita metoda rovnoběžných přímek.

Obrázek 4.2: Ukázka efektivity shlukovaní pomocí určitých metod shlukování

4.4.1 Metoda rovnoběžných přímek

Metoda rovnoběžných přímek, jak už název napovídá, využívá k detekci záznamů přímky rovnoběžné s osou X, pomocí kterých vyhledává mezi polygony hranice záznamů.

V kódu se nám o tuto činnost stará funkce `create_row_edges`:

```
def create_row_edges(polygons):
    # redukce velikosti polygonů pro lepší hledání hran.
    new_polygons = polygons_reduction(polygons, 0.65)
    edges = []
    for polygon in new_polygons:
        interest = False
        # nalezeni nejspodnější hranice polygonu
        _, min_y, _, max_y = polygon.bounds
        # vytvoření rovnoběžné přímky z~tohoto bodu
        edge = sg.LineString([(0, max_y + 0.1), (10000, max_y + 0.1)])
        # detekce průsečiků s~ostatními polygony
        for pol in new_polygons:
            if pol.intersects(edge):
                interest = True
                break
        # uložení hrany pokud neprotíná žádný polygon
        if not interest:
            edges.append(sg.Point(0, max_y))

    return edges
```

Jak můžeme z kódu vidět, funkce není nijak složitá, ale o proti ostatním složitějším metodám je pro nás mnohem lépe využitelná.

Pomocí těchto hran a funkce `create_rows`, rozdělíme tyto polygony do řádků po záznamech.

Tuto metodu a její variace využíváme například i u detekce hlavičky, nebo u zpracování rubrik.

4.5 Zpracování rubrik `headers_analyze`

Zpracování rubrik je základem pro správnou sémantickou analýzu matričních záznamů. V celé aplikaci se využívá jazykový model, který dokáže na základě jeho znalostí rozpoznat v textu vlastí jména, ale neumí je umístit do kontextu. Pro získání kontextu těchto vlastních jmen se nám starají právě rubriky.

Z pole textových polygonů získáme rubriky pomocí už dříve zmiňované **metody rovnoběžných přímek**. Díky této metodě a znalostem typu šablony matriky (ze zpracování hlavičky `page_title_analyze`).

Polygony pomocí adaptivního thresholdu rozdělíme do sloupců (rozdělíme jednotlivé rubriky od sebe) a postupně tyto rubriky hledáme ve slovníku uloženém v souboru **headers.py**. Zde je u každé rubriky uloženo i číslo kategorie rubriky pro další zpracování. Pro

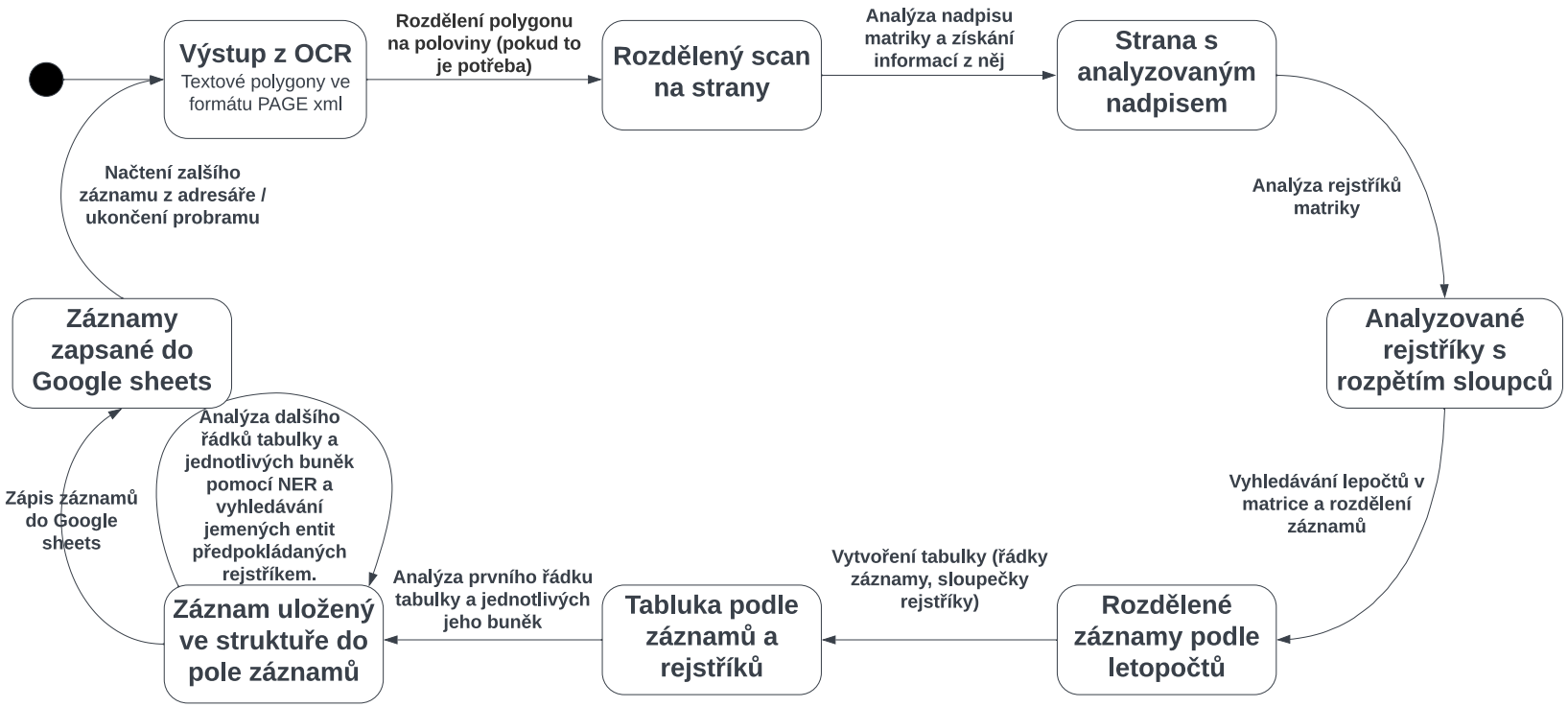
hledání nejpodobnějšího řetězce námi získaného (v tomto případě se získanou rubrikou), používáme funkci `get_close_matches` z knihovny *difflib*.

4.6 Využití jazykového modelu NER

Námi trénovaný model pro rozpoznávání jmenných entit používáme v implementaci na dvou místech, abychom zajistili důležité informace pro výsledné záznamy.

Prvním místem použití je funkce už dříve zmiňovaná `page_title_analyze`, kde hledáme v textu informace o městě, pod které tyto záznamy spadají a o roku, ve kterém byly tyto záznamy zaznamenány.

Druhým využitím je ve funkci `analyze_table`, kde zpracováváme jednotlivé záznamy. Zde pomocí tohoto jazykového modelu můžeme nalézt prakticky všechny informace, které pro záznam hledáme.



Obrázek 4.3: Stavový automat výsledné aplikace

Kapitola 5

Testování

V této kapitole se budeme zabývat úspěšností sémantické analýzy a rozpoznáváním jednotlivých kategorií slov. Dříve něž se však do této části pustíme, musíme nejdříve zohlednit přesnost procesu, na který navazujeme a to je OCR.

5.1 OCR

Jak už bylo dříve řečeno, pro aplikaci bylo vybráno Pero OCR, které je vyvíjeno pracovníky FIT VUT v Brně a je zde do budoucna možná konzultace případných vylepšení.

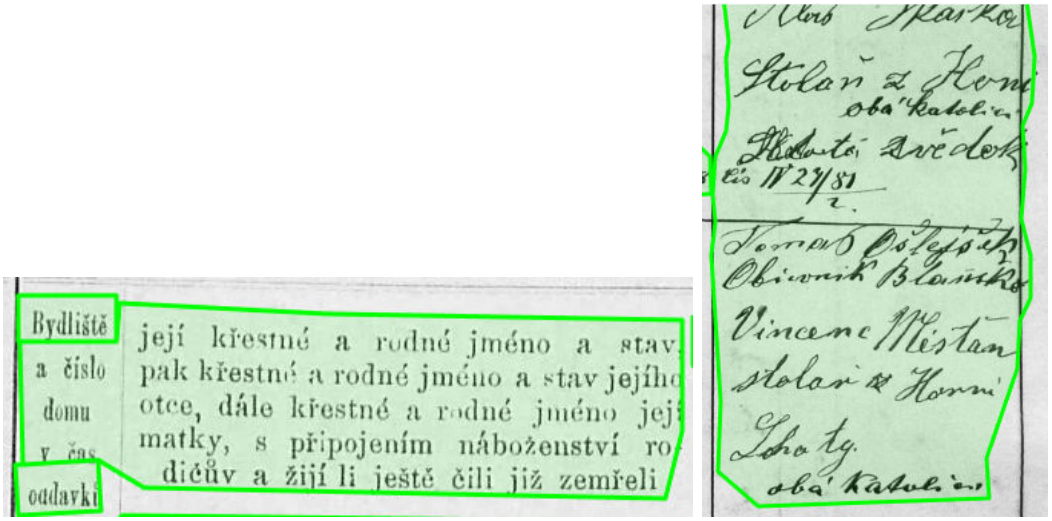
Proces OCR v aplikaci Pero OCR se skládá ze dvou částí. První je **analýza rozložení textu**, která rozpozná textové bloky a odhadne jejich pořadí ve směru čtení a následně od shora dolů. Další částí je **optické rozpoznávání znaků**, kde se pomocí vytrénovaných modelů rozpoznávají jednotlivé znaky, které jsou následně ukládány jako textové bloky (získané v předchozí části) a řádky, které tyto textové bloky obsahují.

5.1.1 Analýza rozložení textu

Analýza rozložení textu je v Pero OCR před-proces samotného optického rozpoznávání znaků, kde se určí textové bloky, které budou v následném procesu rozpoznávány. I přesto, že to na první pohled úplně nevypadá, tento proces je velmi důležitou částí OCR a její chybovost může velmi ovlivnit kvalitu výsledné aplikace. V následujících ukázkách jsou zobrazeny nejčastější chyby analýzy rozložení textu a jejich vliv na kvalitu výsledné aplikace.

Spojené bloky

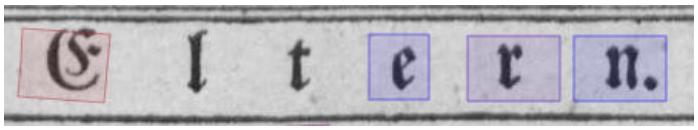
Prvním problémem, který může nastat, je chybné spojení bloků blízko sebe. Tato chyba může negativně ovlivnit následné rozdělení textových bloků do řádků a sloupců, nebo identifikaci Šablony matriky, navíc tato chyba se během následujícího zpracování nedá nijak učině odchytit, nebo opravit a proto zanáší skoro vždy chybu do řešení.



Obrázek 5.1: Špatně spojené dva bloky blízko sebe

5.1.2 Rozdělené bloky

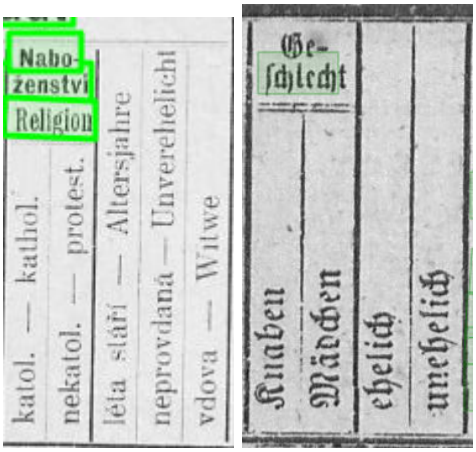
Chyba, která u matričních textů velmi často nastává, je chybou rozdělení bloků, které k sobě mají patřit. U matričních textů se tak nejčastěji jedná o Rubriky, jež jsou psány speciálním druhem textu. Tato chyba je pro nás riziková hlavně z toho hlediska, že se jedná o rubriky a může tak negativně ovlivnit jejích detekci.



Obrázek 5.2: Špatně rozdělen jeden blok do více

Rozpoznávání svislých textů

Další velká chyba, které se Pero OCR dopouští (v tomto případě je to spíše věc, kterou modely Pero OCR neumějí) je nulové rozpoznávání textů, které nejsou psány vodorovně. To samozřejmé velmi sníží možnosti pro rozpoznávání určitých kategorií, které jsou v dané matrice zapsané v hlavičce tímto způsobem.



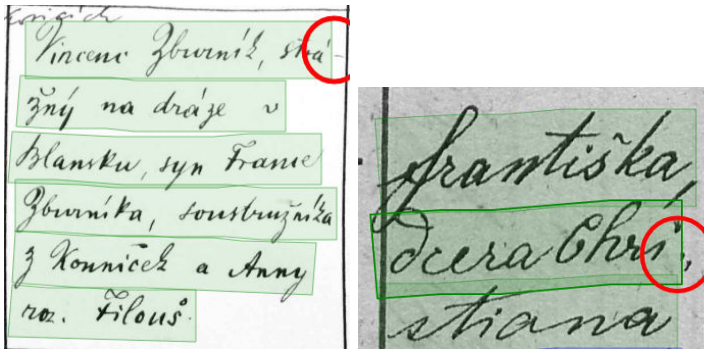
Obrázek 5.3: Nerozeznaný svislý text

Ukončení řádků

Následující chyba je ta, která se v OCR vyskytuje nejčastěji na konci těch řádků, kdy je řádek ukončen čárkou, tečkou, pomlčkou, nebo uvozovkami dole. Jelikož se v textu snažíme nalézt pouze jmenné entity a nijak více text nezkoumáme, tečka a čárka pro nás nejsou nijak zvlášť potřebné znaky, ovšem pomlčka na konci řádku, nebo uvozovky dole, které mají spojit rozdělené části slova, nám můžou zavést do řešení chybu. Takováto chyba může nastat v hned v několika případech.

Za prvé a nejvíce časté je, když nám chybně spojené (nebo spíše nespojené) slovo nedokáže rozpoznat NER a my tak přijedeme o důležitá data.

Další případ také souvisí s rozpoznáním jmenných entit. V tomto případě se však jmenná entita správně rozpoznala, avšak jedná se o vlastní jméno, které rozdělujeme na Jméno a Příjmení. Na tomto místě se nám například může špatně spojené křestní jméno Ali ce, rozdělit jako Jméno: Ali Příjmení: ce.

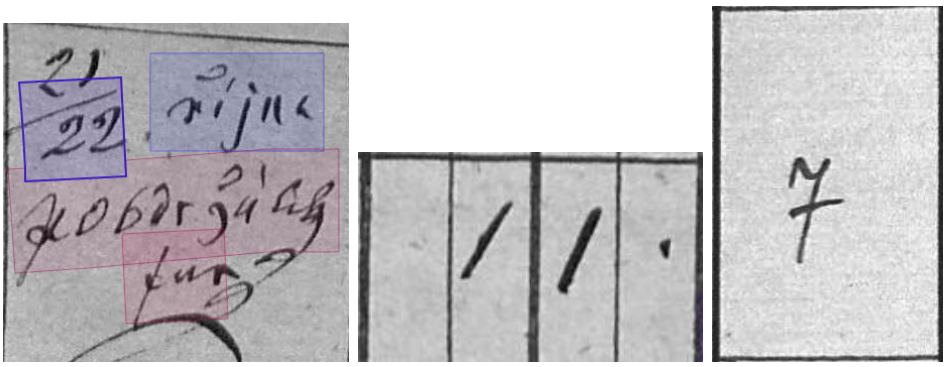


Obrázek 5.4: Nerozeznaný pomlčka, nebo uvozovky dole

Jednoznakové textové bloky

Tato chyba se vztahuje na velmi malé textové bloky, obsahující jeden nebo dva znaky. U matrik tak může tato chyba postihovat lomítka, značící například pohlaví nebo náboženství. Další kategorie, u které se tato chyba vyskytuje, jsou složená data, třeba u matrik narození. Zde jsou pod sebou napsány dva dny v měsíci, kdy jeden den se vztahuje pro narození a

druhý pro křest. Problém však dělá i osamocené číslo měsíce, nenacházející se blízko jména měsíce.



Obrázek 5.5: Nerozeznané malé znaky

Výše popsané chyby, mohou sice svým výskytem znehodnotit nejeden záznam, avšak v OCR se nevyskytují v nějaké příliš vysoké míře. Nesmíme však na jejich výskyt zapomínat, protože mohou negativně ovlivnit výsledky výsledné aplikace.

5.1.3 Optické rozpoznávání znaků

Druhá část procesu OCR u Pero OCR je samotné Optické rozpoznávání znaku, kdy OCR prochází už dříve označené textové bloky a text v nich znak po znaku. Tyto znaky poté postupně zkoumá a pomocí pravděpodobnostního modelu složeného ze známých znaků určuje, o který znak se jedná.

Pro zjištění přesnosti tohoto OCR modelu bylo nutné si nechat vygenerovat originální výstup OCR a k porovnání tento výstup co nejlépe opravit. Zde však nastal problém. Matriky jsou psány různými písmy a očima laika nejsou vždy úplně čitelné. Z důvodu chybějících znalostí starého písma tak mohly být mezi sebou porovnávány pouze výstupy nejnovějších matrik. Pro správné provedení testu byla stanovena hodnota 0.6 (60% podobnost) z důvodu použití této hodnoty při detekci rubrik.

OCR	Podobnost	Správně	Chyba
Narození	89,37%	66,67%	33,33%
Oddání	83,08%	100%	0%
Úmrtí	67,47%	100%	0%
Celkem	**79,97%**	**88,89%**	**11,11%**

Tabulka 5.1: Výsledky testování Optického rozpoznávání znaků

5.2 NER

Named-entity recognition se nám v aplikaci stará o detekci jmenných entit a jejich správnou kategorizaci, proto je také nutné mít znalosti o účinnosti námi vytrénovaného modelu. U takto trénovaných modelů se účinnost nejčastěji uvádí pomocí F-score.

5.2.1 Testování

Většina trénovacích nástrojů, stejně jako námi použitý Spacy, mají výpočet F-score součástí trénování a hotový model má ve svém adresáři `meta.json` soubor obsahující tuto statistiku, kde nalezneme nejen celkovou úspěšnost modelu, ale i úspěšnost rozpoznávání jednotlivých kategorií. Výsledky testování jsou udávány pomocí

Best_model	F-score	Precision	Recall
P - person	0,8840	0,8633	0,9057
O - Artifact names	0,7261	0,7524	0,7015
G - Geographical names	0,8418	0,8438	0,8399
I - Institutions	0,7284	0,7662	0,6941
T - Time expressions	0,9075	0,8873	0,9285
M - Media names	0,7609	0,8974	0,6603
A - Numbers in addresses	0,7922	0,7922	0,7922
Celkem	**0,8219**	**0,8285**	**0,8154**

Tabulka 5.2: Úspěšnost modelu **Best_model** vytrénovaném na korpusu CNEC2-Extended

Best_own_anotaion	F-score	Precision	Recall
P - person	0,8871	0,9024	0,8723
O - Artifact names	0,7273	0,7412	0,7138
G - Geographical names	0,8415	0,8274	0,8415
I - Institutions	0,6944	0,7614	0,6382
T - Time expressions	0,8819	0,8581	0,9074
M - Media names	0,7473	0,8947	0,6415
A - Numbers in addresses	0,8516	0,8462	0,8571
Celkem	**0,8155**	**0,8295**	**0,8019**

Tabulka 5.3: Úspěšnost modelu **Best_with_own_anotation** vytrénovaném na korpusu CNEC2-Extended obohaceném o vlastní o-anotovaná data

Best_CPU	F-score	Precision	Recall
P - person	0,7344	0,7682	0,7033
O - Artifact names	0,5950	0,6429	0,5538
G - Geographical names	0,7002	0,6760	0,7262
I - Institutions	0,5085	0,4891	0,5294
T - Time expressions	0,8702	0,8552	0,8857
M - Media names	0,6809	0,7805	0,6038
A - Numbers in addresses	0,7143	0,9184	0,5844
Celkem	**0,6828**	**0,6929**	**0,6730**

Tabulka 5.4: Úspěšnost modelu **Model_CPU**, vytrénovaném pomocí CPU na korpusu CNEC2-Extended obohaceném o vlastní o-anotovaná data

5.3 Aplikace

Tato část je věnovaná testování výsledné aplikace a jejím dílčím částem. Jak už bylo dříve řečeno aplikace byla testovaná na výstupu z Pero OCR.

Prvotní myšlenkou bylo otestovat aplikaci s dvěma druhy vstupu a to Originálním výstupem bez jakýchkoliv úprav (dále popisován jako **ORIGINAL**) a správně upraveným vstupem (dále popisován jako **FIXED**). Bohužel v této části opět nastal problém. Pero OCR sice má webovou aplikaci, ve které se výstupy dají upravovat, ale po testování bylo zjištěno, že takto upravit jdou pouze obsahy textových polygonů a ne polygony jako takové, protože i přes to, že v aplikaci tato možnost je, úprava polygonů zanese chybu do xml výstupu. Právě proto, že nemohly být chyby v rozložení textu opraveny, je chybovost aplikace rozdělená na chyby způsobené špatným rozložením textu, chyby ve psaní matrik a chyby v aplikaci.

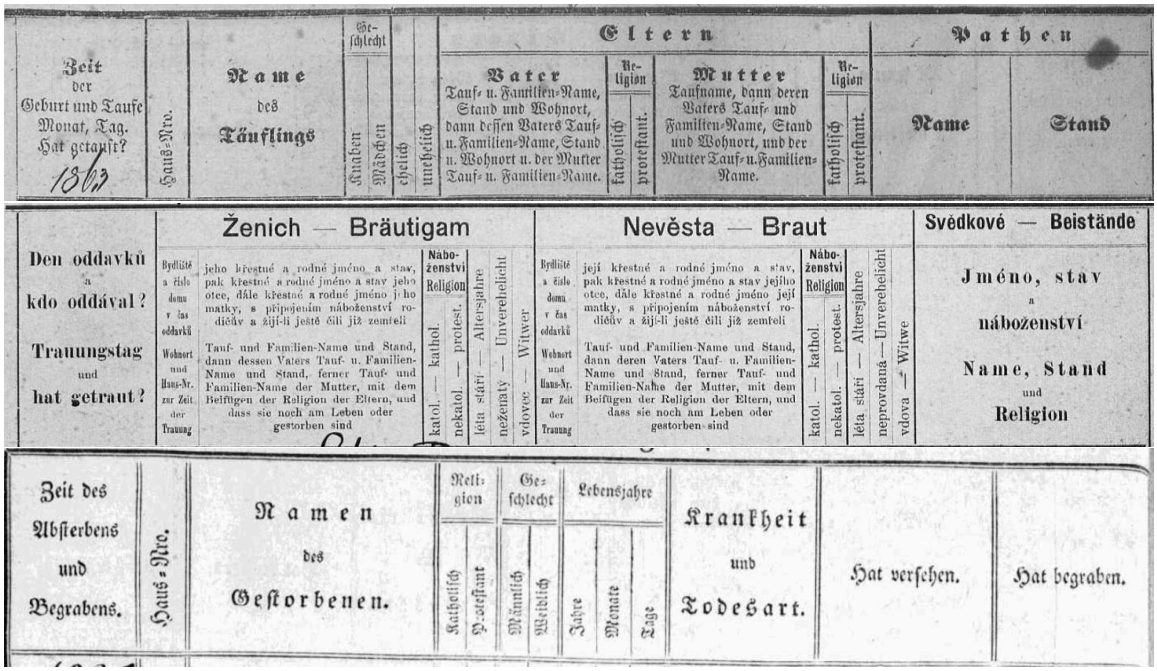
V tabulce s výsledky jsou informace o podobnostech referenčních vstupů a výstupu. Dále jsou uvedeny výsledky testování, tzn. kolik procent testů prošlo bez problému a kde případně se chyba vyskytovala.

5.3.1 Detekce rubrik

Správnost detekce rubrik je klíčovou částí celé aplikace a proto bylo třeba tuto část řádně otestovat. Jak už bylo dříve uvedeno, matriky byly psány v různých formátech, ve spoustě jazycích a v průběhu let se měnil i obsah dat, které se do nich zapisovaly, právě proto byly

pro testování aplikace vybrány matriky v celém průsečíku let když se zapisovaly do rubrik. Využity byly hlavně matriky Města Blansko zahrnující i okolní obce: *Dolní Lhota, Dvůr Žižlavice, Hořice, Horní Lhota, Klepačov, Lažánky, Olešná, Těchov, Olomučany, Ráječko, Suchdol a Veselice.* Čísla použitých matrik: **50, 51, 52, 53, 54, 55, 77, 78, 79, 80, 81, 82, 93, 94, 95, 96, 97**.

Jako starší Latinské byly využity matriky města Křtiny a přilehlých obcí: *Adamov, Babice nad Svitavou, Březina, Bukovina, Bukovinka, Habrůvka, Kanice, Křtiny, Obce, Ochoz u Brna, Proseč a Řicmanice.* Čísla použitých matrik: **417, 418**.



Obrázek 5.6: Pod sebou jsou uvedeny ukázky rubrik matrik z Blanska. Křestní, Oddací, Úmrtní

Test na opraveném výstupu OCR

Nejdříve byla testována detekce rubrik na souboru matrik s tištěnými rubrikami, které byly ve webové aplikaci opraveny, aby byla co nejvíce úspěšnost samotné programové části, kde bude mít co nejmenší vliv chybovost OCR. Jak už však bylo dříve vysvětleno, z důvodu chyby ve webové aplikaci Pero OCR, která znemožnila úpravu rozložení textu, bylo nutné do výsledků specifikovat odkud chyba pochází.

Z výsledků tabulky 4.1 je vidět, že v množině testovaných dat nebyla nalezena žádná chyba, která by přímo souvisela s implementací aplikace. Nejčastější chyby, které bylo možné nalézt ve výsledcích, byly chyby, jenž přímo souvisely se špatnou prací OCR, nebo kvalitou Scanu/zápisu. Konkrétně se v oblasti OCR jednalo o špatné spojení více textových bloků do jednoho, nebo opačným případem, kdy byla jedna rubrika rozdělena do více textových bloků a to třeba i po jednom znaku. Chyby Scanu/Zápisu byly chyby související s nekvalitním nafocením celého bloku matrik.

OLD_ROWS	Podobnost	Správně	Chyba		
			OCR	Scan/Zápis	Aplikace
Narození	100%	100%	0%	0%	0%
Oddání	93,25%	91,17%	8,93%	0%	0%
Úmrtí	95%	90%	1,66%	8,33%	0%
Celkem	**96,08%**	**93,72%**	**3,53%**	**2,78%**	**0%**

Tabulka 5.5: Výsledek testu detekce rubrik s opraveným výstupem OCR

Test na originálním výstupu OCR

Po otestování detekce ve zdánlivě ideálním prostředí bylo nutné matriky otestovat i na výstupu originálním, pro případ, že by nebylo možné chyby vykazující části OCR opravit. K chybám analýzy rozložení textu tak v testovacích datech přibyly chyby související s optickým rozpoznáváním.

Z výsledků testů v tabulce 4.2 vychází, že testovací úspěšnost se změnila a to nejen v procentuální úspěšnosti správně provedených testů, ale i v podobnosti referenčních výstupních hodnot a hodnot získaných. Největší procento chyb, jak se dalo očekávat, přibylo z OCR.

Celkové výsledky testování však mohly být o několik procentních bodů horší. Po prvotním testování byly některé lehko ošetřitelné chyby opraveny. Nejčastěji se jednalo o dodatečné mezery v názvu rubrik.

OLD_ROWS	Podobnost	Správně	Chyba		
			OCR	Scan/Zápis	Aplikace
Narození	85,43%	74%	26%	0%	0%
Oddání	79,32%	66,66%	33,33%	0%	0%
Úmrtí	92,11%	86,66%	3,33%	10,00%	0%
Celkem	**85,62%**	**75,77%**	**20,89%**	**3,33%**	**0%**

Tabulka 5.6: Výsledek testu detekce rubrik s originálním výstupem OCR

Test ručně-psaných rubrik na opraveném výstupu OCR

Poslední rubriky matrik, u kterých detekce byla třeba otestovat, byly rubriky ručně psané. Z prvotního pohledu na výstup OCR těchto Matrik bylo vidět, že testování těchto rubrik bez opravy by nemělo smysl. Výstup z OCR byl totiž ve velmi špatném stavu v obou částech OCR zpracování.

Podle dat získaných testováním v tabulce 4.3 nejsou na první pohled vidět tak obrovské změny v chybovosti jednotlivých testů, bylo však nutné u opravy OCR výstupů z těchto testů vynaložit opravdu velikou snahu, aby bylo dosaženo jejich alespoň textové správnosti. Výsledky takto neupravených testů by se totiž pohybovaly pouze v nízkých jednotkách procent.

OLD_ROWS	Podobnost	Správně	Chyba		
			OCR	Scan/Zápis	Aplikace
Narození	56,90%	40%	60%	0%	0%
Oddání	97,92%	90,00%	10,00%	0%	0%
Úmrtí	80%	50,00%	50,00%	0%	0%
Celkem	**78,27%**	**60,00%**	**40,00%**	**0%**	**0%**

Tabulka 5.7: Výsledek testu detekce ručně-psaných rubrik s opraveným výstupem OCR

5.3.2 Detekce záznamů

Další důležitou částí aplikace je detekce záznamů (řádků v tabulce matriky). Problematice detekce záznamů byla věnovaná velká část kapitoly Návrh a Implementace. Zde jsme zjistili, že shlukování textových polygonů pouze se znalostí poloh těchto polygonů, je pro přesnost rozdělení nedostačující. Úkolem tak bylo nalézt co nejpřesnější řešení, které tento úkol splní. Výsledné řešení metodou rovnoběžných přímek bylo tedy testováno a výsledky jsou zapsány v tabulce.

ROWS	Podobnost	Správně	Chyba		
			OCR	Scan/Zápis	Aplikace
Narození	90,28%	57,90%	23%	0%	13%
Oddání	94,74%	83,33%	16,66%	0%	0%
Úmrtí	90%	42,00%	38,46%	0%	19%
Celkem	**91,83%**	**61,08%**	**26,15%**	**0%**	**10,85%**

Tabulka 5.8: Výsledek testu detekce záznamů

Z výsledků můžeme vidět, že i přes poněkud malou průchodnost testů u matrik narození a úmrtí, dostáváme vysokou úspěšnost v podobnosti referenčního a reálného výstupu. Tato podobnost byla v případě detekce záznamů zjišťována pomocí rozdílu referenčních záznamů na stranu a získaných záznamů na stranu, jejíž hodnota byla porovnávána oproti součtu všech referenčních záznamů.

Chyby se dělily prakticky jen na dva druhy pochybení. První z nich bylo spojení dvou nesouvisejících bloků patřících do různých záznamů, ze strany OCR. Druhé pochybení bylo v tabulce nazváno jako chyba aplikace. Ta však chybou aplikace byla jen částečné.

Jak už bylo několikrát psáno, detekce záznamů pouze pomocí znalosti textových polygonů je složitá a i přesto, že řešení metodou rovnoběžek bylo ze zkoumaných možností nejúspěšnější, existuje jedna možnost, kdy se metoda dopustí chyby. Tato možnost je v případě, že záznam není psán ve velkých textových blocích a tyto bloky jsou rozděleny. V tomto případě mohou vzniknout dva záznamy rozdělením jednoho.

Vzhledem ke skutečnosti, zjištěné z podrobného prozkoumávání chybových výstupů, je nutné výsledky spojit se skutečností, že v ani jednom případě chyby se nejedná o znehodnocení pouze jednoho záznamu, ale vždy se jedná alespoň o dva záznamy.

5.3.3 Celá aplikace

Poslední a nejdůležitější část testování aplikace, je testování aplikace jako celku, tzn. předpokládaný výstup vs. reálný výstup.

Už dříve bylo psáno, že data se ukládají pomocí GoogleAPI na Google drive, kde jsou uchovávány ve formátu xlsx. Bylo tedy nutné vytvořit ke každé testované matrice tabulku referenční, jejíž hodnoty byly ručně co nejpřesněji vypsány z matriky. Zde však bylo nutné, aby tabulky byly totožné, co se týče počtu řádků, aby je bylo možné porovnat. Ovšem jak už bylo dříve zmiňováno OCR se může dopustit chyb, kdy spojí více textových polí do jednoho a v tomto případě by nám počty záznamů neseděly. V reakci na tuto skutečnost musely být referenční vstupy upraveny a vymazány hodnoty, u kterých tato chyba nastávala.

Takto jsme testovali tři rozdílné NER modely popsané v kapitole 4.2, která všechny tři modely více popisuje.

Best_model	**Podobnost**	**Čas**
Narození_fixed	79,03%	22,01s
Oddání_fixed	90,52%	20,06s
Úmrtí_fixed	73,74%	33,73s
Celkem	81,10%	25,27s
Narození_original	47,03%	22,69s
Oddání_original	62,44%	20,18s
Úmrtí_original	53,37%	31,11s
Celkem	54,28%	24,66s

Tabulka 5.9: Výsledky testování úspěšnosti programu běžícím s modelem: **Best_model**

Best_own_annotation	**Podobnost**	**Čas**
Narození_fixed	76,29%	24,32s
Oddání_fixed	76,15%	20,15s
Úmrtí_fixed	74,36%	35,57s
Celkem	75,60%	26,68s
Narození_original	45,54%	26,01s
Oddání_original	59,00%	24,12s
Úmrtí_original	54,12%	32,15s
Celkem	52,89%	27,43s

Tabulka 5.10: Výsledky testování celkové úspěšnosti programu běžícím s modelem: **Best_own_annotaion**

Z výsledků můžeme vidět, že tabulky 4.6 a 4.7 se hlavně v části opravených dat liší docela velkým způsobem a to i přes to, že v druhém modelu byly data manuálně do-anotována. V časech provedení testů se hodnoty zásadním způsobem nelišily.

Už při trénování modelů jsme si všimli jedné dosti zásadní věci. I přesto, že model trénovaný na CPU oproti těm klasickým na GPU nedokazuje vysoké úspěšnosti detekce vlastních jmen, bylo za to jeho trénování vždy mnohem rychlejší. Z tohoto důvodu jsme jeden takto vytrénovaný model do testování zahrnuli, abychom otestovali, jestli jeho rychlost vykompenzuje horší detekci. Výsledek byl poměrně překvapivý.

Best_CPU	Podobnost	Čas
Narození_fixed	51,21%	16,58s
Oddání_fixed	60,68%	16,71s
Úmrtí_fixed	66,34%	20,92s
Celkem	59,41%	18,07s
Narození_original	41,80%	14,53s
Oddání_original	47,38%	13,66s
Úmrtí_original	49,33%	19,88s
Celkem	46,17%	16,02s

Tabulka 5.11: Výsledky testování celkové úspěšnosti programu běžícím s modelem: **Best_CPU**

Snížená kvalita detekce vlastních jmen se sice dostavila, ale s ní přišla samozřejmě i mnohem větší rychlost a to skoro o polovinu. Je tedy na zvážení každého, kdo toto zařízení bude používat, jestli půjde cestou větší kvality, nebo kvantity.

5.3.4 Rozpoznávání kategorií

Poslední část testování byla zaměřena na úspěšnost rozpoznávání jednotlivých kategorií. Při přípravě této části bylo nalezeno hned několik problémů, které byly potřeba řešit.

První z nich bylo rozpoznávání jmen a příjmení. V knihovně Spacy a hlavně v korpusovém formátu CONLL, který byl použit pro trénování modelu je samozřejmostí, že jméno se zadává ve formátu jméno, příjmení a nepočítá s inverzí tohoto formátu. V testování tak bylo nutné z této skutečnosti slevit a nepovažovat otočení jména a příjmení za chybu.

Další věc, kterou nebylo možné prakticky vyřešit, bylo rozpoznávání povolání. Tento problém nastal při trénování modelu s vlastními do-anotovanými daty. Jak můžeme vidět z předchozí části, trénování vlastními do-anotovanými daty, nemělo moc velký úspěch a úspěšnost takto vytrénovaného modelu byla v horší kvalitě, než u originálního data-setu CNEC2-extended.

Výsledky testování si můžeme prohlédnout v tabulce.

Best_model	Datum	Jméno	Příjmení	Město
Fixed	95,32%	83,91%	78,42%	81,85%
Original	90,27%	48,06%	48,27%	63,35%

Tabulka 5.12: Výsledky testování celkové úspěšnosti rozpoznávání jednotlivých kategorií slov

Případná vylepšení

Jak jsme už dříve mohli vidět, přesnost rozpoznávání kategorii jde ruku v ruce s kvalitou vytrénovaného modelu a hlavně s kvalitou korpusu, podle kterého je trénován. Ovšem ze zkušeností získaných pokusem o rozšíření korpusu CNEC2-extended jsme mohli vidět, že rozšíření korpusu o vlastní anotovaná data neznamenalo vylepšení modelu, ale spíše jeho zhoršení. Navíc korpus CNEC2-extended má mnohem více kategorií slov, než můžeme při naší práci s matrikami využít.

Pro rozpoznávání kategorií slov v matričních záznamech se tedy jako nejlepší možnost do budoucna jeví vytvoření vlastního korpusu, vytvořeného přímo pro tento účel. K této části však bude potřeba spoustu vstupních dat, které se stávající úspěšnosti OCR nejdou vytvořit. Pro anotaci jsou totiž potřeba data co možná nejpřesnější, abychom si do korpusu nezaváděly zbytečné chyby.

Kapitola 6

Závěr

Práce na řešení začala studiem problematiky, studiem programu PERO OCR a hlavně možnostmi rozpoznávání předem určených kategoriích slov, vyskytujících se v matričních textech. Z ne čistě informatických částí to pak byla studie matričních knih a různých typů zápisů do nich.

Už při studiu PERO OCR však bylo zjištěno, že na ručně psané texty nedosahuje takové úspěšnosti jakou bychom potřebovali. Nejvíce ohrožující chyby pak pro nás byly ty v analýze textu, které nebyly možné v naší části implementace detekovat, nebo vyřešit. Díky těmto chybám tak mohou být některé záznamy spojeny, nebo může být špatně rozpoznána šablona matriky. Detekce šablon matrik a matričních záznamů, bylo totiž nakonec největší úskalí této práce. PERO OCR nám sice na vstup poskytlo textové polygony a jejich pozice, které by měly být odrazem matriky s tabulkovým rozložením, avšak pouze z geometrických tvarů bez jakékoliv další znalosti strany matriky, je rozdělení takto poskytnutých polygonů velmi složité. I přesto, že se to v této práci nakonec s určitou úspěšností podařilo, bylo by do budoucna lepší přesunout analýzu šablony matriky a rozpoznávání záznamů na stranu OCR.

Rozpoznávání předem určených kategorií slov, bylo prováděno pomocí NER jazykového modelu, vytrénovaného na korpusu CNEC2-extended. Korpus však v základu rozpoznává více kategorií slov, než jsme pro naši práci potřebovali a některé rozpoznávat neumí. Pro tuto práci by tedy bylo lepší vystavět úplně nový korpus specializovaný pouze na práci s matrikami. Bohužel pro tuto činnost v tuto chvíli nemáme dostatek vstupních dat, které by bylo možné anotovat a bohužel ani dostatečně účinné OCR, které by nám tyto data mohlo poskytnout.

Výsledná práce jako celek však ve výsledku nedosahuje na poli umělé inteligence tak špatných výsledků, aby se dala považovat za neúspěch a při zohlednění všech problémů a úskalí se kterými bylo nutné se při tvorbě výsledné aplikace popasovat, můžeme tuto práci brát minimálně jako jakýsi opěrný bod při další práci s touto problematikou.

Literatura

[1] *Studie z korpusové lingvistiky*. Karolinum, 2000. ISBN 80-7184-893-X.

[2] Bartůněk, V. Historický vývoj matrik. *Časopis rodopisné společnosti v Praze*. Praha: Rodopisná společnost v Praze. Duben 1940, sv. 12, č. 1, s. 6–17. ISSN 1805-6490.

[3] David, J. *Souhrn odkazů a užitečných informací k rodopisnému bádání*. 1. vyd. 2014.

[4] Doskočil, K. Vývouj farních matrik v českých zemích. *Časopis rodopisné společnosti v Praze*. Praha: Rodopisná společnost v Praze. Září 1940, sv. 12, č. 2, s. 41–50. ISSN 1805-6490.

[5] DPTeam. *What is a Data Pipeline?* [online]. 2021 [cit. 2023-01-07]. Dostupné z: https://www.datapipelines.com/blog/what-is-a-data-pipeline/.

[6] Grüning, T., Leifert, G., Strauss, T. a Labahn, R. A Two-Stage Method for Text Line Detection in Historical Documents. 2018. Dostupné z: http://arxiv.org/abs/1802.03345.

[7] Jiří Hlavenka, T. B. *Nový výkladový slovník výpočetní techniky*. Praha: Computer Press, 1995. ISBN 80-85896-13-3. Dostupné z: https://ndk.cz/uuid/uuid:6fc9f0a0-fed3-11e8-a5a4-005056827e52.

[8] Josef Miroslav Pražák, J. S. *Latinsko-český slovník*. KLP-Koniasch Latin Press, 1999. ISBN 80-85917-51-3. Dostupné z: https://ndk.cz/uuid/uuid:6fc9f0a0-fed3-11e8-a5a4-005056827e52.

[9] Kiss, M., Benes, K. a Hradis, M. AT-ST: Self-Training Adaptation Strategy for OCR in Domains with Limited Transcriptions. *CoRR*. 2021, abs/2104.13037. Dostupné z: https://arxiv.org/abs/2104.13037.

[10] Kodym, O. a Hradis, M. Page Layout Analysis System for Unconstrained Historic Documents. *CoRR*. 2021, abs/2102.11838. Dostupné z: https://arxiv.org/abs/2102.11838.

[11] Lotko, E. *Slovník lingvistických termínů pro filology*. Univerzita Palackého, 1999. ISBN 80-7067-965-4.

[12] Maur, E. Vývoj matričního zápisu v Čechách. *Historická demografie*. Praha: Historický úst. ČSAV. 1972, sv. 12, č. 2, s. 40–55. ISSN 0323-0937.

[13] Melkesová, M. Církevní matriky českých zemí v pozornosti badatelů. *Historická demografie*. Praha: Historický úst. ČSAV. 2008, sv. 32, s. 5–56. ISSN 0323-0937.

[14] Nečas, J. E. *Deutsch-böhmische juridische Terminologie =: Německo-české názvosloví právnické*. Brünn: C. Winkler, 1893. Dostupné z: https://www.digitalniknihovna.cz/mzk/uuid/uuid:8835a880-a21f-11e2-bc29-005056825209.

[15] Papadopoulos, C., Pletschacher, S., Clausner, C. a Antonacopoulos, A. The IMPACT Dataset of Historical Document Images. In: *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. New York, NY, USA: Association for Computing Machinery, 2013, s. 123–130. HIP '13. DOI: 10.1145/2501115.2501130. ISBN 9781450321150. Dostupné z: https://doi.org/10.1145/2501115.2501130.

[16] Pilát, M. *Neuronové sítě - konvoluční sítě a zpracování obrazu* [online]. [cit. 2023-01-07]. Dostupné z: https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-konvolucni-site-zpracovani-obrazu/.

[17] Sánchez, J.-A., Romero, V., Toselli, A. H. a Vidal, E. ICFHR2014 Competition on Handwritten Text Recognition on Transcriptorium Datasets (HTRtS). *2014 14th International Conference on Frontiers in Handwriting Recognition*. 2014, s. 785–790.

[18] Sánchez, J.-A., Romero, V., Toselli, A. H., Villegas, M. a Vidal, E. ICDAR2017 Competition on Handwritten Text Recognition on the READ Dataset. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 2017, sv. 01, s. 1383–1388.

[19] Ševčíková, M., Žabokrtský, Z. a Krůza, O. Named Entities in Czech: Annotating Data and Developing NE Tagger. In: Matoušek, V. a Mautner, P., ed. *Text, Speech and Dialogue*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, s. 188–195. ISBN 978-3-540-74628-7.

[20] suvratarora06. *Understanding Semantic Analysis – NLP* [online]. 2021 [cit. 2023-01-08]. Dostupné z: https://www.geeksforgeeks.org/understanding-semantic-analysis-nlp/.

[21] Wolff, R. *Semantic Analysis, Explained* [online]. 2020 [cit. 2023-01-08]. Dostupné z: https://monkeylearn.com/blog/semantic-analysis/#:~:text=Semantic%20Analysis%20Techniques-,What%20Is%20Semantic%20Analysis%3F,words%20in%20a%20particular%20context./.

[22] Wood, T. *What is a F-score?* [online]. 2023 [cit. 2023-04-24]. Dostupné z: https://deepai.org/machine-learning-glossary-and-terms/f-score.

[23] Česko. *Fragment f2075913 zákona č. 301/2000 Sb., o matrikách, jménu a příjmení a o změně některých souvisejících zákonů - znění od 1. 2. 2022.* [online]. AION CS, 2010-2022 [cit. 25. 12. 2022]. Dostupné z: https://www.zakonyprolidi.cz/cs/2000-301#f2075913.

Příloha A

Instalace a použití aplikace

Aplikace je navržena tak, aby vstup ve formátu .xml z PERO OCR, kterým byly zpracovány matriční knihy v tabulkovém rozložení, zpracovaly a pomocí semantické analýzy a NER jazykového modelu vrátili důležité informace, které se v matrice vyskytují.

A.1 Instalace

Program byl vyvíjen v jazyce **Python 3.10.11**. Nejdříve je tedy nutné si tuto verzi Python nainstalovat. K práci s knihovnami bude využívána knihovna knihovna **virtualenv** proto je nejdříve nutné tuto knihovnu nainstalovat.

```
sudo apt install python3-venv
```

Pomocí této knihovny si následně vytvoříme venv pro knihovny naší aplikace v kořenovém adresáři aplikace.

```
python3.10.11 -m venv venv
```

A aktivujeme ji.

```
source test/test_env/bin/activate
```

Poté už stačí pouze stáhnout to této virtual enviroment knihovny potřebné pro práci s aplikací.

```
pip3 install -r src/requirements.txt
```

A.1.1 Nastavení exportu na Google Disk

Před exportem je nutné získat vlastní **google_credentials.json** pro přístup na google disk. Nastavování probíhá na stránce: https://console.cloud.google.com/iam-admin/ kde je potřeba nastavit si servisní účet a pro tento účet si vygenerovat již zmíněný **google_credentials.json** a povolit aplikace **Google Sheets API** a **Google Drive API**. **Nesmíte také zapomenou přidělit práva u servisního účtu i svému klasickému google účtu.**

A.2 Použití programu

Použití: **main.py** [-h] [-f F] [-d D] -o O

-h, –help Zobrazí pomocnou zprávu pro spuštění programu.

-f F Vstupní xml soubor v PAGE formátu.

-d D Vstupní složka obsahující xml soubory v PAGE formátu.

-o O Jméno výstupního souboru.

A.2.1 Přílkad pužití

```
python3.10 src/main.py -f
./Scans/fixed/50/216000010-000253-003368-000000-000050-000000-00-B08716-01510.xml
-o 50_test
```

A.3 Testování

Na testování aplikace byla použita knihovna **pytest** a testy jsou uloženy v adresáři **test/**. Testování je dobré spouštět s argumenty **-v -s** pro lepší přehlednost a hlavně všechny výstupy.

Příklad použití

```
pytest -v -s test/
```

Pro **test_output_category** je nejdříve nutné do používané složky v Google Disk vložit soubory ze složky **Scans/final_output**

Příloha B

Obsah adresáře

- `/`
  - `output/` ..... Složka s vytrénovanými jazykovými modely
    - `model-best`
    - `model_CPU`
    - `model_own_anotation`
  - `Scans/` ..... Složka se vstupními soubory xml ve formátu PAGE
    - `accuracy_test`
    - `final_output`
    - `fixed_fixed` ..... Složka s opravenými výstupy OCR, které je možné použít pro testování aplikace
    - `ocr_text_fixed`
    - `ocr_text_original`
    - `original` ..... Složka s originálními OCR výstupy, které je možnéí použít pro testování aplikace
  - `src/` ..... Složka se zdrojovými kody aplikace
    - `data_structures`
    - `pero_ocr` ..... Složka s částí aplikace PERO OCR využívaná pro parsování xml souboru
  - `tests/` ..... Složka s testy k aplikaci
  - `README.md` ..... Soubor s návodem k použití a zprovoznění aplikace