ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F3

Fakulta elektrotechnická
Katedra počítačů

Diplomová práce

Metody strojového učení pro extrakci novinových článků ze souborů PDF

Bc. Tomáš Zach

Studijní program: Otevřená Informatika, Obor: Softwarové inženýrství

Květen 2021

Vedoucí práce: Ing. Jan Drchal, Ph.D.

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Zach Tomáš** Personal ID number: **420656**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Software Engineering**

II. Master's thesis details

Master's thesis title in English:

Machine Learning for News Article Layout Extraction from PDF Files

Master's thesis title in Czech:

Metody strojového učení pro extrakci novinových článků ze souborů PDF

Guidelines:

The goal of the thesis is to develop, implement, and evaluate machine learning methods to analyze news page layouts to extract separate articles. The input data involve rendered pages as well as metadata extracted from source PDF files (e.g., text segments, textbox or figure positions, font sizes, etc.)
1) Explore the state-of-the-art methods of layout extraction.
2) Design methods to extract separate news articles based on structured PDF data. Approach the varying counts of page elements utilizing methods of multiple instance learning. Consider improving the quality of segmentation using NLP or possibly image recognition methods.
3) Evaluate the proposed methods on own dataset or possibly other dataset supplied by the supervisor. Focus mainly on the assessment of feature importance.

Bibliography / sources:

[1] Eskenazi, Sébastien, Petra Gomez-Krämer, and Jean-Marc Ogier. "A comprehensive survey of mostly textual document segmentation algorithms since 2008." Pattern Recognition 64 (2017): 1-14.
[2] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
[3] Using Neural Network Formalism to Solve Multiple-Instance Problems, Tomáš Pevný, Petr Somol, 2016
[4] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. In Advances in neural information processing systems (pp. 3391-3401).
[5] Lee, Juho, Yoonho Lee, and Yee Whye Teh. "Deep amortized clustering." arXiv preprint arXiv:1909.13433 (2019).

Name and workplace of master's thesis supervisor:

Ing. Jan Drchal, Ph.D., Department of Theoretical Computer Science, FIT

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **10.02.2021** Deadline for master's thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Ing. Jan Drchal, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt	Student's signature

Poděkování / Prohlášení

Chtěl bych poděkovat vedoucímu práce Ing. Janu Drchalovi, Ph.D. za možnost zabývat se zajímavým tématem a též za vstřícný přístup, cenné rady a čas, který mi věnoval při psaní této práce. Dále děkuji své rodině za podporu v průběhu celého studia.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 21. 5. 2021

.....

Abstrakt / Abstract

Tato práce se zabývá metodami strojového učení pro extrakci rozložení novinový článků ze souborů PDF. Nejdříve jsou představeny některé stávající metody extrakce následované teoretickým základem týkajícím se umělé inteligence a klasifikace. Dále je zde zmíněna tvorba veřejné datové sady, včetně nástrojů k jejímu zpracování. Následuje představení použitých technologií a konečně samotný Multiple Instance Learning model neuronové sítě, na kterém je provedena řada experimentů zakončená jejich vyhodnocením.

Klíčová slova: umělá inteligence, strojové učení, Multiple Instance Learning, neuronová síť, PyTorch, Python

This work deals with machine learning methods for layout extraction of newspaper articles from PDF file. At first, some existing methods of layout extraction followed by theoretical background concerning artificial intelligence and clasification are introduced. Next creation of public dataset is done, including tools for its processing. Then used technologies are introduced and finally the Multiple Instance Learning model of neural network, which undergoes series of experiments finished with evaluation.

Keywords: Artificial Intelligence, Machine Learning, Multiple Instance Learning, Neural Network, PyTorch, Python

Title translation: Machine Learning for News Article Layout Extraction from PDF Files

Obsah /

Tabulky / Obrázky

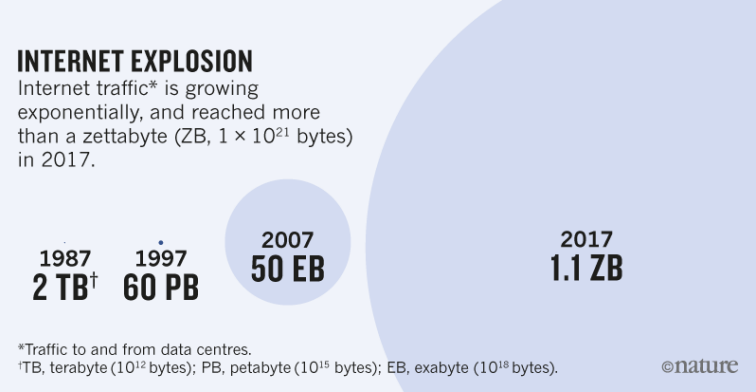
Kapitola 1
Úvod

Žijeme v době informační. Z původního ARPANETu, coby vojenského projektu, se vyvinul dnešní internet. Kde se dříve pohybovaly stovky uživatelů, jsou jich nyní připojeny miliardy každý den.

Spolu s uživateli internetu a technickým rozmachem vznikla spousta nových problémů. Kde data ukládat, jak k nim přistupovat, jak minimalizovat spotřebu energie datacenter, jak účinně bránit zneužívání dat a služeb, jak co nejrychleji najít v miliardách webových stránek kýženou informaci.

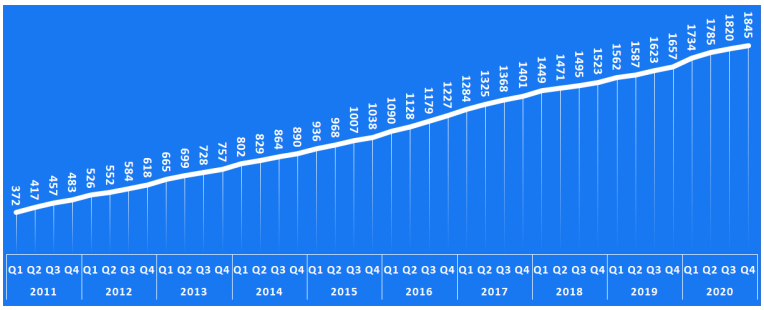
Podívejme se na následující časovou osu, která ve zkratce dokládá překotný vývoj objemu síťového provozu, počtu uživatelů na internetu, denně posílaných zpráv apod.

- 1979 protokoly TCP/IP, na kterých internet stojí, získávají finální podobu
- 1987 síťový provoz na internetu - do a z datacenter - 2TB(tera 10^{12})
- 1997 síťový provoz na internetu - do a z datacenter - 60PB (peta 10^{15}) / 30.000x více než před 10 lety
- 2002/10 Torch první vydání - knihovna pro ML (Machine learning) a DL (Deep learning)
- 2004/12 Facebook má zhruba 1 milion uživatelů
- 2007 síťový provoz na internetu - do a z datacenter - 50EB(exa 10^{18}) / 833x více než před 10 lety
- 2011/Q1 Facebook DAU (Daily active users) 372 milionů uživatelů
- 2012/6 Google X Lab implementoval ML algoritmus, který autonomně procházel YouTube a detekoval videa s kočkami
- 2014/5 Google experimentuje s použitím ML pro optimalizaci chlazení (tedy spotřeby elektrické energie) svých datacenter
- 2015/Q1 Facebook nasazuje DeepFace (DL systém na rozpoznávání lidských tváří na digitálních fotografiích) - přesnost 97,35%
- 2016/7 Goggle testuje svou „DeepMind AI“ v živém provozu v jednom datacentru - dosahuje redukce spotřeby el. energie na chlazení až o 40%
- 2016/Q3 Facebook DAU 1.179 milionů uživatelů - tj. 3,2x více než v 2011/Q1
- 2016/9 je vydán PyTorch alpha od Facebook's AI Research lab (FAIR)
- 2017 síťový provoz na internetu - do a z datacenter - 1,1ZB(zetta10^{21}) / 22x více než před 10 lety



Obrázek 1.1. Vývoj síťového provozu.[1]

- 2017/2 Torch poslední stabilní release - vývoj ukončen
- 2018/8 Google nasazuje svůj AI systém na řízení chlazení datacenter do ostrého provozu - lidská obsluha figuruje jen v úloze supervizora (úspora el.energie a tedy i snížení produkce CO2)
- 2019 pomocí Facebook Messengeru uživatelé pošlou cca 20 miliard zpráv měsíčně
- 2019/6 uživatelé na Facebooku zveřejní 4,3 miliardy zpráv denně
- 2020/Q4 Facebook DAU 1.845 milionů uživatelů - tj. 5x více než v 2011/Q1



Obrázek 1.2. Vývoj počtu denně aktivních uživatelů FB (v mil.)[2]

- 2021/3 PyTorch aktuální stabilní release
- 2021/4 Facebook/WhatsApp doručuje 100 miliard zpráv denně

Nelze se totiž divit tomu, že konvenční algoritmy v určitou chvíli už nestačily na zpracování takového objemu generovaných dat. To přirozeně vyvolalo poptávku po nových metodách řešení těchto problémů. Ačkoliv teoreticky byly koncepty například strojového učení rozpracovány dlouho (Arthur Samuel např. prvně použil termín „Machine Learning“ v roce 1952)[3], aby vývoj v této oblasti skutečně akceleroval, muselo nastat několik příznivých okolností dohromady:

- nárůst výkonnosti hardware
- dříve nepředstavitelné objemy dat uložené v datacentrech (nikdo nemá tak rozsáhlou sbírku fotografií jako Facebook nebo videí jako Google/YouTube) či denně pořizovaných (žádná aplikace nemá téměř 2 miliardy denně aktivních uživatelů, kteří denně publikují několik textových příspěvků)

[1] převzato z `https://www.nature.com/articles/d41586-018-06610-y`
[2] převzato z `https://www.businessofapps.com/data/facebook-statistics/`
[3] `https://www.dataversity.net/a-brief-history-of-machine-learning/`

- a v neposlední řadě existence společností, které jsou ochotny do AI investovat obrovské finanční prostředky a mají je na účtech (např. v roce 2014 Google koupil startup DeepMind za 400 mil. USD, v roce 2019 Microsoft oznámil plán investovat do AI 1 mld. USD)[4]

To je, dle mého názoru, vysvětlení proč se tzv. „technologičtí giganti“ jali velmi aktivně implementovat a rozvíjet koncepty AI - (Artificial Intelligence - umělá inteligence), ML - (Machine Learning - strojové učení). Anebo proč za jednou z nejpoužívanějších open source knihoven pro strojové učení, kterou využívám i ve své práci, stojí Facebook.

1.1 Motivace

Jednou z praktických úloh kterou s sebou přinesl rozvoj internetu a výpočetní techniky, je digitalizace knihoven. Vesměs je realizována pořízením digitální podoby „analogové“ předlohy (např. knihy, časopisu) ve formě bitmapového souboru. Ten se relativně snadno pořídí, avšak pro další zpracování není pochopitelně moc vhodný. Jistě lze bitmapové podklady převézt do textového formátu pomocí OCR (Optical Character Recognition), ale v takovém případě se jednak ztratí informace o rozložení textu a v případě složitějšího rozložení podkladu nemusí být triviální zachovat jeho integritu.

Odtud už je jen krůček k motivům zadání pro mou práci. V dnešní době existují celkem běžně elektronické archivy dokumentů - některé jsou komerční, ale lze najít i veřejně dostupné. Kdyby existoval postup, jak z této digitální předlohy extrahovat obsah a metadata (např. o layoutu) do textového, strojově dobře dále zpracovatelného formátu, dali bychom vědcům z řad sociologů či žurnalistů možnost výstupy dále analyzovat a zpracovávat. Uveďme jako příklad dvě zajímavá témata: analýza za účelem sledování trendů v „bulvarizaci“ periodik - jsou titulky v čase větší a větší nebo zůstávají stejně velké po několik posledních let? nebo s využitím NLP (systémů zpracování přirozeného jazyka) - vykazuje výběr témat na titulní straně deníku, vlastněného přes svěřenecké fondy předsedou vlády, nějaké „anomálie“ v porovnání s jinými deníky?

Kupříkladu už od roku 2019 spolupracují odborníci z ČVUT - Fakulty elektrotechnické a UK - Fakulty sociálních věd na projektu TAČR[5] (Technologická agentura ČR): „Proměna etických aspektů s nástupem žurnalistiky umělé inteligence“. Zde se prostředky strojového učení a umělé inteligence používají při mezioborové spolupráci, integrující žurnalistiku s počítačovými vědami a v rozvíjení principů společenské odpovědnosti novinářů.

1.2 Formulace problému

Máme k dispozici neveřejný dataset deníku Právo, poskytnutý firmou Newton Media ve formátu PDF. Součástí datasetu je i vektorová reprezentace, popisující layout - tedy počet i umístění textových polí na stránce s jejich souřadnicemi a velikostí použitého fontu. Každá stránka může obsahovat rozdílný počet textových polí. Sémantická data (tj. např. rozdělení na články nebo klasifikaci elementů) však nemusí být dostupná.

Záměrem je

- dokázat vytvořit dataset v identickém formátu z veřejně dostupných dat
- extrahovat z tohoto datasetu články v textovém formátu pro možnost dalšího strojového zpracování

[4] https://www.techadvisor.com/feature/small-business/tech-giants-investing-in-artificial-intelligence-3788534/

[5] https://starfos.tacr.cz/cs/project/TL02000288#project-main

1.3 Cíle práce

První část práce

Je vytvořit vlastní VBAS (Vector-Based Article Segmentation) dataset a implementovat nástroje pro vytvoření datové sady z veřejně dostupných dat, která bude vycházet z formátu poskytnutého Newton Media. To umožní zapojit se do tohoto, pokud víme, ještě neřešeného problému vědcům po celém světě. Nutno podotknout, že se zaměřujeme na anglická periodika. PDF soubory obsahují kromě textových polí i obrázky, přičemž každá stránka má různý počet textových polí. Jednotlivá textová pole jsou na sobě navzájem závislá. Nadpis například ovlivňuje rozložení dalších článků. Např. se typicky bude nacházet vlevo nad většinou textových polí daného článku. Při přípravě veřejného datasetu je třeba rozpoznat, které textové boxy patří k jednomu článku.

Druhá část práce

Je vytvořit model neuronové sítě, který pomocí metod strojového učení dokáže ve vstupních PDF nacházet shluky (clustery) textových polí, které tvoří jednotlivé články. Pro zadanou úlohu, které se říká tzv. „learning to cluster“ využiji semi-supervised strojového učení. Naučený model bude pak schopen ze vstupního souboru a jeho vektorové reprezentace extrahovat jednotlivé články v textové podobě, které by bylo možné pak dále strojově zpracovat. K tomu využiji tzv. MIL (Multiple Instance Learning), do kterého vstupují vektory ve formě množiny, což se přesně hodí, protože textových polí na každé stránce může být různý počet.

Úkolem je rozhodnout zda textové boxy patří ke stejnému článku či nikoliv. Do učení vstupují souřadnice textového pole spolu s délkou textu ve znacích. Na základě toho se vyhodnotí dané kritérium a použije se clusterovací algoritmus jehož výsledkem je výčet článků spolu s jejich příslušností do daného shluku (clusteru, čili článku).

Následně budu experimentovat se vstupními daty připravenými s pomocí NER (Named entity recognition). Pro každou dvojici vybraných textových boxů vyextrahuji seznam pojmenovaných entit, který přidám k vlastnostem textových polí zmíněných výše. Pojmenovaná entita je jméno osoby, místa nebo třeba organizace. Zde je malá ukázka:

```
Jde o reakci na turecký nákup ruského protiraketového systému S-400,
který podle Washingtonu ohrožuje bezpečnost USA a NATO.

op S - 400
gr Washingtonu
gc USA
io NATO
```

Ve výše zobrazené větě byly nalezeny čtyři pojmenované entity různých typů, první je typu produkt, druhá a třetí jsou geografické názvy a poslední je název organizace.

Nakonec porovnám přesnost obou přístupů, délky běhu trénovací fáze a přesnosti učení.

Kapitola 2
Teoretický základ

2.1 Existující řešení

V této sekci si shrneme některá existující řešení v problému extrakce rozložení PDF souboru.

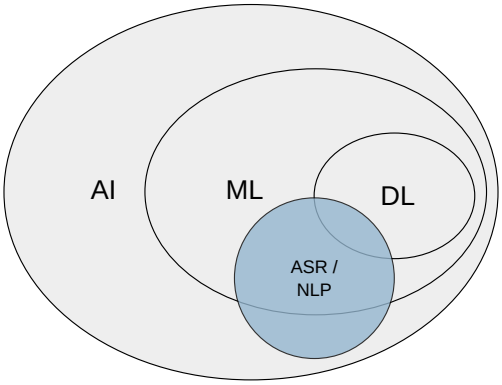
V [1] požívají extrakci rozložení PDF ke klasifikaci diagramů. Nutno podotknout, že exrahují vektory grafických objektů. V další práci [2] rozdělí vstupní PDF soubor na tři různé dokumenty. PDF pouze s textem a dokument s obrázky a poslední s grafikou. Zde z jednotlivých písmen skládají slova a z nich textové řádky postavené na zarovnání. Následující práce [3] představuje systém DOMINUS (DOcument Management INtelligent Universal System). A nakonec [4] používá opensource knihovnu JPedal sloužící k získání textových polí.

2.2 AI - Artificial Intelligence (Umělá Inteligence)

Termín Umělá inteligence (AI) je často používán jako souhrnné označení technik, směřujících k co nejvěrnějšímu napodobení chování lidského mozku, respektive projevů lidské inteligence, prostředky výpočetních techniky. Někdy je termín AI také chápán jako označení vědního oboru, jehož cílem je vytvoření takového (umělého) systému. Nejobecnějším popisem cíle takového systému je, řekněme, „schopnost řešit komplexní úlohy a problémy samostatně, bez aktivní lidské participace“.

Abych použil pro dokreslení konkrétní příměr - představme si robotizaci nějakého parciálního technologického úkonu na výrobní lince průmyslového závodu - třeba transport plechu do lisu a přesun výlisku na transportní pás ve Škodě Auto. To bychom mohli označit termínem „automatizace výrobního procesu“ - neboli nahrazení konkrétní lidské činnosti či úkonu strojem. Na opačném pólu složitosti bychom si mohli představit implementaci „systému řízení plně autonomního vozidla pro běžný provoz za jakýchkoliv povětrnostních podmínek“ to je přesně ona „komplexní úloha“, jejíž úspěšné řešení evidentně není možné bez využití poznatků oboru tzv. Umělé inteligence. Někde mezi těmito dvěma body, na úsečce komplexnosti problému, bychom si mohli představit zadání „implementovat plně automatizovanou linku pro lakování automobilů bez ohledu na konkrétní model“.

Chceme-li, aby se umělý systém podobal lidské inteligenci, musí, zcela nepochybně, zvládnout také proces „učení“. Tj. schopnost, s přibývajícími datovými vstupy a s délkou trvání jeho provozu, optimalizovat své „chování“ - dosahovat například rychlejší a přesnější odezvu. A právě této podmnožině světa AI se věnuje Strojové učení - Machine learning (ML).

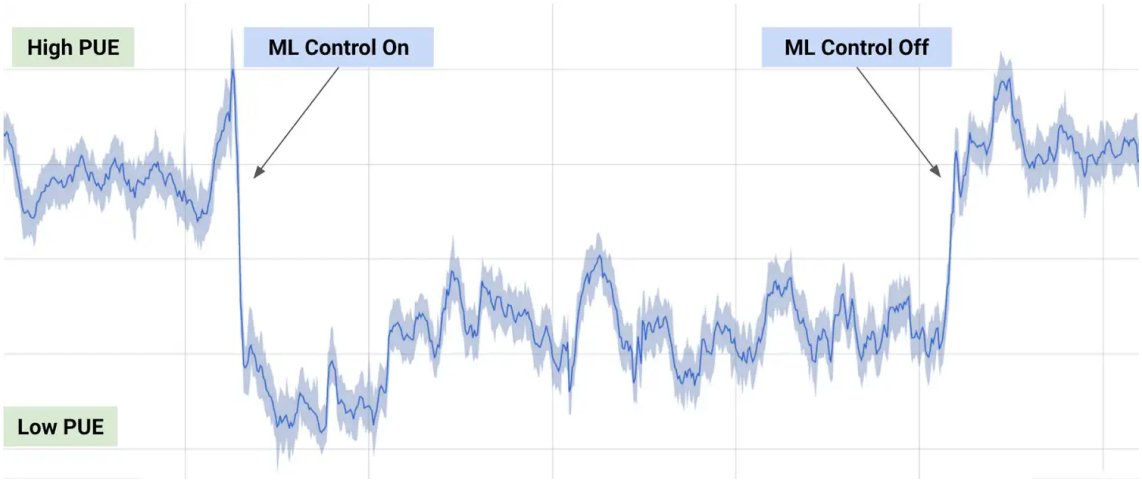


Obrázek 2.1. AI-ML-DL

2.3 ML - Machine Learning (Strojové učení)

Obecně řečeno je Strojové učení (ML) obor, zabývající se algoritmy a technikami, pomocí kterých se stroje (počítače) dokáží zlepšovat v řešení úloh, které plní a to na základě předešlé zkušenosti. Pracuje se s obrovským množstvím dat, které jsou analyzovány s pomocí specifických algoritmů a postupně je budován přesnější a přesnější model. Účinnost se také zvyšuje opakováním algoritmů, kdy jsou vstupní data postupně optimalizována. Své znalosti tedy dokáže ML dále rozšiřovat bez lidské pomoci. O využití ML se už stará umělá inteligence AI - lze tedy říct, že Strojové učení (ML) je nástrojem Umělé inteligence (AI). Tato oblast AI v současné době vykazuje největší příslib do budoucna ve smyslu poskytování již nyní reálně a průmyslově použitelných výstupů.

Na obrázku níže je vidět velmi přesvědčivý doklad o účinnosti opatření, navrhovaných AI s pomocí strojového učení (ML) „DeepMind“, při testování AI/ML v rámci pilotního projektu snižování spotřeby energie pro chlazení Google datacenter. Obrázek zobrazuje hodnotu ukazatele efektivity využívání energie v datacentru (PUE) v čase - čím nižší, tím lepší. Vyznačeny jsou chvíle, kdy bylo řízení spotřeby přepnuto na opatření doporučovaná ML a kdy bylo přepnuto zpět na původní systém regulace.



Obrázek 2.2. Index spotřeby energie při využití ML[1]

[1] převzato z `https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40`

Princip fungování strojového učení můžeme shrnout těmito čtyřmi body:

- **shromažďování a příprava dat;** algoritmus identifikuje zdroje a na základě sestavených dat vytvoří struktury. Data se rozdělí do dvou segmentů – tréninkového a testovacího / validačního
- **trénování modelu** s využitím tréninkové sady dat dochází k vyladění na výkonosti a přesnosti zpracování
- **ověření modelu** probíhá pomoci testovací sady, která vyhodnocuje efektivitu algoritmu

Typické způsoby jak Machine learning zpracovává data jsou například:

- **klasifikace**; rozpoznávané objekty jsou rozdělované do tříd
- **regrese**; odhaduje číselnou hodnotu výstupu podle vstupu
- **shlukování (clusterování)**; vstupní data seskupuje do skupin podobných vlastností, typicky při učení bez učitele - bez plné znalosti obsahu. O tomto způsobu budu ještě dále řeč

Obdobně jako při „lidském“ učení i při strojovém lze využívat různé metody:

2.3.1 Supervised learning - učení s učitelem

V tomto případě je pro daný vstup jasně definován výstupní parametr. Tj. „učitel“ klasifikuje vstupy tak, aby algoritmus později věděl jak se zpracovávaným vstupem naložit. Strojové učení trénované tímto způsobem tak umí pouze to, co jej naučil učitel.

2.3.2 Unsupervised learning - učení bez učitele

Ve druhé nejobvyklejší metodě ke vstupním datům není známý výstup a na postup, řešení i výsledek si algoritmus přichází sám metodou pokus - omyl. Pěkným příkladem je například segmentace zákazníků nebo detekce anomálií síťového provozu.

2.3.3 Semi-supervised learning - kombinované

Jak název napovídá v tomto případě dochází ke kombinaci obou předchozích postupů. Použije se učení bez učitele, bez nutnosti anotace každého treninkového příkladu a zároveň učení s učitelem - typicky menší část vstupních dat je se známým výstupem a větší část bez něj. O tomto typu Strojového učení bude řeč ještě později.

2.3.4 Reinforcement learning - posilované (někdy též zpětnovazební) učení

Algoritmus shromažďuje zkušenosti bez přesné informace o požadovaném výstupu.

2.4 DL - Deep Learning (Hluboké učení)

Techniky Hlubokého učení (DL) jsou to nejmodernější, co AI nabízí. DL se ještě více zaměřuje na podmnožinu nástrojů a technik Strojového učení (ML) a zaměřuje se na řešení reálných problémů s využitím neuronových sítí, napodobujících lidské rozhodování. Aplikuje je na řešení komplexních problémů, které vyžadují „myšlení“ lidské nebo umělé. Deep Learning ke své činnosti využívá vícevrstvé neuronové sítě, vycházející z modelů fungování lidského mozku. Hluboké učení lze použít ke zpracování jakékoliv formy dat - audio, video, řeč, psaná slova - a rychlému vytváření výstupů které vypadají, jako by byly výsledkem lidského zpracování.

Pěknými příklady je třeba rozpoznávání lidských tváří na digitálních fotografiích Facebook/DeepFace, tzv. virtuální asistenti Siri / Cortana / Alexa, chatboti

Specifickými aplikacemi AI/DL jsou například viz. obr. 2.1

- ASR - Automatic Speech Recognition - systémy pro automatické rozpoznávání řeči;
- NLP - Natural Language Processing - systémy zpracování přirozeného jazyka.

2.5 MIL - Multiple Instance Learning

Multiple Instance Learning (MIL) je typem učení s učitelem tzv. supervised learning, kde instance datových vzorků jsou uspořádány ve formě množiny, kterým se říka tzv. „bagy“. Na rozdíl od standardního strojového učení, kde každý vzorek má formu vektoru fixní délky. Vektorům tvořícím bag se říká instance. Anotace jsou dostupné na úrovni bagu a ne pro každou jednotlivou instanci. Tradičně, MIL se používal na problémy binární klasifikace, kde bag je označen jako pozitivní, pokud obsahuje alespoň jednu pozitivní instanci a jinak je označen jako negativní, což je, když všechny instance jsou negativní. Tento typ nerovnoměrného rozdělení má četné aplikace v různých oblastech, kde anotace pro jednu z tříd nejsou známé nebo chybí úplně. Například klasifikace spamových e-mailů, detekce podvodů kreditních karet, klasifikace medicínských dat, klasifikace obrázků a video analýza. Cílem MIL je naučit se základní koncept, který správně předpovídá anotaci bagu daných neanotovaných instancí.

Formálně, buď $\mathcal{X}$ vstupní tzv. feature prostor a $\mathcal{Y}$ buď výstupní anotační prostor. Standardní proces učení s učitelem zahrnuje naučení se mapování $\mathcal{X} \mapsto \mathcal{Y}$ z množiny n trénovacích vzorků $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, kde $x_i \in \mathcal{X}$ je trénovací vzorek a $y_i \in \mathcal{Y}$ je anotace asociovaná s x_i.

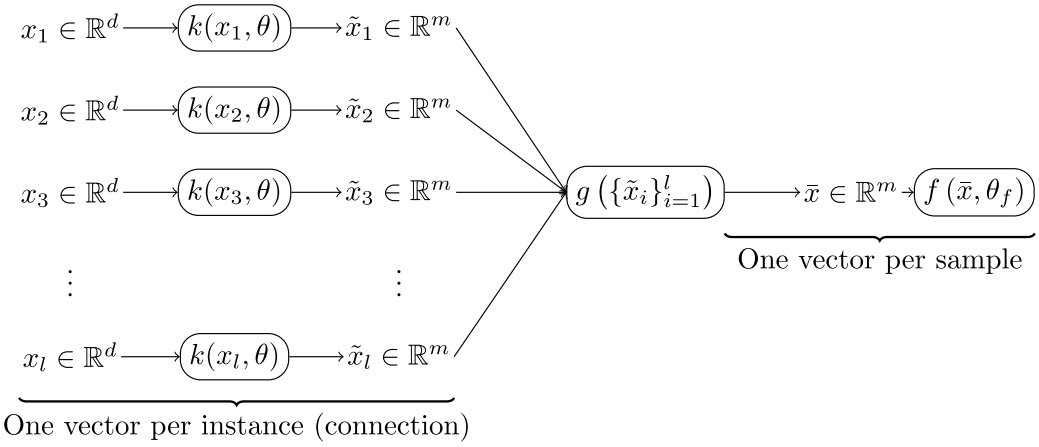
V MIL, trénovací množina $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_m, Y_m)\}$ se skládá z m anotovaných bagů $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ik}\}$ a cílem je naučit se mapování $f_{MI}: 2^{\mathcal{X}} \mapsto \mathcal{Y}$.

Nedávný úspěch v hlubokém učení vedl k dalšímu vývoji umělých neuronových sítí schopných řešit MIL problémy. Bylo prokázáno, že provádění úloh s učitelem na množinách je nutná a postačující podmínka použít permutačně-invariantní funkci, která mapuje vstupní bagy na cílovou proměnnou[5]. Permutačně-invariantní mapování zachovává výstup s ohledem na různá uspořádání prvků vstupní množiny na vstupech sítě.

Neuronová architektura uvedená v [6] zapojuje vrstvu zpracování na úrovni instancí s permutačně-invariantní agregační vrstvou následovanou klasifikátorem. Buď $\mathcal{X}$ neprázdný prostor instancí. Formulace předpokládá, že každý bag $B \subseteq \mathcal{X}$ je realizací nějaké náhodné proměnné s pravděpodobnostním rozdělením produkujícím jednotlivé instance a $y \in \mathcal{Y}$ je anotace bagu. Cílem modelu je naučit se rozlišovací funkci $f: \mathcal{B} \to \mathcal{Y}$, kde $\mathcal{B}$ je množina všech možných realizací všech možných rozdělení nad $\mathcal{X}$. Dopředný průchod danou architekturou je formulován následovně:

$$F(B; \theta, \theta_f) = f(g(\{\phi(x_i; \theta) | x_i \in B\}), \theta_f) \tag{2.4}$$

kde $\phi: \mathcal{X} \to \mathbb{R}$ je parametrizované mapování, které promítá jednotlivé instance do prostoru reálných vektorů, $g: \mathcal{P}^{\mathbb{R}^m} \to \mathbb{R}^m$ je agregační funkce a $f: \mathbb{R}^m \to \mathcal{Y}$ je síť klasifikátoru.



Obrázek 2.3. Náznak neuronové sítě optimalizující vektorovou reprezentaci v embedding-space paradigmatu.[2]

V této práci vyžiji MIL přístupu kvůli předem neznámému počtu textových polí na stránce, přičemž nezáleží na jejich pořadí.

2.6 K-Means

Jedná se o formu učení bez učitele. Metoda využívá jednoduchý způsob, jak rozdělit data pomocí předem definovaného počtu shluků. Hlavní myšlenkou metody je nějakým způsobem definovat K těžišť (centroidů) každého shluku. Dalším krokem je spojit každý jednotlivý bod s nejbližším těžištěm. Pokud takto spojíme všechny body, je první seskupování hotové. Po tomto seskupení se přepočtou těžiště. Jednotlivé body se opět spojí s nejbližšími těžišti. Tím vzniká cyklus, ve kterém v každém kroku všech k těžišť mění své umístění. Cyklus je ukončen ve chvíli, kdy přestanou probíhat změny, jinými slovy ve chvíli, kdy se těžiště přestanou hýbat. Obecně se dá postup metody popsat čtyřmi kroky:

1. Urči k bodů a označ je jako těžiště shluků.
2. Každý bod přiřaď nejbližšímu těžišti. Body přiřazené jednomu těžišti označ za shluk.
3. Pro každý shluk najdi nové těžiště.
4. Kroky 2 a 3 se opakuj, dokud se těžiště nepřestanou pohybovat.

Pro více informací odkazuji čtenáře na publikaci [7].

[2] převzato z [6]

Kapitola 3
Základní kontext

Motivace

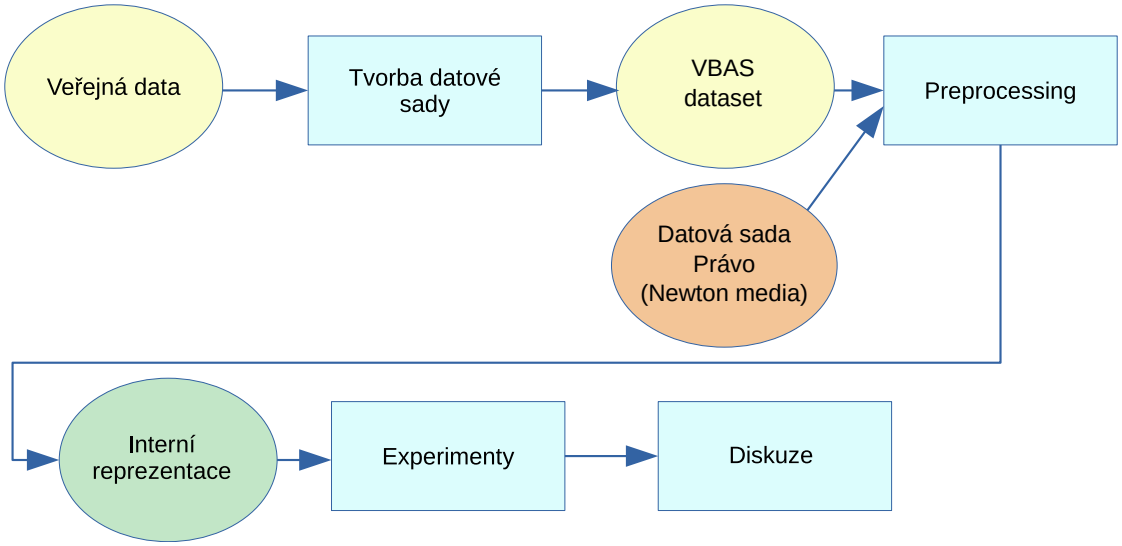
Cílem práce je implementace extrakce novinových článků z jejich elektronické verze, uložené ve formátu PDF, s využitím metod strojového učení. Vedlejším produktem je sada transformačních skriptů a nástrojů, které umožní data z otevřených veřejných zdrojů zkonvertovat do podoby VBAS (Vector based article segmentation) data setu, kompatibilního se vzorkem Právo (poskytnutého společností Newton Media). Což otvírá možnost dále prezentovaným postupem extrahovat textovou podobu takřka libovolného elektronického archivu deníků či časopisů. Získanou textovou podobu článků lze pak dále zpracovávat technikami NLP (Natural language processing).

Formát PDF, používaný pro ukládání článků, novin, časopisů, katalogů, vědeckých prací i knih, postupně získal obrovskou popularitu a etabloval se jako de-facto standard pro výměnu a distribuci popsaného obsahu po internetu. Jeho obrovskou výhodou je platformová nezávislost - distributor obsahu má (bezmála) jistotu, že si uživatel soubor otevře a přečte úspěšně ať už na desktopových systémech s operačním systémem Windows, Mac i Linux či na mobilních zařízeních (tabletech a chytrých telefonech). Bez ohledu na to, zda na nich běží Android či iOS. Popularita formátu je patrná i z jeho podpory, integrované v hlavních internetových prohlížečích - Chrome, Firefox nebo Edge. Mezi další důvody jeho obliby patří, že si poradí jak s textovým tak grafickým obsahem - fotky, obrázky, grafy anebo také možnost odkazovat se, například z obsahu, na jednotlivé kapitoly.

Nepřekvapí tedy, že elektronické verze novinových článků jsou distribuovány a archivovány - převážně - ve formátu PDF. Problém ovšem nastává ve chvíli, kdy je třeba extrahovat textový obsah z těchto souborů pro další strojové / počítačové zpracování. Vstupních souborů je mnoho, mají různou vnitřní strukturu stránek (layout) a není triviální detekovat, které textové segmenty na stránce náleží k témuž článku. A to je problémová situace „jak dělaná“ pro nasazení algoritmů strojového učení (Machine learning), použití softwarové implementace neuronových sítí (Neural network) - tedy pro umělou inteligenci (Artificial intelligence), řečeno populárně.

3.1 Schéma zpracování

Na následujícím obrázku je naznačena logická posloupnost kroků na jejichž konci je porovnání výstupních metrik neuronové sítě mnou zvolené metody strojového učení se dvěma reprezentacemi textových polí (bez a s NER).

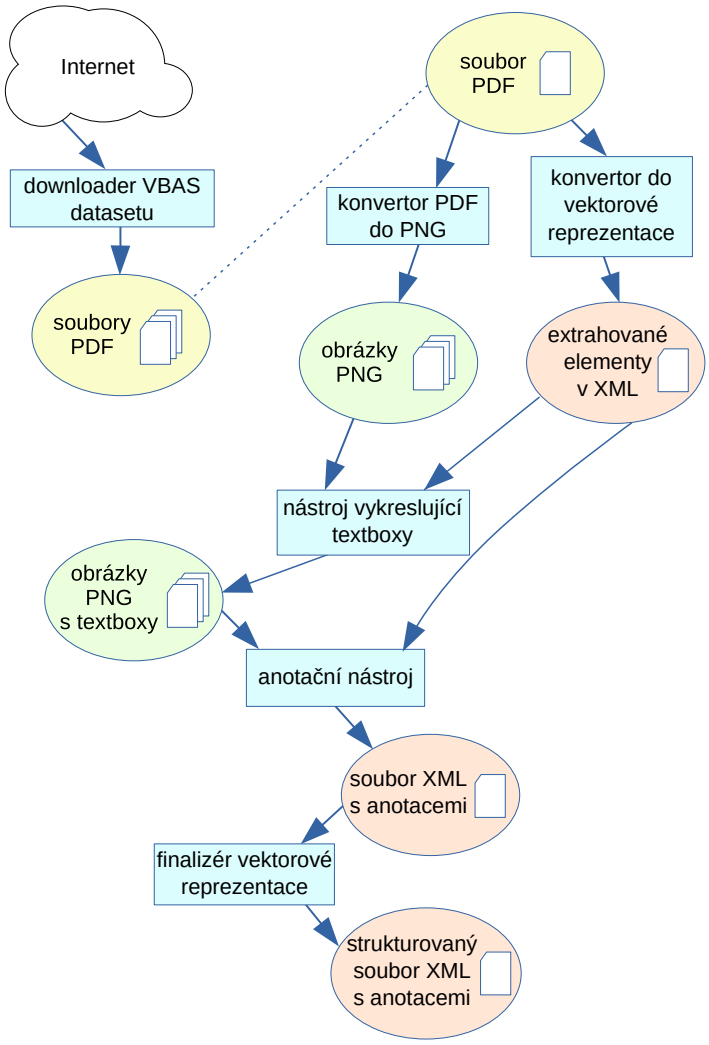


Obrázek 3.1. Logické schéma práce

- oválné prvky ve schématu odpovídají různým datovým formátům v průběhu zpracování
- modře podbarvené prvky odpovídají následujícím kapitolám práce, kde vysvětlím další detaily

Kapitola 4
Tvorba datové sady

Tato kapitola pojednává o vytvoření VBAS (Vector-Based Article Segmentation) datasetu. Dále vysvětlím logiku zpracování, implementované skripty a nástroje, použité datové formáty. Cílem je získat data ve formátu stejném jako formát datové sady Právo poskytnutý Newton Media (na obr. 4.1 označený jako strukturovaný soubor XML s anotacemi).



Obrázek 4.1. Schéma přípravy anotovaného XML souboru

4.1 Použité programy a skripty

Následující sekce shrnuje použité aplikace v celém procesu přípravy VBAS datasetu.

4.1.1 Downloader VBAS datasetu

Skript na stažení PDF souborů VBAS datasetu je napsán pro BASH, využívá program **wget** a jmenuje se **downloadpub.sh**.

4.1.2 Konvertor do vektorové reprezentace

Pro extrakci elementů z PDF do XML souboru jsem napsal skript založený na balíčku **Pdfminer** s názvem **pdfextract.py**.

4.1.3 Konvertor PDF do PNG

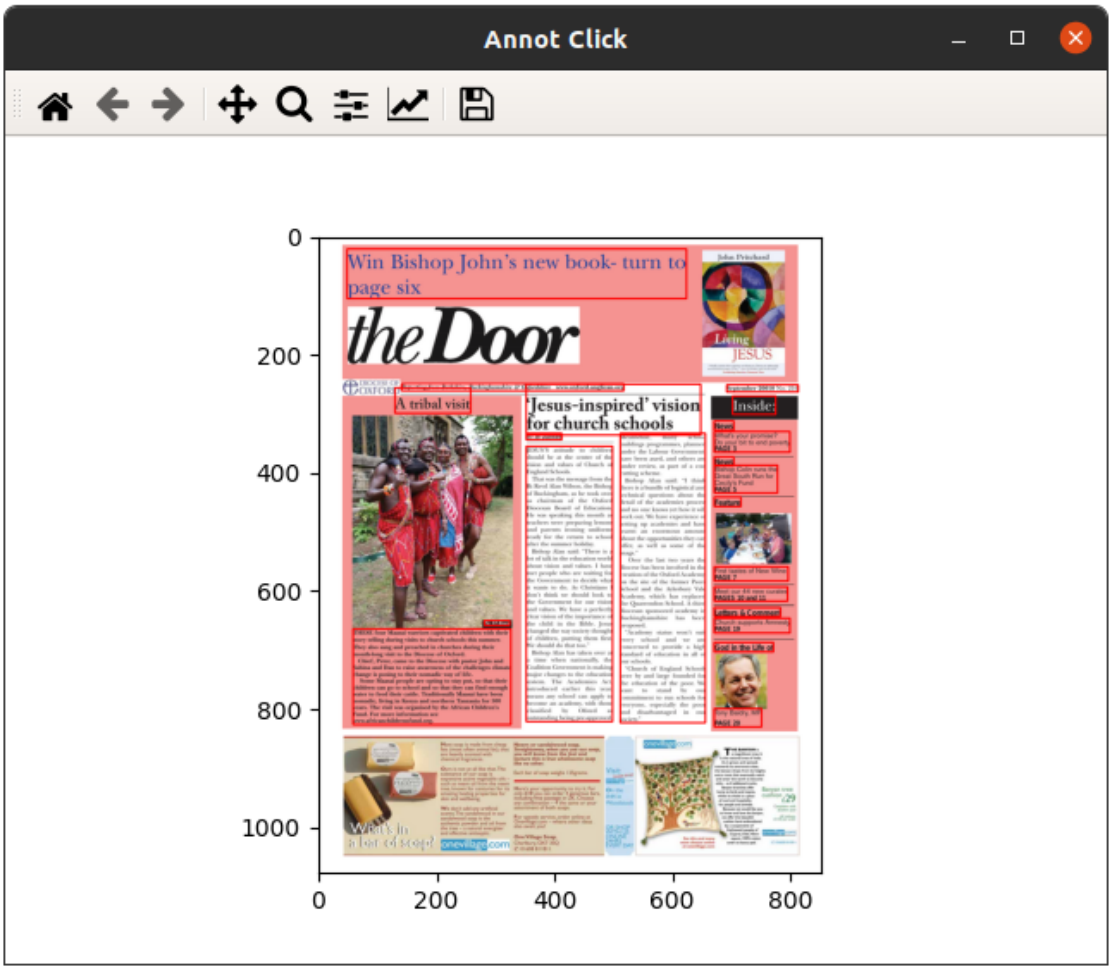
Ke konverzi PDF do obrázků PNG stránek využívám program **pdftoppm** - „PDF to portable pixmap converter“

4.1.4 Nástroj vykreslující textová pole

Do skriptu k vykreslení textových boxů do obrázků stránek - **XMLParser.py** vždy vstupuje soubor s extrahovanými elementy XML a obrázky stránek ve formátu PNG.

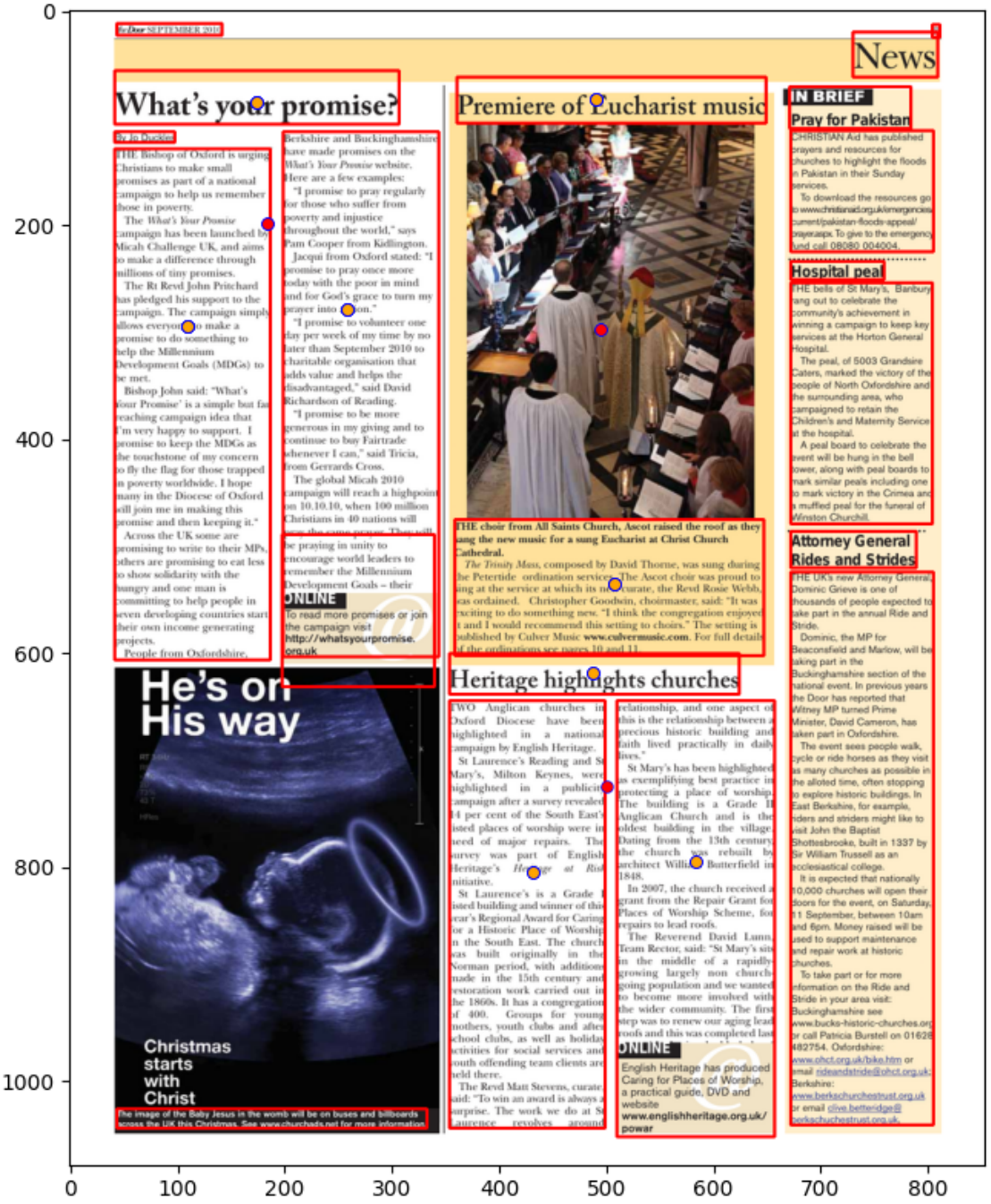
4.1.5 Anotační nástroj

Anotační nástroj načítá data ve formátu, který je výstupem skriptu **XMLParser.py**. Jedná se tedy o obrázky stránek ve formátu PNG s vykreslenými textovými boxy a soubor XML s extrahovanými elementy, jehož formát bude popsán níže, který je upravován anotačním nástrojem. Program jsem naprogramoval v jazyce python a je založen na knihovně **matplotlib** a interaktivním grafickém backendu **Qt5Agg**.



Obrázek 4.2. Hlavní okno anotačního nástroje

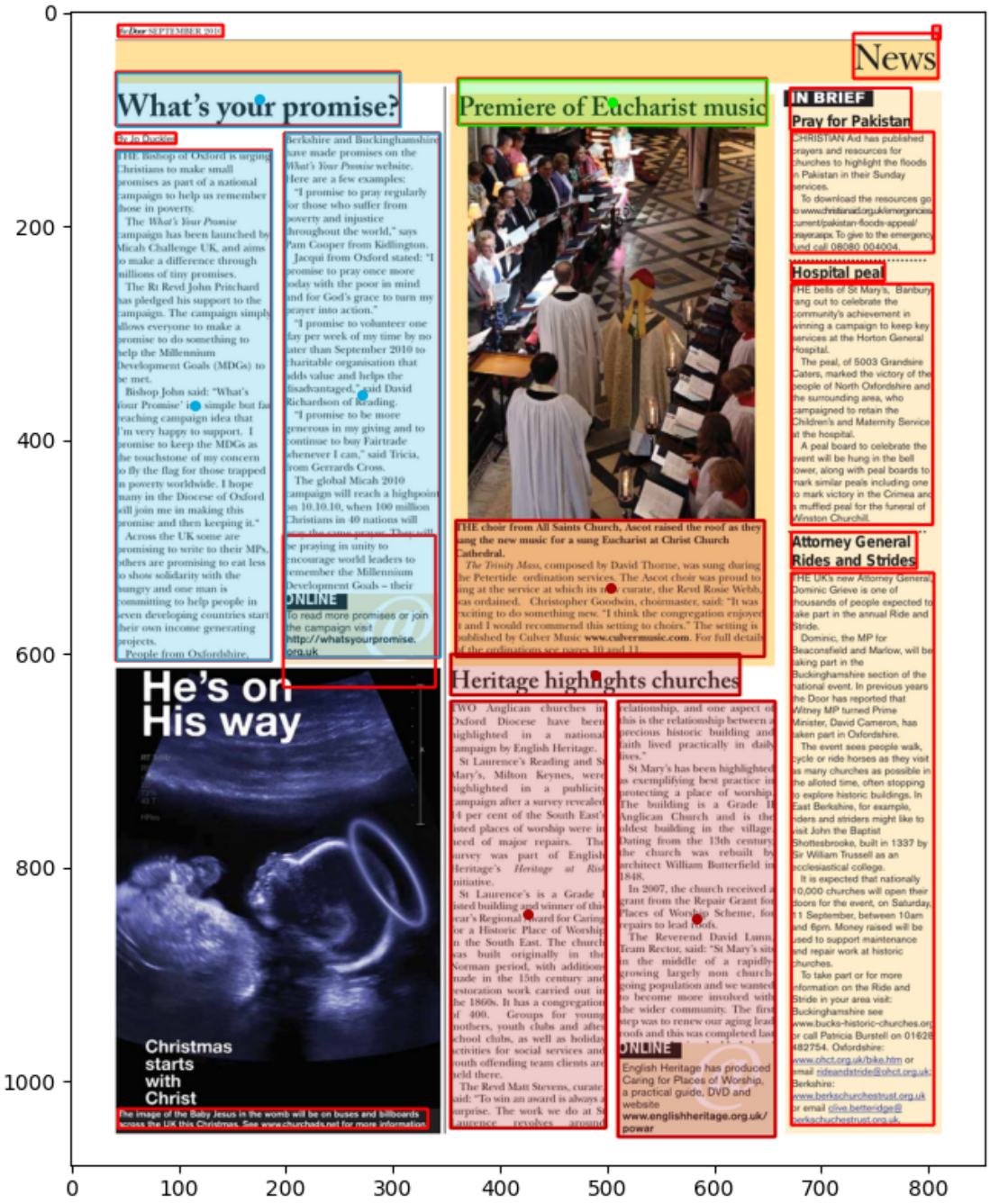
Po spuštění nástroje se otevře hlavní okno s prvním obrázkem stránky PDF souboru jak zobrazuje obr. 4.2. Anotátor se ovládá následovně: šipkami vlevo, vpravo si vyberete požadovanou stránku. Potom kliknutím levého tlačítka myši zvolíte bod v textovém boxu indikovaný oranžovou tečkou. Pokud chcete přidat tzv. „clusterovací bod“ kliknete spolu s klávesou shift. Ten je indikován červeně. Používá se jako střed clusteru pro následující krok.



Obrázek 4.3. Anotační nástroj po manuálním vybrání třech článků levým tlačítkem myši

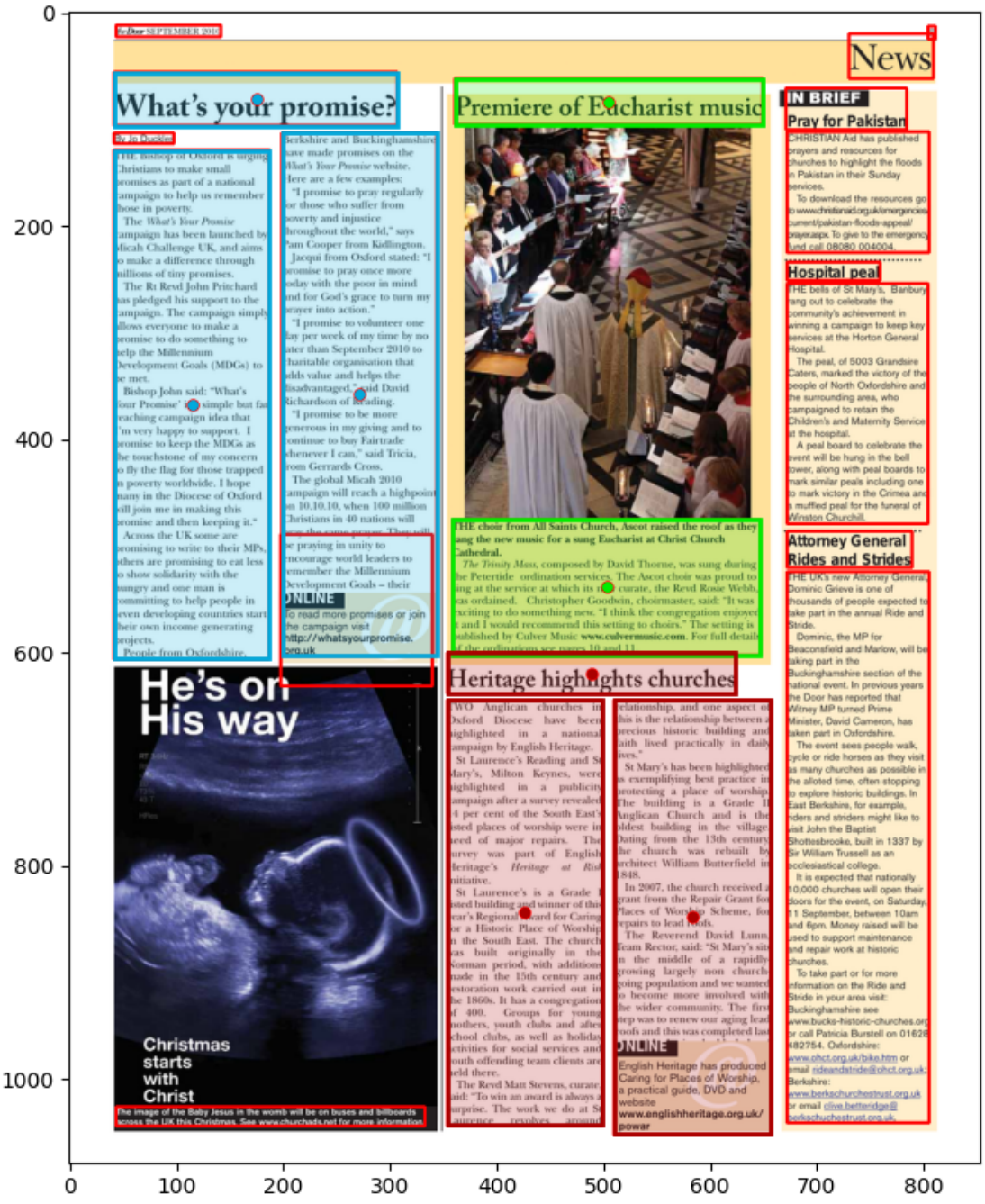
Pokud mám vybrané všechny požadovaná textová pole, mohu spustit algoritmus **K-MEANS**, podrobněji popsán v [8]. Jeho smyslem je inicializace vybraných článků, aby člověk nemusel volit u každého jednotlivého textového pole příslušnost ke clusteru. Do klasifikačního algoritmu vstupují souřadnice vybraných bodů. Algoritmus na základě jejich vzdálenosti od reprezentanta shluku (zobrazen červeně) vytvoří informaci o tom, zda textové pole patří do stejné skupiny jako jiné. To je indikováno barevně. Textové boxy stejné skupiny jsou vykresleny stejnou barvou. Současně s tím se ukládá informace do souboru XML, který se pak používá dále. Pokud nastane situace, že některé body

patří do nesprávné skupiny, dá se to vyřešit kliknutím pravého tlačítka myši na daný bod, tím změním jeho příslušnost ke clusteru a změní se tak i jeho barva.



Obrázek 4.4. Anotační nástroj po provedení alogritmu K-Means, sloužícímu k prvotní inicializaci

Jak je vidět z obr. 4.4, článek na žlutém pozadí uprostřed je v nesprávném clusteru, počítá se jako součást článku pod ním. Zde je potřeba provést manuální korekci, jejíž výsledek zobrazuje následující obrázek 4.5



Obrázek 4.5. Anotační nástroj po provedení manuální korekce

4.1.6 Finalizér vektorové reprezentace

Data, která jsou výstupem z anotačního nástroje „soubor XML s anotacemi“, se zpracují skriptem **annotMerge** . Pokud se na stránce objevují texboxy ze stejné skupiny tj. se stejným id, uzavře je všechny do právě jednoho elementu **<news>** . Dříve totiž každé jednotlivé textové pole bylo uzavřeno v tomto elementu. Výstup bude vypadat následovně „strukturovaný soubor XML s anotacemi“:

```
<?xml version='1.0' encoding='utf-8'?>
<pages>
   <page bbox="0,0,819.213013,1034.645996">
```

```
      <news id="0">
         <textbox bbox="38.6155,723.3741,549.7675000000002,805.07009">
            The first ‘real’Easter Egg
         </textbox>
         <textbox bbox="38.6155,238.16234000000034,179.7168,696.58869">
         ...
         </textbox>
         ...
      </news>
      <news id="1">
         <textbox ...>
         ...
      </news>
      ...
   </page>
</pages>
```

4.2 Datové formáty

4.2.1 Finální anotovaný soubor XML

Představme si finální formát souboru XML viz. „strukturovaný soubor XML s anotacemi“ v obr. 4.1. Formát XML použil Ing. Radek Mařík, CSc.[1] pro tvorbu uzavřeného datasetu nad deníkem Právo (poskytnuto Newton Media). Toto je formát, do kterého potřebuji transformovat data veřejné datové sady, abych je mohl dále zpracovat přesně tak, jako neveřejnou datovou sadu. Ukázka tohoto XML formátu vypadá následovně:

```
<?xml version="1.0" encoding="utf-8" ?>
<pages>
  <page id="1" bbox="0.0,0.0,841.89,1190.551" rotate="0.0">
    <news id="0" bbox="56.693,610.109,787.461,1100.986">
      <title>
        <textbox bbox="57.167,1040.786,784.731,1100.986" fontSize="60.2"
        tags="headline">
          Jihočeský kraj: zóny kůrovce nezastaví
        </textbox>
      </title>
      <body>
        <textbox id="5" bbox="88.812,1025.77,138.89,1035.94" fontSize="10.17"
        tags="body">
          Pavel Orholz
        </textbox>
        <textbox id="6" bbox="56.693,917.003,173.274,1020.836" fontSize="10.836"
        tags="body">
          Jihočeský kraj zásadně nesouhlasí s  novými zónami v  Národním parku Šumava.
          Podle hejtmanky Ivany Stráské (ČSSD) se rozšíří kůrovcová kalamita a nebude
          ji možné zastavit. Nové zóny schválila v pátek rada parku a poslala je
          na ministerstvo životního prostředí, které bude mít poslední slovo.
        </textbox>
        ...
      </body>
    </news>
```

[1] https://comtel.fel.cvut.cz/cs/users/marikr

```
  </page>
</pages>
```

Jednotlivé stránky jsou uzavřeny v elementu **<pages>** dále pak následuje samostatná stránka - element **<page>** a v ní element **<news>**, který obsahuje všechny textové boxy jednoho článku (bez ohledu na to jestli se nachází v elementu <title> nebo <body>). Každý texbox obsahuje identifikátor, souřadnice tzv. „bounding box“ - zde bbox, velikost fontu a atribut tags, zda se jedná o nadpis (headline) či článek (body).

4.2.2 Extrahované elementy v XML

Z finálního formátu jsem vycházel při tvorbě transformačního skriptu. Prvním krokem v transformaci dokumentu PDF je extrahovat elementy do souboru XML viz. obr 4.1. Výsledek po transformaci PDF do XML (pomocí pdfextract.py) vypadá následovně:

```
<?xml version="1.0" encoding="utf-8" ?>
<pages>
  <page bbox="0,0,819.213013,1034.645996">
    <textbox bbox="55.2415,936.4876,519.9806308000002,1017.3396">
      Explore Sandhurst in our latest prayer walk - page eleven
    </textbox>
    <textbox bbox="134.3162,787.018,496.0521501,798.798">
      Reporting from Berkshire, Buckinghamshire &amp;amp; Oxfordshire
      www.oxford.anglican.org
    </textbox>
    <textbox bbox="38.6155,723.3741,549.7675000000002,805.0700999999999">
      The first ‘real’Easter Egg
    </textbox>
    <textbox bbox="38.6155,707.0708000000001,95.72409999999999,718.3838000000001">
      By Jo Duckles
    </textbox>
    ...
  </page>
  <page>
  ...
  </page>
</pages>
```

Tento formát je o poznání odlehčenější, neobsahuje např. indentifikátory textových polí, rozdělení na title a body, a prozatím ani elementy news. Ty budou přidány až následně pomocí anotačního nástroje. Co se týče problému s více stránkami (původní formát obsahoval vždy jednu stránku), zachoval jsem veškeré náležitosti předchozího formátu jen s tím rozdílem, že element **<page>** je použit vícekrát.

4.2.3 Soubor XML s anotacemi

Výstupní soubor XML po anotaci viz.„soubor XML s anotacemi“ v obr. 4.1. Vypadá např. takto:

```
<?xml version='1.0' encoding='utf-8'?>
<pages>
   <page bbox="0,0,819.213013,1034.645996">
      <news id="0">
         <textbox bbox="38.6155,723.3741,549.7675000000002,805.0700999999999">
            The first ‘real’Easter Egg
         </textbox>
      </news>
      <news id="0">
```

```
            <textbox bbox="38.6155,238.16234000000034,179.7168,696.5886999999">
            ...
            </textbox>
        </news>
        ...
        <news id="1">
            <texbox ...>
            ...
    </page>
</pages>
```

Informace o tom do jaké skupiny textový box patří označuje element **<news id=0>** například. To znamená, že každé jednotlivé anotované textové pole je uzavřené v tomto elementu.

4.3 Datové sady

4.3.1 Vector-Based Article Segmentation dataset

VBAS dataset [2] jsem vytvořil pomocí anotačního nástroje, o kterém jsem hovořil v sekci 4.1.5.

Veřejná datová sada obsahuje anotované články čísel magazínů **The Door** a **Galway Advertiser**. Jména datových sad jsou ve formátu YYYYMM, kde YYYY je rok, MM je měsíc. Výjímku tvoří sada TheDoor_20102011. Ta zahrnuje magazíny roku 2010 a 2011.

- TheDoor_20102011
- TheDoor_201003
- TheDoor_201009
- TheDoor_201011
- TheDoor_201103
- TheDoor_201106
- GA_2010_01_07
- GA_2010_01_14

4.3.2 Statistika - VBAS dataset

Datová sada	Počet stránek	Počet anotovaných textových boxů	Průměrný počet text. polí na stránku
TheDoor_20102011	96	105	31.88
TheDoor_201003	20	26	30.9
TheDoor_201009	20	15	29
TheDoor_201011	16	17	31
TheDoor_201103	20	24	33
TheDoor_201106	20	23	34
GA_2010_01_07	21	47	42
GA_2010_01_14	68	124	44

pozn. nejsou anotována všechna textová pole

[2] skript ke stažení veřejné datové sady `https://github.com/zachtoma/pdf-mil`

4.3.3 Dataset Právo

Privátní datová sada poskytnutá Newton Media obsahuje anotované články deníku Právo. Sada se skládá ze 349 jednostránkových PDF souborů.

4.3.4 Statistika - dataset Právo

Datová sada	Počet stránek	Počet anotovaných textových boxů	Průměrný počet text. polí na stránku
Deník Právo (Newton Media)	349	2388	36.51

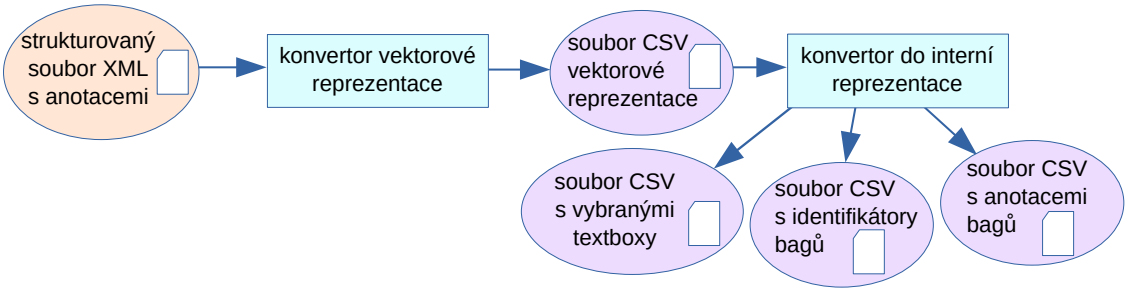
Kapitola 5
Preprocessing

V této práci využívám dva přístupy k reprezentaci textových polí pro strojové učení:

- reprezentace pomocí souřadnic
- reprezentace pomocí souřadnic a pojmenovaných entit

Následující sekce popisují postup preprocessingu pro tyto dva různé přístupy.

5.1 Souřadnicová reprezentace



Obrázek 5.1. Vytvoření interní reprezentace datasetu

Máme tedy data anotovaná způsobem popsaným v předchozí kapitole. Nyní následuje transformace pomocí skriptu **annot2csv.py**. Jak název napovídá provede konverzi dat z formátu XML do formátu CSV. Na první řádku je hlavička se jmény sloupců následovaná daty, každý řádek zaznamenává informaci o jednom textovém poli. Následuje ukázka:

```
x0,y0,x1,y1,length,nid,pid
490,197,629,367,171,0,1
340,240,478,354,280,0,1
189,240,328,354,278,0,1
39,238,180,697,1122,0,1
39,723,550,805,26,0,1
604,477,780,930,1695,0,2
41,466,215,498,132,0,2
42,516,201,588,64,0,2
41,595,212,855,796,0,2
43,847,212,931,138,0,2
227,929,595,976,23,0,2
191,408,331,592,504,0,3
40,397,179,592,550,0,3
40,598,291,636,28,0,3
40,28,180,364,951,1,3
40,365,245,405,21,1,3
491,445,630,587,352,2,3
339,445,479,587,389,2,3
...
```

Kromě souřadnic a délky textu textového pole ve znacích obsahuje soubor informaci o identifikátoru článku ke kterému textový box patří (sloupec nid) a nakonec nese informaci o stránce (sloupec pid - page id), na které se nachází.

Nyní jsou data připravena pro finální transformaci do tzv. „interní reprezentace“. Tu provede skript **csv2intRepr.py**. Ta se sestává ze tří souborů:

- soubor CSV s vybranými textboxy (PREFIXi.csv)
- soubor CSV s identifikátory bagů (PREFIXids.csv)
- soubor CSV s anotacemi bagů (PREFIXlabels.csv)

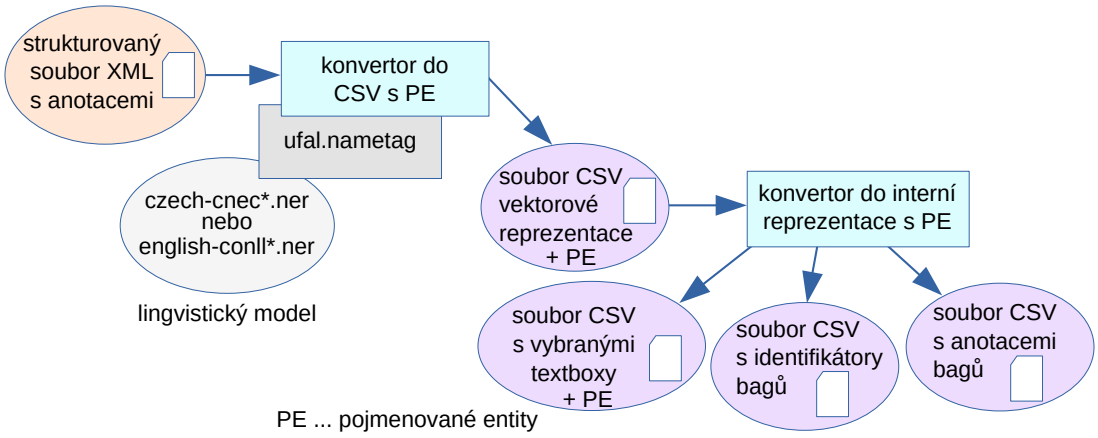
První soubor (PREFIXi.csv) obsahuje souřadnice textových boxů spolu s výběrem dvojice ve sloupci **pair**. V rámci stránky (page id - pid) jsou vybrány všechny možné kombinace dvojic textových polí. Zde například má první stránka **5** textových polí. Ukázka je zde:

```
x0,y0,x1,y1,length,pid,pair
490,197,629,367,171,1,1
340,240,478,354,280,1,1
189,240,328,354,278,1,0
39,238,180,697,1122,1,0
39,723,550,805,26,1,0
...
```

Další soubor (PREFIXids.csv) nese informaci o identifikátorech tzv. „bagů“. Kde každý řádek obsahuje informaci o tom do kterého bagu textové pole spadá. Více formálně: buď $\mathcal{X} = \mathbb{N}_0$ množina všech identifikátorů. Pak pro každé textové pole platí, že je označeno identifikátorem z této množiny.

A konečně poslední soubor (PREFIXlabels.csv) zaznamenává zda dvojice vybraných textových polí v rámci bagu je ze stejného článku. Počet řádků odpovídá počtu bagů. Pokud vybraná dvojice textových polí patří do stejného článku, obsahuje řádek 1 jinak 0. To se zjistí snadno porovnáním odpovídajících hodnot ze sloupce nid u původního souboru CSV vektorové reprezentace zobrazeného na začátku kapitoly 5.

5.2 Reprezentace pomocí souřadnic a pojmenovaných entit



Obrázek 5.2. Vytvoření interní reprezentace datasetu s použitím NER

Druhý typ interní reprezentace datasetu jsem připravil s použitím NER (Named Entity Recognition). Tato sada CSV souborů, z pohledu neuronové sítě jde o vstupní data MIL algoritmu, je obohacena o pojmenované entity

Ukázka souboru CSV vektorové reprezentace - reprezentace pomocí souřadnic viz. sekce 5.1

```
x0,y0,x1,y1,length,nid,pid
491,303,633,426,318,0,1
341,235,479,426,493,0,1
190,236,332,426,438,0,1
40,237,181,703,1255,0,1
...
```

Ukázka souboru CSV vektorové reprezentace - reprezentace pomocí souřadnic a pojmenovaných entit

```
x0,y0,x1,y1,length,nid,pid,ners
491,303,633,426,318,0,1,Synod,Christian,UK
341,235,479,426,493,0,1,Trident,BishopStephen
190,236,332,426,438,0,1
40,237,181,703,1255,0,1,AWE,Christian,Bishop,BishopMike,Trident,Bristol
...
```

Ukázka souboru PREFIXi.csv se zakódovanými pojmenovanými entitami

```
x0,y0,x1,y1,length,pid,ner_int,ner_diff,ner_int_norm,ner_diff_norm,pair
491,303,633,426,318,1,0,5,0.0,0.2777777777777778,1
341,235,479,426,493,1,0,5,0.0,0.2777777777777778,1
190,236,332,426,438,1,0,5,0.0,0.2777777777777778,0
40,237,181,703,1255,1,0,5,0.0,0.2777777777777778,0
40,714,94,724,13,1,0,5,0.0,0.2777777777777778,0
40,729,596,793,34,1,0,5,0.0,0.2777777777777778,0
491,303,633,426,318,1,0,3,0.0,0.16666666666666666,1
341,235,479,426,493,1,0,3,0.0,0.16666666666666666,0
190,236,332,426,438,1,0,3,0.0,0.16666666666666666,1
40,237,181,703,1255,1,0,3,0.0,0.16666666666666666,0
40,714,94,724,13,1,0,3,0.0,0.16666666666666666,0
40,729,596,793,34,1,0,3,0.0,0.16666666666666666,0
...
```

V rámci stránky vždy spočítám velikost průniku pojmenovaných entit vybraných textových polí - sloupec ner_int. Dále pak jejich symetrickou diferenci - sloupec ner_diff. A nakonec přidám počet dříve zmíněných vydělený tj. normalizovaný celkovým počtem pojmenovaných entit vybraných textových polí.

Kapitola 6
Implementace

Skript pro stažení VBAS datasetu je implementován v BASHi, ostatní nástroje v jazyce Python 3.X. Pro import a manipulaci s daty jsem použil datové rámce z knihovny **Pandas**. V anotačním nástroji využívám knihovnu **matplotlib** pro práci s obrázky.

Vlastní (MIL) multiple-instance-learning model je implementován v jazyce Python pomocí knihovny **PyTorch** [9] . Dále využívám knihovnu **mil_pytorch**[1] pro MIL modely implementovanou v PyTorchi.

6.1 Použité technologie

6.1.1 Python

Python je interprepovaný, interaktivní, objektově-orientovaný programovací jazyk. Zahrnuje moduly, vyjímky, dynamické typování, dynamické datové typy a třídy. Podporuje mnoho programovacích paradigmat kromě objektově-orientovaného programování jako procedurální a funkcionální programování. [10].

Veškeré programy a skripty používají verzi Pythonu 3.X.

6.1.2 Pandas

Pandas je rychlý, robustní, flexibilní a snadno použitelný open source nástroj na analýzu a manipulaci dat založený na programovacím jazyce Python.

V této práci jsem knihovnu Pandas použil k tvorbě, importu, úpravě a uložení souborů CSV, kde jsem použil datové rámce z knihovny Pandas [11] konkrétně verzi 1.2.3.

6.1.3 Ufal.nametag

Jak zmiňuje [12] *NameTag je open-source nástroj pro named entity recognition (NER). NameTag identifikuje správná jména v textu a klasifikuje je do předdefinovaných kategorií jako jsou jména osob, míst, organizací atd. NameTag je distribuovaný jako samostatný nástroj nebo knihovna, spolu s trénovanými lingvistickými modely.*

Tuto knihovnu jsem použil při vytváření datové reprezentace se souřadnicemi a pojmenovanými entitami viz. sekce 5.2, která byla vyextrahována z textových boxů. Verze knihovny byla 1.1.2.1.

6.1.4 Matplotlib

Matplotlib je rozsáhlá knihovna pro vytváření statických, animovaných a interaktivních vizualizací v Pythonu.

Tuto knihovnou využívám v anotačním nástroji k načítání PNG obrázků s texboxy a jejich anotování. Použitá verze knihovny je 3.3.4.

[1] odkaz na veřejný repozitář knihovny `https://github.com/jakubmonhart/mil_pytorch`

6.1.5 PyTorch

PyTorch je open source knihovna pro strojové učení založená na knihovně Torch, vyvinutá Facebookovou AI Research lab (FAIR). První alfa verze knihovny PyTorch byla publikována v září 2016. Torch je vědecký výpočetní framework s širokou podporou pro algoritmy strojového učení probíhající na GPU. Na PyTorchi jsou založené některé deep-learningové softwary jako například Tesla autopilot. Je tedy dostupná na RCI clusteru.

6.1.6 mil_pytorch

Tato knihovna je implementací modelu multiple-instance-learningu nad knihovnou PyTorch.

Knihovnu využívám k vytvoření modelu neuronové sítě a jejímu učení. Je tedy používána na RCI clusteru.

6.1.7 CUDA

CUDA byla představena společností NVIDIA v listopadu roku 2006. Jedná se o paralelní výpočetní a programovací model. Umožňuje efektivnější řešení výpočetních úloh s využitím GPU, než pomocí klasických procesorů. CUDA podporuje různé programovací jazyky, API a je multiplatformní. Návrh CUDA modelu zabezpečuje, že se zvyšujícím se počtem výpočetních jader udržuje vysoký stupeň paralelizace bez neúměrného zvyšování složitosti kódu. Toho je docíleno těmito třemi klíčovými abstrakcemi:

- hierarchie vláken
- sdílená paměť
- bariérový synchronizační vzor - tzn. všechna vlákna čekají v určitém bodě na poslední

Každý blok vláken může být spuštěn na libovolném dostupném multiprocesoru GPU. O počet fyzicky přítomných multiprocesorů na GPU se stará běhové prostředí.

6.1.8 Hardware - RCI cluster

ČVUT FEL a FIT založily Výzkumné centrum informatiky (Research Center for Informatics – RCI). Byl vybudován nejvýkonnější počítačový cluster pro výzkum umělé inteligence v ČR v hodnotě přes 40 milionů korun. Cluster má 20 výpočetních uzlů,(celkem 480 procesorových jader), 12 NVIDIA GPU výpočetních uzlů.

6.2 Vyvinuté nástroje a scripty

Tabulka 6.1 shrnuje implementované nástroje a jejich použití.

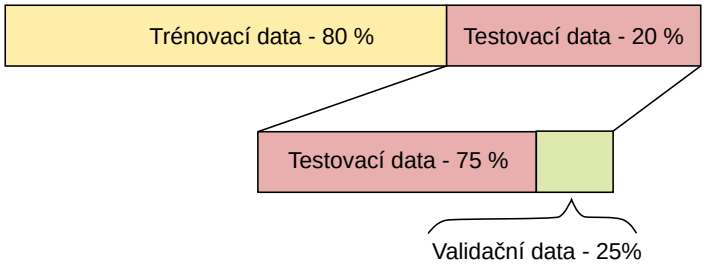
6.3 Schéma zpracování neuronovou sítí

Interní reprezentace, jejíž vznik jsme si popsali v kapitole 5 vstupuje do implementace modelu neuronové sítě. Skládá se ze tří souborů. První soubor nese informaci o vybraných dvojicích textových polí. Druhý soubor obsahuje informaci o rozdělení dat z prvního souboru do bagů. A Konečně ve třetím souboru jsou anotace jednotlivých bagů.

Dříve než data vstoupí do neuronové sítě je proveden tzv. „three-way split“, kde se data rozdělí do skupin dat trénovacích, testovacích a validačních.

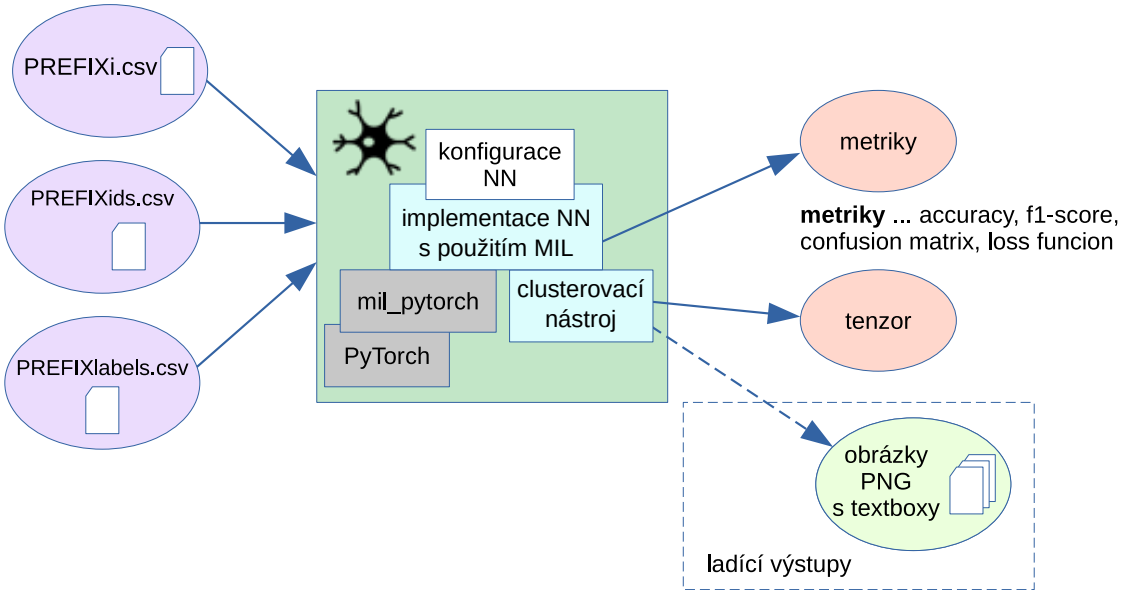
Název	Programovací jazyk	Účel
downloadpubpdf.sh	BASH	stažení veřejné datové sady
pdfextract.py	python 3.X	konverze PDF do souboru XML
XMLParser.py	python 3.X	trasformace obrázků PNG a souboru XML na obrázky s text. poli
annotClick.py	python 3.X	nástroj k anotování XML souboru
annotMerge.py	python 3.X	konverze souboru XML s anotacemi do strukturovaného souboru XML
annot2csv.py	python 3.X	konverze XML s anotacemi do vektorové reprezentace CSV
csv2intRepr.py	python 3.X	konverze CSV do interní reprezentace CSV
annot2csvner.py	python 3.X	konverze anotovaného XML do souboru CSV s pojmenovanými entitami
csv2intReprNer.py	python 3.X	konverze CSV do interní reprezentace CSV s pojmenovanými entitami
mil_pdf_rci.py	python 3.X	implementace MIL modelu neuronové sítě

Tabulka 6.1. Vyvinuté nástroje a scripty.



Obrázek 6.1. Rozdělení dat na trénovací, testovací a validační data

Data se rozdělí na 80 % trénovacích a 20 % testovacích dat. Ze 20 % testovacích dat je rozděleno ještě 25 % dat validačních.



Obrázek 6.2. Schéma zpracování neuronovou sítí

6.3.1 Konfigurace neuronové sítě

Po několika hrubých experimentech jsem došel k této konfiguraci neuronové sítě.

```
# Defining neural networks for proccesing inputs before and after aggregation function
prepNN = torch.nn.Sequential(
    torch.nn.Linear(input_len, 256, bias = True),
    torch.nn.ReLU(),
    torch.nn.Linear(256, 256),
    torch.nn.ReLU(),
    torch.nn.Linear(256, 256),
    torch.nn.ReLU(),
)

afterNN = torch.nn.Sequential(

    torch.nn.Linear(256, 256),
    torch.nn.ReLU(),
    torch.nn.Linear(256, 256),
    torch.nn.ReLU(),
    torch.nn.Linear(256,1)
)
```

Obrázek 6.3. Konfigurace neuronové sítě

Výše uvedený fragment kódu zobrazuje definici neuronové sítě, která se skládá z několika vrstev a aktivačních funkcí ReLU (Rectified Linear Unit).

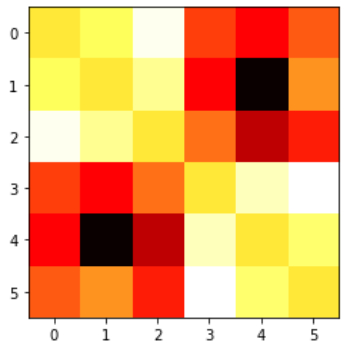
Ukázka definuje neuronovou síť s 256 neurony. Jde o části mezi něž je vložena agregační funkce.

6.3.2 Výstupy neuronové sítě

Ukázka výstupní matice z neuronové sítě $\mathbb{A}$, kde na pozici $a_{i,j}$ je předvídaná anotace textových polí i,j v rámci stránky. Nutno podotknout, ze $a_{i,j} >= 0$ se vezme jako ano - textová pole jsou ze stejného článku, $a_{i,j} < 0$ ne.

```
[[  0.          11.91335939  30.15411898 -26.79672066 -35.11393958  -22.22375691]
 [ 11.91335939   0.          19.10858552 -35.80860663 -77.17523238  -13.27553699]
 [ 30.15411898  19.10858552   0.         -18.88737416 -47.75146932  -31.21111581]
 [-26.79672066 -35.80860663 -18.88737416   0.          23.91523637   31.95702503]
 [-35.11393958 -77.17523238 -47.75146932  23.91523637   0.           14.30624292]
 [-22.22375691 -13.27553699 -31.21111581  31.95702503  14.30624292   0.        ]]
```

Na následujícím obrázku je pro představu teplotní mapa této matice. Bílé pole v teplotní matici odpovídá maximální hodnotě v matici výstupu. Černá odpovídá minimální hodnotě. Jednotlivá pole jsou hodnoceními jednotlivých bagů tj. posuzované kombinace dvojice textových polí ze stránky.



Obrázek 6.4. Teplotní mapa matice NN

Výstupní tenzor obsahuje identifikátory textových polí a teprve na jeho podkladě lze zrekonstruovat jednotlivé články ze vstupní datové sady v textové podobě.

```
0
0
0
1
1
1
```

Na začátku je shrnutí vstupních a výstupních souborů a parametrů. Dále pak už jsou vypsány metriky učení a to při každé sté epoše, které se skládají z výpisu ztrátové funkce na trénovacích datech - train_loss. Následují parametry na validačních datech a to: ztrátová funkce (Loss), Přesnost (Accuracy), F1 skóre a nakonec tzv. confusion matrix, jejíž příklad je na obr. 6.5.

n=165	Předpověď: NE	Předpověď: ANO
Pravda: NE	50	10
Pravda: ANO	5	100

Obrázek 6.5. Ukázka confusion matrix pro 165 výsledků

V ideálním případě bychom chtěli, aby na vedlejší diagonále byly 0. To by znamenalo, že nedošlo k žádným falešně negativním a falešně pozitivním výsledkům.

Kapitola 7
Experimenty

V této kapitole projdu výsledky učení modelu MIL (Multiple Instance Learning) neuronové sítě na datasetech VBAS a Právo.

7.1 Obecné nastavení

Při experimentech jsem použil následující nastavení:

- optimizer: Adam
- batch size: 16
- learning rate: 0,001
- weight decay: 0,0001
- loss function: BCEWithLogitsLoss

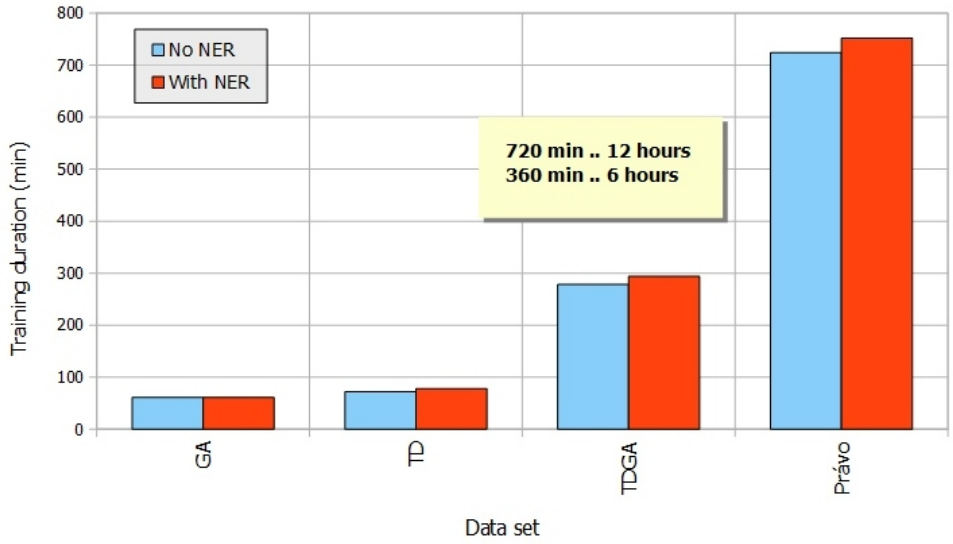
7.2 Učení neuronové sítě - tréninková fáze

Na čas tréninkové fáze měl hlavní vliv počet anotovaných textových polí na stránce. Z následující tabulky se dá vyčíst, že nejkomplexnější rozložení měl dataset Právo.

Název datové sady	Čas
GalwayAdvertiser	1h 1m
TheDoor	1h 12m
TheDoor & GalwayAdvertiser	4h 38m
Právo	12h 4m

Tabulka 7.1. Porovnání délek běhu tréninkové fáze

Následuje srovnání časů učení různých datových sad a různých reprezentacích - bez a s pojmenovanými entitami.



Obrázek 7.1. Délka běhu tréninkové fáze neuronové sítě na různých datasetech

Legenda:

- GA - Galway Advertiser
- TD - TheDoor_20102011
- TDGA - The_Door_20102011 & Galway Advertiser

Jak je vidět z grafu výše, reprezentace s NER zabere o něco déle. Nejdelší dobu trvalo učení datové sady Právo kvůli zmíněné komplexnosti rozložení.

7.3 Výstupy - tréninková fáze

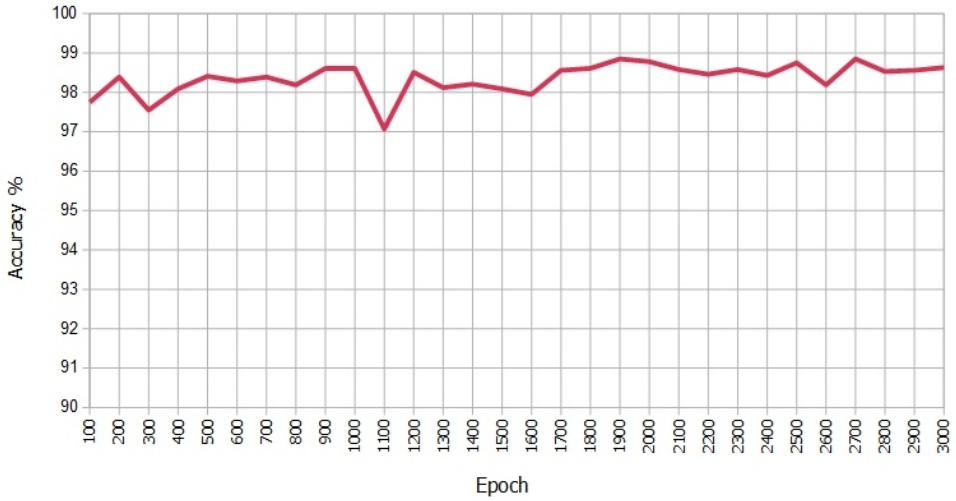
Tato sekce shrnuje výstupy učení modelu MIL neuronové sítě nad různými data sety s různou reprezentací textových polí a to bez a s NER pojmenovanými entitami.

7.3.1 Dataset Právo

Jedná se o neveřejný dataset poskytnutý firmou Newton Media. Dosažené výsledky učení nad datasetem právo zobrazuje následující tabulka.

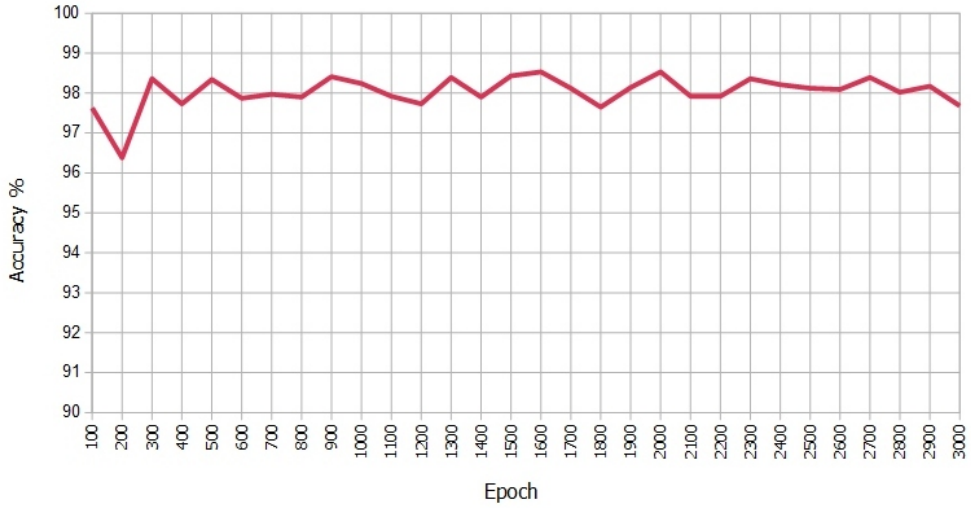
Datová sada	Počet epoch	Přesnost trénovacích dat	Přesnost testovacích dat	Přesnost validačních dat
newton30	3000	99,76 %	98,43 %	98,63 %
newton30ner	3000	99,41 %	97,46 %	97,68 %

Tabulka 7.2. Výsledky experimentu na datasetu Právo



Obrázek 7.2. Vývoj přesnosti validačních dat u datasetu Právo

Graf na obr. 7.2 zobrazuje vývoj přesnosti na validačních datech. Můžeme si všimnout výkyvu kolem epochy 1100, ale jinak se model učil dobře.



Obrázek 7.3. Vývoj přesnosti validačních dat u datasetu Právo s využitím NER

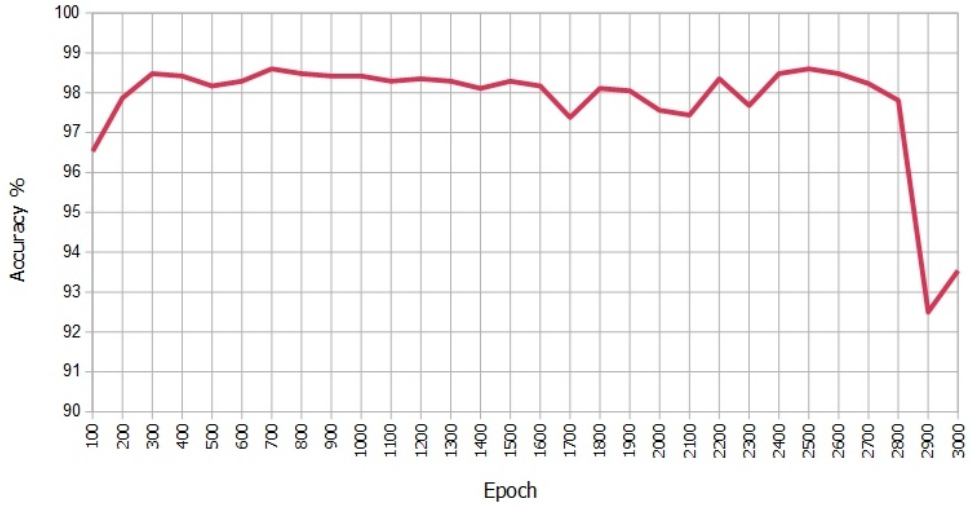
V případě grafu na obr. 7.3 k výraznějším výkyvům nedocházelo, nicméně výsledky jsou v porovnání s variantou bez pojmenovaných entit o něco horší z důvodu toho, že ne pro každé textové pole jsou pojmenované entity dostupné.

7.3.2 Dataset VBAS

Jedná se o vlastní veřejný dataset vytvořený v rámci práce. Výsledky experimentu zobrazuje následující tabulka.

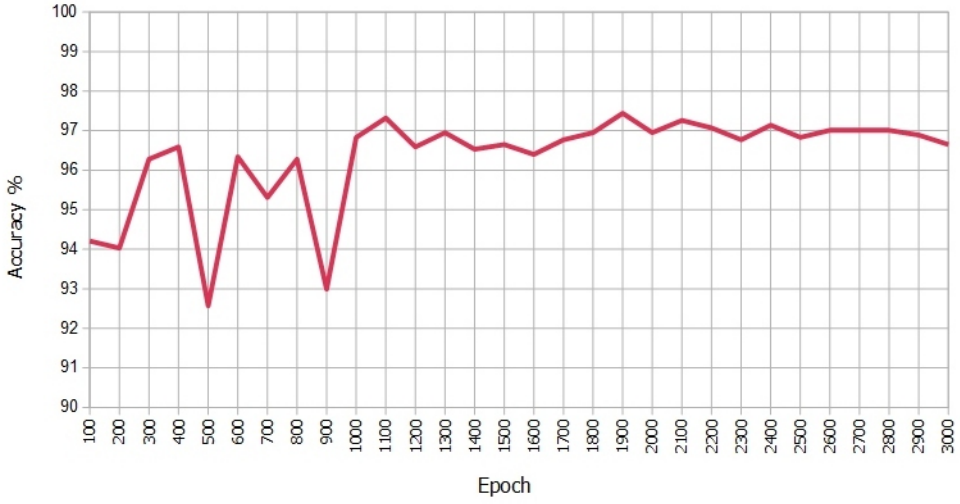
Datová sada	Počet epoch	Přesnost trénovacích dat	Přesnost testovacích dat	Přesnost validačních dat
GA	3000	100 %	99,39 %	99,39%
GAner	3000	100 %	97,56 %	98,17%
TheDoor_20102011	3000	100 %	96,55 %	97,93 %
TheDoor_20102011ner	3000	100 %	95,40 %	94,70 %
TDGA	3000	96,42 %	94,03 %	93,54 %
TDGAner	3000	100 %	96,65 %	96,65 %

Tabulka 7.3. Výsledky experimentu na VBAS datasetu



Obrázek 7.4. Vývoj přesnosti u datasetu TDGA

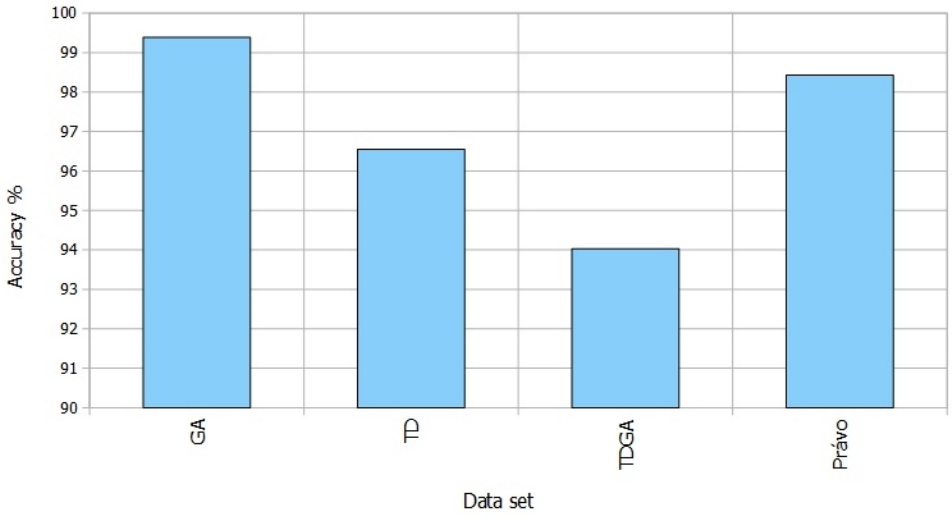
U grafu na obr. 7.4 si můžeme všimnout výrazného propadu u epochy 2900, což dávám za vinu přeučení modelu.



Obrázek 7.5. Vývoj přesnosti u datasetu TDGA s využitím NER

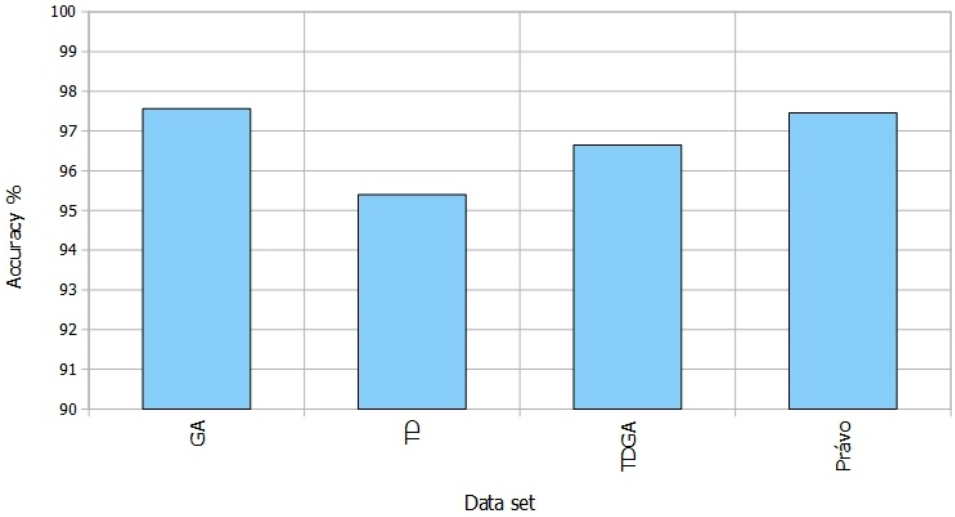
7.4 Přesnost na testovacích datech

Následující grafy srovnávají přesnosti validačních dat na jednotlivých datasetech.



Obrázek 7.6. Přesnosti testovacích dat jednotlivých datasetů

Nejlepšího výsledku dosáhla data Galway Advertiser spolu s datasetem Právo. Naopak nejhůře dopadl dataset TheDoor & Galway advertiser, což dávám za vinu rozdílnějšímu rozložení obou novin.



Obrázek 7.7. Přesnosti testovacích dat jednotlivých datasetů s použitím NER

Obecně se výsledky při použití NER zhoršily z důvodů toho, že ne každý textbox obsahuje pojmenované entity. Nicméně nejlepšího výsledku dosáhla opět datová sada Galway Advertiser a Právo. Déle se pak výrazně zlepšila přesnost u sady dat TheDoor & Galway Advertiser oproti reprezentaci bez NER.

Kapitola 8
Diskuze

V první části práce šlo vlastně o návrh a softwarovou implementaci datové transformace a naprogramování nástroje pro přípravu VBAS datové sady. Vstupní i výstupní formáty transformačních skriptů byly známé.

Tvorba anotačního nástroje už byla komplikovanější, protože jde o program s uživatelským rozhraním, jehož ovládání musí být pro uživatele příjemné a intuitivní. Nástroj navíc kromě frontendu obsahuje i součást pro uživatele skrytou - s pomocí algoritmu tzv. „učení bez učitele“ provádějící, bez jeho zásahu, shlukování (clustering) zdrojových dat. Čili iniciální návrh klasifikace textových polí - která patří ke shodnému článku. Nástroj jsem sám použil při přípravě VBAS datové sady, takže jsem v roli uživatele na vlastní kůži pocítil, kde je třeba na jeho ovládání a fungování ještě zapracovat. V každém případě, ačkoliv nástroj pomůže s iniciálním rozdělením textových boxů do clusterů, je práce supervizora (učitele) i tak poměrně úmorná - a to jsem anotoval jen asi 130 stran. Uživatel musí zkontrolovat každou stránku a určit, zda k-means algoritmus detekoval články správně a pokud ne, provést manuální korekci.

Jako nejzajímavější hodnotím druhou fázi práce, kdy došlo na experimenty s algoritmy strojového učení, s laděním konfigurace neuronové sítě, kdy jsem napjatě čekal (třeba 12 hodin) na výsledky dalšího kola učení a testování modelu na datech, která jsem sám dříve připravil. I na nejkomplexnějším datasetu v češtině - Právo.

A to jsem - proti miliardám příspěvků uživatelů například na Facebooku, jejichž analýza se ukazuje jako více a více potřebná - pracoval s neporovnatelně menšími daty. Pochopil jsem, že teze, popisující strojové analýzy grafických dat jako ten „jednodušší problém“ oproti porozumění a zpracování přirozeného jazyka / textu strojem jsou naprosto pravdivé.

Každopádně je úžasné, jakých úspěchů umělá inteligence a algoritmy strojového učení dosáhly už v tuto chvíli. A je fantastická představa, čeho se v nejbližší budoucnosti na tomto poli ještě dočkáme. Z technického hlediska mne problematika ML a AI velmi zaujala.

A je tu ještě jeden rozměr, který považuji za důležitý: práce mi pomohla uvědomit si, že stejně zásadní, jako hledat a nacházet, jak problémy vyřešit z technického pohledu, je klást si i filosofičtější otázky. Otázky, pokoušející se předvídat například etické aspekty a společenské dopady případné příliš překotné implementace chladných strojových a neosobních algoritmů na život společnosti. Doufejme ale, že přínosy dalšího zavádění ML a AI do praxe, převáží nad riziky i jejich negativními dopady.

Kapitola 9
Závěr

V první kapitole jsem popsal, co podle mého názoru, přispělo k růstu nasazení umělé inteligence. Dále jsem se věnoval popisu motivace mé práce a jejich cílů.

První částí zadání práce bylo vytvořit veřejný VBAS (Vector-Based Article Segmentation) dataset podobného charakteru jako neveřejný dataset deníku Právo, poskytnutý firmou Newton Media. K tomu jako „vedlejší produkt“ jsem musel implementovat skupinu nástrojů pro získání vektorové reprezentace novinových článků ze souborů PDF. Nejdůležitější aplikací v rámci této části práce je anotační nástroj, který má grafické rozhraní pomocí kterého uživatel upraví vybrané shluky textových polí tvořící články inicializované algoritmem K-Means. V rámci práce jsem použil anotační nástroj k anotování asi 130 stránek novin.

Dalším cílem bylo využít metod strojového učení k extrakci textových polí z vektorové reprezentace. K tomu jsem využil MIL (Multiple Instance Learning) model neuronové sítě, do které místo vektorů fixní délky vstupují tzv. bagy, které obsahují množinu textových polí v rámci stránky. Na každé stránce se nachází proměnný počet textových polí. V práci jsem vyhodnotil dvě podoby vstupní reprezentace, a to pomocí souřadnic a dále navíc s extrahovanými pojmenovanými entitami získanými metodou NER (Named Entity Recognition). Ukázalo se, že výsledky obou dvou přístupů mají poměrně vysokou přesnost. Varianta s NER zlepšila výsledky pouze u jednoho datasetu. Což mohlo být způsobeno menší velikostí datasetu a taky tím, že ne pro každé textové pole jsou pojmenované entity dostupné.

Dalším úkolem bylo popsat teoretický základ sestávající se ze shrnutí existujících metod. Které bylo zakončeno pojmem MIL - Multiple Instance Learning.

Dále jsem představil schéma zpracování spolu s tvorbou veřejné datové sady, nástrojů na její vytvoření a datových transformací k dosažení formátu totožného s neveřejnou datovou sadou Právo.

V neposlední řadě jsem představil použitý model neuronové sítě pro metodu MIL a výsledky experimentu na různých datových sadách a reprezentacích. Ukázalo se, že souřadnice textových polí jsou důležitější než vyextrahované pojmenované entity. Na základě provedených experimentů by se dala práce rozšířit následujícím způsobem:

- vyzkoušení na větší datové sadě
- využití anotačního nástroje pro tvorbu větší sady
- naučit se předvídat velikost fontu
- rozhodovat zda se jedná o nadpis či článek
- vyhledávat články nejen v rámci stránky (např. pokud je článek na více stránek)

Literatura

[1] Futrelle, Robert P., Mingyan Shao, Chris Cieslik a Andrea Elaina Grimes. Extraction, Layout Analysis and Classification of Diagrams in PDF Documents. In: *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*. USA: IEEE Computer Society, 2003. s. 1007. ICDAR '03. ISBN 0769519601. Dostupné na `http://www.ccs.neu.edu/home/futrelle/pubs37/diagrams/DiagramPapers/ExtractionLayout2003.pdf`.

[2] Chao, Hui a Jian Fan. Layout and Content Extraction for PDF Documents. In: Simone Marinai a Andreas R. Dengel, editoři. *Document Analysis Systems VI*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. s. 213–224. ISBN 978-3-540-28640-0. Dostupné na `https://link.springer.com/content/pdf/10.1007/978-3-540-28640-0_20.pdf`.

[3] Esposito, Floriana, Stefano Ferilli, Teresa Basile a Nicola Di Mauro. Machine Learning for Digital Document Processing: from Layout Analysis to Metadata Extraction. In: 2007. s. 105-138. ISBN 978-3-540-76279-9. Dostupné na DOI 10.1007/978-3-540-76280-5_5.

[4] Ramakrishnan, Cartic, Abhishek Patnia, Eduard Hovy a Gully APC Burns. Layout-aware text extraction from full-text PDF of scientific articles. *Source Code for Biology and Medicine*. May, 2012, ročník 7, č. 1, s. 7. ISSN 1751-0473. Dostupné na DOI 10.1186/1751-0473-7-7. Dostupné na `https://doi.org/10.1186/1751-0473-7-7`.

[5] Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov a Alexander J. Smola. Deep Sets. *CoRR*. 2017, ročník abs/1703.06114. Dostupné na `http://arxiv.org/abs/1703.06114`.

[6] Pevný, Tomáš a Petr Somol. Using Neural Network Formalism to Solve Multiple-Instance Problems. *CoRR*. 2016, ročník abs/1609.07257. Dostupné na `http://arxiv.org/abs/1609.07257`.

[7] Jin, Xin a Jiawei Han. K-Means Clustering. In: Claude Sammut a Geoffrey I. Webb, editoři. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. s. 563–564. ISBN 978-0-387-30164-8. Dostupné na DOI 10.1007/978-0-387-30164-8_425. Dostupné na `https://doi.org/10.1007/978-0-387-30164-8_425`.

[8] Hamerly, Greg a Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*. MIT Press, 2004, ročník 16, s. 281–288.

[9] [online]. Dostupné na `https://pytorch.org/`.

[10] *What is Python* [online]. Dostupné na `https://docs.python.org/3.9/faq/general.html#what-is-python`.

[11] *Pandas dataframe* [online]. Dostupné na `https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html`.

[12] [online]. Dostupné na `https://ufal.mff.cuni.cz/nametag/1`.

Příloha A
Zkratky

AI	Artificial Intelligence - Umělá inteligence
ASR	Automatic Speech Recognition - Automatické rozpoznávání řeči
DAU	Daily Active Users
DL	Deep Learning
MIL	Multiple Instance Learning
ML	Machine Learning - Strojové učení
NER	Named Entity Recognition
NLP	Natural Language Processing - Zpracování do přirozeného jazyka

Příloha B
Seznam souborů přílohy

`data-for-model`	Ukázková interní reprezentace dat k vyzkoušení modelu.
`Readme.md`	Popis jednotlivých částí přílohy.
`src`	Zdrojové soubory skriptů a nástrojů.
`src-model`	Zdrojový soubor modelu MIL neuronové sítě.