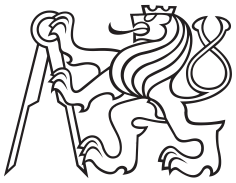
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Hodnocení obsahu mediálních zpráv

Bc. Ondřej Mézl

Vedoucí: Ing. Radek Mařík, CSc.
Květen 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mézl** Jméno: **Ondřej** Osobní číslo: **474578**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**

Studijní program: **Otevřená informatika**

Specializace: **Počítačová grafika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Hodnocení obsahu mediálních zpráv

Název diplomové práce anglicky:

Media News Content Evaluation

Pokyny pro vypracování:

1/ Vytvořte rešerši metod, které umožňují hodnotit obsah textových mediálních zpráv a jeho časový vývoj.
2/ Proveďte rešerši a vytvořte přehled metod vizualizace výsledků souvisejících se zpracováním přirozeného jazyka a vývojem obsahu zpráv.
3/ Vyberte vhodnou sestavu metod a vytvořte implementaci příslušného řetězce zpracování.
4/ Experimenty proveďte jak s anglickými, tak i českými texty se zaměřením na vizualizaci výsledků.
5/ Proveďte diskusi získaných výsledků a identifikujte kritické body zpracování.

Seznam doporučené literatury:

[1] Hapke, Hannes, Cole Howard, and Hobson Lane. 2019. Natural Language Processing in Action. Simon and Schuster.
[2] Bird, Steven, Ewan Klein, and Edward Loper. 2009. Natural Language Processing with Python. “O’Reilly Media, Inc.”
[3] Kubat, Miroslav. 2018. Introduction to Machine Learning. S.L.: Springer International Pu.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Radek Mařík, CSc. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **17.02.2023** Termín odevzdání diplomové práce: **26.05.2023**

Platnost zadání diplomové práce: **22.09.2024**

Ing. Radek Mařík, CSc. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.	
Datum převzetí zadání	Podpis studenta

Poděkování

Děkuji Ing. Radku Maříkovi, Csc. za vedení, odbornou pomoc, ochotu a cenné zkušenosti, díky kterým byl tento projekt realizován.

Dále také děkuji své rodině a přátelům, kteří mě motivovali a činili můj život radostnější.

Děkuji.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 25. května 2023

Abstrakt

Strojové učení je podobor umělé inteligence, který je možné využít pro zpracování přirozeného jazyka. Je to velmi mocný nástroj, díky kterému je možné zpracovávat slova i věty a mimo jiného reprezentovat jejich sentiment jako vektory, případně je klasifikovat.

Interpretovat výsledky může usnadnit vizualizace, která člověku různými technikami umožňuje rychle pochopit získaná data.

Tato práce se zabývá metodami zpracování přirozeného jazyka (především reakcí čtenářů na obsah mediálních zpráv) českého a anglického, a vizualizačními technikami aplikovanými k zobrazování výsledků.

Jedná se o diplomovou práci.

Klíčová slova: hodnocení, vizualizace, strojové učení, zpracování přirozeného jazyka

Vedoucí: Ing. Radek Mařík, CSc.

Abstract

Machine learning is a subfield of artificial intelligence, which is possible to be used for natural language processing. It is a very powerful tool, which is able to process words, sentences and furthermore represent their sentiment as vectors, eventually classify them.

The interpretation of the results may facilitate visualization, which enables a human with various techniques to understand acquired data quickly.

This thesis focuses on methods of Czech and English natural language processing (especially classification of readers' reactions to the content of media news) and visualization techniques applied to display the results.

This is a diploma thesis.

Keywords: classification, visualization, machine learning, natural language processing

Title translation: Media News Content Evaluation

Obsah

Obrázky

Tabulky

Část I

Úvodní část

Kapitola 1

Úvod

Pochopit význam lidského jazyka není pro počítač triviální úkol. Pomocí relativně nových metod strojového učení je však možné jazyk zpracovávat a reprezentovat sentiment slov i vět ve vektorových prostorech.

Při přenosu zpracovaných dat člověku může pomoci obor vizualizace, který dokáže různými technikami značně zlepšit schopnosti člověka vnímat a pochopit data.

Tato práce se zabývá metodami zpracování přirozeného jazyka, především hodnocení čtenářů na obsah mediálních zpráv - českých i anglických - a relevantními vizualizačními technikami.

1.1 Úvod do zpracování přirozeného jazyka

Běžná úloha zadaná pro počítač spočívá v hledání výsledku. Stroj dostane zadání (vstup) a recept (algoritmus), podle kterého má postupovat a dospět tak k výsledku.

Ve strojovém učení však stroj dostane příklady zadání s výsledky a jeho úkolem je doplnit co nejlépe hodnoty do algoritmu. Hledá tedy zobecnění pravidel, jak získat ze zadání výsledek.

Tento koncept je hojně užíván například pro zpracování obrazu, kde můžeme třeba detekovat jisté objekty v rastrové matici či zpravování zvuku, kde je možné provést například rozpoznávání řeči. Dá se však použít i pro zpracování lidského jazyka, například k získání sentimentu slov, ale mezi další typická využití patří například překladače, vyhledávače, chatovací asistenti nebo spam filtry. Tento podobor, kde se stroj snaží porozumět lidskému jazyku, se označuje zkratkou *NLP (Natural Language Processing)*. V této oblasti byly nedávno učiněny značné pokroky, podstatným milníkem je například jazykový model GPT-4 [Ope23].

1.2 Úvod do vizualizace

Vizualizace podporuje zrakový vjem člověka. Je to relativně nová disciplína, která mu umožňuje v datech "vidět více"pomocí reprezentace dat grafickými prvky. Lidský zrak je velmi efektivní kanál pro získávání informací, proto se vyplatí jej podporovat. Lidé mají zároveň relativně dobré kognitivní schopnosti, dokáží rozpoznávat vzory, barvy a tvary, což umožňuje vhodnou reprezentací předávat i data s více atributy, kde následně člověk dokáže vykonávat úlohy typu vyhledávání na základě jisté hodnoty či rozsahu, zjišťování hodnot, odhad distribuce, nalezení vzorů, shluků či korelace v datech, a podobně. Různými interaktivními technikami a obrazovými reprezentacemi tak člověk dokáže z dat získat cenné informace s výrazně nižším úsilím.

Vizualizace však nenahrazuje odborníka, pouze mu usnadňuje práci, interpretace je stále ponechána lidskému uvážení. Expert pracující s vizualizací dokáže z reprezentace dat získat cenné informace, dále je na základě získaných znalostí možné provést intervenci, upravit data či model a opět je vizualizovat, což vede k efektivnímu procesu vytěžení nových informací a hlubšímu porozumění pro analýzu či výzkum i z komplexních dat.

Dnes v mnoha vědeckých odvětvích hraje vizualizace díky jejímu širokému uplatnění klíčovou roli při lidském chápání velkých dat, například medicínských. Mimo jiné hraje důležitou roli v analýze, těžbě dat, prezentaci výsledků nebo interakci s uživatelem.

Dat je opravdu mnoho, a jejich množství roste stále rychleji. Proto je při zpracování vizualizace stále důležitější, jelikož dokáže rychle předat informaci o velkém množství komplexních dat člověku.

1.3 Cíle práce

Cílem práce je prozkoumání metod zpracování přirozeného jazyka a hodnocení textů (českých a anglických), obzvláště pak mediálních zpráv a jejich časového vývoje. Dále také vytvoření přehledu metod pro vizualizaci výsledků.

Metody pak budou v experimentální části vyzkoušeny a implementovány, výsledky budou diskutovány.

Vznikne tak řetězec zpracování a přehled technik, které je dále možné využít pro výuku či usnadnění práce při vizualizaci nebo zpracování přirozeného jazyka.

Část II

Teoretická část

Kapitola 2

Strojové učení

Strojové učení je podoborem umělé inteligence, který je hojně užívaný v datových vědách. Stroj se může učit tzv. s učitelem, tedy dostane příklady, typicky vstup - výstup, který byl předem vytvořen a ohodnocen odborníkem. Úkolem stroje je najít zobecněná pravidla, jak přiřazovat vstupy ke správným výstupům.

Zmíněný stroj typicky pracuje nad nějakou sítí (tzv. *Neuronovou sítí*), jejíž architektura (model) byla navržena pro daný problém a stroj se snaží upravovat její parametry tak, aby výstupy sítě byly co nejlépe přiřazeny správným výstupům z příkladů. Po úspěšné konfiguraci parametrů sítě je pak daný stroj schopen vykonávat predikce i na datech, které nebyly součástí původních příkladů.

Na počátku je síť prakticky nahodilá, ale matematickými cenovými funkcemi je možné vyhodnotit, jak velká chyba při výpočtu vznikla. Díky tomu je možné využít gradientní sestup a upravit tak parametry sítě směrem k dosažení menší chyby. Takto se stroj "učí". Tento proces může trvat relativně dlouhou dobu, obzvláště v případech, kdy je architektura neuronové sítě komplexní a dat je mnoho, proto se v dnešní době takové učení provádí typicky na grafických procesorech v paralelních dávkách. Trénování neuronových sítí je natolik důležité, že architektury moderních grafických procesorů mu jsou uzpůsobeny. Výsledky učení však mohou být nejisté.

2.1 Rešerše

Teoretickým úvodem do strojového učení mi byla kniha Miroslava Kubata [Kub17], která popisuje základní principy, ale také problémy strojového učení i dat, na kterých se učí.

Typickým problémem je přeučení. V případě přeučení má daný klasifikátor excelentní výsledky na datech, pomocí kterých je trénován, což může přivést iluzi úspěšného učení. Problémem je, že takový klasifikátor má dobré výsledky pouze na těchto datech, jelikož parametry použité k vyhodnocení nejsou dostatečně obecné a na dosud neviděných vstupech stroj výrazně častěji selže. Tento problém může nastat, když je daná síť trénována příliš dlouho na stejných datech. K zamezení se dají použít tzv. validační data, která nejsou přímo použita k trénování, ale je na nich možné model vyhodnotit, jelikož jsou více nezaujatá.

Je však vhodné podotknout, že data, na kterých je model testován by měla být dosud neviděna, ani by neměla mít vliv na trénování, jinak zanikají dobré vlastnosti generalizace a síť začne být spíše zaujatá na konkrétní podmnožinu vstupů. Proto by měly být množiny trénovacích, validačních a testovacích dat zcela disjunktní.

Ke strojovému učení se běžně používá populární programovací jazyk Python. Jedná se o multiparadigmatický interpretovaný programovací jazyk. Pro efektivní práci typicky používá různé knihovny (moduly), které jsou napsány v jazycích jako C++. Pro strojové učení se běžně využívá různých knihoven, které provádějí výpočty na grafických procesorech vyšší rychlostí paralelně (obvykle pomocí platformy CUDA). Takovou populární knihovnou je například Tensorflow [ABC+16], kterou je možné použít pro trénink neuronových sítí. Jelikož tato knihovna je poměrně nízkoúrovňová, existují další knihovny, které nad ní staví.

Dalším frameworkem pro strojové učení je například knihovna PyTorch [PGM+19], kterou ve své knize užili autoři Delip Rao a Brian McMahan [RM19]. Tato kniha mi byla experimentálním průvodcem ve strojovém učení.

Pro své experimenty jsem však použil knihovnu Keras[GP17]. Tato knihovna je rozhraním nad knihovnou Tensorflow), ale je více přímočará a vysokoúrovňová a obsahuje implementace užitečných algoritmů a stavebních bloků neuronových sítí. Při experimentech byla také užita dokumentace knihovny Tensorflow Keras [tfd].

Zpracování přirozeného jazyka (NLP) se obvykle zabývá porozumění jazyku strojem, analýzou a interpretací, ale zahrnuje i úlohy generování textu. Získání hodnocení textových dat, což je oblast zájmu této práce, je problém řešitelný pomocí metod NLP extrakcí emočního významu. V oboru zpracování přirozeného jazyka hrají důležitou neuronové sítě složené z různých vrstev, například hustých či rekurentních. Můžeme se setkat i s vrstvami konvolučními, které jsou typicky užívané pro zpracování obrazu, ovšem mají dobré vlastnosti jako vyšší odolnost vůči šumu při zpracování řeči [BBGC16].

Při zpracování přirozeného jazyka se stroj může potýkat s různými problémy. Jako příklad je možné uvést slova, která mění význam na základě kontextu - uvažme slovo *los*, které má ve frázích *los aljašský* a *výherní los* výrazně odlišný význam. Dalšími problematickými prvky mohou být sarkasmus či ironie, neformální fráze, idiomy, a podobně. [KKKS23]

2.1.1 Základní stavební prvky strojového učení pro NLP

V NLP hrají důležitou roli neuronové sítě. Základním stavebním prvkem je *neuron*, který získal název podle nervových buněk na základě podobné funkce, tedy přijímá, zpracovává a předává signály na základě podráždění. V neuronových sítích neurony provádějí jednoduchou matematickou operaci nad váženým součtem sil signálů předchozích neuronů a mohou pak předat signál dalším neuronům. Takto se v síti předávají mezivýsledky. Neuronová síť se skládá obvykle z několika stavebních bloků různých funkcí, které jsou vzájemně propojeny a jsou tvořeny různými množstvími neuronů. Tyto bloky abstraktně nazýváme jako *vrstvy*. První vrstvu pak nazýváme *vstupní* a poslední *výstupní*.

Prvním krokem zpracování přirozeného jazyka je předzpracování dat, která jsou následně předána vstupní vrstvě sítě. Je-li síť komplexnější, předává data dalším vrstvám, kterým říkáme *skryté*. Je relativně obtížné zjistit, jak síť došla k výsledkům z těchto vrstev. Tyto výsledky jsou nakonec předány vrstvě *výstupní*, která může například reprezentovat distribuci pravděpodobnosti pro klasifikaci.

2.1.2 Vektorová reprezentace slov

Nejprve je nutné zavést kódování do vektorů, nad kterými se dají provádět matematické operace. Je možné provést vektorizaci na úrovni znaků. Vzhledem

k tomu, že doména znaků je relativně malá, z této perspektivy se může jevit tato reprezentace výhodně. Pro věty se však běžně používá kódování na úrovni slov. K tomuto účelu se vytváří slovník, který obsahuje známá slova. Každé slovo pak primitivně může být zakódováno jednotkovým vektorem, který obsahuje 1 pouze na pozici odpovídající indexu ve slovníku. Takové reprezentaci se říká *One-Hot* [RM19]. Problémem je, že takto zakódovaná slova jsou mnohodimenzionální a nemají žádnou sémantiku, jelikož jsou tyto vektory v prostoru na sebe navzájem kolmé.

2.1.3 Embedding

Embedding vrstva je jeden z možných stavebních bloků neuronové sítě. Smyslem embeddingu je naučení se mapovat prvky (jednoduše indexem zakódovaná slova) na vektory v prostoru fixní dimenze.[CW08] Řeší tak problém s vysokou dimenzionalitou vektorově reprezentovaných slov. Tyto vektory mají typicky menší velikost než slovník, tedy i neefektivní *One-Hot* reprezentace. Výhodou je, že takto reprezentovaná slova zároveň zachovávají jistou sémantiku.

Slova, která jsou si sémanticky podobná budou v této reprezentaci pravděpodobně blíže u sebe. Pro příklad, můžeme předpokládat, že kosinová vzdálenost vektorů *psa* a *kočky* bude menší, než kosinová vzdálenost vektorů *psa* a *cihly*, jelikož *pes* a *kočka* se pravděpodobně nacházejí běžněji v podobném kontextu věty, za předpokladu že se takové věty objevovaly v trénovacích datech.

Existují již předtrénované modely pro vektorizaci slov, například *GLOVE* [PSM14] s různě velkými variantami.

2.1.4 Hustá vrstva

Hustá vrstva je nejjednodušším a fundamentálním stavebním blokem neuronových sítí. Tato vrstva je plně propojena s neurony vrstvy předchozí. Její funkcí je provedení jednoduché matematické operace (tzv. aktivační funkce) nad váženým součtem signálů neuronů z předchozí vrstvy, ke kterému je přičtena konstanta (tzv. *bias*). Různě komplexní husté vrstvy je možné stavět za sebe nebo je kombinovat s jinými vrstvami a umožnit tak síti naučit se složitější vzory.

2.1.5 Rekurentní vrstva

Důležitým stavebním prvkem pro zpracování jazyka je rekurentní vrstva, která si při vyhodnocování dokáže udržet "časový kontext". Rekurentní povaha těchto sítí pochází z postupné aplikaci stejné funkce na datové elementy, jako jsou slova ve větě. Při zpracovávání slova ve větě získá jakýsi kontext, který reprezentuje shrnutí významu předchozích slov, sekvenčně se s každým dalším zpracovaným slovem aktualizuje a ovlivňuje vyhodnocení následujících slov. Implementací těchto vrstev jsou různé varianty, ať už *LSTM*[HS97] či moderní *GRU*[CVMG+14].

Pomocnými mechanismy mohou být například *attention* (pozornost) [VSP+17], která slovům přidává na vážené důležitosti v kontextu dynamicky při zpracování. Rekurentní vrstvy mají samy o sobě potíž s pamětí, je-li text velmi dlouhý, kontext z počátku věty se může brzy ztratit. Použití pozornosti produkuje lepší výsledky ve srovnání s klasickou krátkodobou pamětí rekurentních vrstev.

Existují i hotové modely jako *BERT* [DCLT18] (tzv. transformátor) představený roku 2018 výzkumníky společnosti Google, společně s různě masivními variantami, které již pracují na úrovni celého textu, místo sekvenčního zpracování prováděného běžnými rekurentními vrstvami. V moderní době jsou hojně používány (nejen) k úkolům textového zpracování jako překlad nebo shrnutí.

Kapitola 3

Vizualizace

V dnešní době se mnohdy setkáváme s daty, která nejsou pro člověka triviálním úkolem pochopit. Taková data mohou být komplexní svým obsahem či množstvím.

Vizualizace vylepšuje kognitivní schopnosti člověka zrakovým vjemem vytvořením vhodné obrazové reprezentace dat v pozadí. Vizualizace typicky užívá širokou škálu diagramů, ve kterých jsou atributy zakódovány do různých grafických elementů, například bodů, čar a křivek, geometrických tvarů, které mohou být dále rozlišeny například barvou, velikostí či dalšími transformacemi. Barvou je přitom myšlen především odstín, ale je možné využít i dalších vlastností barev. S pomocí vizualizace, která je k danému úkolu adekvátní, dokážeme najít zajímavé vlastnosti či vzory v datech za relativně krátkou dobu, což umožňuje zkušenému uživateli vykonávat danou práci efektivněji.

Tyto dobré vlastnosti vizualizace jsou velmi užitečné a hojně využívané například při vědeckých výzkumech, těžbě dat nebo i k ověření výsledků. Vizualizace byla například užita i při verifikaci operací přístrojů na americké planetární sondě *New Horizons*[KHMB16].

Vizualizace textu je podobor vizualizace informace, který je v současné době stále významnější. Dnes může být obtížné vyhledávat relevantní textová díla či získat nadhled nad množinou dat, zvážíme-li, že každý rok vychází miliony publikací a na internetu jsou každou sekundu napsány miliony nových zpráv.

Text je ovšem poměrně široký pojem. Při vizualizaci jsou obvykle textem myšleny dokumenty, korpusy, zprávy, metadata či dokonce zdrojový kód software. Obsah takového textu se pak může významně lišit a zdroj může hrát významnou roli, například z pohledu spisovnosti, gramatické korektnosti, jazyka a významu. Zároveň text nepostihuje pouze přepis lidské mluvy, neboť se může týkat například o instrukce pro stroj, které mají výrazně odlišné vlastnosti od textu užitého pro komunikaci informace mezi lidmi.

Obecně, text není jednoduché vizualizovat. Jedním z problémů je vysoká dimenzionalita textu. Slov je mnoho a mohou nabývat různého významu na základě kontextu. Zakódujeme-li slova do vektorů, můžeme očekávat, že budou dlouhé. Vizualizovat pak takové vektory na dvourozměrné rovině - jako počítačové obrazovce či stránce knihy - je komplikované a typicky se ztrácejí informace. Význam textu lze však považovat za kategorická data, jejichž kvantitativní vlastnosti jsou lépe vizualizovatelné.

Je vhodné podotknout, že úkol hledat v textu není pro člověka snadný, například vyhledávání konkrétního slova na stránce textu je obtížné, tedy text sám o sobě není vhodná vizualizace.

Při vizualizaci časově orientovaných dostáváme typicky data ve spojení s časovými body, intervaly nebo jako posloupnost událostí. Taková data jsou implicitně seřazena, jelikož chápeme čas jako osu jedním směrem. Jisté úseky se však opakují, například dny v týdnu či měsíce v roce, což může být vhodné při vizualizaci uvážit, chceme-li na základě nich hledat vzory.

Forma vizualizace se odráží od našeho zájmu nad danými daty. Data jsou typicky obohacována (například interpolací), filtrována, mapována na vizualizační elementy a následně vykreslena. Výsledek je nakonec prezentován uživateli.

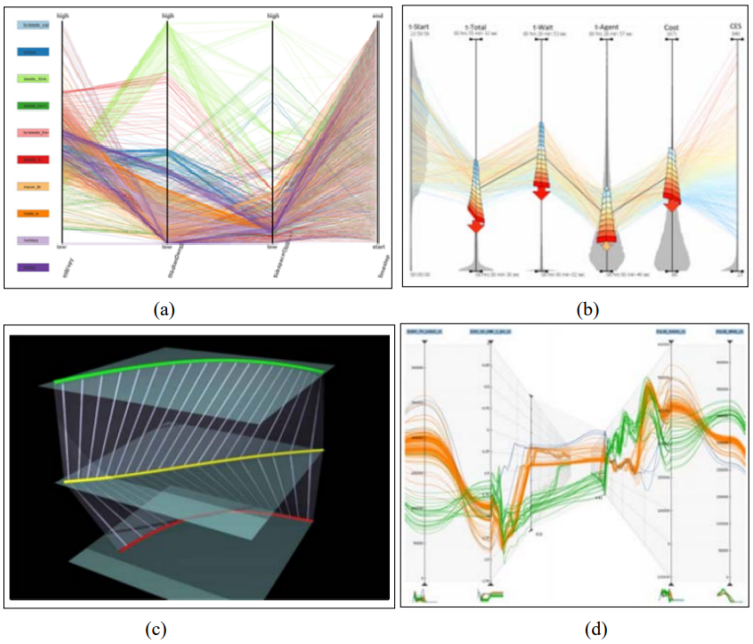
3.1 Úloha

Úkolem je analýza sentimentu mediálních zpráv. Vstupem jsou tabulková data, která byla předzpracována a ohodnocena v předchozích krocích. Sentiment zpráv se typicky rozděluje do kategorií: pozitivní či negativní (případně neutrální). Dále je možné, že tyto zprávy budou temporální data, tedy dalším atributem je časová známka, například kdy daná zpráva byla napsána.

3.2 Rešerše

Porozumění typu dat je klíčovým krokem efektivní vizualizace. Analyzovaná data můžeme zhruba rozlišit do následujících kategorií, jak navrhují Dr. Avishek Pal a Dr. PKS Prakash[PP17]. Prvním z těchto typů jsou data průřezová, která jsou výsledkem pozorování v jediném časovém bodě. Mezi taková data mohou patřit například statistiky jedné zprávy. V tomto ohledu časovou osu není nutné uvažovat, neboť je konstantní. Dále uvažujeme časové řady, což jsou data získaná pozorováním jedné entity v čase. Mezi taková data by například patřil vývoj konkrétní zprávy v čase. Nakonec rozlišujeme panelová data, která jsou získána pozorováním množiny entit v čase. Pro každou z těchto kategorií vznikají různé úkoly odrážející se ve vizualizaci. Data můžeme chtít vidět jako vlastnosti individuální entity, vidět časovou osu a pozorovat vývoj jediné entity či srovnávat časový vývoj vícero různých entit.

Podstatným krokem při vytváření vizualizace nad časově orientovanými daty je vhodné umístění grafických prvků na časovou osu. Uživatel tak může zjišťovat a porovnávat hodnoty v různých časových bodech. Andrew U. Frank popisuje relevantní časové koncepty [Fra98]. Časovou osu můžeme uvažovat lineární či cyklickou, která se opakuje po smysluplných intervalech, například ročních období, kde umožňuje sledovat sezónní trendy. Yujie Fanga, Hui Xub a Jie Jiang ve svém článku [FXJ20] píší, že jednotlivé datové atributy můžeme chápat jako body, které představují okamžik - událost bez ohledu na její trvání. Alternativně, časová data mohou být intervaly, které představují (menší) časový úsek. Dále je možné události uspořádat sekvenčně, či jako větvící se strukturu. Standardem je zvolit osu x jako časovou, zatímco osa y reprezentuje další atribut. Tímto způsobem vidíme časový vývoj, je ovšem obtížné zachytit periodické vzory. Pro vizualizaci atributů je doporučen *Spirálový diagram*, který je vhodný k analýze periodických dat. Tento diagram chápe časovou osu jako spirálu, kde každý cyklus reprezentuje periodu. Hodnoty atributů je možné zobrazovat různými cestami, například barvou, čarami a podobně. Dále je doporučen *Kalendářový pohled*, který zobrazuje data jako shluky tvořené na základě periodicky formátované granularity, podobně jako kalendář, například dny v každém týdnu jsou zobrazeny jako samostatné řádky. Další vizualizační technikou je *ThemeRiver*, kde je časový tok zobrazen jako řeka. Tato technika je podrobněji popsána později. Pro potřeby vizualizace je možné techniky kombinovat, nebo použít dynamickou vizualizaci, kde pro každý časový okamžik překreslujeme body do pozice v daném čase (například *Gapminder Trendalyzer*). Jsou-li data mnohodimenzionální, můžeme využít *paralelních souřadnic*, které jsou podobné čárovým diagramům, kde atributy můžeme reprezentovat jako hodnoty na uspořádaných paralelních souřadných osách a spojit je čarou. Pro taková data je opět možné použít již představenou vizualizační techniku *ThemeRiver*.



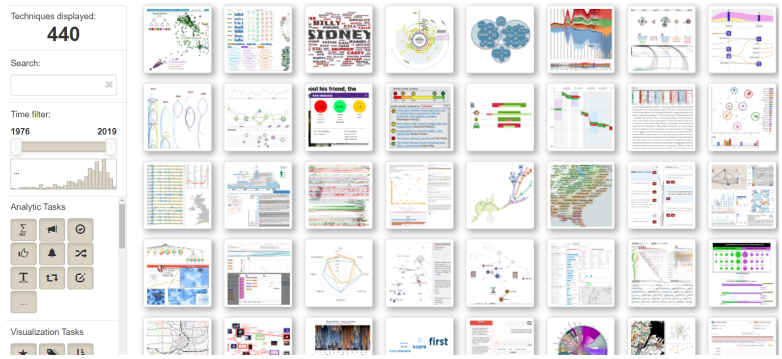
Obrázek 3.1: Ukázka různých variant *Paralelních souřadnic* z článku *A Survey of Time Series Data Visualization Research*

Dle článku autorů Aigner, W., Miksch, S., Müller, W., Schumann, H., a Tominski, C. [AMM+07], existuje mnoho vizualizací specializovaných konkrétní analýze, ale pro prostá data jsou k základním analytickým úlohám vhodné standardní vizualizační techniky jako více či méně sofistikované grafy a diagramy, které v takových případech překonávají specializované techniky, neboť je snadné jim porozumět.

Užitečným nadhledem vizualizačních technik je *Text Visualization Browser* skupiny *ISOVIS*, který ve svém článku prezentují autoři Kucher a Kerren[KK15]. Tento online prohlížeč zobrazuje drobné náhledy mnoha (v době psaní této práce 440) vizualizačních technik pro text s několika možnými filtry a vyhledáváním.

1

[1]Teoretickým i praktickým úvodem do vizualizace mi byl mimo jiné také předmět Vizualizace.



Obrázek 3.2: Ukázka *Text Visualization Browser*

3.2.1 Obecné vizualizační techniky

Pro vizualizaci se běžně používá široká škála nástrojů a metod. Tyto metody existují pro vylepšení kognitivních schopností člověka obrazovou reprezentací, která umožňuje získání smysluplného nadhledu nad nějakými daty. Metody se liší na základě dat a prováděného úkolu. Typicky se setkáváme s technikami jako jsou diagramy (sloupcové, koláčové, spojnicové, ...), chceme-li vyjádřit jisté kvantitativní vlastnosti, pro geografická data můžeme očekávat mapy, pro sítě vztahů různé grafy a podobně.

Elementy (dokumenty, autoři, ...) je možné umisťovat do prostoru dle vybraných atributů. Takto nám mohou vznikat přirozené shluky. Jelikož finální obraz je pouze dvourozměrný, typicky takto lze reprezentovat jen kombinace dvou (až tří) atributů.

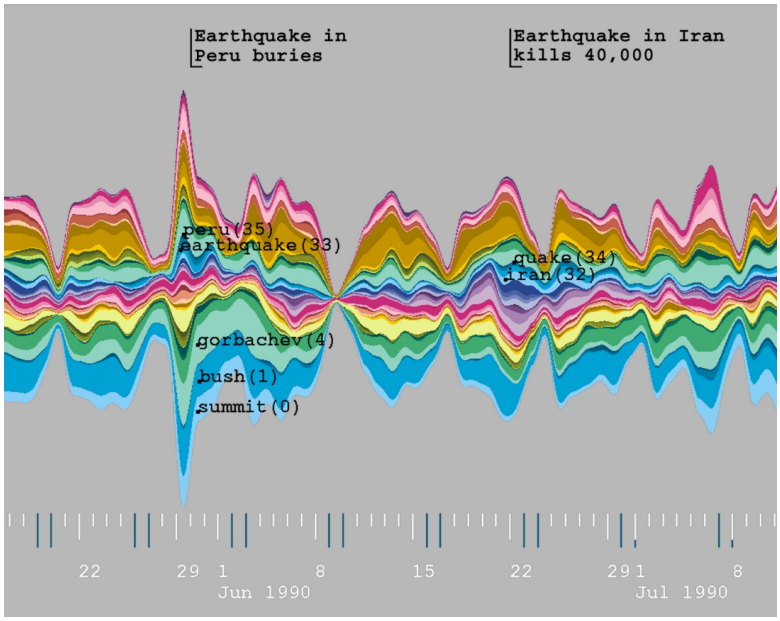
Další atributy pak můžeme vizualizovat například barvou, velikostí či tvarem elementu. Použít je rovněž možné *glyfy*, což jsou malé ikonky, které nabývají různých vzhledů na základě dat. Rovněž je možno využít projekčních technik. [Spe99]

Obrázek 3.3: Ilustrace vizualizace atributů.

3.2.2 ThemeRiver

ThemeRiver [HHN00] je vizualizační technika původně navržena pro vizualizaci změn témat v dokumentech. V *ThemeRiver* čas plyne ve směru osy x. Velikosti zastoupení jednotlivých témat v daném časovém bodě jsou akumulovány a mapovány na osu y tak, aby byly souměrné podle osy x. Můžeme tedy říct, že obrys je symetrický podle osy x. Jednotlivá témata jsou v této vizualizaci barevně rozlišena, aby jejich vývoj byl možný sledovat uživatelem.

Vizualizace pomocí *ThemeRiver* je metafora řeky a umožňuje vidět změny sil témat v čase. To je možné využít pro vizualizaci časového vývoje sentimentu. Uvážíme-li, že každou kategorii reprezentujeme jako téma (a náležitě barevně rozlišíme), je pak možné vizualizovat i síly sentimentu v čase.

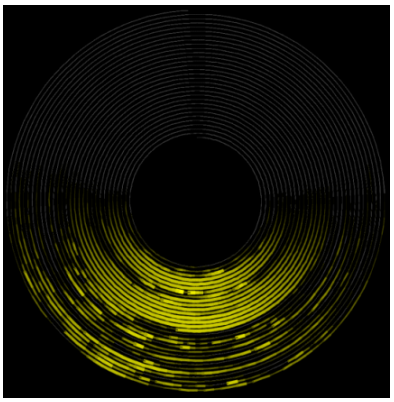


.

Obrázek 3.4: Ukázka *ThemeRiver* z článku *ThemeRiver: Visualizing Theme Changes over Time*

3.2.3 Spirálový diagram

Vizualizace časově orientovaných dat na spirálu [WAM01] je vhodná k analýze sezónního vývoje, kdy pozorujeme hodnoty na periodických časových intervalech, například dny v týdnu. Časová osa, která je obvykle lineární, je transformována do spirály a data jsou mapována pomocí barev, textur či vzorů.

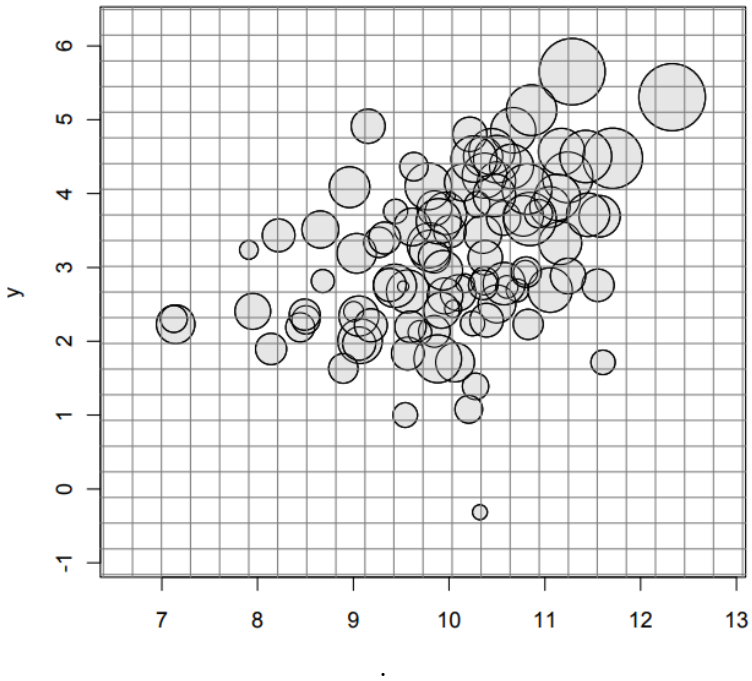


.

Obrázek 3.5: Ukázka vizualizace dat na spirálu z článku *Visualizing Time-Series on Spirals*

3.2.4 Bublinový diagram

Bublinový diagram je vylepšení bodového diagramu. V bodovém diagramu jsou do dvourozměrného prostoru umisťovány body, reprezentovány jako tečky fixní velikost. V případě bublinového diagramu jsou však zobrazovány elementy, které se mohou lišit velikostí (bubliny)[MG10]. Tímto způsobem je možné vizualizovat dodatečný atribut ve dvourozměrném prostoru. Ačkoliv je vhodné podotknout, že velikost tvaru není pro člověka příliš přesný kanál, je například relativně snadné najít velké body.



Obrázek 3.6: Ukázka *Bublinového diagramu* z knihy *Advances in social science research using R*

3.2.5 Orange

Vizualizace však nemusí být nutné vytvářet manuálně, existuje různý software, který umožňuje uživateli snadno vytvářet obrazové reprezentace dat s nižším úsilím, což je využitelné například pro efektivní těžbu dat. Jedním z takových softwarů je *Orange* [DCE+13], který obsahuje různé nástroje pro vizualizaci, zpracování dat a strojové učení. Zároveň je snadno pochopitelný pro uživatele. Na následujícím obrázku je ukázka prostředí programu, kde uživatel načítá datový soubor a následně může data vizualizovat různými připravenými technikami.

.

Obrázek 3.7: Ukázka práce v programu *Orange*

Část III

Experimentální část

Kapitola 4

Použitý software

Pro vývoj byl použit editor *Visual Studio Code* společně s *Jupyter Notebook*.

Jelikož byl vývoj proveden v jazyce *Python*, byl také použit standardní *Python interpreter* a *pip* pro instalaci knihoven.

Dále pak *CUDA Toolkit* pro výpočty na grafické kartě pomocí knihoven zmíněných dříve.

Pro psaní tohoto dokumentu bylo použito *TeXstudio*.

Kapitola 5

NLP experimenty

V této sekci jsou popsány experimenty, které se týkali zpracování přirozeného jazyka. Zpracování přirozeného jazyka je důležitým bodem řetězce hodnocení mediálních zpráv, neboť umožňuje textům jako jsou například reakce přiřazovat sentiment. Ne všechny experimenty se týkají analýzy sentimentu, neboť sloužily jako úvod do problematiky zpracování přirozeného jazyka. První takové příklady byly uvedeny v knize o NLP v knihovně PyTorch[RM19], ale při implementaci byly přepsány do knihovny Keras, kterou jsem se rozhodl používat pro své experimenty.

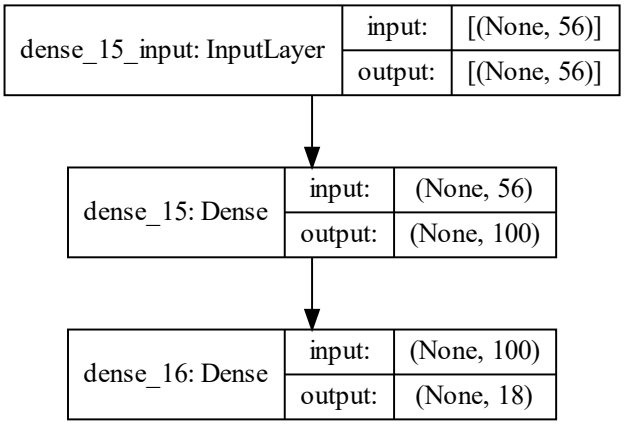
5.1 Klasifikace příjmení

Prvním jednoduchým experimentem je klasifikace příjmení. Je dán dataset obsahující různá příjmení označená 18 možnými národnostmi. Úkolem algoritmu je přečíst si příjmení a zjistit, které národnosti z 18 možností dané příjmení je. Pro tento počáteční příklad byly textové řetězce vektorizovány na úrovni znaků pomocí *One-Hot* kódování.

5.1.1 Naivní řešení hustými sítěmi

První řešení bylo proložení vstupní a výstupní vrstvy jednou skrytou hustou vrstvou. Architektura sítě se tedy skládala z vrstvy vstupní, husté vrstvy a

husté výstupní vrstvy. Výstupní vrstva obsahuje 18 neuronů a reprezentuje pravděpodobností rozložení jednotlivých kategorií národnosti.

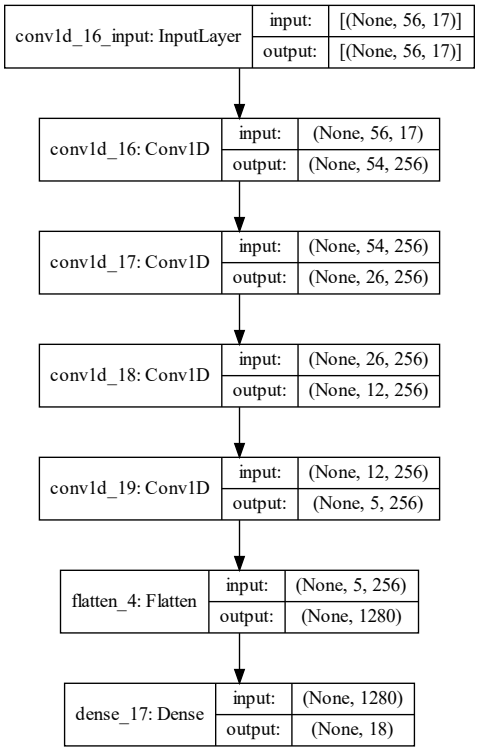


Obrázek 5.1: Naivní architektura sítě pro klasifikaci příjmení

Po trénování na 30 *epoch* (opakování) tato síť dokázala správně ohodnotit 59.94 % případů.

5.1.2 Řešení konvolučními sítěmi

Druhé řešení zapojilo tzv. *konvoluční vrstvy*. Konvoluční vrstvy jsou v principu podobné rekurentním, ale místo *časového* kontextu mají kontext *prostorový*. Při výpočtu tyto vrstvy postupně zpracovávají malé části vstupu v okénku, které se postupně posouvá a aplikují na ně naučenou konvoluční matici. Myšlenkou je, obsahuje-li příjmení například sekvenci *ová*, redukuje se množství národností, kterému může patřit.



Obrázek 5.2: Architektura sítě pro klasifikaci příjmení pomocí konvolučních sítí

Po trénování na 50 epoch síť dokázala správně ohodnotit 71.02 % případů.

Příklad interakce:

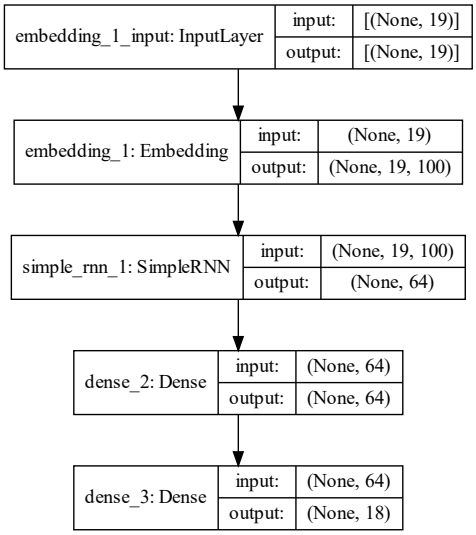
```
(Formát: VSTUP VÝSTUP PRAVDĚPODOBNOST)

Vu Vietnamese 0.9985569
Putin Russian 0.82910234
Novák Czech 0.99961495
```

5.1.3 Řešení rekurentními sítěmi

Třetí řešení zapojilo *rekurentní sítě* a také *embedding*. Řetězec je tedy zpracováván sekvenčně a zároveň je snaha přiřadit jednotlivým prvkům význam ve

vektorovém prostoru.



Obrázek 5.3: Architektura sítě pro klasifikaci příjmení pomocí rekurentních sítí

Po 30 epochách síť korektně oklasifikovala 67.23 % případů.

5.1.4 Diskuse ke Klasifikaci příjmení

Osobně považuji za zajímavé, že hustá síť dokázala relativně dobře oddělit kategorie příjmení.

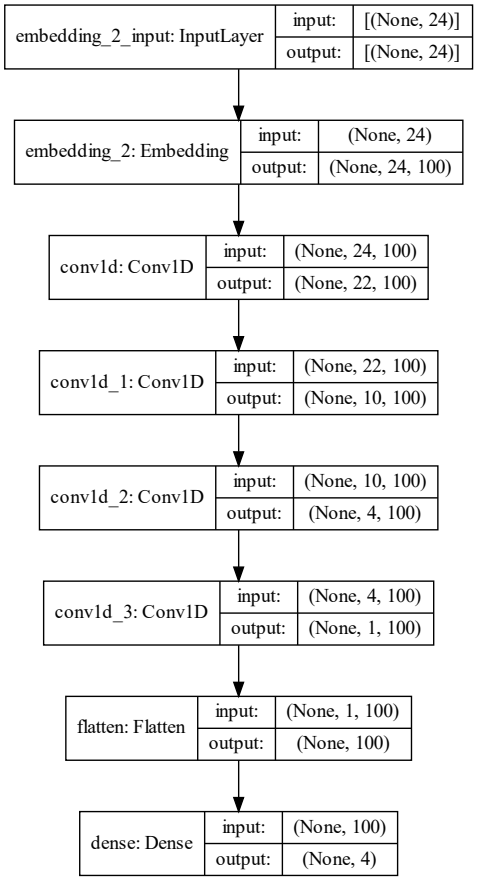
Na druhou stranu je nutné podotknout, že dodaná data byla nevyvážená: V 10 980 vzorcích byla data z 18 kategorií označkovaná čísly 0 - 17, ovšem některé kategorie byly reprezentovány lépe (až 2972 vzorků), než jiné (jen 55 vzorků), což může způsobit zkreslení výsledků.

Architektura v tomto případě byla velmi prostá a embedding pro tento formát není příliš efektivní.

5.2 Klasifikace titulků dokumentů s pomocí GLOVE a konvolučních sítí

V tomto úkolu síť klasifikuje titulky dokumentů mediálních zpráv. Titulky jsou v tomto datasetu označkovány 4 kategoriemi: Business, Sci/Tech, Sports a World.

Nyní jsou klasifikovány titulky dokumentů, ale na rozdíl od předchozího experimentu budou vektorizována jsou celá slova. Pro zachování sémantiky slov je použit *embedding*, konkrétně předpřipravený slovník *GLOVE* [PSM14]: *glove.6B.100d*. Jelikož je slovník předpřipravený, může nastat případ, že se dané slovo nenachází ve slovníku. Opravdu tomu tak je, *Hit rate* (poměr slov nalezených ve slovníku) je 86.57 %.



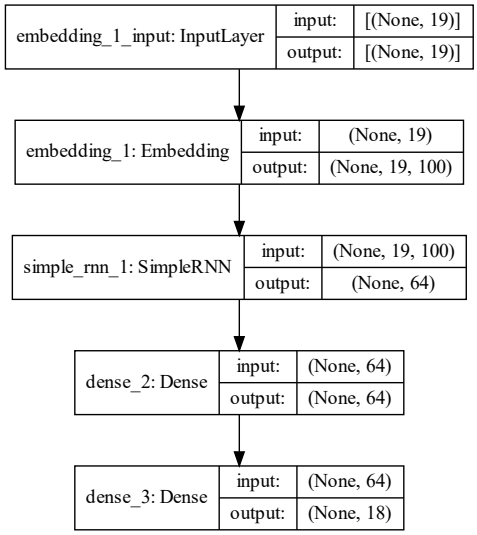
Obrázek 5.4: Architektura sítě pro klasifikaci titulků dokumentů s GLOVE a konvolučními sítěmi

Pomocí této sítě je například titulek *"boeing expects to pay a lot of money"* je klasifikován správně jako *Business*.

5.3 Generování příjmení

Cílem této úlohy je generovat nová příjmení požadované národnosti a délky. V principu se jedná o odlišný úkol, ovšem stále je založená na klasifikačních schopnostech neuronové sítě. Síť dostane na vstup 3 parametry: Prefix (prvních několik písmen vstupního řetězce), národnost a počet znaků, které má doplnit. Například z (*Nov, Czech, 2*) by mohl vzniknout *Novák*.

Tato úloha se dá převést na úlohu generování dalšího znaku. Jako vstup je dáno zřetězení embeddingu prefixu s embeddingem národnosti a výstup sítě je po výpočtu přes rekurentní a hustou vrstvu 1 znak, který je zvolen na základě nejvyšší hodnoty v pravděpodobnostním rozložení známých symbolů. Tento výsledný znak se připojí za prefix a tento proces se opakuje, dokud řetězec nedosáhne požadovaného počtu znaků.



Obrázek 5.5: Architektura sítě pro klasifikaci titulků dokumentů s GLOVE a konvolučními sítěmi

V architektuře sítě je přidána také vrstva *dropout* s 50 % silou, která

částečně randomizuje výsledky. Myšlenkou této vrstvy je učinit síť méně deterministickou, což podporuje "kreativitu"a síť následně produkuje zajímavější výsledky.

Příklady vygenerovaných příjmení:

```
(FORMÁT: PREFIX, NÁRODNOST, #ZNAKŮ, VÝSTUP)

'm', 'English', 10, martersonya
'cli', 'English', 7, clinderton
'trum', 'English', 7, trumenderso
'novakov', 'Czech', 10, novakovanovskyava
'mal', 'Czech', 10, mallanovskyan
'espo', 'Italian', 10, esporninininin
'boti', 'Italian', 10, botinininininini
'i', 'Vietnamese', 10, ingahanghan
'v', 'Vietnamese', 10, vanganengha
'kuros', 'Japanese', 10, kuroshishinakak
```

Síť byla při testu přesná na 33.77 %, což není příliš relevantní. Některá vygenerovaná příjmení působila relativně přirozeně, jiná méně, ovšem vygenerované příjmení mnohdy mělo typické národnostní rysy.

5.3.1 Diskuse ke generování příjmení

Některá vygenerovaná příjmení považuji za relativně zdařená. V některých případech generátor začal opakovat znaky, což je pravděpodobně způsobeno nedostatečnou reprezentací jmen dané národnosti.

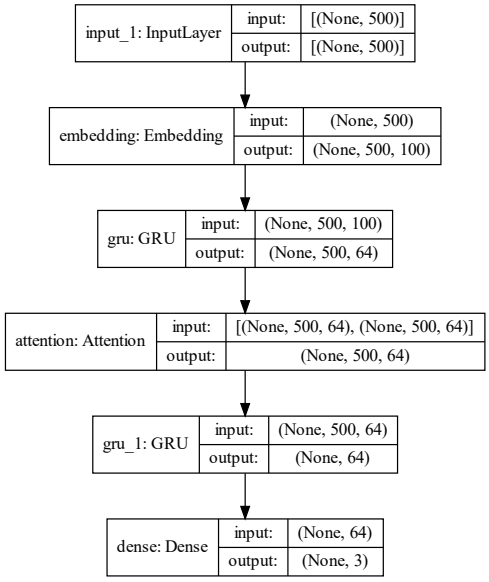
Některá jména se také jeví být chybně klasifikována v datasetu, jelikož například spousta jmen jako *Widerlechner* se rozhodně nezdají být podle své značky česká.

5.4 Klasifikace recenzí na Yelp

Tato, ani žádná další sekce se již v příkladech neinspiruje literaturou.

Pro tento úkol mi byl dán dataset, který obsahoval ohodnocené recenze firem z platformy Yelp značkami pozitivní, negativní a neutrální. Úkolem sítě je nyní přečíst si recenzi (textový řetězec) a odhadnout, zda je daná recenze pozitivní, negativní či neutrální.

Jako zajímavý experiment jsem zkusil použít 2 rekurentní vrstvy proložené pozorností. Jednotlivá slova jsou opět vektorizována embeddingem a výstupní vrstvou je vrstva hustá.



Obrázek 5.6: Architektura sítě pro klasifikaci recenzí na Yelp

Ačkoliv se síť trénovala velmi dlouho, po 10 epochách dosáhla přesnosti při testu 84.61 %.

Příklad interakce s uživatelem:

```
I cannot recommend this place. The food was overcooked.
```

```
Prediction: NEGATIVE
Probability: 96.33377205093953 %

I had quite a good time!
Prediction: POSITIVE
Probability: 79.6690411663691 %
```

5.5 Klasifikace recenzí na ČSFD

Pro tento úkol mi byl dán dataset recenzí z filmové databáze ČSFD. Na rozdíl od předchozí sítě, která klasifikovala recenze na platformě Yelp, tato pracuje s češtinou, což přináší řadu dalších problémů. Jelikož čeština je výrazně komplikovanější jazyk, než angličtina, dá se předpokládat, že klasifikátor nebude dosahovat tak dobrých výsledků. Důležitým bodem při klasifikaci českých textů je předzpracování, jelikož menší sítě nejsou schopny efektivně pojmout takto komplexní jazyk, chápat mnoho různých forem slov a gramatických pravidel.

Pro usnadnění práce klasifikátoru bylo provedeno několik předzpracování, které jsou klíčovým bodem pro efektivní zpracování českých textových dat:

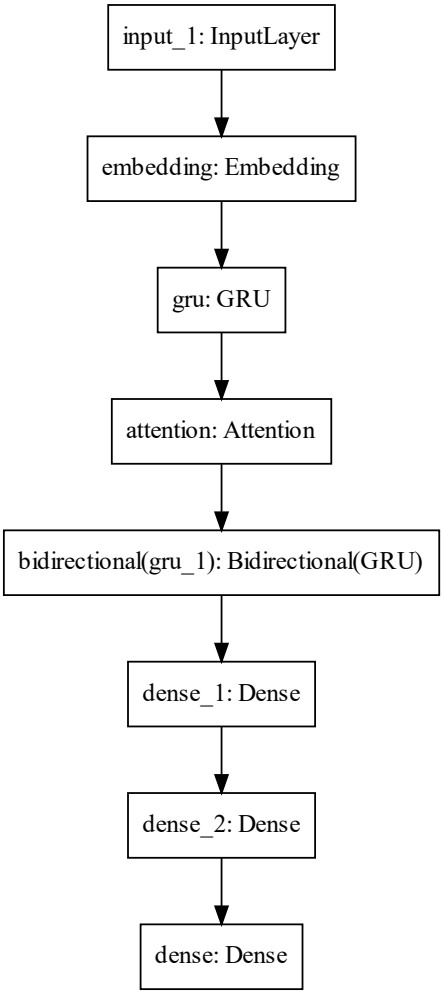
- 1. Odebrání oddělovačů
- 2. Konverze na malá písmena
- 3. Odebrání diakritických znamének
- 4. Stemming (redukce na kořen slova)

Příklad preprocessingu začátku jedné recenze:

- 0. *Tenhle film můžu opravdu kdykoliv. A to nejen*
- 1. *Tenhle film můžu opravdu kdykoliv A to nejen*
- 2. *tenhle film můžu opravdu kdykoliv a to nejen*
- 3. *tenhle film muzu opravdu kdykoliv a to nejen*
- 4. *tenhl film muz opravd kdykoliv a to nejen*

Pro stemming byla použitá knihovna *czech_stemmer* [Gom10], kterou ovšem bylo nutné upravit pro daný úkol, jelikož knihovna předpokládá, že vstupní slova obsahují výhradně korektní češtinu. Problémem však je, že lidé na internetu mnohdy nepíší korektně a často ani nepoužívají diakritiku, proto byly přidány rozšíření, které se pokouší tyto případy zachytit.

Architektura této sítě je podobná, jako v předchozím klasifikátoru recenzí, ale tentokrát jsem se snažil experimentálně dodat další podpůrné prostředky: 2. Rekurentní vrstva je obousměrná a přidal jsem další husté vrstvy jako lehkou podporu pro oddělování kategorií.



Obrázek 5.7: Architektura sítě pro klasifikaci recenzí na ČSFD

Bohužel, dle očekávání, ani tento model se nemohl srovnávat s angličtinou. Ve správných klasifikacích se mé modely pro češtinu pohybovaly při testu zpravidla kolem 70 % po 10 epochách.

Příklad interakce s uživatelem:

```
Tak nudný film jsem už dlouho neviděl.
Prediction: NEGATIVE
Probability: 83.73191952705383 %

Už se těším, až to uvidím znovu.
Prediction: POSITIVE
Probability: 42.69120395183563 %
```

5.6 Klasifikace reakcí na zprávy týkající se onemocnění Covid 19

Pro tuto sekci mi byl dán malý oklasifikovaný dataset obsahující reakce čtenářů na zprávy týkající se onemocnění Covid 19 na úrovni slov. Vyhodnocený sentiment je v datasetu označen těmito kategoriemi: "silné -", "slabé -", "neutrální", "slabé +", "silné +"; kde "+"znamená pozitivní a "-"negativní.

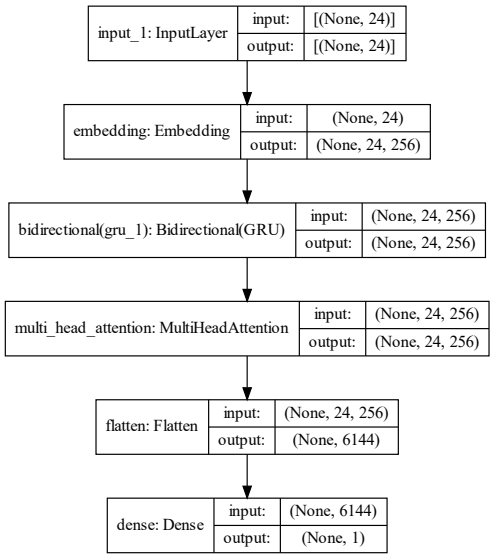
5.6.1 Analýza sentimentu slov v reakcích čtenářů

Dokážeme-li analyzovat sentiment slov ve větě, můžeme pak odhadnout i celkový sentiment reakce, například na základě většinového počtu. V této části se bude neuronová síť snažit klasifikovat slova v reakcích čtenářů do 5 kategorií: Silně negativní, slabě negativní, neutrální, slabě pozitivní a silně pozitivní.

Jelikož je dat velmi málo, síť není možné příliš dlouho trénovat, jinak by došlo k přeučení. Zároveň pravděpodobně nebude možné nastavit dostatečně dobře nastavit velké množství parametrů. Proto jsem na výstupní vrstvě místo 5 neuronů, které by obvykle odpovídaly jednotlivým kategoriím, použil jediný, který reprezentuje osu od 0 do 1. Tuto osu lze rozdělit na 5 částí:

- 0: Silně negativní
- 0.25: Slabě negativní
- 0.5: Neutrální
- 0.75: Slabě pozitivní
- 1: Silně pozitivní

Kategorie je nyní vybrána podle nejbližšího bodu. Myšlenkou je, že se spíše podaří generalizovat směr sentimentu než rozdělování do jednotlivých kategorií, a například chybně označení silně negativního vzorku za slabě negativní považuji za výrazně menší, než označení za silně pozitivní. Slova tentokrát nejsou nijak zvláště předzpracována, jelikož je dat velmi malé množství, zdá se vhodné neztrácet informace.



Obrázek 5.8: Architektura sítě pro klasifikaci reakcí čtenářů na zprávy týkající se onemocnění Covid 19

Po 10 epochách síť dokázala správně klasifikovat pouze 31.94 % případů. V případech, kdy síť klasifikovala alespoň *přibližně správně* (tzn. vzdálenost od správné kategorie je $< \epsilon = 0.3$) vyšla přesnost 78.32 %.

Tento příklad je v této práci později vizualizován.

5.7 Analýza sentimentu komentářů u videí na YouTube souvisejících s onemocněním Covid 19

Pro tuto sekci byl použit veřejně dostupný dataset[seu], jehož data jsou tvořena z komentářů a jejich metadat u vybraných videí (tématicky se týkající onemocnění Covid 19) na platformě YouTube. Podstatná část komentářů tvořící tento dataset pochází z videí zpravodajských kanálů jako NBC News, BBC, CNN a mnoho dalších. Naprostá většina komentářů je v jazyce anglickém. Dataset obsahuje videa nahraná v časovém období leden 2021 a jejich komentáře z období leden 2021 až březen 2021. Tento dataset je obzvláště zajímavý, jelikož na rozdíl od předchozích datasetů obsahuje temporální data, tedy komentářům jsou přiřazeny časové známky, což je zajímavé z pohledu vizualizace, jelikož uživatel nyní může zkoumat časový vývoj.

Dataset ovšem neobsahuje analýzu sentimentu. Proto tento krok reprezentuje propojení předchozího úsilí s novými daty. Sentiment komentářů byl vyhodnocen obdobně jako recenze na platformě Yelp a k existujícímu datasetu byly přidány nové atributy - kategorie sentimentu. Každý komentář byl na základě predikce klasifikátoru označkován jedné z kategorií: pozitivní, neutrální nebo negativní.

Výsledky budou použity ve vizualizační sekci.

Kapitola 6

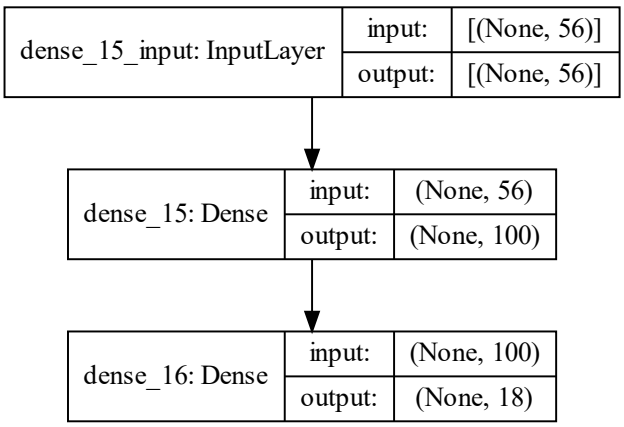
Vizualizační experimenty

Vizualizační experimenty v této sekci představují propojující krok mezi zpracováním přirozeného jazyka a vizualizací. Různými technikami jsou zobrazovány výsledky získané v předchozích krocích.

6.1 Vizualizace architektury neuronové sítě

Při práci s neuronovými sítěmi, nejen při zpracování přirozeného jazyka, je užitečné získat nadhled. Struktura sítě je obvykle zapsána kódem nebo uložena v binárním formátu, který pro člověka není čitelný. Pochopit tak architekturu takto reprezentované neuronové sítě není triviální. Proto je vhodné strukturu vizualizovat a usnadnit tak čtenáři nebo vývojáři jejímu porozumění.

Jednou z výhod knihovny Keras[GP17] je, že umožňuje snadnou vizualizaci architektury neuronové sítě. Tyto vizualizace byly mnohdy využity při vývoji neuronových sítí, ale také přímo v této práci - v experimentální části při zpracování přirozeného jazyka. Použití je přitom velmi jednoduché, formát výstupního obrázku určuje koncovka jména souboru. Je možné zvolit i rastrové formáty.



Obrázek 6.1: Příklad vizualizace architektury sítě v knihovně Keras

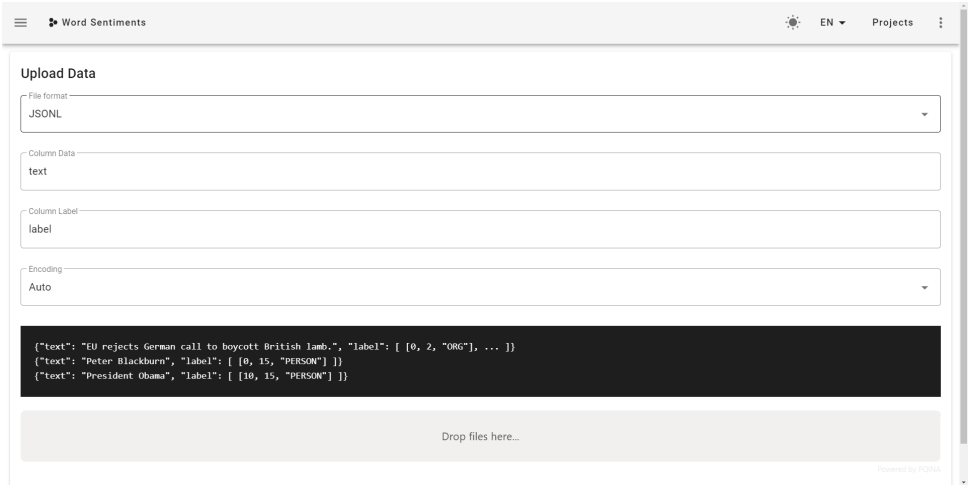
6.2 Vizualizace reakcí na zprávy týkající se onemocnění Covid 19

6.2.1 Vizualizace sentimentu slov v reakcích čtenářů v nástroji Doccano

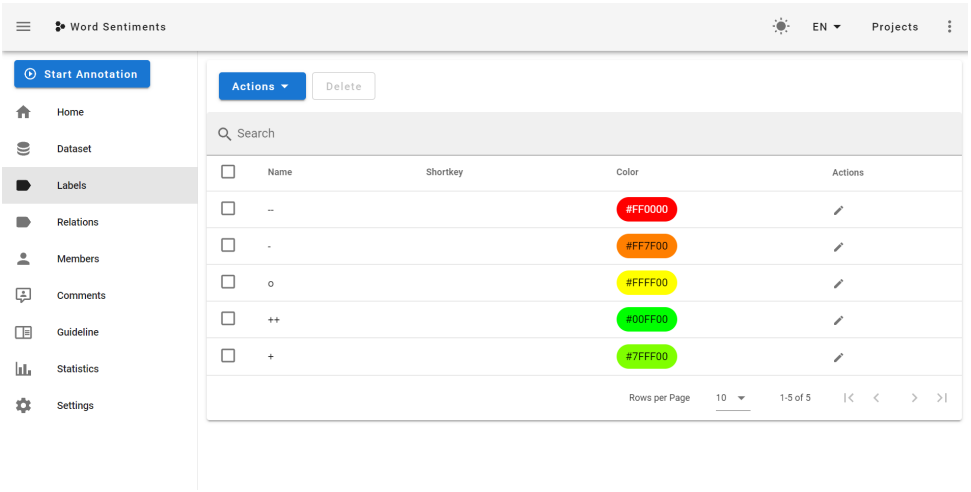
Toto je vizualizační pokračování NLP experimentu Klasifikace reakcí na zprávy týkající se onemocnění Covid 19.

Pro vizualizaci sentimentu slov byl použit nástroj *Doccano* [NKK+18]. Tento software umožňuje provádět jednoduché vizualizace na úrovni textu pomocí webového rozhraní. Stačí vytvořit značky (*labels*), do kterých se budou textové entity rozřazovat a také dataset, který bude vizualizován ve validním formátu *json* či *jsonl* a je nutné jej do Doccana importovat.

Dataset ve formátu *jsonl* dokáže Python snadno vyrobit díky knihovně *json*. Značky je možné vyrobit buď interaktivně uživatelem přímo v Doccanu, ale existuje i možnost hotové značky importovat, opět v *json* formátu. Výhodou *json* formátu je, že jej mnoho knihoven dokáže snadno vyrobit a přečíst. Zároveň je relativně čitelný i pro člověka, tady opravit případné chyby na úrovni syntaxe či obsahu není obtížné.



Obrázek 6.2: Ukázka rozhraní pro import datasetu do nástroje Doccano

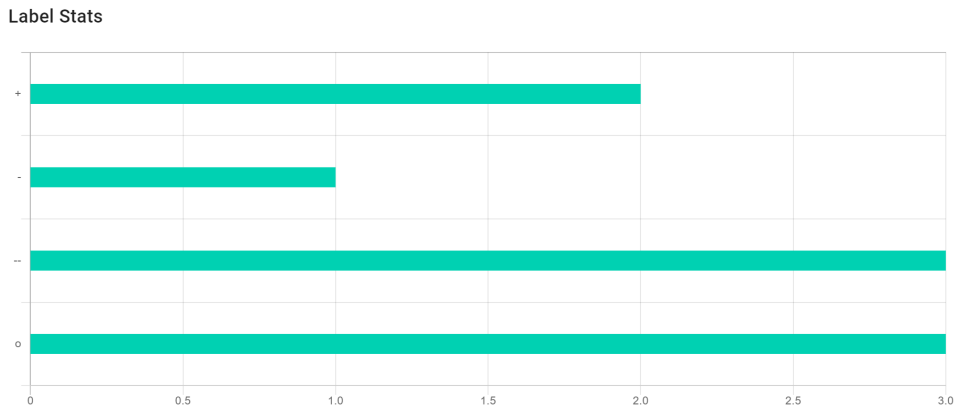


Obrázek 6.3: Ukázka rozhraní pro zvorbu značek v nástroji Doccano

Vizualizace textu pak vypadá následovně:

Obrázek 6.4: Ukázka vizualizace sentimentu slov v nástroji Doccano

Sentimenty slov jsou nyní vizualizovány barevným podtržením, červená barva reprezentuje negativní sentiment, zelená pozitivní, žlutá neutrální. Celá věta je současně zobrazena jako text. Díky statistice, kterou nástroj Doccano nabízí, můžeme dokonce odhadnout sentiment celého textu na základě většinového sentimentu slov:



Obrázek 6.5: Ukázka vizualizace sentimentu slov v nástroji Doccano

Je vidět, že většina slov je buď silně negativní nebo neutrální. Silně pozitivní slova se v textu ani nevyskytují. Na základě toho lze tedy odhadnout, že text má celkově negativní sentiment.

6.3 Vizualizace sentimentu komentářů u videí na YouTube souvisejících s onemocněním Covid 19

Toto je pokračování NLP experimentu Analýza sentimentu komentářů u videí na YouTube souvisejících s onemocněním Covid 19.

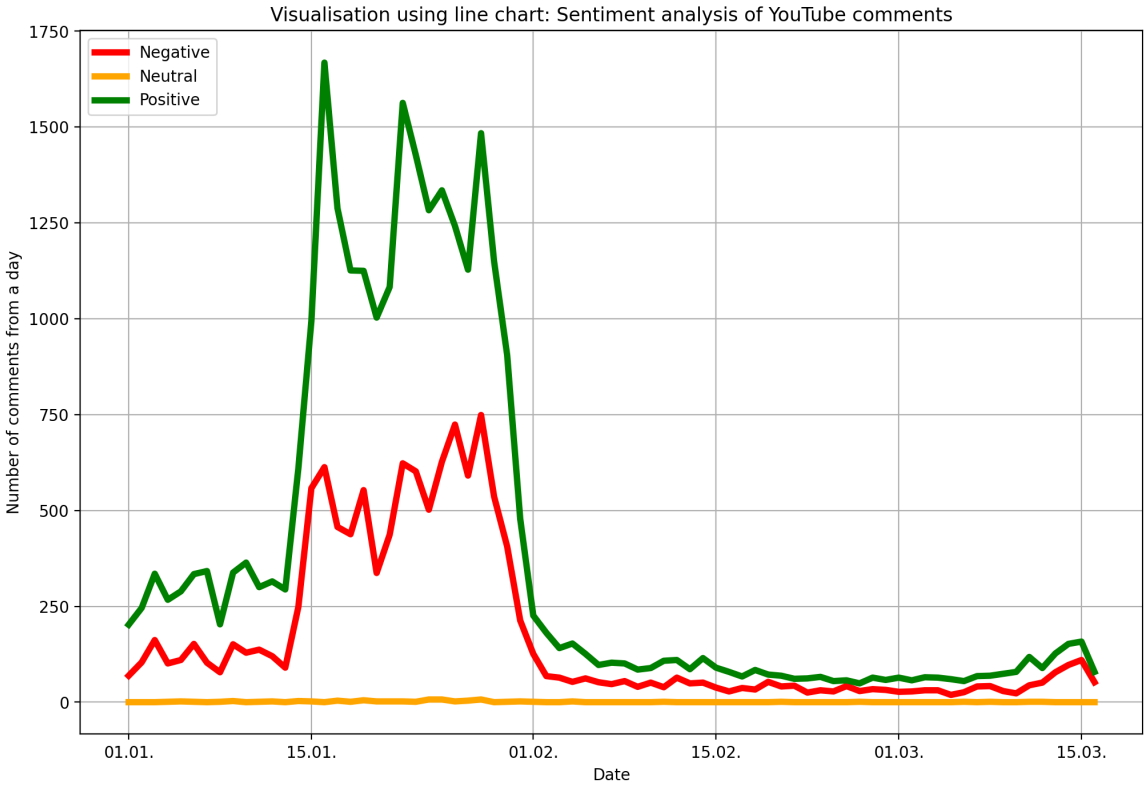
V experimentální sekci byl popsán dataset komentářů u videí na YouTube s metadaty (obsahující mimo jiné čas), který bude použit pro vizualizace v této sekci. Data pocházejí z doby kolem ledna 2021. Komentáře nebyly v tomto datasetu ohodnoceny, namísto toho byl jejich sentiment odhadnut neuronovou sítí, která byla natrénována na jiných datech. Dataset komentářů tedy nyní navíc obsahuje kategorický atribut sentiment. Sentiment může nebývat hodnot pozitivní, neutrální nebo negativní.

Tento dataset je zajímavý z pohledu vizualizace, neboť kromě samotných komentářů obsahuje i mimo jiné adresu videa, hodnocení videa (pozitivní a negativní), hodnocení komentáře (pouze pozitivní), počet zhlédnutí, ale především časové značky - datum vytvoření komentáře a nahrání videa. To otevírá nové úlohy, zkoumání časového vývoje sentimentu. Je vhodné podotknout, že hodnocení videa nemusí nutně odpovídat hodnocení dané zprávy.

Vizualizace v této sekci byly realizovány převážně pomocí knihoven *matplotlib*[Hun07] a *seaborn*[Was21].

6.3.1 Vizualizace čárovým diagramem

Jednoduchou, nicméně efektivní vizualizací je čárový diagram. Pro každou kategorii sentimentu jsou v každém relevantním časovém bodě (den) vyneseny počty komentářů spadající do dané kategorie a následně propojeni čárou, tedy jsou implicitně interpolovány. Jednotlivé čáry reprezentující dané kategorie jsou různě obarveny pro rozlišitelnost, čára pozitivního sentimentu zeleně, negativního červeně a neutrálního oranžově. Na ose x se nacházejí česové termíny na úrovni dnů a na ose y počty komentářů v dané kategorii pro daný den.



Obrázek 6.6: Vizualizace sentimentu komentářů v čase pomocí čárového diagramu

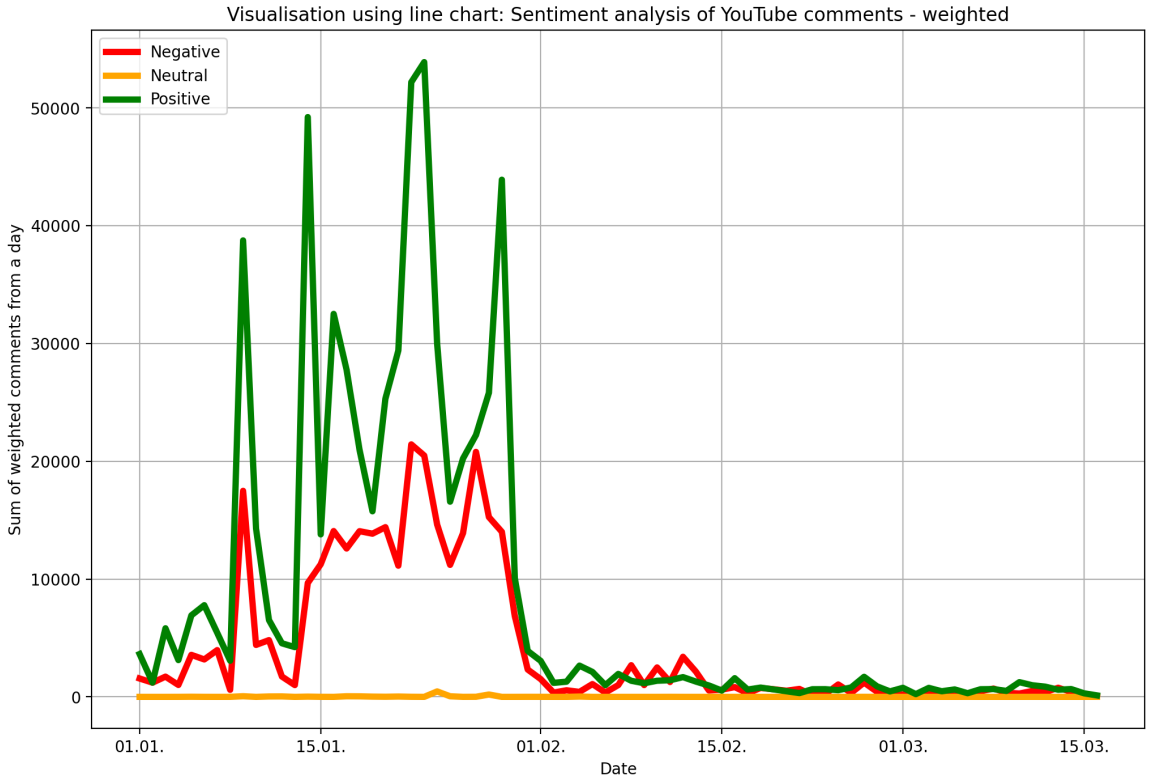
Díky této vizualizaci je možné zodpovědět dotazy:

1. Kolik komentářů ze dne X bylo ohodnoceno sentimentem Y?
2. Dne X, komentářů které kategorie bylo nejvíce/nejméně?
3. Vývoj počtu komentářů dané kategorie v čase.

Na základě této vizualizace můžeme vypozorovat, že například v druhé polovině ledna byl značný nárůst komentářů, nebo že většina komentářů je ohodnocena jako pozitivní. Dále je možné si povšimnout, že neutrálních komentářů je velmi málo. Dále je vidět, že počet pozitivních a negativních komentářů v čase spolu pravděpodobně souvisí, jelikož tvar čar těchto kategorií je podobný.

Z datasetu je však možné vyzískat ještě více informací, jelikož známe počet pozitivních hodnocení komentářů, dále je možné například vizualizovat vážené

komentáře, kde každý komentář má váhu $(1 + likes(c))$, kde $likes(c)$ je počet pozitivních hodnocení komentáře. Tyto váhy jsou sečteny. Myšlenkou je započítat souhlasné reakce s komentářem daného sentimentu, které reprezentují stejný sentiment jako hodnocený komentář.



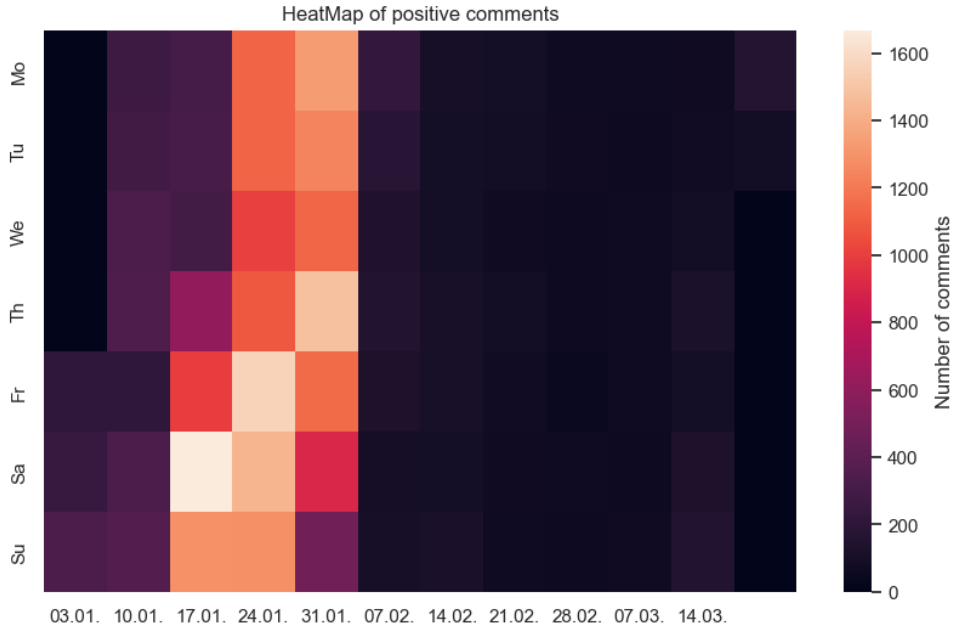
Obrázek 6.7: Vizualizace sentimentu vážených komentářů v čase pomocí čárového diagramu

Tato vizualizace nyní není odrazem počtu komentářů, ale zahrnuje i reakce získané z metadat, které nebyly součástí samotného textu. Můžeme vypozorovat podobné informace jako v předchozím případě.

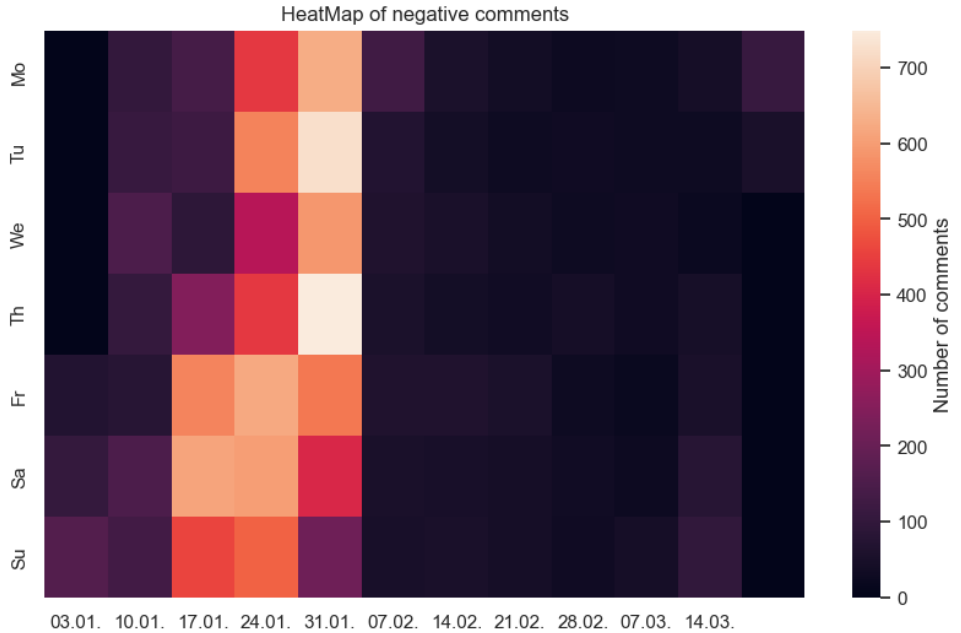
6.3.2 Vizualizace pomocí HeatMap

HeatMap je dvourozměrná barevná matice, jejíž buňky jsou obarveny na základě četnosti zastoupení v daném bodě. Jelikož data komentářů pokrývají několik měsíců, následující vizualizaci jsou časová data rozdělena na týdny. To umožňuje sledovat vzory v rámci dnů v týdnu, které by byly jinak obtížně viditelné v klasické lineární reprezentaci. Na ose x jsou namapovány týdny,

na ose y dny v týdnu. Je to tedy jistým způsobem kalendářní pohled. Buňky jsou obarveny na základě počtu komentářů, světlé barvy značí vyšší počet, tmavší nižší.



Obrázek 6.8: Vizualizace sentimentu pozitivních komentářů v čase pomocí *HeatMap*

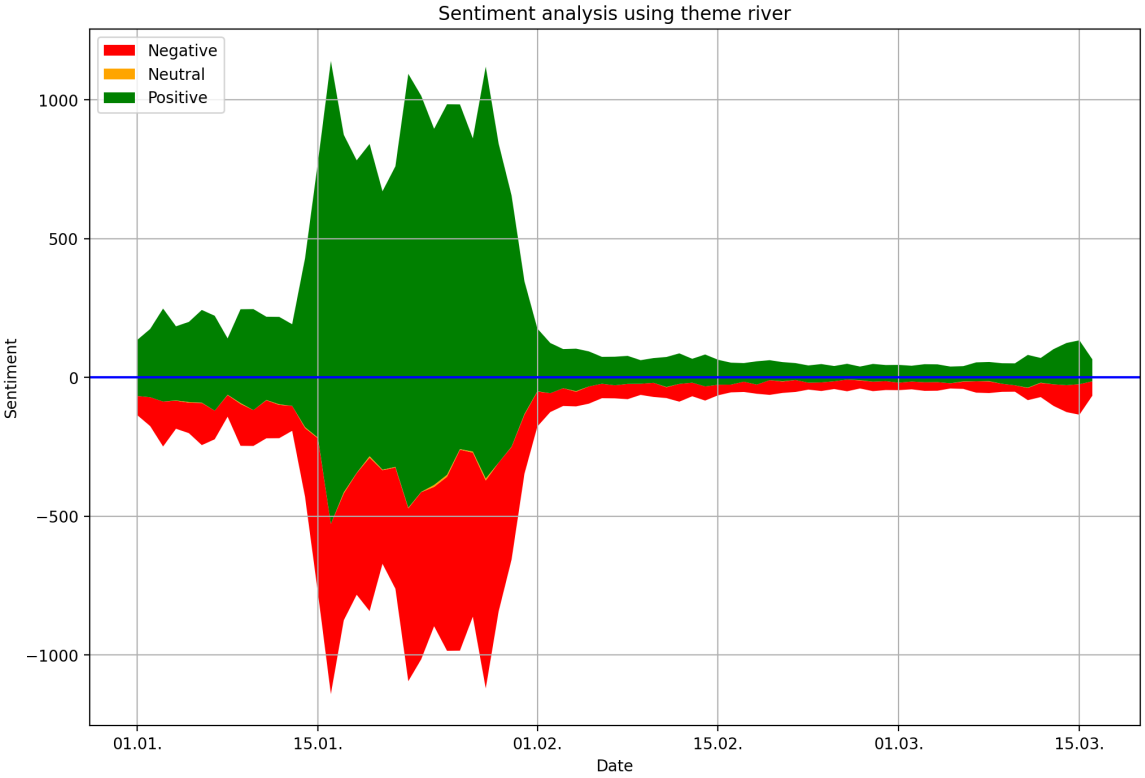


Obrázek 6.9: Vizualizace sentimentu negativních komentářů v čase pomocí *HeatMap*

Na základě této vizualizace můžeme například vypozorovat, že nejvíce negativních komentářů bylo napsáno v posledním lednovém týdnu.

6.3.3 Vizualizace pomocí ThemeRiver

Pomocí ThemeRiver je možné vidět časový vývoj v datech a jednotlivých kategoriích. V každém relevantním časovém bodě (den) jsou vyneseny počty komentářů symetricky podle osy x. Jednotlivé kategorie jsou v této vizualizaci rozlišeny barevně, pozitivní je zelená, neutrální je oranžová a negativní je červená.



Obrázek 6.10: Vizualizace sentimentu komentářů v čase pomocí *ThemeRiver*

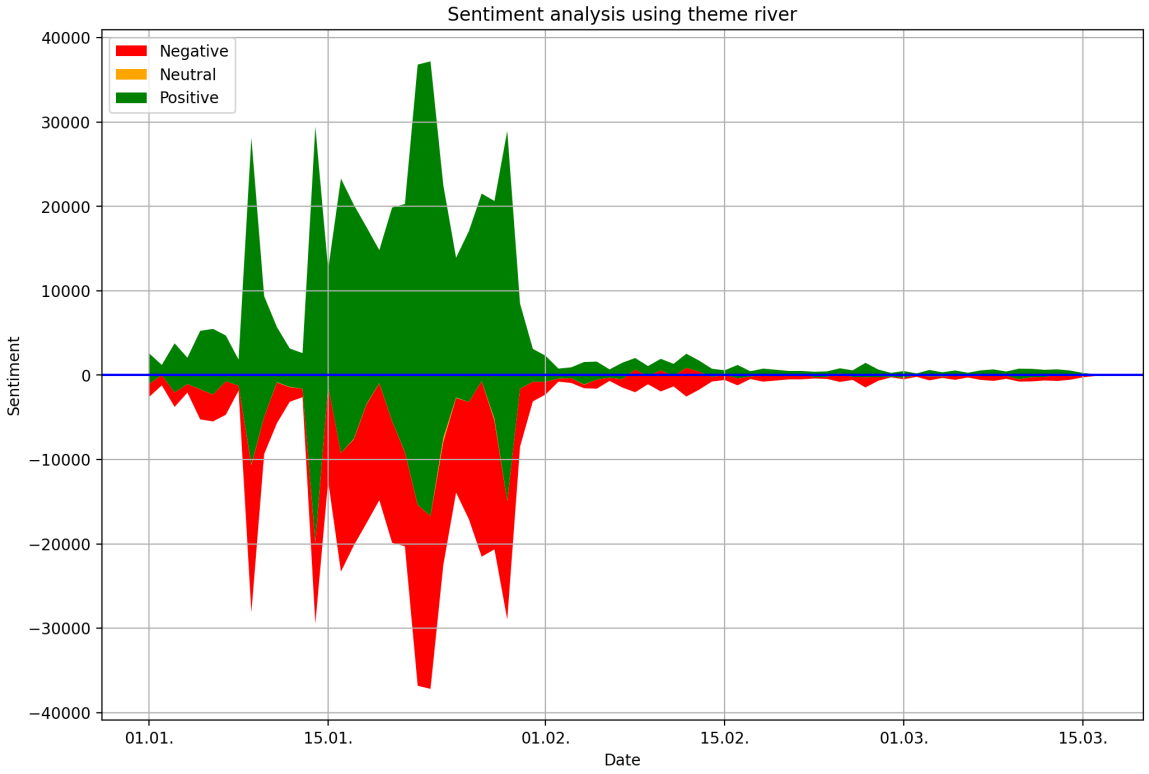
Díky této vizualizaci je možné zodpovědět dotazy:

1. Vývoj a porovnávání počtu komentářů v různých časových intervalech.
2. Vývoj počtu komentářů dané kategorie v čase.

3. Proporce komentářů dané kategorie vůči celkovému počtu komentářů v čase.

V této vizualizaci si můžeme například opět povšimnout nárůstu počtu komentářů v druhé polovině ledna. Dále je vidět, že pozitivních komentářů je více než negativních a jakou proporci komentářů přibližně tvoří.

Pro účel získání dalších informací je možné opět obohatit data vyvážením komentářů vzorcem $(1+likes(c))$, kde $likes(c)$ je počet pozitivních hodnocení komentáře.

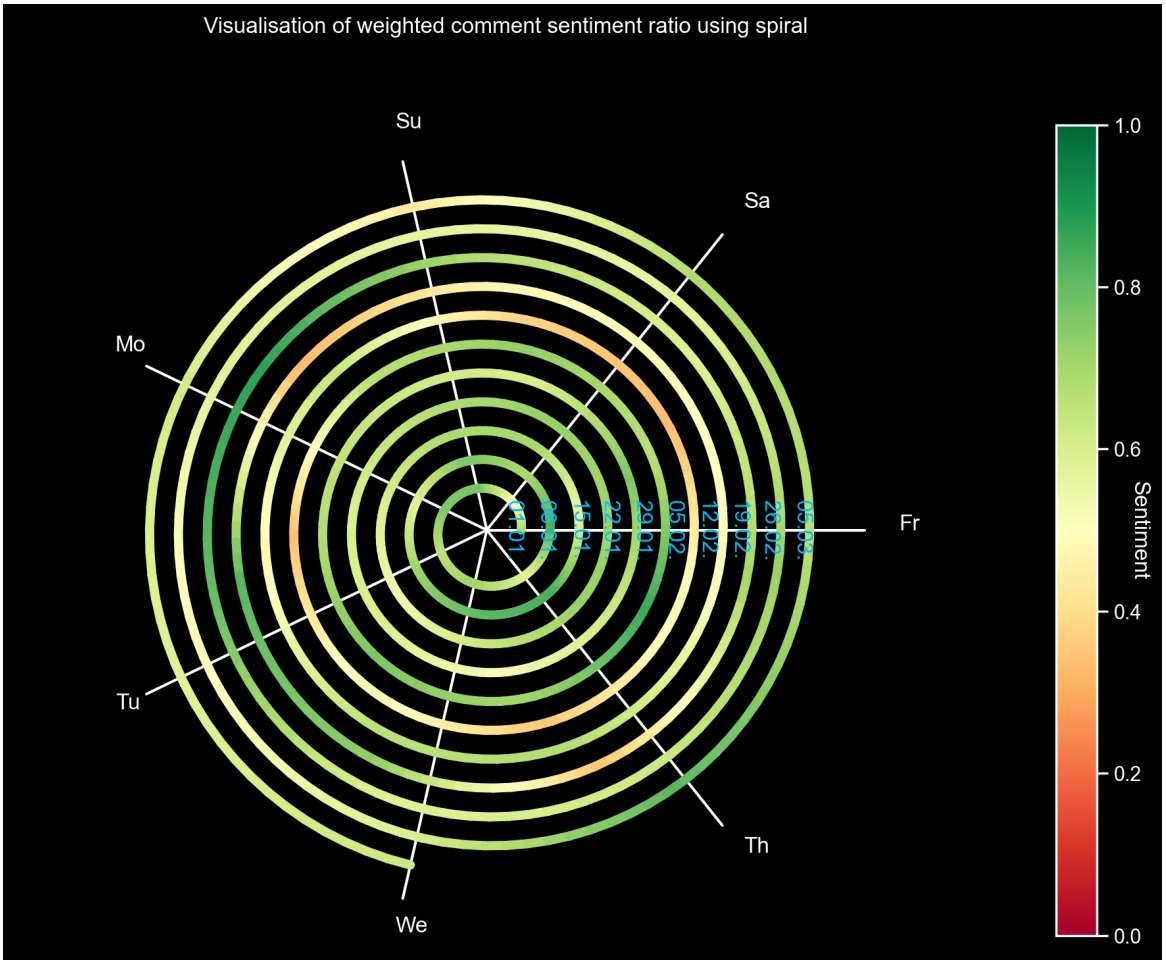


Obrázek 6.11: Vizualizace sentimentu vážených komentářů v čase pomocí *ThemeRiver*

6.3.4 Vizualizace spirálovým diagramem

Vizualizace na spirálu umožňuje vidět v časově orientovaných datech vývoj i sezónní trendy. Tento typ vizualizace pracuje s cyklickou časovou osou. Časová osa se postupně rozvíjí do šířky v periodách odpovídajících danému

časovému intervalu. V této vizualizaci jsou na časovou osu (spirálu) naneseny barevně údaje o poměru sentimentu vážených komentářů v daném dni, vypočteno pomocí vzorce $\frac{positive(d)}{positive(d) + neutral(d) + negative(d)}$, kde tyto funkce vyjadřují počet vážených komentářů dané kategorie v určitý den d.



Obrázek 6.12: Vizualizace sentimentu vážených komentářů v čase pomocí spirály

Díky této vizualizaci je možné zodpovědět dotazy:

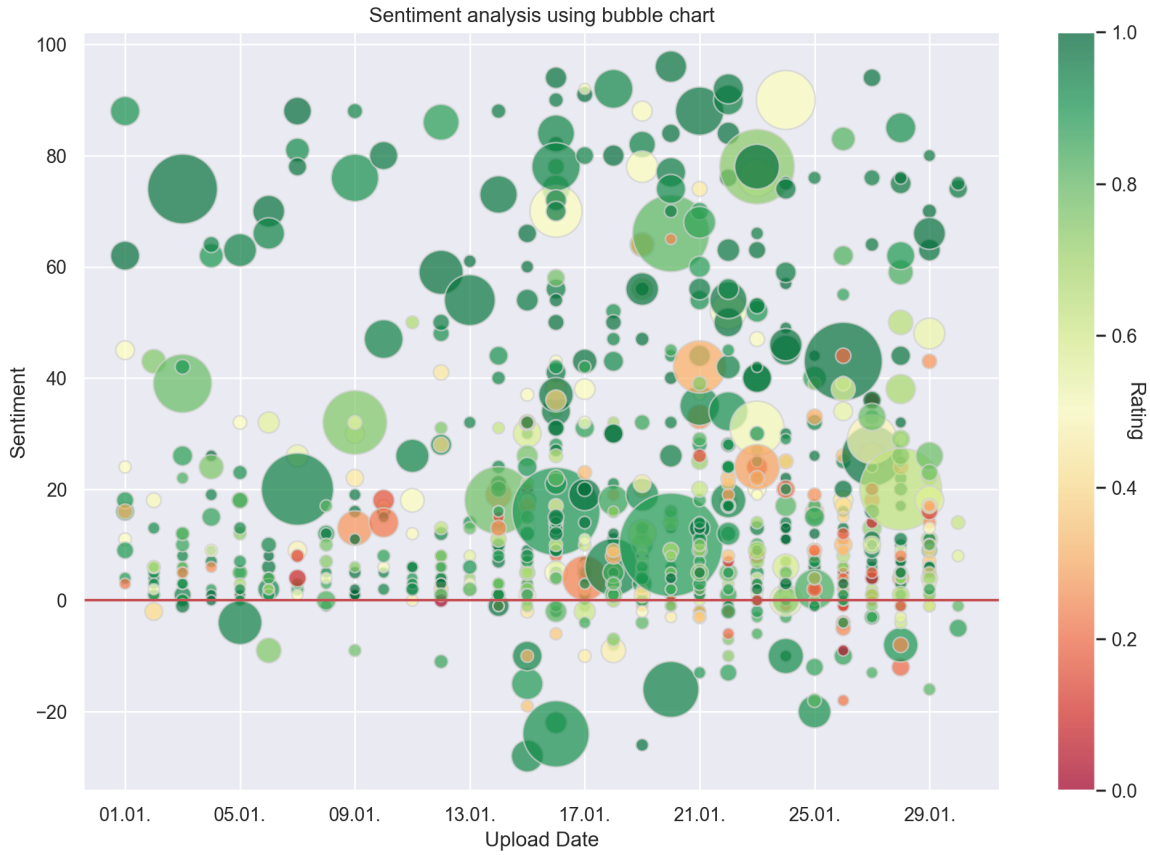
1. Vývoj a porovnávání počtu komentářů v periodických časových intervalech.
2. Vývoj poměru váženého sentimentu komentářů v čase.
3. Jaká je hodnota sledovaného atributu v daném časovém bodě?

Na základě této vizualizace můžeme vypozorovat časový vývoj: Na počátku převažovaly pozitivní reakce, v první polovině února byly dominantní negativní ohlasy, ovšem ve druhé polovině února opět pokračoval trend pozitivních reakcí. Můžeme také odhadnout, že v pátek vznikají pozitivnější reakce než v pondělí, neboť páteční osa (označená *Fr*) protíná zelenější barvy než osa pondělní (označená *Mo*).

6.3.5 Vizualizace pomocí Bublinového diagramu

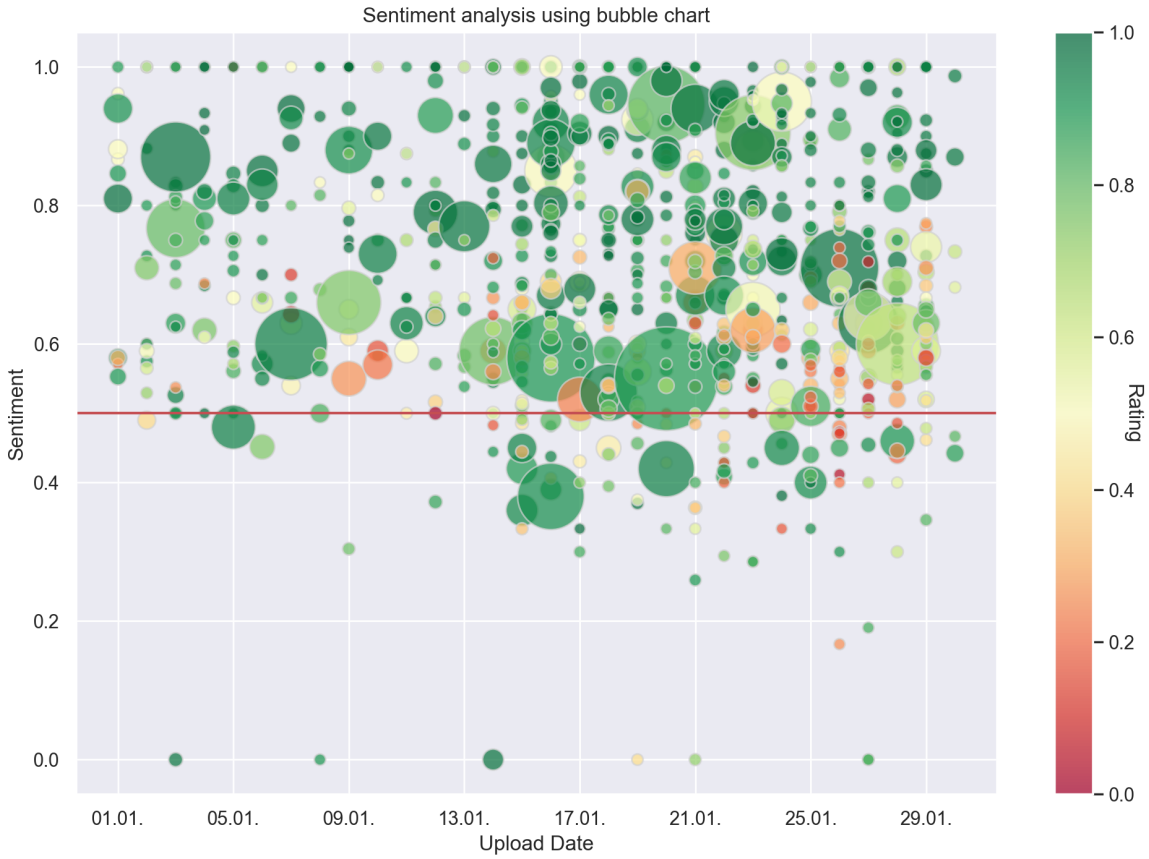
Bublinový diagram, jakožto rozšíření bodového diagramu, umožňuje zapojení dalších atributů. V této sekci jsou vizualizována videa jako jednotlivé body, nikoliv komentáře. Na časové ose x je datum nahrání. Na ose y je relativní sentiment komentářů jako rozdíl, tedy $(positive(v) - negative(v))$, kde $positive(v)$ vyjadřuje počet komentářů daného videa, které byly vyhodnoceny jako pozitivní a $negative(v)$ negativní. Neutrální komentáře nejsou započítány, jelikož nenesou žádný sentiment.

Je rovněž možno využít metadat; v následující vizualizaci vyjadřuje barva poměr pozitivních reakcí na video jako $\frac{likes(v)}{likes(v) + dislikes(v)}$, kde $likes(v)$ vyjadřuje počet pozitivních hodnocení videa a $dislikes(v)$ počet negativních hodnocení videa. Velikost bodu vyjadřuje počet zhlédnutí videa (tedy dosah).



Obrázek 6.13: Vizualizace hodnocení videa nahraného v čase na základě sentimentu komentářů a poměru pozitivních reakcí pomocí Bublinového diagramu

Sentiment komentářů můžeme rovněž získat jako poměr pozitivních komentářů oproti celkovému počtu komentářů podobným vzorcem:
$\frac{positive(v)}{positive(v) + neutral(v) + negative(v)}$.
V tomto případě již neutrální komentáře zapojujeme, například video, které by mělo 100 neutrálních komentářů a 1 negativní komentář neodpovídá silně negativnímu sentimentu. Taková vizualizace by pak vypadala takto:



Obrázek 6.14: Vizualizace hodnocení videa nahraného v čase na základě poměru sentimentu komentářů a poměru pozitivních reakcí pomocí Bublinového diagramu

Na vizualizacích výše je osa x časová (datum nahrání videa), osa y vyjadřuje sentiment komentářů (rozdíl nebo poměr), velikost bodů počet zhlédnutí, barva poměr hodnocení videa.

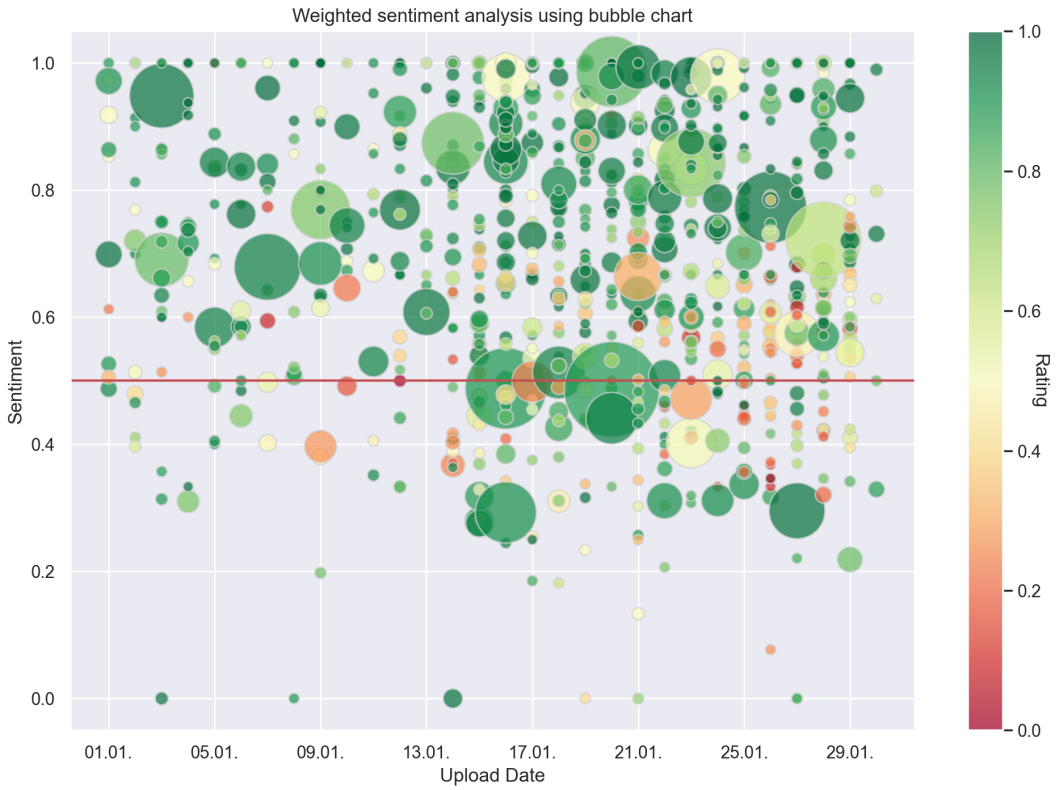
Díky této vizualizaci je možné zodpovědět dotazy:

1. Vývoj sledovanosti videí.
2. Vzory v hodnocení, sentimentu komentářů, počtu zhlédnutí a čase.

Z této vizualizace můžeme vypozorovat, že videa z velkým dosahem (mají mnoho zhlédnutí, velké body) jsou typicky dobře hodnoceny na základě hodnocení uživatelů, ale značná část komentářů může být negativní. Dále

je vidět, že videa, která jsou dobře hodnocena mají mnohdy i pozitivní komentáře (většina zelených bodů je umístěna nahoře, zatímco červené body jsou koncentrovány níže). Také si můžeme povšimnout, že větší body jsou umístěny spíše vpravo, což může značit vyšší sledovanost v druhé polovině ledna.

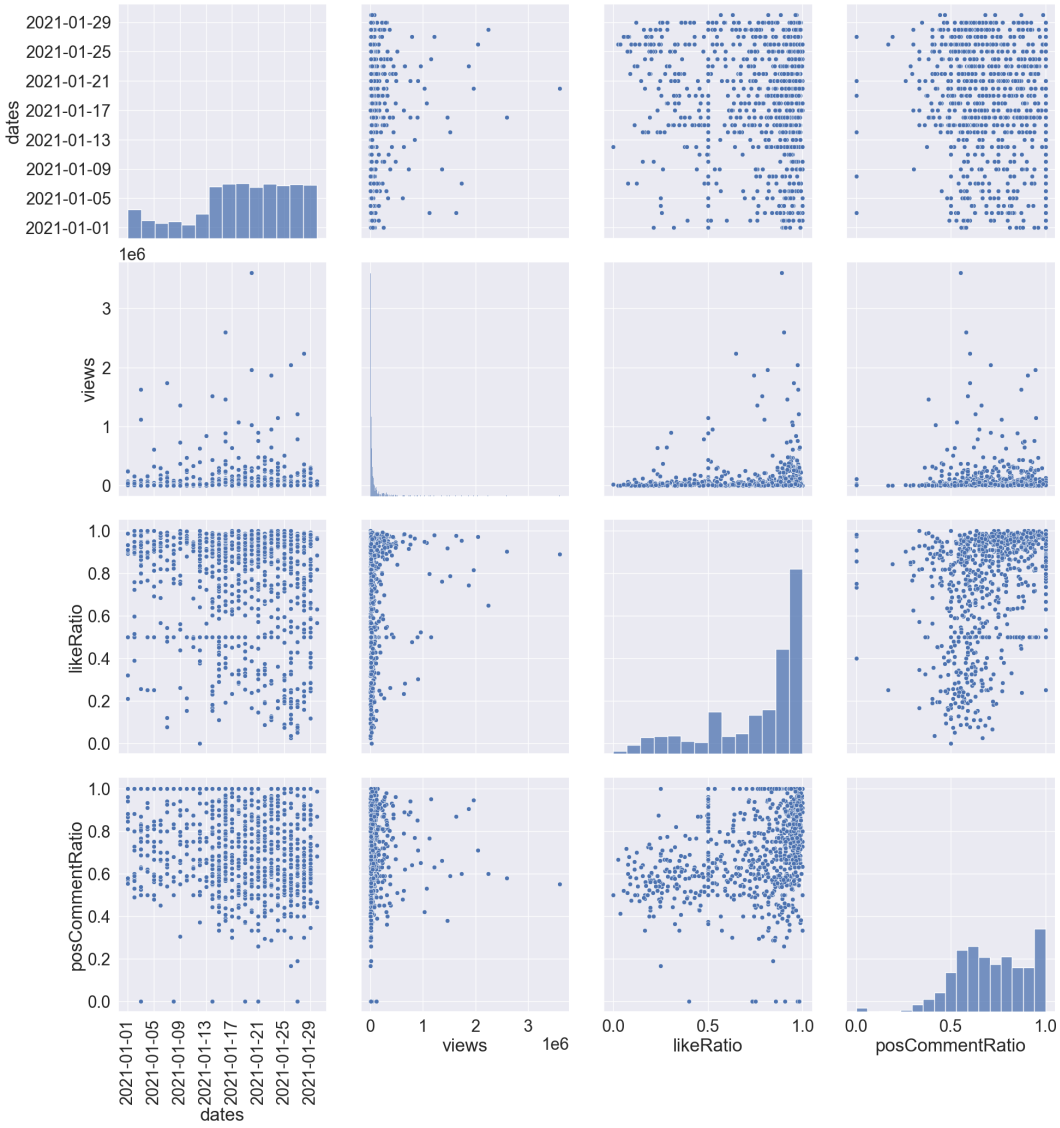
Komentáře je opět možné vyvážit na základě jejich hodnocení:



Obrázek 6.15: Vizualizace hodnocení videa nahraného v čase na základě poměru sentimentu vážených komentářů a poměru pozitivních reakcí pomocí Bublinového diagramu

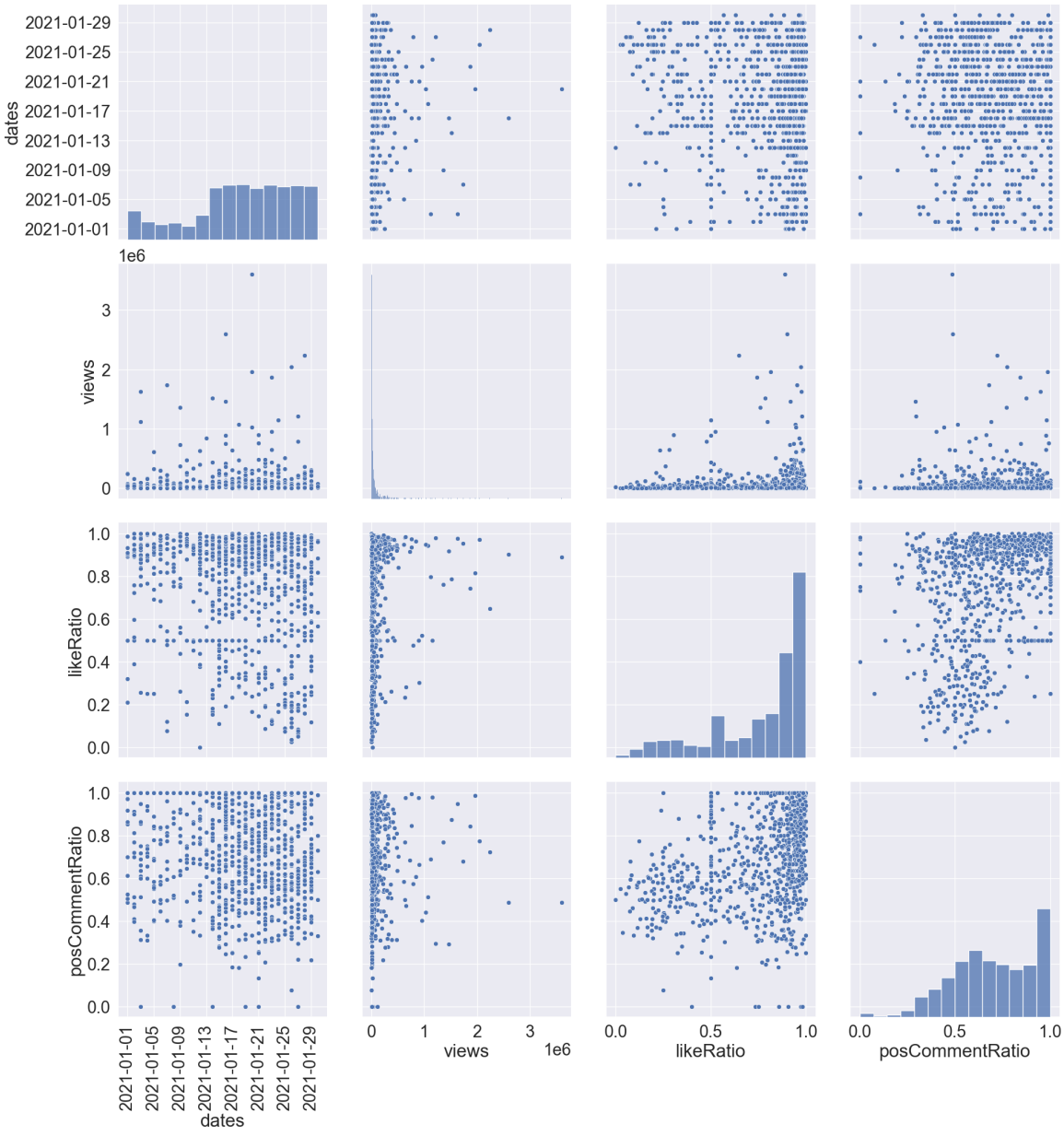
6.3.6 Vizualizace párovou maticí

Párová matice umožňuje vytvořit přehled vztahů mezi dvojicemi atributů. Jedná se o kolekci diagramů uspořádaných do mřížky, kde osy x a y tvoří jednotlivé atributy. Získáváme tedy $|a|^2$ vizualizací, kde $|a|$ je počet atributů, uspořádaných v matici.



Obrázek 6.16: Vizualizace párovou maticí získaných atributů videa

Komentáře je opět možné vyvážit na základě jejich hodnocení a získat následující vizualizaci:



Obrázek 6.17: Vizualizace párovou maticí získaných atributů videa s váženými komentáři

Díky této vizualizaci je možné zodpovědět dotazy:

1. Vztah mezi dvojicemi atributů.

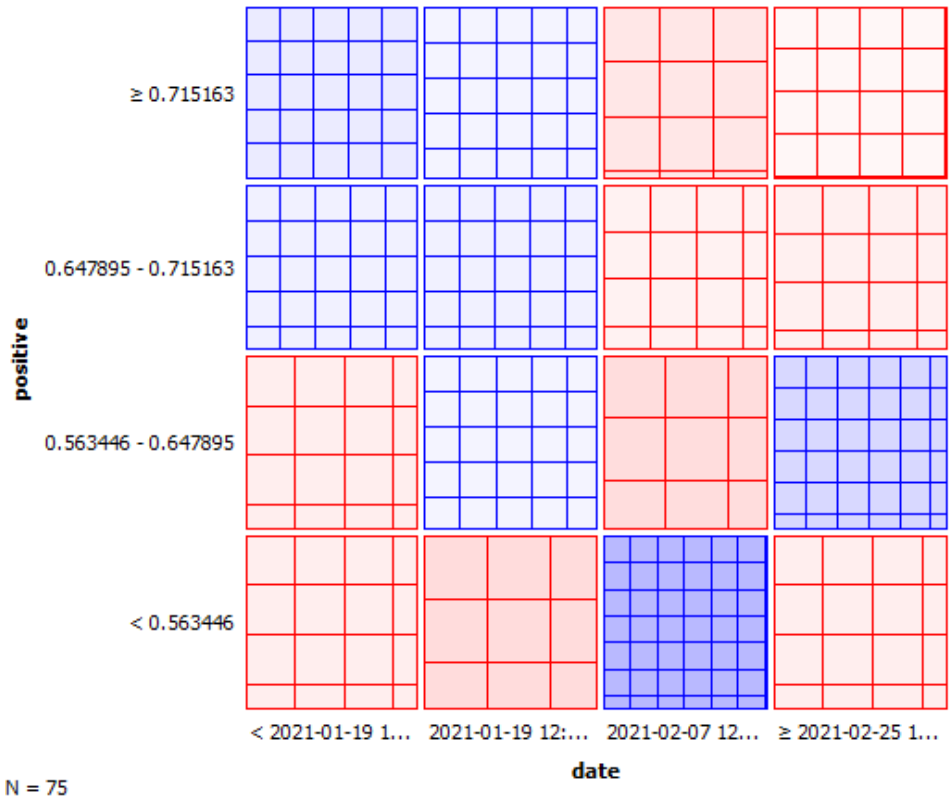
2. Rozložení dat (například v čase).

3. Hledání vzoru mezi různými atributy.

Na základě této vizualizace si můžeme povšimnout, že více sledovaných videí je v druhé polovině ledna (vyšší koncentrace bodů v *dates-views*). Dále také, že pozitivně hodnocená videa mívají komentáře, jejichž sentiment byl také vyhodnocen jako pozitivní (koncentrace bodů v pravém horním rohu v *likeRatio-posCommentRatio*).

6.3.7 Vizualizace sítovým diagramem

Zajímavým způsobem vizualizace dat je *sítový diagram* [RS94], který na základě 2 kategorických atributů vytváří mřížku, jejíž buňky mají velikost podle očekávané frekvence. Buňky jsou dále vyplněny mřížkou. Je-li pozorovaná frekvence vyšší, než očekávaná frekvence, buňka je obarvena modře, je-li nižší, buňka je obarvena červeně. Roli v barvě hraje také hustota. Buňky jsou dále rozděleny mřížkou odpovídající barvy, která vizualizuje hustotu. Tato vizualizace byla realizována v nástroji *Orange*, ovšem data byla připravena v jazyce *Python*. Jsou na ní zobrazen poměr vážených pozitivních komentářů v čase.



Obrázek 6.18: Vizualizace vážených komentářů v čase síťovým diagramem

Je zřejmé, že tato vizualizace pracuje s diskrétními hodnotami. Časová osa byla rovnoměrně rozdělena na 4 části, osa pozitivních komentářů byla rozdělena na základě rozložení dat.

Na základě této vizualizace můžeme pozorovat, ve kterých časových intervalech byl nečekaně vysoký/nízký vážený poměr pozitivních vážených komentářů. Například si můžeme povšimnout, že mezi 7. a 25. únorem bylo nejvíce nečekaně nízký poměr těchto komentářů (čtverec ve 4. řádku a 3. sloupci je modře zbarven a jeho mřížka je relativně hustá).

Část IV

Závěrečná část

Kapitola 7

Závěr

7.1 Zhodnocení výsledků

V rámci této práce byl vytvořen řetězec pro zpracovávání textových zpráv.

Byl vytvořen klasifikátor sentimentu recenzí na platformě Yelp, který pracuje s anglickými texty. V této oblasti se dají očekávat výsledky s přesností kolem 70-80 %[Liu20], především u sítí, které nejsou velmi komplexní, proto považuji přesnost 84.61 % za relativně dobrou.

Pro zpracování českého jazyka byl zapojen *stemming*, jehož účelem je zjednodušovat jednotlivá slova do jejich základů a usnadnit tak klasifikátoru práci. Pro jazyky, které mají mnoho tvarů jednoho slova může být vhodné jej použít, ovšem slova částečně ztrácejí význam či mohou získat význam jiný, což může ovlivnit přesnost klasifikátoru.

Navržené vizualizace se jeví vhodné pro daný úkol, jelikož je na jejich základě možné v datech pozorovat vzory, časový vývoj, apod.

7.2 Závěr

Tato práce se zabývá metodami zpracování přirozeného jazyka českého i anglického za použití strojového učení a dále aplikací vizualizačních technik pro zobrazování výsledků.

Během její realizace jsem se naučil základy strojového učení a rozšířil tak své znalosti z oboru umělé inteligence.

Byl navržen řetězec pro analýzu sentimentu textových zpráv a vizualizaci. Práce navrhuje předzpracování textových řetězců, analýzu pomocí neuronových sítí a různé vizualizační techniky aplikované na výsledky získané v předchozích krocích.

V experimentální části bylo provedeno četné množství pokusů v oblasti zpracování přirozeného jazyka za použití neuronových sítí, některé dosahovaly relativně dobrých výsledků.

Dále jsou prezentovány různé vizualizační techniky, které umožňují člověku snazší pochopení výsledků, pozorování vývoje a hledání vzorů v datech. Vizualizace se zpravidla zabývaly hodnocením mediálních zpráv.

Přílohy

Příloha A

Rejstřík

Příloha B

Literatura

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., *Tensorflow: A system for large-scale machine learning*, 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[AMM+07] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski, *Visualizing time-oriented data—a systematic view*, Computers & Graphics **31** (2007), no. 3, 401–409.

[BBGC16] Ashwin Bhandare, Maithili Bhide, Pranav Gokhale, and Rohan Chandavarkar, *Applications of convolutional neural networks*, International Journal of Computer Science and Information Technologies **7** (2016), no. 5, 2206–2215.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078 (2014).

[CW08] Ronan Collobert and Jason Weston, *A unified architecture for natural language processing: Deep neural networks with multitask learning*, Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.

[DCE+13] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar,

Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan, *Orange: Data mining toolbox in python*, Journal of Machine Learning Research **14** (2013), 2349–2353.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).

[Fra98] Andrew U Frank, *Different types of "times" in gis*, Spatial and temporal reasoning in geographic information systems (1998), 40–62.

[FXJ20] Yujie Fang, Hui Xu, and Jie Jiang, *A survey of time series data visualization research*, IOP Conference Series: Materials Science and Engineering, vol. 782, IOP Publishing, 2020, p. 022013.

[Gom10] Luís Gomes, *czech stemmer*, Nov 2010.

[GP17] Antonio Gulli and Sujit Pal, *Deep learning with keras*, Packt Publishing Ltd, 2017.

[HHN00] Susan Havre, Beth Hetzler, and Lucy Nowell, *Themeriver: Visualizing theme changes over time*, IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings, IEEE, 2000, pp. 115–123.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.

[Hun07] J. D. Hunter, *Matplotlib: A 2d graphics environment*, Computing in Science & Engineering **9** (2007), no. 3, 90–95.

[KHMB16] Hong Kyu Kang, Ann P Harch, Nicole Martin, and Emma M Birath, *Using computer visualization as a verification tool for new horizons' pluto encounter instrument operations*, 14th International Conference on Space Operations, 2016, p. 2400.

[KK15] Kostiantyn Kucher and Andreas Kerren, *Text visualization techniques: Taxonomy, visual survey, and community insights*, 2015 IEEE Pacific visualization symposium (pacificVis), IEEE, 2015, pp. 117–121.

[KKKS23] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh, *Natural language processing: State of the art, current trends and challenges*, Multimedia tools and applications **82** (2023), no. 3, 3713–3744.

[Kub17] Miroslav Kubat, *An introduction to machine learning*, Springer, 2017.

[Liu20] Siqi Liu, *Sentiment analysis of yelp reviews: a comparison of techniques and models*, arXiv preprint arXiv:2004.13851 (2020).

[MG10] Keith A Markus and Wen Gu, *Bubble plots as a model-free graphical tool for continuous variables*, Advances in social science research using R, Springer, 2010, pp. 65–94.

[NKK+18] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang, *doccano: Text annotation tool for human*, 2018, Software available from https://github.com/doccano/doccano.

[Ope23] OpenAI, *Gpt-4 technical report*, 2023.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, *Pytorch: An imperative style, high-performance deep learning library*, Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.

[PP17] Avishek Pal and PKS Prakash, *Practical time series analysis: master time series data processing, visualization, and modeling using python*, Packt Publishing Ltd, 2017.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, *Glove: Global vectors for word representation*, Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[RM19] D. Rao and B. McMahan, *Natural language processing with pytorch: Build intelligent language applications using deep learning*, O'Reilly Media, 2019.

[RS94] Hans Riedwyl and M Schüpbach, *Parquet diagram to plot contingency tables*, Softstat **93** (1994), 293–299.

[seu] seungguini, *Youtube comments for covid-19 related videos [dataset]*.

[Spe99] William M Spears, *An overview of multidimensional visualization techniques*, Evolutionary Computation Visualization Workshop, 1999.

[tfd] *Module: tf.keras | TensorFlow Core v2.7.0*.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, Advances in neural information processing systems **30** (2017).

[WAM01] Marc Weber, Marc Alexa, and Wolfgang Müller, *Visualizing time-series on spirals.*, Infovis, vol. 1, 2001, pp. 7–14.

[Was21] Michael L. Waskom, *seaborn: statistical data visualization*, Journal of Open Source Software **6** (2021), no. 60, 3021.