

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Obsahová analýza webových stránek a hlasová syntéza

Diplomová práce

Bc. Radek Hiess

Vedoucí práce: Ing. Jiří Jelínek, CSc.

České Budějovice 2020

ZADÁVACÍ PROTOKOL MAGISTERSKÉ PRÁCE

Student: Bc. Radek Hiess

Obor – zaměření studia:

1802T001 / Aplikovaná informatika - specializace Informační systémy a technologie

Katedra/ústav PŘF JU, kde bude práce vypracována a obhájena:

Ústav aplikované informatiky

Školitel: Ing. Jiří Jelínek, CSc.

(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PŘF:

(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:

(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma magisterské práce: Obsahová analýza webových stránek a hlasová syntéza

Cíle práce:

- Rešerše současného stavu na poli SW pro osoby se zrakovými problémy s důrazem na zvukový přepis popisu a obsahu webové stránky.
- Popis algoritmů pro analýzu textu a obrazu s cílem zjištění jejich sémantiky. Popis metodik použitých v konkrétních službách typu Microsoft Cognitive Services, Google Cloud Platform, IBM Watson, atd.
- Návrh metodiky pro zpracování a integraci sémantické informace z dostupných služeb.
- Implementace vytvořené metodiky v pilotní aplikaci určené pro užití osobami se specifickými potřebami.
- Realizace pilotní testování aplikace cílovou skupinou (dotazník).
- Shrnutí výsledků, jejich analýza a závěry.

Základní doporučená literatura:

- BAHARUDIN, Baharum, Lam Hong LEE a Khairullah KHAN. A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology. 2010, 1(1), 69. DOI: 10.4304/jait.1.1.4-20. ISSN 1798-2340.
- SONKA, Milan., Vaclav. HLAVAC a Roger BOYLE. Image processing, analysis, and machine vision. Fourth edition. Stamford, CT, USA: Cengage Learning, 2015. ISBN 978-1-133-59360-7.
- HAYKIN, Simon S. Neural networks and learning machines. 3rd ed. New York: Prentice Hall, 2009. ISBN 978-0-13-147139-9.

Bibliografické údaje

Hiess, R., 2020: Obsahová analýza webových stránek a hlasová syntéza.

[Websites content analysis and speech synthesis. Mgr. Thesis, in Czech.] – 83 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Diplomová práce řeší návrh metodiky pro zpracování obsahu webové stránky a řečovou syntézu získané informace. Zaměřuje se na analýzu klíčových slov v textu pomocí služeb Microsoft Cognitive Services, IBM Watson a Amazon Comprehend, sumarizaci textu s využitím modelu neuronové sítě BERT a klasifikaci obrazu službou Microsoft Computer Vision. Vytvořená metodika je implementována v pilotní aplikaci. Prostřednictvím této aplikace je následně otestována. Na závěr je metodika vyhodnocena na základě komparace s jinými službami pomocí dotazníkového šetření a metrikami ROUGE.

Annotation

The diploma thesis deals with creation of the methodology for processing the content of a website and the speech synthesis of the received information. The thesis focuses on the analysis of keywords in the text with help of Microsoft Cognitive Services, IBM Watson and Amazon Comprehend services. It also focuses on the text summarization by using the BERT neural network model and image classification by Microsoft Computer Vision. The created methodology is implemented in a pilot application. With help of this application the methodology is tested. Finally, based on the comparison with other services, using survey questionnaire and ROUGE metric, the methodology is evaluated.

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 20. 5. 2020

Poděkování

Děkuji panu Ing. Jiřímu Jelínkovi, CSc. za cenné rady, věcné připomínky a konzultace. Rád bych také poděkoval za spolupráci všem, kteří se podíleli na testování metodiky a sdílení dotazníku. Velký dík patří především mé rodině a mé přítelkyni, jež mě po celou dobu studia podporovali.

Obsah

Úvod	1
Cíle práce.....	2
1. Rešerše software pro osoby se specifickými potřebami	3
1.1. Osoby se specifickými potřebami.....	3
1.2. Odečítač obrazovky	5
1.3. Porovnání odečítačů obrazovky.....	8
1.4. Východiska práce	9
2. Služby pro analýzu textu a obrazu.....	11
2.1. Dolování z textu	11
2.2. Analýza obrazu	13
2.3. Microsoft Cognitive Services	14
2.4. IBM Watson.....	16
2.5. Google Cloud Platform.....	17
2.6. Amazon Web Services	18
3. Návrh vlastní metodiky	20
3.1. Popis řešení.....	20
3.2. Popis zvolených služeb pro analýzu textu.....	21
3.3. Popis zvoleného algoritmu pro sumarizaci textu.....	24
3.4. Popis zvolené služby pro analýzu obrazu.....	25
3.5. Popis zvolené služby pro syntézu řeči.....	26
3.6. Popis vlastní metodiky	27
4. Implementace metodiky v pilotní aplikaci	34
4.1. Platforma pro vývoj	34
4.2. Požadavky na aplikaci	36
4.3. Aplikační logika	37
5. Testování.....	44
5.1. Funkční testování aplikace	44
5.2. Uživatelské testování.....	45
6. Vyhodnocení.....	47
6.1. Výsledky uživatelského testování	47
6.2. Hodnocení sumarizace.....	48
6.3. Porovnání popisu obrázku	51
6.4. Shrnutí	51
6.5. Potenciální rozvoj metodiky.....	51
Závěr.....	53
Seznam použité literatury	54
Seznam obrázků.....	59
Seznam tabulek.....	60
Seznam ukázek zdrojového kódu	61
Seznam příloh.....	62
Příloha 1: Rešerše odečítačů obrazovky	63
Příloha 2: Srovnání služeb.....	64
Příloha 3: Testovací scénáře	67
Příloha 4: Dotazník.....	79
Příloha 5: Vyhodnocení dotazníkového šetření	80
Příloha 6: Zdrojové kódy aplikace.....	83

Úvod

Průzkum z roku 2017 odhaduje, že v celosvětové populaci žije 217 milionů osob se zrakovým postižením, z toho 36 milionů nevidomých. [1] Život s těžkým zrakovým postižením bývá velmi komplikovaný a omezený, avšak díky moderním asistivním technologiím jednodušší než kdy dříve. Nabídka na trhu s asistivními technologiemi a speciálním softwarem pro osoby se specifickými potřebami je bohatá. Příkladem budiž odečítač obrazovky. To je program, který pomáhá zrakově postiženým uživatelům při práci s počítačem. Prostřednictvím odečítače obrazovky mohou ovládat počítačové programy, psát e-maily nebo číst webové stránky.

Prohlížení webových stránek je činnost, kterou dělají zrakově postižení uživatelé internetu každý den. Průchod webovou stránkou pomocí odečítače obrazovky je však zdoluhavý. Aby uživatel prozkoumal strukturu stránky, musí projít stránku od shora dolů. Určité nedostatky mají odečítače obrazovky také v interpretaci grafických prvků. Při popisu obrázku se totiž spoléhají jen na alternativní popis vyplněný autorem, nehledě na to, co je na obrázku skutečně znázorněno.

Zmíněná problematická místa, s nimiž se zrakově znevýhodnění uživatelé při prohlížení webových stránek potýkají, lze však v současné době řešit pomocí prvků umělé inteligence. Díky zvýšené poptávce po aplikacích využívajících umělou inteligenci dnes technologické korporace nabízejí služby, díky kterým mohou používat umělé neuronové sítě i běžní vývojáři. Právě využitím těchto služeb pro zpracování přirozeného jazyka a klasifikaci obrazu se zabývá tato diplomová práce.

Práce je rozdělena do šesti kapitol. První kapitola definuje cílovou skupinu, porovnává funkce odečítačů obrazovky a formuluje východiska práce. Druhá kapitola vysvětluje terminologii, jež s probíraným tématem souvisí, a shrnuje nabídku služeb v oblastech analýzy textu, analýzy obrazu a zpracování řeči. Třetí kapitola, jakožto klíčová pasáž diplomové práce, popisuje zvolené služby, dále algoritmus použitý pro sumarizaci textu a podrobně řeší návrh vlastní metodiky pro zpracování obsahu webové stránky. Čtvrtá kapitola představuje vývojovou platformu Node-RED a především se zabývá implementací metodiky v pilotní aplikaci. Pátá kapitola se zabývá funkčním testováním aplikace a pak také uživatelským testováním, jehož cílem je validace navržené metodiky. Poslední kapitola uzavírá diplomovou práci shrnutím výsledků testování, shrnutím výsledků dotazníkového šetření a nakonec zhodnocením kvality metodiky na základě komparace výstupů s využitím metrik ROUGE.

Cíle práce

- Rešerše současného stavu na poli SW pro osoby se specifickými potřebami s důrazem na zvukový přepis popisu a obsahu stránky.
- Popis algoritmů pro analýzu textu a obrazu s cílem zjištění jejich sémantiky. Popis metodik použitých v konkrétních službách typu Microsoft Cognitive Services, Google Cloud Platform, IBM Watson atd.
- Návrh metodiky pro zpracování a integraci sémantické informace z dostupných služeb.
- Implementace metodiky v pilotní aplikaci určené pro užití osobami se specifickými potřebami.
- Realizace testování pilotní aplikace cílovou skupinou (dotazník).
- Shrnutí výsledků, jejich analýza a závěry.

1. Rešerše software pro osoby se specifickými potřebami

První kapitola definuje cílovou skupinu a představuje asistivní technologie, jež zrakově postižení uživatelé využívají. Podrobně rozebírá software zvaný odečítač obrazovky, jeho funkce a práci s ním. Popisuje aktuální situaci na trhu s odečítači obrazovky a porovnává jejich významné funkce. Poslední část kapitoly poukazuje na problematická místa, která nevidomým uživatelům znepříjemňují čtení webu, a formuluje východiska práce.

1.1. Osoby se specifickými potřebami

Za osoby se specifickými potřebami lze obvykle považovat osoby s určitým zdravotním znevýhodněním. Zdravotní postižení se dělí do několika kategorií. Osoby se specifickými potřebami lze podle toho rozdělit do základních skupin:

- osoby s tělesným postižením,
- osoby s mentálním postižením,
- osoby s řečovým postižením,
- osoby se sluchovým postižením,
- osoby se zrakovým postižením,
- osoby s kombinovaným postižením. [2]

Každý druh postižení se vyznačuje jinými specifiky. V závislosti na druhu postižení má každá skupina jiné potřeby a požadavky. Práce se zaměřuje na osoby se zrakovým postižením, následující kapitola proto popisuje tuto cílovou skupinu.

1.1.1. Osoby se zrakovým postižením

Osoby se zrakovým postižením mají různé typy a různé stupně snížených zrakových schopností. „*Úžeji se tímto termínem rozumí osoby, u nichž poškození zraku nějak ovlivňuje činnosti v běžném životě a u nichž běžná optická korekce nepostačuje.*“ [3] Do této definice nelze zahrnout osoby, které užívají dioptrické brýle, s jejichž pomocí mohou bez omezení fungovat v každodenním životě.

Z definice vyplývá, že se jedná o osoby těžce zrakově postižené, tedy osoby slabozraké či nevidomé. Přesněji lze zrakové postižení klasifikovat do pěti kategorií:

1. střední slabozrakost,
2. silná slabozrakost,
3. těžce slabý zrak,
4. praktická nevidomost,
5. úplná nevidomost. [2]

Generalizovat potřeby všech osob se zrakovým postižením je téměř nemožné, jelikož spektrum vad je opravdu široké. Předně je nutno podotknout, že každý jedinec je specifický a má jiné potřeby. Odlišné požadavky má člověk středně slabozraký, který vidí jen předměty v těsné blízkosti, a to velmi rozostřeně; a odlišné požadavky má člověk úplně nevidomý, který dokáže rozeznat pouze světlo a tmu. Je proto nezbytné zaměřit se na okruh osob, pro něž je vyvíjená aplikace určena.

Cílovou skupinou jsou lidé, kteří používají osobní počítač, získávají informace na internetu, navštěvují webové stránky, ale při práci na počítači již nemohou využívat svého zraku. Převážně to jsou uživatelé nevidomí a uživatelé se zbytky zraku (dále jen *nevidomí uživatelé*).

1.1.2. Asistivní technologie

Přese všechna úskalí mají zrakově znevýhodněné osoby podobné příležitosti, jako „zdraví“ lidé. Díky moderním kompenzačním pomůckám mohou vykonávat stejné aktivity. Moderní technologie v čele s umělou inteligencí jim umožňují chodit do zaměstnání, případně pracovat z domova, chodit k lékaři či na nákup, podnikat, účastnit se konferencí, aktivně zasahovat do veřejného dění a podobně. Mohou běžně využívat počítač nebo mobilní telefon. Mohou pracovat s aplikacemi a počítačovými programy, psát e-maily, spojovat se na sociálních sítích s ostatními lidmi, komunikovat s úřady nebo nakupovat v e-shopech. Ke všem těmto činnostem však zpravidla potřebují nějaké asistivní technologie.

V kontextu práce na PC představují asistivní technologie speciální software a hardware, který slouží jako vstupní zařízení nebo výstupní zařízení (předává obsah ve zvukové či jiné podobě). Zde je krátký popis technologií, které nejvíce usnadňují práci s počítačem.

Odečítač obrazovky

Odečítač obrazovky je speciální program, který zachytává informace zobrazené na obrazovce a interpretuje je uživateli v alternativní podobě. Alternativní podobou je myšlen hlasový nebo hmatový výstup. Podrobný popis odečítače obrazovky a práce s ním je uvedený v kapitole 1.2. Odečítač obrazovky.

OCR program

Tištěné dokumenty, letáky, knihy, digitální grafiku obsahující text a jiné informační materiály si musí nevidomé osoby převádět do digitální podoby, aby s nimi mohly pracovat. Převod naskenovaných dokumentů do přístupné podoby zajišťuje metoda OCR – Optical Character Recognition, česky optické rozpoznávání znaků. Digitalizovaný text už může uživatel zpracovat řečovou syntézou nebo v Braillském řádku.

Braillský řádek

Uživatelé, kteří ovládají Braillovo písmo, mohou ke svému počítači připojit Braillský řádek. Toto vstupně-výstupní zařízení zobrazuje textové informace v podobě Braillova bodového písma. Navíc má také několik tlačítek, kterými uživatel vykonává jednoduché textové operace.

1.2. Odečítač obrazovky

Odečítač obrazovky (screen reader) je software, který identifikuje dění na obrazovce a předává uživateli informace, v jaké části programu či systému se právě nachází. Nevidomý uživatel, který pracuje na PC, jej ovládá výhradně běžnou klávesnicí s využitím klávesových zkratk. Má-li k dispozici Braillský řádek, může PC ovládat tlačítka na něm. Výstup odečítače je uživateli interpretován hlasem pomocí technologie text-to-speech a/nebo pomocí Braillského řádku.

1.2.1. Syntéza řeči

Text-to-speech (TTS), česky též syntéza řeči, je technika, která převádí text do uměle vytvořeného řečového signálu, tzn. do mluvené řeči. Kvalita syntetizátoru řeči se určuje podle dvou základních vlastností: přirozenosti (podobnosti s lidskou řečí) a srozumitelnosti. Syntetizátory se obvykle snaží tyto dvě vlastnosti maximalizovat a přiblížit se tak lidskému hlasu. V odečítačích obrazovky se dnes nejčastěji používají syntetizátory řeči založené na

formantové syntéze nebo hlubokém neuronovém učení.

Formantová syntéza

Formantová syntéza vytváří řečová data podle pravidel definujících akustické vztahy různých zvuků. Pravidla a parametry formantů¹ modelují výsledný hlasový trakt. Parametry syntézy (frekvence, fonace, hladina šumu...) se průběžně odhadují a mění. Díky velkému množství parametrů lze různě modelovat i citové zabarvení řeči. Takto generovaná řeč je však stále relativně nepřirozená; dala by se přirovnat k mluvení robota. [5] Formantová syntéza však má své nesporné výhody, jako jsou vysoká variabilita či nízké nároky na výkon. Používá se např. v odečítači NVDA.

Syntéza pomocí hlubokého neuronového učení

Hluboká umělá neuronová síť se skládá z několika skrytých vrstev. Síť zpracovává vstupy a počítá pravděpodobnost výskytu (podoby) výstupů. [6] Pro účely syntézy je neuronová síť vytrénovaná na záznamech reálného lidského hlasu. Naučená neuronová síť pak může brát v úvahu jednak vztah mezi pravopisnou reprezentací slov a jejich akustickou podobou, jednak okolní kontext sdělení. Tím je zajištěno vhodné užití přízvuku, intonace či citového zabarvení. [7] Syntéza řeči s užitím hluboké neuronové sítě je přirozenější a srozumitelnější než Formantová syntéza. Třeba výstup konvoluční neuronové sítě WaveNet se svou věrohodností blíží lidské řeči. [8] Hlubokou neuronovou síť dnes využívá např. odečítač Jaws.

1.2.2. Práce s odečítačem obrazovky

Odečítač obrazovky zpřístupňuje nejen samotný operační systém, ale i veškeré aplikace, které jsou v PC nainstalované. Nevidomí uživatelé totiž při práci s PC používají běžně dostupný software bez speciálních úprav. Přístupnost všech aplikací zajišťuje právě odečítač obrazovky. Díky tomu mohou nevidomí uživatelé běžně pracovat s internetovým prohlížečem, e-mailovým klientem, textovým editorem a jinými programy.

Odečítač zachytává tzv. „focus“. Focus určuje, který prvek v grafickém uživatelském prostředí je aktivní. Je-li focus např. na tlačítku, lze vybrané tlačítko potvrdit; je-li např. na textovém poli, lze do vybraného textového pole psát. Podle pozice focusu vyčítá odečítač informace z obrazovky.

¹ Formant – tón tvořící akustický základ hlásky

Po zapnutí PC se automaticky spustí odečítač obrazovky, je-li software takto nastaven, a začne komunikovat s uživatelem. „*Při spuštění nějaké aplikace získá focus spuštěné okno a zároveň první ovládací prvek v něm. Odečítač obrazovky tuto skutečnost komentuje tím, že přečte titulkový pruh aplikace a pokračuje oznámením aktuálního ovládacího prvku.*“ [9] V otevřeném okně aplikace se uživatel pohybuje pomocí šipek na klávesnici a dalších klávesových zkratk mezi ovládacími prvky. Tím se předává focus jednotlivým prvkům. Odečítač pak nahlas předčítá popisky vybraných ovládacích prvků (tlačítka, rozbalovací nabídky, dialogová okna...), stisky kláves a další změny na obrazovce. Na základě personifikovaného nastavení může odečítač oznamovat změnu formátování textu, začátek a konec tabulky, umístění grafického prvku v textu apod. „*Je-li okno aplikace zavřeno, je focus předán naposledy použité aplikaci, a to ovládacímu prvku, jenž byl v aplikaci naposledy aktivní.*“ [9] Takto uživatel pracuje s operačním systémem a nainstalovanými aplikacemi.

Práce s webovým prohlížečem je specifická. Pro práci s webovou stránkou nabízí odečítače obrazovky rozšířené možnosti a různé přístupy, které ulehčují její prohlížení. Základním režimem je plynulé čtení webové stránky od shora dolů podle toho, jak se elementy nacházejí v DOM² stránky. To je však velice zdlouhavé. Pokud chce uživatel vyhledat konkrétní informaci (např. telefonní číslo), musí tímto způsobem projít a nechat si přečíst celou stránku, než se k požadované informaci dostane. Z tohoto důvodu umí odečítače přeskakovat mezi jednotlivými elementy, a urychlit tak průchod stránkou.

Důležitou funkcí pro efektivní čtení webové stránky je rychlá navigace mezi jejími elementy. Odečítače obrazovky umí pomocí klávesových zkratk skákat po:

- nadpisech,
- odkazech,
- formulářových prvcích,
- oblastech stránky,
- seznamech,
- obrázcích,
- a dalších prvcích.

² DOM – Document Object Model – reprezentace HTML dokumentu v podobě stromové struktury, kterou si prohlížeč vytvoří po zpracování webové stránky.

Uživatel tak má možnost číst stránku rychlým pohybem po nadpisech či odkazech a zaměřit se jen na tu část, která ho zajímá. Nalezení konkrétní požadované informace je tak mnohem rychlejší a pohodlnější.

Dalším režimem je „focus mód“. Na rozdíl od režimu čtení focus mód umožňuje manipulovat s prvky na stránce. Stisky kláves jsou posílány přímo do aplikace. Po aktivaci elementu může uživatel např. vkládat text do textových polí, vybírat položky v rozbalovacím seznamu, potvrzovat tlačítka apod.

1.3. Porovnání odečítačů obrazovky

Odečítačů obrazovky je celá řada. Většina odečítačů je komerční, lze však najít i odečítače zdarma. Základní funkce odečítačů se neliší a i základní ovládání se výrobci snaží udržet jednotné. Mezi standardní prvky odečítačů obrazovky patří např. režim čtení, focus mód, nastavení hlasu, navigace mezi prvky webové stránky, podpora WAI-ARIA.

Ani ve zpracování obrázků se odečítače příliš neliší. Narazí-li odečítače při průchodu webovou stránkou na grafický prvek (element ``), přečtou uživateli text v atributu `alt`. Pokud má `alt` prázdnou hodnotu, odečítač pouze oznámí existenci grafiky.

Odečítače obrazovky samozřejmě také nabízejí mnoho funkcí, které uživatelům pomáhají nejen při práci s webovou stránkou. Tabulka v Příloha 1: Rešerše odečítačů obrazovky shrnuje funkce vybraných odečítačů obrazovky, jež se v současné době na trhu vyskytují. Za zmínku stojí zvláštní funkce v odečítačích JAWS a VoiceOver, které jsou relevantní k tématu této práce.

1.3.1. Odečítač obrazovky JAWS

JAWS obsahuje funkci pojmenovanou Text Analyzer. Tato funkce upozorní uživatele na nekonzistence při psaní textu. Jedná se o chyby, které mohou být při čtení řečovou syntézou přehlédnuté – např. neuzavřené závorky, neúmyslné změny formátování, zdvojená interpunkce aj.

Druhou funkcí je Skim Reading. Ta umožňuje nechat si přečíst jen první řádek každého odstavce. Uživatel tak nemusí číst celé odstavce, což významně zrychluje čtení dlouhých dokumentů. S touto funkcí souvisí Skim Reading Summary.

Skim Reading Summary je shrnutí dokumentu obsahující text, který by byl přečten funkcí Skim Reading. Funkce vezme první řádky jednotlivých odstavců, pouhým zřetěžením z nich vytvoří jednoduché shrnutí a toto shrnutí pak přečte uživateli. [10] Nevidomý uživatel díky tomu nemusí procházet celý dokument, ale přečte si pouze toto shrnutí.

1.3.2. Odečítač obrazovky VoiceOver

VoiceOver obsahuje funkci poskytující stručný přehled prvků na stránce. Nevidomý uživatel si může nechat přečíst, kolik je na webové stránce nadpisů, odkazů, tabulek, oblastí, regionů a dalších prvků. Tyto statistické údaje uživateli pomohou posoudit rozsah a strukturu webové stránky. Sám se pak může rozhodnout, jak projít stránku nejrychleji.

1.4. Východiska práce

Záměrem řešerše software pro osoby se zrakovým postižením bylo nalezení problematických míst, se kterými si tyto programy při prohlížení webových stránek neumí poradit. Z řešerše vyplynula dvě negativa, jež nevidomým uživatelům znepříjemňují čtení webu.

Zaprvé: zdlouhavý průchod webovou stránkou. Po vstupu na stránku má nevidomý uživatel dvě možnosti, jak zjistit její strukturu.

1. Projít stránku v běžném režimu čtení a přečíst si vše od shora dolů.
2. Projít stránku pohybem po nadpisech nebo oblastech.

První možnost je neefektivní a trvá velmi dlouho. Druhá možnost je možná v případě, kdy jsou nadpisy stránky správně strukturované nebo jsou oblasti definované. Bohužel, ne vždy autoři stránek nadpisy H1-H6 a oblasti používají. Pokud jsou nadpisy tvořené jen naformátovaným textem uvnitř tagu <p>, zbývá nevidomému uživateli jen první možnost – projít celou stránku.

Zadruhé: problém odečítačů obrazovky s interpretací grafických prvků. Odečítače se spoléhají jen na alternativní popisek vyplněný autorem. Pokud však autor stránek atribut alt nevyplní, nevidomý uživatel se o obrázku nic nedozví. Žádný z odečítačů dosud neumí rozpoznat obsah obrázku. Nemohou proto kompenzovat špatný či chybějící alternativní popisek. Technologie pro rozpoznání obrazu však již existují. Stačí je jen využít.

Nejnovější průzkum z roku 2017 odhaduje, že ve světové populaci žije celkem 217 milionů osob se zrakovým postižením, z toho 36 milionů nevidomých. [1] V České republice žije podle Českého statistického úřadu 102 tisíc osob s postižením zraku, z toho 18 tisíc prakticky nevidomých. [4] V porovnání s celkovým počtem všech uživatelů internetu se jedná jen o desetiny procent, avšak i tito uživatelé si zaslouží pohodlné a rychlé prohlížení webu.

Život s těžkým zrakovým postižením bývá velmi komplikovaný a omezený, avšak díky moderním technologiím jednodušší, než kdy dříve. Mezi zrakově postiženými lidmi lze nalézt nadšence, kteří jsou otevření všem novým digitálním pomůckám. Rádi je zkoušejí, testují a navzájem si doporučují ty pomůcky, které jim opravdu pomáhají. Tato diplomová práce si neklade za cíl být revoluční. Měla by být jen dalším střípkem do mozaiky asistivních technologií, jež pomáhají v každodenním životě lidem, kteří to opravdu potřebují.

2. Služby pro analýzu textu a obrazu

Převážná část aktivit, které uživatelé na internetu dělají, generuje nestrukturované texty a obrázky. Příkladem jsou e-maily, diskuze, různé články, zprávy, příspěvky na sociálních sítích aj. S přibývajícím množstvím textových a obrazových dat narůstá také potřeba tato data rychle a efektivně číst, zpracovávat a třídit. Z tohoto důvodu se do popředí zájmu dostávají algoritmy pro automatickou analýzu textu a obrazu. Učení a nasazení algoritmů strojového učení pro podobné účely byl v minulosti složitý proces, který vyžadoval tým odborníků a výkonné hardwarové prostředky. Avšak, vzhledem k vysoké poptávce po aplikacích využívajících umělou inteligenci, dnes technologické korporace nabízejí služby, díky kterým mohou používat umělé neuronové sítě i běžní vývojáři. Právě nabídka služeb v oblastech analýzy textu, analýzy obrazu a zpracování řeči je shrnuta v následujících podkapitolách. Podrobná tabulka obsahující srovnání služeb je k nahlédnutí v Příloha 2: Srovnání služeb. Před samotným představením služeb je však potřeba nejprve vysvětlit terminologii a teorii, jež s problematikou zpracování přirozeného jazyka a analýzy obrazu souvisí.

2.1. Dolování z textu

Dolování z textu (text mining) nebo též analýza textu (text analysis) je proces zpracování přirozeného jazyka, který z běžného nestrukturovaného textu extrahuje požadované informace ve strukturované podobě. [11] Zjednodušeně řečeno, dolování dokáže z předloženého textu vytáhnout samostatné elementy nebo odvodit význam textu.

Tento proces zahrnuje vše od extrakce informací, jmen, kontaktních údajů či událostí přes shlukování a klasifikaci textu až po přiřazování entit a relací. Z přirozeného textu tak lze odvodit, kdo/co je předmětem textu nebo kdy, kde, jak a proč se v textu něco děje. Typicky se k tomuto rozpoznání používají lingvistické koncepty, jako je rozdělení slovních druhů (podstatné jméno, sloveso, přídavné jméno) či gramatická struktura vět (podmět, přísudek, vložené věty, předložkové fráze, vyjádření syntaktického vztahu apod.). [12] Dolování textu se skládá z následujících dílčích kroků, které jsou pro získání informací nezbytné:

- předzpracování textových dat,
- analýza dat,
- zpracování výsledků.

Obecně lze při dolování z textu uplatnit několik metod, které slouží k analyzování, porozumění a vytažení informace; např.:

- shlukování textů,
- analýza sentimentu,
- analýza tématu,
- rozpoznání entit,
- klasifikace textu,
- extrakce klíčových slov,
- sumarizace textu.

Při návrhu metodiky pro obsahovou analýzu webové stránky jsou uplatněny především poslední tři metody. V kontextu této práce je proto lze považovat za nejdůležitější.

2.1.1. Klasifikace textu

Klasifikace textu je proces, při kterém se nestrukturovanému textu přiřazují předdefinované třídy nebo kategorie. Kategorie jsou zpravidla členěny podle témat, názvů apod. Díky všestrannosti, kterou nabízí, se považuje za nejužitečnější metodu zpracování přirozeného jazyka. Používá se mimo jiné pro předzpracování dat při dalších metodách, jako je analýza sentimentu, rozpoznání tématu či detekce jazyka. [13] Mezi algoritmy pro klasifikaci textu s využitím hluboké neuronové sítě lze zahrnout: Word2vec, Long Short-Term Memory nebo grafové konvoluční sítě.

2.1.2. Extrakce klíčových slov

Extrakce klíčových slov je metoda, která v nestrukturovaném textu identifikuje nejrelevantnější slova a výrazy tak, aby extrahovaná množina co nejlépe vystihovala hlavní téma textu. [14] Detekce klíčových slov může být provedena dvěma způsoby. Buď jsou klíčová slova vybraná z předem daného slovníku, nebo se přímo extrahují slova zmíněná ve vstupním textu. Běžně používané algoritmy jsou term frequency–inverse document frequency, Support-vector machines či konvoluční neuronové sítě.

2.1.3. Sumarizace textu

Sumarizace textu je proces, při kterém se dlouhý nestrukturovaný text zestruční do kratší podoby. Výsledný souhrn má formu nestrukturovaného textu a obsahuje nejdůležitější informace z původního textu. [15] Význam textu tak zůstává zachován. Existují dva přístupy, jak souhrn textu vytvořit: extraktivní a abstraktní.

V případě extraktivní sumarizace se výstupní souhrn poskládá ze zkopírovaných frází a vět, které se v původním textu identifikují jako nejvýznamnější. Může se jednat o první věty odstavce, tučně zvýrazněné fráze či věty obsahující vysokou frekvenci klíčových slov. Abstraktně sumarizovaný souhrn je automaticky vygenerovaná interpretace původního textu. Věty nejsou vyjmuté z původního textu, ale jsou uměle vytvořené. Díky tomu má abstraktně sumarizovaný souhrn lepší kohezi textu a jednotlivé věty na sebe lépe navazují. [16] Moderní přístupy pro sumarizaci textu jsou postavené na rekurentních neuronových sítích. Jedná se např. o algoritmy Sequence-to-Sequence, Long Short-Term Memory nebo Gated Recurrent Unit.

2.2. Analýza obrazu

Úkolem analýzy obrazu je extrakce informací z obrazu. Při analýze obrazu se počítač snaží najít vztah mezi vstupním obrazem a předem definovanými modely reálného světa. Během tohoto procesu se zredukuje informace obsažené v obraze na relevantní množství potřebné pro danou doménu znalostí a tyto informace se pak dále zpracovávají. [17] Tímto způsobem je možné obraz klasifikovat do konkrétní kategorie, detekovat v obraze objekty zájmu, případně tyto objekty segmentovat a klasifikovat je samostatně.

2.2.1. Zpracování obrazu

Při zpracování obrazu se nad obrazem provádí operace, jejímž účelem je vylepšení vstupního obrazu nebo extrakce požadovaných informací z obrazu. V případě digitálního obrazu se jedná o zpracování digitálního signálu vybranými algoritmy, kdy vstupem je obraz a výstupem může být obraz nebo vlastnosti spojené s daným obrazem. [18]

Tři základní fáze, kterými musí při zpracování obrazu projít všechny typy dat, jsou:

- předzpracování,
- vylepšení obrazu a zobrazení,
- extrakce informací.

Během zpracování obrazu je vstupní obraz zpracováván několika algoritmy. Může se jednat jak o jednoduché transformace, tak o složitější specializované operace, např.:

- změna měřítka,
- odstranění šumu,
- detekce hran,
- segmentace obrazu,
- rozpoznání vzorů,
- klasifikace obrazu.

2.2.2. Klasifikace obrazu

Klasifikace obrazu je proces, spadající do specializace strojového vidění, který rozpozná vizuální obsah obrazu a přiřadí obrazu relevantní kategorie. Výstupem procesu je množina kategorií. [19] Základní klasifikátor umí poznat, zda je na obrázku člověk, pes nebo číslice, ale už nedokáže rozlišit jednotlivé segmenty (části) obrázku. Pokročilé algoritmy založené na neuronových sítích však už dnes umí rozpoznat popředí a pozadí; segmentovat rozdílné objekty a pojmenovat je.

2.3. Microsoft Cognitive Services

Microsoft Cognitive Services je služba poskytující funkce strojového učení k řešení obecných úkolů v oblastech zpracování přirozeného jazyka, analýzy textu, strojového vidění, analýzy obrazu apod. Je součástí platformy Microsoft Azure, která obsahuje mnoho dalších služeb pro vývojáře. Platforma se stará o vše, co je potřebné pro práci se strojovým učením: data, algoritmus i naučený model. Díky tomu nemusí vývojáři aplikací složitě učit vlastní neuronovou síť.

Ke službě se přistupuje prostřednictvím REST API³ a sady SDK⁴. Cognitive Services lze rozdělit do pěti stěžejních segmentů:

- vidění,
- řeč,
- jazyk,
- vyhledávání na webu,
- rozhodování.

2.3.1. Analýza textu

Text Analytics API poskytuje pokročilé zpracování přirozeného jazyka a nestrukturovaného textu. Zahrnuje čtyři hlavní funkce: analýzu sentimentu, extrakci klíčových frází, detekci jazyka a rozpoznání entit. Model je předem naučený rozsáhlým jazykovým korpusem. Využívá různé kombinace technik pro analýzu, jako např. zpracování textu, pořadí slov, vzájemné asociace slov a podobně. [20]

Extrakce klíčových frází

Key Phrase Extraction API vyhodnocuje nestrukturovaný text a pro každý vstupní dokument vrací seznam klíčových frází. To může být užitečné v případě, kdy je potřeba identifikovat hlavní body vyjadřující obsah textu.

2.3.2. Analýza obrazu

Computer Vision API obsahuje pokročilé algoritmy pro zpracování obrazu. Výstup je ovlivněn vstupními parametry podle toho, jaké informace chce vývojář z obrázku extrahovat. Lze jej např. použít k detekci objektů v obraze, nalezení obličejů, identifikaci loga, detekci dominantních a sekundárních barev, klasifikaci obsahu obrázku; získání popisku, detekci textu v obraze a dalším operacím.

³ REST API definuje sadu funkcí, pomocí kterých mohou vývojáři posílat požadavky (GET/POST/...) a přijímat odpovědi přes protokol HTTP, a komunikovat tak se serverem.

⁴ SDK je sada vývojových nástrojů umožňující vytváření aplikací pro určité softwarové balíčky, frameworky a jiné platformy.

2.3.3. Zpracování řeči

API pro zpracování řeči umožňuje převod textu na přirozeně znějící řeč (text-to-speech), převod mluvené řeči na text (speech-to-text), překlad z jednoho jazyka do druhého nebo rozpoznání a ověření mluvčího. Podporované jazyky se mezi jednotlivými funkcemi liší. Funkce text-to-speech umí 48 řečí, funkce speech-to-text podporuje 40 jazyků a dialektů, překlad je možný mezi 66 jazyky a jejich dialekty. [21]

2.4. IBM Watson

IBM Watson je otevřená platforma, která umožňuje vytvářet vlastní modely umělé inteligence anebo využívat předem vytvořené aplikace. Nabízí kompletní sadu funkcí pro zpracování přirozeného jazyka, zpracování řeči, strojové vidění a porozumění. Vedle dalších služeb jsou k dispozici API pro:

- rozpoznání obrazu,
- převod textu na řeč,
- překlad jazyků,
- porozumění přirozenému jazyku.

2.4.1. Analýza textu

Pro analýzu textu lze použít čtyři služby: Natural Language Understanding, Natural Language Classifier, Tone Analyzer, Personality Insights.

Porozumění přirozenému jazyku

Natural Language Understanding API je služba pro komplexní analýzu nestrukturovaného textu. Součástí tohoto rozhraní jsou funkce pro detekci entit (lidé, místa, události aj.), kategorizaci textu, rozpoznání hlavní myšlenky, extrakci emocí, extrakci klíčových slov, analýzu sentimentu, porozumění vztahu mezi entitami, extrakci metadat (autor, název, datum publikování atd.) a detekci sémantických rolí (podmět, přísudek, předmět).

2.4.2. Analýza obrazu

Také služba Watson Visual Recognition používá algoritmy hlubokého učení. Předem naučené modely zvládají detekci objektů, obličejů, barev, nevhodného explicitního obsahu a dalších prvků v obraze. Zvláštností služby je model speciálně vytrénovaný na

rozpoznávání obrázků s jídlem. Navíc je možné vytvářet vlastní modely určené pro detekci specifického obsahu. Výstup obsahuje kromě dalších parametrů i klíčová slova, jež podávají informaci o obrázku.

2.4.3. Zpracování řeči

Nabídka IBM v oblasti zpracování řeči není široká. K dispozici jsou dvě API: převod textu na řeč (text-to-speech) a převod mluvené řeči na text (speech-to-text). Text-to-speech API podporuje 13 řečí. K tomu samozřejmě obsahuje několik syntetických hlasů. Speech-to-text API podporuje 17 jazyků a dialektů v několika zvukových formátech. [22]

2.5. Google Cloud Platform

Google Cloud Platform nabízí širokou škálu řešení pro práci s velkými daty, datovou analýzu, síťová řešení, bezpečnost, strojové učení a vývoj aplikací. Stejně jako ostatní zmíněné společnosti, také Google umožňuje hostování virtuálních strojů a strojového učení. Primární model služeb je však založen na vývoji a nasazení kontejnerových aplikací. Do Cloud Platform náleží, spolu s jinými, rovněž API pro:

- zpracování přirozeného jazyka,
- počítačové vidění,
- zpracování řeči,
- překlad jazyků.

2.5.1. Analýza textu

Natural Language API umí extrahovat informace o obecných entitách, lidech, místech a událostech. Zvládá i klasifikaci obsahu, analýzu sentimentu či analýzu syntaxe. Zajímavostí je, že k pochopení struktury a významu přirozeného textu využívá stejné modely hlubokého učení, jaké Google používá ve službách vyhledávání a Google Asistent.

2.5.2. Analýza obrazu

Cloud Vision API dává vývojářům možnost integrovat do aplikací funkce pro klasifikaci obrázků, detekci objektů, log a obličejů. Dále též funkce pro označování nevhodného obsahu nebo rozpoznávání tištěného a rukou psaného textu. I v této službě lze vytrénovat vlastní model pro specifickou množinu kategorií.

2.5.3. Zpracování řeči

Zpracování řeči je rozděleno do dvou rozhraní: Text-to-Speech pro syntézu řeči a Speech-to-Text pro překlad mluvené řeči do textu. Text-to-Speech API je postavené na již zmíněné neuronové síti WaveNet a podporuje 31 řečí. [23] Speech-to-Text API podporuje 120 jazyků a jejich dialektů.

2.6. Amazon Web Services

Amazon Web Services zahrnuje rozsáhlou nabídku služeb, jako např. vývojářské, analytické, bezpečnostní a databázové nástroje nebo výpočetní nástroje založené na neuronových sítích. Většina služeb není určena přímo koncovým uživatelům, nýbrž vývojářům, kteří je mohou využít ve svých aplikacích. Amazon Web Services jsou přístupné přes REST API. Mezi mnoha dalšími jsou k dispozici API pro:

- zpracování přirozeného jazyka,
- analýzu obrazu a videa,
- práci s řečí,
- překlad jazyků.

2.6.1. Analýza textu

Amazon Comprehend API pomůže z nestrukturovaného textu extrahovat klíčové fráze, místa, lidi, obchodní značky nebo události. Dále identifikuje jazyk textu, vyhodnotí sentiment nebo roztrídí sbírku dokumentů podle tématu. Specialitou je služba Amazon Comprehend Medical, která je zaměřena na extrahování souhrnných zdravotních informací z lékařských dokumentů.

2.6.2. Analýza obrazu

S pomocí Amazon Rekognition API lze v obraze detekovat objekty či nevhodný obsah, rozpoznávat text, scénérie a aktivity. Služba obsahuje pokročilé funkce pro analýzu obličeje (detekce a porovnání obličejů, rozpoznání emocí, pojmenování celebrit, počítání lidí apod.). Amazon poskytuje v rámci Rekognition API také analýzu videa. Kromě všech výše uvedených funkcí pro analýzu obrazu lze ve videu využít i další funkce, jako např. sledování pohybu lidí ve videu či identifikaci konkrétních aktivit.

2.6.3. Zpracování řeči

Zpracování řeči zajišťují dvě služby: Amazon Polly a Amazon Transcribe. Amazon Polly slouží jako řečová syntéza – převádí text na umělou řeč. Amazon Transcribe naopak transformuje mluvené slovo na text. První z nich podporuje 93 řečí, druhý rozumí 31 jazykům a jejich dialektům.

3. Návrh vlastní metodiky

Stěžejním cílem diplomové práce je návrh metodiky pro zpracování obsahu webové stránky. Vytvořená metodika definuje obecný postup, jak využít dostupné služby pro analýzu textu a analýzu obrazu k získání sémantických informací z webové stránky, jak takové informace použít a jak je předat uživateli přívětivou formou. Na následujících stránkách je nejprve uveden nástin eventuálních řešení problémů, se kterými se zrakově znevýhodnění uživatelé na webu potýkají. Dále popis vybraných služeb pro analýzu textu, analýzu obrazu i syntézu řeči a popis zvoleného algoritmu pro extraktivní sumarizaci textu. Klíčovou částí kapitoly je popis metodiky, který detailně rozebírá její dílčí kroky.

3.1. Popis řešení

Na úplném začátku byla provedena rešerše software pro osoby se zrakovým postižením. Tato rešerše ukazuje problematické aspekty, jejichž zlepšení může mít pozitivní vliv na prohlížení webových stránek zrakově znevýhodněnými uživateli. Konkrétní problémy jsou vysvětleny v kapitole 1.4. Východiska práce. Nyní přichází na řadu řešení.

3.1.1. Shrnutí textového obsahu

První otázkou je, jak uživateli efektivně předat obsah webové stránky, aniž by musel projít celou stránku. Řešení tohoto problému je několik. Lze kupř. extrahovat první větu z každého odstavce a z těchto vět vytvořit základní výtah. Jedná se o poměrně jednoduchý postup. V prostředí webových stránek ale takto poskládaný výtah nemusí být dostatečně vypovídající o obsahu celé stránky. Navíc, u velmi dlouhých textů je i samotný výtah dlouhý. Další možností je extraktivní sumarizace obsahu. Při té se nejprve provede obsahová analýza celého textu, identifikují se nejvýznamnější věty a pak se z nich vytvoří krátké shrnutí. Takové shrnutí vystihuje hlavní body celého textu lépe než předchozí varianta. Stále však trpí špatnou soudržností vět. Ideálním řešením je abstraktní sumarizace obsahu. Ta je založena na analýze sémantiky. Výstupem algoritmu je vygenerované, přirozeně znějící shrnutí. Nevýhodou abstraktní sumarizace jsou vysoké požadavky na výkon, protože vyžaduje hlubší porozumění textu.

3.1.2. Interpretace grafických prvků

Druhým bodem je interpretace grafických prvků, typicky obrázků, v textové, resp. zvukové podobě. Přímočarým řešením problému s obrázkem je přečtení jeho atributu alt. Ten ale může být vyplněn nesmyslně nebo nemusí být vyplněn vůbec. Spolehlivějším řešením je provedení analýzy obrazu. Na základě klasifikace obrazu se rozpoznají objekty v obraze a vygeneruje se popis toho, co je na obrázku opravdu znázorněno.

3.1.3. Zvolená řešení

Při řešení vytyčených problémů výborně posloužily již existující služby a programy. V případě interpretace grafických prvků se jako nejlepší řešení ukázala služba Microsoft Computer Vision. Analýzu klíčových frází obstarávají služby Microsoft Text Analytics, IBM Watson Natural Language Understanding a Amazon Comprehend. Syntéza řeči je zpracovaná službou Amazon Polly. Shrnutí textového obsahu webové stránky řeší Python knihovna Bert Extractive Summarizer. Výběr této knihovny je zdůvodněný v podkapitole 3.3. Popis zvoleného algoritmu pro sumarizaci textu.

Vybrané služby jsou propojeny pomocí funkcí, které zajišťují režii a práci s daty. To vše tvoří obecný algoritmus, jehož dílčí kroky řeší zmíněné problémy a obstarávají i další činnosti vedoucí k požadovanému výsledku. Podrobný popis metodiky pro zpracování sémantické informace z webové stránky je rozebrán v podkapitole 3.6. Popis vlastní metodiky. Předtím je potřeba popsat vybrané služby.

3.2. Popis zvolených služeb pro analýzu textu

Druhá kapitola Služby pro analýzu textu a obrazu ukazuje, že dostupné služby pohlížejí na analýzu textu všeobecně. Jsou naučené tak, aby dokázaly pochopit různě zaměřené texty z různých odvětví. To je velmi užitečné právě v případě dolování z textu webových stránek. Pro získání základní představy o tématu webové stránky využívá metodika extrakci klíčových slov. Aby bylo téma textu určeno co nejlépe, kombinuje metodika výstupy ze tří služeb: Microsoft Text Analytics, IBM Watson Natural Language Understanding a Amazon Comprehend. Výsledkem porovnání výstupů z více služeb je přesnější určení klíčových slov a lepší pochopení, do jakého tématu text spadá.

3.2.1. Microsoft Text Analytics

Vstupem do Microsoft Text Analytics KeyPhrases API je textový řetězec zabalený společně s nastavovacími parametry do JSON⁵ objektu. Výstupem je pole klíčových frází, opět ve formátu JSON. Na rozdíl od konkurenčních služeb, Microsoft Text Analytics KeyPhrases API nevrací skóre, s jak velkou pravděpodobností jsou klíčové fráze relevantní k analyzovanému textu. Kromě extrakce klíčových frází se Text Analytics API používá i pro detekci jazyka textu. V tomto případě je výstupem kód rozpoznávaného jazyka.

```
1. {  
2.   "documents": [  
3.     {  
4.       "keyPhrases": [  
5.         "single area",  
6.         "ancient oak trees date",  
7.         "world wars",  
8.         "single type of soil",  
9.         "unique group of oaks"  
10.      ],  
11.      "id": "1"  
12.    }  
13.  ],  
14.  "errors": []  
15. }
```

Ukázka 1: Výstup extrakce klíčových slov Microsoft Text Analytics API

3.2.2. IBM Watson Natural Language Understanding

Watson Natural Language Understanding API přijímá na vstupu textový řetězec a parametry, jež nastavují požadované vlastnosti analýzy. Na výstupu vrací JSON objekt obsahující všechny vyžádané atributy textu. Výhodou tohoto API je zevrubná analýza. V rámci jednoho požadavku umí vrátit klíčová slova (včetně relevance), sémantické role, jazyk, kategorie aj.

⁵ JSON – JavaScript Object Notation – je způsob zápisu dat určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Zápis má vždy textovou podobu.

```

1. {
2.   "usage": {
3.     "text_units": 1,
4.     "text_characters": 1536,
5.     "features": 1
6.   },
7.   "keywords": [
8.     {
9.       "text": "920-years-old",
10.      "sentiment": {
11.        "score": -0.648166,
12.        "label": "negative"
13.      },
14.      "relevance": 0.83577,
15.      "emotion": {
16.        "sadness": 0.342449,
17.        "joy": 0.260081,
18.        "fear": 0.298228,
19.        "disgust": 0.184471,
20.        "anger": 0.142352
21.      },
22.      "count": 1
23.    }
24.  ],
25.  "language": "en"
26. }

```

Ukázka 2: Výstup extrakce klíčových slov Watson NLU API

3.2.3. Amazon Comprehend

Do Amazon Comprehend API se na vstup posílá JSON objekt, který v sobě nese textový řetězec a jazyk textu. Na výstupu služba vrací seznam klíčových frází, pravděpodobnostní skóre relevance a pozice klíčových frází v textu. Výstup je ve formátu JSON.

```

1. {
2.   "KeyPhrases": [
3.     {
4.       "Score": 0.9999639987945557,
5.       "Text": "People",
6.       "BeginOffset": 0,
7.       "EndOffset": 6
8.     },
9.     {
10.      "Score": 0.9999907612800598,
11.      "Text": "a Blenheim Palace",
12.      "BeginOffset": 18,
13.      "EndOffset": 35
14.    }
15.  ]
16. }

```

Ukázka 3: Výstup extrakce klíčových slov Amazon Comprehend

3.3. Popis zvoleného algoritmu pro sumarizaci textu

Algoritmů pro sumarizaci textu je nespočet. Při výběru vhodného řešení směřovaly první úvahy k použití abstraktní sumarizace. Od této varianty však bylo záhy ustoupeno. Jak dokazují srovnání sumarizačních metod [24] a také praxe [25], současné algoritmy pro abstraktní sumarizaci dosud nejsou zcela robustní a kladou značné nároky na výkon výpočetního zařízení.

Hledání adekvátního algoritmu se tedy zaměřilo na extraktivní sumarizaci. Přijatelným řešením se zdála být Python knihovna pysummarization [26], která pro zpracování textu využívá konvoluční neuronovou síť, konkrétně model Encoder/Decoder postavený na architektuře hlubokého učení Long short-term memory se zpětnou propagací chyby. Testování knihovny však nepřineslo uspokojivé výsledky. Zpracování odstavce o délce cca 775 znaků trvalo v průměru 132 sekund. Sumarizace textu z webové stránky o délce cca 12 480 znaků zabrala v průměru 164 minuty. Kvůli vysoké latenci byla knihovna pysummarization vyhodnocena jako nevyhovující. Dobře si oproti tomu v testování vedla knihovna Bert Extractive Summarizer. Odstavec textu zpracovala v průměru za 10 sekund. Sumarizace textu z webové stránky jí trvala v průměru 29 sekund. Právě rychlost zpracování představovala hlavní důvod, proč byla tato knihovna vybrána.

3.3.1. Knihovna Bert Extractive Summarizer

Knihovna Bert Extractive Summarizer [27] je napsaná v jazyce Python. Pro zpracování textu používá model neuronové sítě naučený pomocí techniky BERT. Pro identifikaci vět, které mají být extrahovány, používá metodu shlukování K-Means.

BERT, celým názvem Bidirectional Encoder Representations from Transformers, je technika pro učení umělých neuronových sítí zaměřených na zpracování přirozeného jazyka. BERT vyvíjí vědci ze společnosti Google jako open-source projekt. Vedle architektury pro učení neuronových sítí poskytl Google komunitě také předem vytrénované modely. Architektura je založená na vícevrstevném, obousměrném Transformer enkodéru. Jedná se o model hlubokého neuronového učení bez učitele, ve kterém je každý výstupní prvek propojený s každým vstupním prvkem. Váhy mezi těmito prvky se vypočítávají dynamicky podle jednotlivých propojení. [28]

K-Means je typický algoritmus strojového učení pro shlukovou analýzu dat. Slouží k hledání podobností v neznámých datech (např. při dolování z dat). Funguje na principu tzv. centroidů, kdy se nalezené prvky zařazují do větších shluků podle toho, ke kterému centroidu jsou nejbližší.

Knihovnu Bert Extractive Summarizer lze přizpůsobit vlastním potřebám pomocí několika parametrů. Lze kupř. použít vlastní model, vybrat skrytou vrstvu pro zpracování vektorů, nastavit handler pro zpracování vět, nastavit poměr délky souhrnu vůči původnímu textu, ignorovat příliš krátké nebo příliš dlouhé věty apod.

Text se do knihovny předává jako čistý řetězec. Nejprve se odstavce textu rozdělí na věty a ty pak na tzv. tokeny (zpravidla slova). Věty ve formě tokenů se transformují modelem BERT na vektorovou reprezentaci. Vektorová reprezentace vět je dále zpracovaná K-Means algoritmem. Vektorové reprezentace vět, které jsou nejbližší centroidu, jsou vybrány do výstupního souhrnu. [29] Knihovna na výstupu vrací sumarizovaný souhrn ve formě řetězce.

3.4. Popis zvolené služby pro analýzu obrazu

Všechny čtyři služby pro analýzu obrazu poskytují širokou škálu funkcí, díky kterým umí v obrázku rozpoznat a pojmenovat rozličné objekty. Prvotním záměrem bylo použít více služeb a jejich výstupy zkombinovat. Uživatel by byl předložen výčet klíčových slov. Z takového výčtu by uživatel pochopil, že je na obrázku např. dům se stromy, lidé anebo graf. Nakonec padla volba na využití jen jedné služby – Microsoft Computer Vision. Ta dokáže, jako jediná, popsat obrázek přirozeně znějící větou.

3.4.1. Microsoft Computer Vision

Do Microsoft Computer Vision API lze poslat buď samotný obrázek v binární podobě, nebo URL obrázku uloženého na webovém serveru. V tomto případě se posílá URL obrázku extrahované z atributu src elementu . Služba si obrázek stáhne a analyzuje ho. Rozhraní vrací poměrně rozsáhlý JSON objekt, který obsahuje pravděpodobné kategorie, rozpoznané barvy, klíčová slova, titulek a další vlastnosti. Pro potřeby metodiky je nejdůležitější právě titulek. Jde o krátký popis psaný přirozeným jazykem, který s určitou pravděpodobností vystihuje vizuální obsah obrázku. Je-li na obrázku např. československý vlčák stojící na louce, služba titulek formuluje jako „a dog standing on top of a grass covered field“. Takový popis obrázku má, dle testování, pro zrakově postiženého uživatele největší

hodnotu. V tomhle ohledu vykazuje služba Computer Vision, v porovnání s ostatními, nejlepší výsledky.

```
1. {
2.   "requestId": "71bcf696-d953-4e2a-ba2d-e97cbe7c8dbc",
3.   "metadata": {
4.     "width": 729,
5.     "height": 486,
6.     "format": "Jpeg"
7.   },
8.   "tags": [{
9.     "name": "grass",
10.    "confidence": 0.9999997615814209
11.  }, {
12.    "name": "dog",
13.    "confidence": 0.9998142719268799
14.  }, {
15.    "name": "outdoor",
16.    "confidence": 0.998653769493103
17.  }],
18.   "description": {
19.     "tags": [
20.       "grass",
21.       "dog",
22.       "outdoor"
23.     ],
24.     "captions": [{
25.       "text": "a dog standing on top of a grass covered field",
26.       "confidence": 0.9323118271511521
27.     }]
28.   }
29. }
```

Ukázka 4: Zkrácený výstup Microsoft Computer Vision API

3.5. Popis zvolené služby pro syntézu řeči

Syntéza řeči se zdá být vcelku přímočarý proces – na vstupu je text a na výstupu zvuk. Opak je však pravdou. Za tvorbou syntetického řečového signálu se skrývají složité algoritmy umělých neuronových sítí, které generují poměrně kvalitní řeč. Hodnocení kvality syntézy poslechem je proto spíše subjektivní záležitostí. Výběr služby pro syntézu řeči lze nicméně postavit i na objektivních parametrech, jako je podpora jazyků, latence nebo cena. Dle těchto parametrů byla vybrána služba Amazon Polly ze sady Amazon Web Services.

3.5.1. Amazon Polly

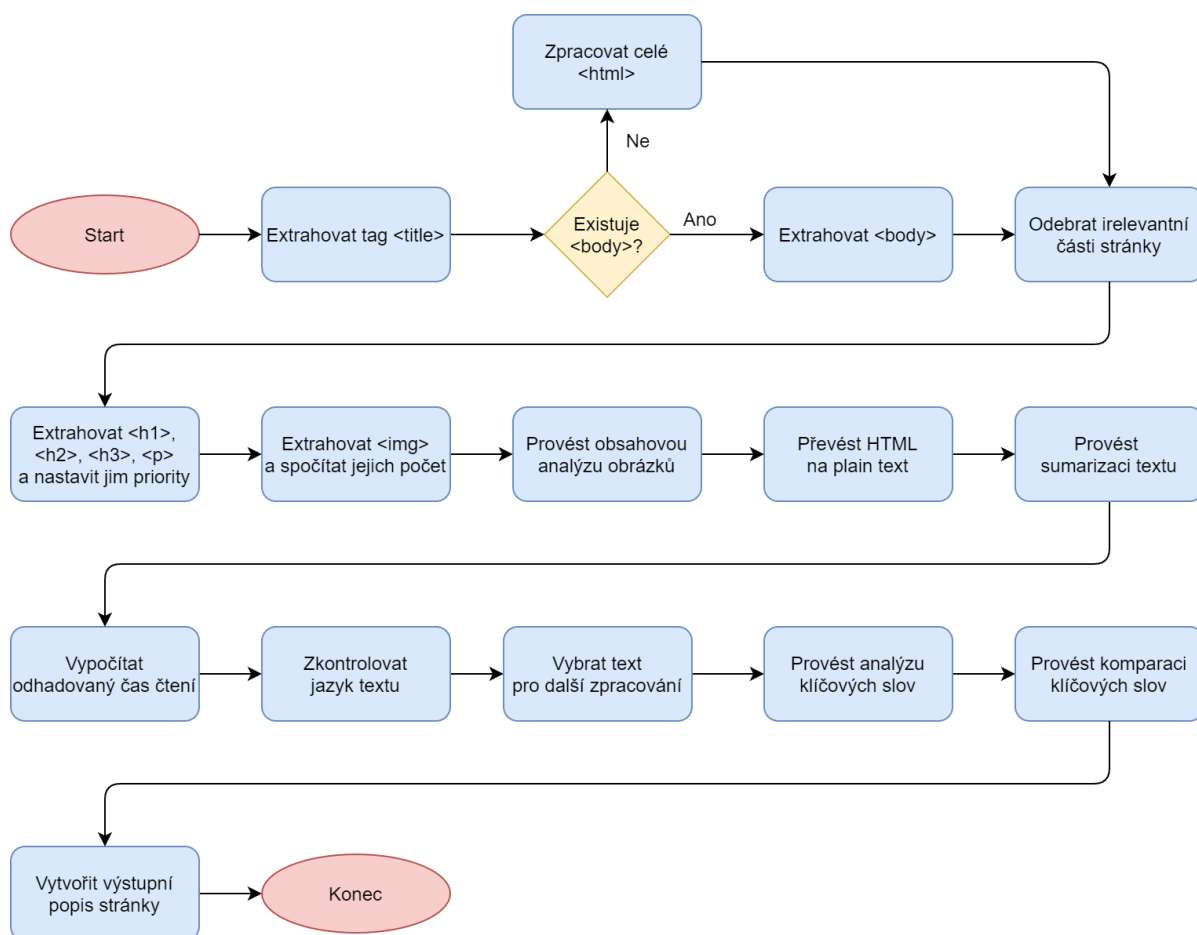
Vstupem do Amazon Polly SynthesizeSpeech API je vždy text, který je součástí JSON objektu spolu s dalšími parametry syntézy. Vstupní text může mít podobu prostého textu

nebo může být obohacený o SSML⁶ značky. Pomocí SSML lze nastavit různé aspekty řeči jako např. výslovnost, hlasitost, tón hlasu či rychlost řeči. Na výstupu služba vrací zvukový stream v binární podobě, přičemž si lze zvolit zvukový formát výstupu (mp3, Ogg, PCM). Vedle standardních hlasů si lze vybrat také mezi hlasy generovanými umělou neuronovou sítí. Možnost výběru mezi standardní syntézou řeči a syntézou pomocí umělé neuronové sítě nabízejí i ostatní služby. Amazon Polly však podporuje nejvíce řeči (93) a nabízí výhodnější cenu za počet znaků (\$4, resp. \$16 za milion znaků) než jiné služby.

3.6. Popis vlastní metodiky

Záměrem metodiky je identifikovat relevantní elementy na webové stránce, na základě vybraných elementů analyzovat obsah stránky, vytvořit popis stránky a interpretovat jej uživateli. Metodika v tomto případě definuje obecný postup jak dosáhnout požadovaného výstupu. Při návrhu metodiky se nebere v úvahu, jakým způsobem bude metodika implementovaná. Díky tomu je nezávislá na platformě či programovacím jazyku. Během implementace pak mohou být v jednotlivých krocích metodiky uplatněna různá řešení. V případě potřeby může být dokonce využita jen část metodiky, a výsledek bude i přesto dávat smysl. Popis metodiky je proto rozdělen na dvě části: Analýza textu a Klasifikace obrázků. Každou část lze implementovat samostatně. Stejně tak je ovšem možné tyto dvě části prolnout a implementovat metodiku jako celek, protože úvodní kroky jsou stejné. Celkový pohled na metodiku znázorňuje diagram na další straně. Následuje také ukázka výstupu metodiky.

⁶ SSML – Speech Synthesis Markup Language – je značkovací jazyk založený na formátu XML. Poskytuje standardizovaný zápis parametrů pro syntézu řeči.



Obrázek 1: Vizuální schéma metodiky

1. Title: Prague Info: An online travel guide about Prague.
2. Keywords: Prague Castle, Charles Bridge, famous Wenceslas Square, Powder Tower.
3. Summary: Prague is one of the most visited cities in Europe. Some areas in the new town are also very touristy in Prague. Prague Castle Besides the Old Town Square and Charles Bridge, Prague Castle is the third major monument in Prague. Shopping Center Palladium Of course, one can argue about whether one should list a modern shopping centre among the sightseeings of Prague. One thing is for sure, it is one of the most visited buildings of the city by tourists.
4. Estimated reading time: 5 minutes.
5. Images count: 7 images.
6. Images: a bridge over a body of water; an aerial view of a city; a large clock mounted to the side of a building; a bridge over a body of water with a city in the background; a castle with a clock at the top of a building; a group of people walking in front of a building; a large clock tower towering over a city.

Ukázka 5: Příklad výstupu metodiky

3.6.1. Analýza textu

Analýza textového obsahu je definovaná jako sekvence kroků, které jsou na sobě navzájem závislé.

Krok 1

Z HTML kódu stránky se extrahuje text z elementu <description> (dále jen *Description*).

Z HTML kódu stránky se extrahuje text z elementu <title> (dále jen *Title*).

Krok 2

Pokud stránka obsahuje element <body>, extrahuje se obsah elementu <body>. Pokud element <body> neexistuje nebo je prázdný, zpracovává se HTML kód celé stránky.

Krok 3

Odeberou se části stránky, které nejsou relevantní pro hlavní obsah. Zpravidla se jedná o hlavičku, navigaci, patičku aj. Z HTML kódu stránky se odstraní HTML komentáře. Dále se z HTML kódu odstraní elementy: header, nav, footer a další nadbytečné elementy. Výběr elementů, které se mají odstranit, lze udělat pomocí selektorů s využitím regulárních výrazů:

- header, `*[class='head']`, `*[class*='header']`, `*[id='head']`,
`*[id*='header']`, `*[id*='banner']`, `*[class*='banner']`
- nav, `*[role='navigation']`, `*[class='nav']`, `*[id='nav']`,
`*[class*='navigation']`, `*[id*='navigation']`, `*[class*='navbar']`,
`*[id*='navbar']`, `*[class*='menu']`, `*[id*='menu']`,
`*[class*='subnav']`, `*[id*='subnav']`, `*[class*='submenu']`,
`*[id*='submenu']`
- footer, `*[role='contentinfo']`, `*[class*='footer']`, `*[class='foot']`,
`*[id*='footer']`, `*[id='foot']`, `*[class*='copyright']`,
`*[id*='copyright']`
- aside, `*[class*='comment']`, `*[id*='comment']`, `*[class*='social-
buttons']`, `*[id*='social-buttons']`, `*[class*='breadcrumb']`,
`*[id*='breadcrumb']`, `*[class*='skip-link']`, `*[id*='skip-link']`,
`*[class*='skip']`, `*[id*='skip']`, `*[class*='screen-reader']`,
`*[id*='screen-reader']`, `*[class*='accessible']`, `*[id*='accessible']`,
`*[class*='meta']`, `*[id*='meta']`, `*[class*='reflist']`,
`*[id*='reflist']`

Po odstranění elementů zůstane HTML kód, ve kterém by měl být jen podstatný obsah.

Krok 4

Z HTML kódu zbylého po kroku 2 se odstraní HTML značky a převede se na prostý text. Vznikne textová podoba stránky (dále jen *Textová podoba stránky*). Během testování vyšlo najevo, že toto je velmi důležitý krok pro sémantickou analýzu textu. Je potřeba extrahovat slova z HTML tagů a správně je poskládat do vět.

Z HTML kódu zbylého po kroku 2 se extrahuje text z nadpisů H1, H2, H3 a text z odstavců. Jednotlivým prvkům se nastaví priorita podle tabulky. Priority nejsou přiřazeny náhodně. Hodnoty vychází z výzkumu [30].

Element	<h1>	<h2>	<h3>	<p>
Priorita	1	0,9	0,8	—

Tabulka 1: Nastavení priority HTML elementům podle důležitosti

Krok 5

Textová podoba stránky se zpracuje knihovnou Bert Extractive Summarizer. Do knihovny se posílají dva vstupní parametry:

1. *Textová podoba stránky*,
2. poměr vět mezi vstupním textem a sumarizovaným textem. V metodice je tento poměr nastavený na 7 %. Testování dokázalo, že 7% sumarizace původního textu v sobě nese dostatečnou informaci, přičemž délka sumarizovaného textu je ještě snesitelná.

Výstupem knihovny je extraktivní shrnutí stránky (dále jen *Shrnutí*).

Krok 6

V *Textové podobě stránky* se spočítá celkový počet slov (dále jen *Počet slov*). Podle počtu slov se dle vzorce vypočítá *Odhadovaný čas čtení*.

$$\text{Odhadovaný čas čtení} = \frac{\text{Počet slov}}{\text{Počet slov za minutu}}$$

Kde *Počet slov za minutu* je průměrná rychlost čtení. V metodice se počítá s 250 slovy za minutu, což je průměrná rychlost čtení dospělého člověka. Výsledek rovnice se zaokrouhlí na celé číslo nahoru. Zůstane *Odhadovaný čas čtení* v minutách.

Krok 7

Textová podoba stránky se zkrátí na 5000 znaků (kvůli limitu) a pošle se do služby Microsoft Text Analytics – Language detection, která rozpozná jazyk textu. Pokud není jazyk podporovaný, uživatel je o tom informován a proces se zastaví. Pokud služby pro analýzu textu jazyk podporují, pokračuje se na další krok. Za podporované jazyky lze považovat angličtinu, španělštinu, francouzštinu a němčinu.

Krok 8

Pokud bylo ve stránce nalezeno více než pět odstavců (elementy <p>), spojí se do jednoho dlouhého textu. Prázdné odstavce se ignorují. Pokud bylo nalezeno méně odstavců obsahující text, posílá se dál *Textová podoba stránky*.

Krok 9

Výstup z předešlého kroku se zpracuje službami pro analýzu textu:

- Microsoft Text Analytics,
- Watson Natural Language Understanding,
- Amazon Comprehend.

Krok 10

Porovnájí se mezi sebou výstupy ze služeb pro analýzu textu a nadpisy (připravené ve 4. kroku). Porovnávají se:

- klíčová slova z Microsoft Text Analytics a klíčová slova z Watson Natural Language Understanding,
- klíčová slova z Watson Natural Language Understanding a klíčová slova z Amazon Comprehend,
- klíčová slova z Microsoft Text Analytics a klíčová slova z Amazon Comprehend,
- klíčová slova z Watson Natural Language Understanding a nadpisy,
- klíčová slova z Microsoft Text Analytics a nadpisy,
- klíčová slova z Amazon Comprehend a nadpisy.

Při porovnání se vybírá průnik ekvivalentních frází. Klíčovým slovům se nastaví skóre relevance. To se vypočítá jako aritmetický průměr relevancí klíčových slov podle jednotlivých služeb. Pokud se z některé služby vrátí prázdný výstup, do porovnání

klíčových slov se výstup z této služby nezahrne. Množina vybraných frází se seřadí sestupně podle skóre relevance a odeberou se duplicity. Pokud množina obsahuje více než šest frází, klíčová slova s nejnižším skóre relevance se odeberou. Množina vybraných klíčových frází se pak pošle do posledního kroku (dále jen *Klíčová slova*).

Krok 11

Sestaví se finální souhrn informací o webové stránce, který obsahuje údaje z předchozích kroků. Výstupní souhrn má tuto podobu:

Title: *Title*.

Keywords: *Klíčové slovo 1, Klíčové slovo 2, ... , Klíčové slovo 6*.

Summary: *Shrnutí*.

Estimated reading time: *Odhadovaný čas čtení*.

3.6.2. Klasifikace obrázků

Také analýza obrázků na stránce je definovaná jako sekvence kroků, které jsou na sobě navzájem závislé.

Krok 1

Odeberou se části stránky, které nejsou relevantní pro hlavní obsah. Zpravidla se jedná o hlavičku, navigaci, patičku aj. Z HTML kódu stránky se odstraní HTML komentáře. Dále se z HTML kódu odstraní elementy: header, nav, footer a další nadbytečné elementy. Výběr elementů, které se mají odstranit, lze udělat např. pomocí selektorů s využitím regulárních výrazů. Zůstane HTML kód, ve kterém by měl být jen podstatný obsah.

Krok 2

Z HTML kódu zbylého po kroku 1 se extrahují všechny obrázky, tzn. všechny elementy ``. U každého obrázku se z atributu `src` extrahuje URL adresa obrázku (dále jen *URL obrázku*). *URL obrázku* se upraví do podoby absolutní URL adresy.

Spočítá se celkový počet elementů `` na stránce (dále jen *Počet obrázků*).

Krok 3

URL obrázku se pošle do služby Microsoft Computer Vision. Výstup z Microsoft Computer Vision obsahuje atribut `captions.text` (dále jen *Titulek obrázku*). *Titulek obrázku* se předá do posledního kroku.

Krok 4

Iteruje se nad všemi obrázky a sestaví se souhrn popisků všech obrázků na webové stránce. Obrázky, které mají prázdný popis (,,“ nebo NULL) se ignorují. Výstupní souhrn popsání obrázků má tuto podobu:

Images count: *Počet obrázků.*

Images description: *Titulek obrázku 1, Titulek obrázku 2, ... , Titulek obrázku N.*

4. Implementace metodiky v pilotní aplikaci

Aby bylo možné ověřit, zda je navržená metodika funkční, je potřeba ji nejprve aplikovat v praxi. K tomuto účelu slouží pilotní aplikace. Prostřednictvím pilotní aplikace se metodika testuje a vyhodnocuje. Metodika je na předchozích stránkách rozdělená na dvě nezávislé části. Stejně tak je ovšem možné tyto dvě části prolnout a implementovat metodiku jako celek, protože úvodní kroky jsou stejné. V pilotní aplikaci je proto metodika implementovaná jako celek. V této kapitole je představena platforma Node-RED, která je použita pro vývoj pilotní aplikace. Taktéž jsou zde definované požadavky na aplikaci a je zde vysvětlena implementace důležitých funkcí aplikace. Některé části funkcí jsou pro lepší pochopení objasněny ukázkou kódu.

4.1. Platforma pro vývoj

Způsobů, jak metodiku převést do podoby softwarové aplikace, je nespočet. Může se jednat o klasický program pro PC, jednoduchý program spustitelný v konzoli příkazového řádku, mobilní aplikaci nebo webovou aplikaci běžící jako služba na serveru. Jak již bylo řečeno, metodiku lze implementovat v jakémkoli programovacím jazyku. Na začátku připadaly v úvahu dvě možnosti jak pilotní aplikaci vytvořit.

1. Využít nástroj pro tvorbu aplikací, který obsahuje předdefinované funkce.
2. Naprogramovat celou aplikaci od nuly, třeba i s využitím nějakého frameworku.

Protože pilotní aplikace slouží jen jako nástroj pro testování metodiky, nejsou na ni kladeny vysoké nároky. Nasazení pilotní aplikace veřejně do ostrého provozu není, vyjma testování, v plánu. Při výběru technologií proto lze upřednostnit rychlost a jednoduchost tvorby před např. bezpečností či rychlostí aplikace. S ohledem na tato fakta byla zvolena první možnost – vytvořit aplikaci v některém z nástrojů, které už obsahují předem definované funkce pro tvorbu aplikací.

V době výběru platformy pro vývoj aplikace padla volba na nástroj Node-RED. Alternativou jsou nástroje NoFlo a Losant, které poskytují podobný způsob tvorby aplikací. Losant je proprietární software, který nabízí společnost Losant svým klientům pro správu IoT⁷ zařízení. NoFlo je volně dostupný nástroj pro flow-based programování. Není přímo

⁷ IoT – Internet of Things – koncept připojení věcí do existující internetové infrastruktury.

zaměřen jen na programování aplikací pro IoT zařízení. NoFlo se prezentuje jako prostředí pro jednoduché programování jakýchkoli aplikací v JavaScriptu. Důvodů, proč byl zvolen právě Node-RED, je několik. Node-RED je open-source projekt a má na sebe navázanou širokou komunitu vývojářů. Díky tomu nabízí rozsáhlou knihovnu modulů a funkcí, které lze volně použít. Využití již existujících modulů umožňuje oprostít se od tvorby základních funkcí aplikace a soustředit se na funkce důležité pro implementaci metodiky.

4.1.1. Flow-based programování

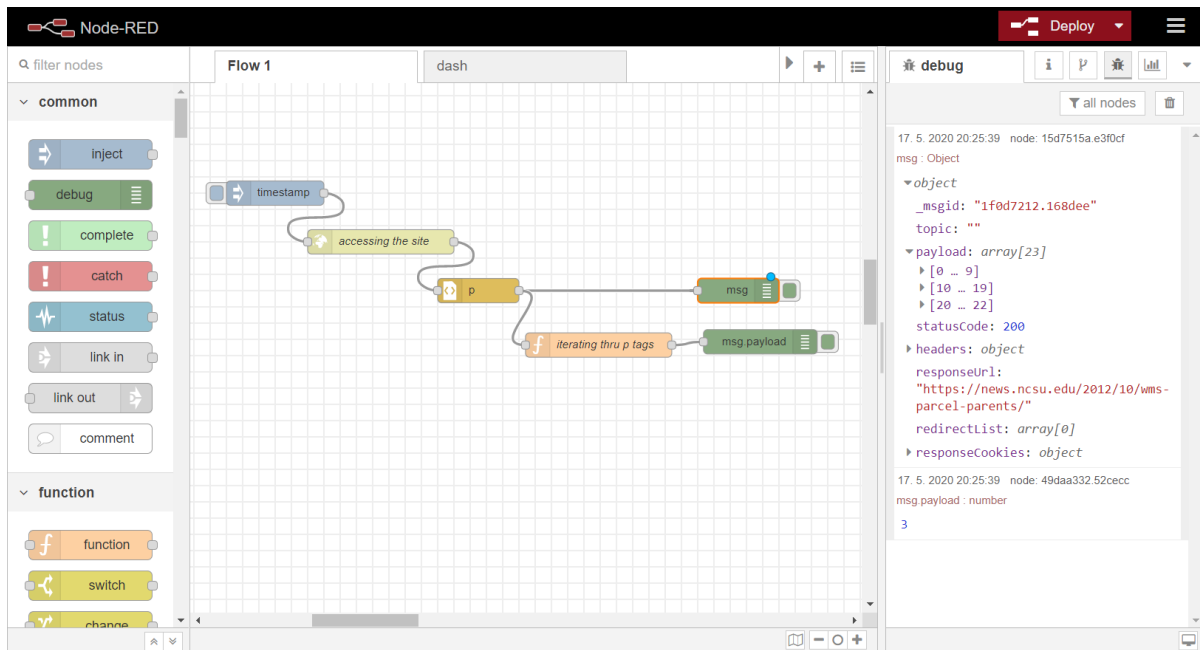
Flow-based programování je programovací paradigma pro vývoj aplikací. Aplikaci definuje jako síť uzlů chovajících se jako tzv. black box. Uzly spolu komunikují prostřednictvím zpráv, tzv. informačních paketů, které proudí skrze předem definovaná spojení. Tato spojení se nacházejí mimo uzly. Pomocí spojů lze uzly donekonečna propojovat a vytvářet tak libovolně spleť aplikace, aniž by bylo nutné zasahovat do logiky jednotlivých uzlů. [31]

4.1.2. Node-RED

Node-RED je nástroj pro flow-based vizuální programování. Běhové prostředí je postavené na technologii Node.js a funkce se píše v JavaScriptu. Původně je určený pro tvorbu aplikací pro Internet věcí. Umožňuje snadné propojení fyzických IoT zařízení, potřebných API rozhraní a online služeb. Jednotlivé funkce aplikace jsou reprezentovány uzly, tzv. nody. Může se jednat o jednoduché tlačítko, výpis textu do uživatelského rozhraní nebo komplexní službu třetí strany. Tyto uzly lze libovolně ukládat, sdílet či upravovat. Navzájem propojené uzly tvoří tok, tzv. flow. Toky jsou uloženy ve formátu JSON, díky čemuž lze jednoduše sdílet celé toky včetně uzlů. Uzly spolu komunikují pomocí zpráv (messages), které si mezi sebou posílají. Zprávy mohou nést libovolné množství vlastností. Vlastnosti zprávy mohou nabývat všech datových typů, které JavaScript podporuje. Nutno zmínit, že každá zpráva má dvě základní vlastnosti:

- `_msgid` – unikátní identifikátor zprávy,
- `payload` – vlastnost zpravidla nesoucí hlavní obsah zprávy.

Součástí Node-RED je přívětivý editor, ke kterému se přistupuje skrze webový prohlížeč. Editor obsahuje základní sadu uzlů, jež lze ihned používat, ladící konzoli, Git a další vývojářské nástroje. Node-RED se spouští jako Node.js server. Po spuštění Node.js serveru se k editoru přistupuje přes webový prohlížeč zadáním adresy 127.0.0.1:1880.



Obrázek 2: Vývojové prostředí Node-RED

4.2. Požadavky na aplikaci

Než se přejde k samotné tvorbě jakékoli aplikace, je potřeba provést alespoň základní analýzu požadavků a navrhnout adekvátní řešení. V případě pilotní aplikace vychází požadavky z navržené metodiky. Velká část požadavků proto koreluje s funkcemi, které vyžaduje metodika.

4.2.1. Funkční požadavky

Funkční požadavky specifikují, jak by se měla aplikace chovat v určitých situacích.

- **Zadání URL adresy webové stránky** – aplikace má textové pole pro zadání vstupu uživatelem.
- **Řečová syntéza** – výstup aplikace je uživateli předáván v podobě hlasu; aplikace komunikuje prostřednictvím API se službou Amazon Polly.
- **HTML parser** – aplikace dokáže provést syntaktickou analýzu HTML kódu stránky, vydolovat z něj požadované elementy a textový obsah.
- **Nastavení priority HTML elementům** – vydolovaným elementům se nastaví priority.
- **Sumarizace textového obsahu** – aplikace v sobě integruje knihovnu Bert Extractive Summarizer, případně s ní komunikuje prostřednictvím API.

- **Analýza klíčových slov** – aplikace komunikuje prostřednictvím API se službami Microsoft Text Analytics a Watson Natural Language Understanding. Výstupy služeb porovnává mezi sebou.
- **Výpočet odhadovaného času čtení** – na základě počtu slov na stránce se vypočítá odhadovaný čas čtení stránky.
- **Klasifikace obrázků** – aplikace komunikuje prostřednictvím API se službou Microsoft Computer Vision.
- **Vytvoření výstupního popisu stránky** – sestaví se finální souhrn informací o webové stránce.

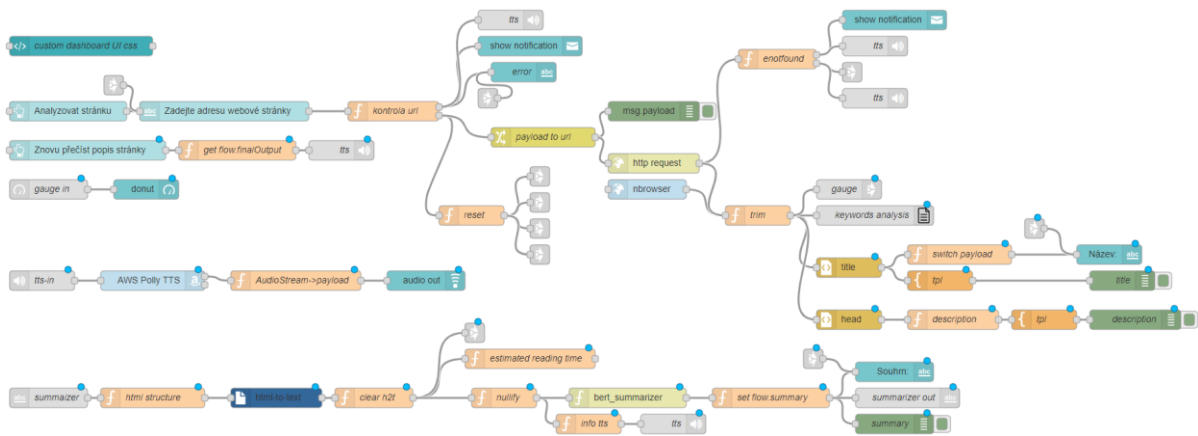
4.2.2. Nefunkční požadavky

Nefunkční požadavky specifikují vlastnosti a omezení aplikace.

- **Přístupnost pro nevidomé** – aplikace je plně přístupná pro zrakově postižené osoby.
- **Nezávislost na platformě** – aplikace není závislá na typu zařízení, ze kterého uživatel přistupuje, na rozlišení obrazovky ani na verzi systému.
- **Sériové zpracování požadavků** – aplikace vyřizuje požadavky od uživatele sériově.
- **Přijatelná doba odezvy** – aplikace průběžně poskytuje zpětnou vazbu uživateli a požadavky vyřizuje maximálně v řádu desítek sekund.

4.3. Aplikační logika

Protože je pilotní aplikace postavená na platformě Node-RED, je tvořena několika uzly propojenými do jednoho toku. Pro lepší pochopení by se dalo říct, že každý uzel v toku reprezentuje určitou funkci aplikace. Celkem je v aplikaci více než 150 uzlů. Velkou část aplikační logiky zajišťují uzly, které jsou základní součástí editoru Node-RED nebo pocházejí od jiných tvůrců. Část aplikační logiky je napsaná na míru. Pro psaní vlastních funkcí má Node-RED uzel zvaný Function. Ten lze vložit kamkoli do toku mezi jiné uzly a naprogramovat mu jakoukoli funkci.



Obrázek 3: Podoba pilotní aplikace v Node-RED

4.3.1. Grafické uživatelské rozhraní

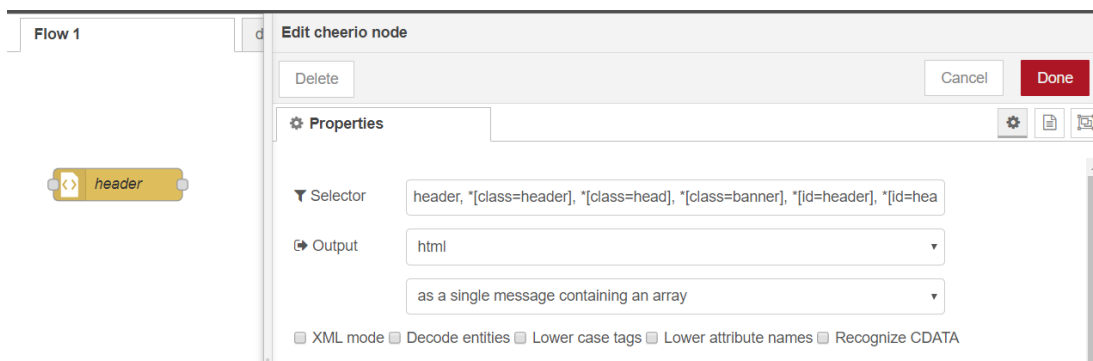
Grafické uživatelské rozhraní aplikace je vytvořeno pomocí sady uzlů Node-RED Dashboard. Každý uzel ze sady reprezentuje jeden ovládací prvek v grafickém rozhraní. Aplikace využívá uzly pro textové pole, tlačítko, textový výpis, notifikaci a měřidlo. Ke grafickému uživatelskému rozhraní se přistupuje přes webový prohlížeč zadáním adresy 127.0.0.1:1880/app.



Obrázek 4: Grafické uživatelské rozhraní aplikace a Node-RED Dashboard uzly

4.3.2. Syntaktická analýza HTML kódu

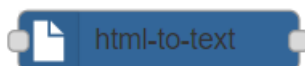
Aby bylo možné z HTML kódu extrahovat konkrétní elementy, je potřeba kódu porozumět. Pro práci s HTML kódem se používá uzel Cheerio. Ten parsuje HTML kód a extrahuje z něj elementy. Pracuje s objektovým modelem dokumentu (DOM). Požadované elementy lze specifikovat kombinací CSS selektorů a jQuery selektorů.



Obrázek 5: Node-RED uzel Cheerio a jeho nastavení

4.3.3. Převod HTML kódu do prostého textu

Převod HTML kódu do prostého textu je funkce využívající syntaktickou analýzu kódu. Samotný převod zajišťuje uzel `html-to-text`. Uzel parsuje HTML kód, extrahuje text z HTML tagů a vytváří dokument s prostým textem. Výsledný text obsahuje jen mezery a odřádkování – žádné další prvky navíc. Původní uzel `html-to-text` musel být mírně upraven, protože výstupní prostý text byl špatně zřetězen.



Obrázek 6: Node-RED uzel `html-to-text`

4.3.4. Nastavení priority specifickým prvkům

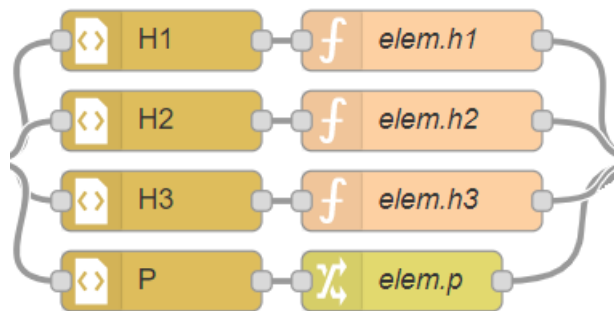
Metodika nastavuje některým prvkům prioritu. To je v aplikaci realizováno dvěma uzly. Nejprve uzel `Cheerio` extrahuje požadované prvky a vytvoří z nich pole. Vzápětí se toto pole projde běžnou JavaScript funkcí a prvkům v poli se přidělí priorita. Funkce pro přidělování priority je v toku vložena jako uzel `Function`.

```

1. var newPayload = {};
2. newPayload.priority = 0.9;
3. newPayload.elements = msg.payload;
4. msg.elem = 'h2';
5. msg.payload = newPayload;
6. return msg;

```

Ukázka 6: Nastavení priority prvkům H2



Obrázek 7: Node-RED uzly Cheerio a Function

4.3.5. Sumarizace textového obsahu

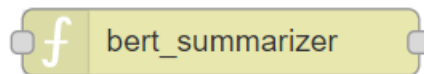
Sumarizaci textového obsahu řeší externí knihovna Bert Extractive Summarizer. Ta je do aplikace implementovaná jako zvláštní uzel prostřednictvím programu bert_summarizer.py. Program bert_summarizer.py vytvořil autor diplomové práce a je napsaný v Pythonu. Program v sobě obsahuje knihovnu Bert Extractive Summarizer. Navíc také funguje jako emulátor Node-RED uzlu v prostředí Python. Zadáním příkazu pynodered bert_summarizer.py v příkazové řádce se spustí program bert_summarizer.py. Pomocí programu bert_summarizer.py a knihovny pynodered se v editoru Node-RED vytvoří speciální uzel, který zajišťuje funkce na straně Node-RED. Současně se také spustí Python server, který obstarává komunikaci mezi Node-RED a programem bert_summarizer.py. Kdykoli přijde zpráva do Node-RED uzlu bert_summarizer (obrázek 8), knihovna pynodered předá požadavek do programu bert_summarizer.py, ten provede sumarizaci textu a přes pynodered vrátí výsledek zpět do aplikace v Node-RED.

```

1. from pynodered import node_red
2. from summarizer import Summarizer
3.
4. @node_red(category="advanced")
5. def bert_summarizer(node, msg):
6.     input_text = msg['payload']
7.     model = Summarizer()
8.
9.     # 0.07 = The ratio of sentences that we want for the final summary.
10.    output_text = model(input_text, 0.07)
11.    output_text = output_text.replace('\n', '')
12.
13.    msg['payload'] = output_text
14.    return msg

```

Ukázka 7: Kód programu bert_summarizer.py



Obrázek 8: Node-RED uzel bert_summarizer

4.3.6. Analýza klíčových slov

Před voláním rozhraní pro analýzu textu se prvně rozhoduje, jaký text bude analyzován. Rozhodovací podmínka je vložena v uzlu Function. Pokud syntaktická analýza kódu alespoň šest tagů <p>, zkontroluje se zde, zda nejsou prázdné. Text odstavců, které nejsou prázdné, se zřetězí do jednoho dlouhého textu. Pokud se v kódu žádné odstavce nevyskytovaly, posílá se dál výstup z uzlu html-to-text.

```
1. if(msg.payload.p !== undefined) {
2.     paragraphs = msg.payload.p;
3.     if(paragraphs.length > 5) {
4.         for(var i=0; i < paragraphs.length; i++) {
5.             if(paragraphs[i] !== "" && paragraphs[i] !== " ") {
6.                 out = out + paragraphs[i] + "\n";
7.             }
8.         }
9.         msg.payload = out;
10.    } else {
11.        // load cleaned body code
12.        msg.payload = msg.h2t;
13.    }
14. } else {
15.    // load cleaned body code
16.    msg.payload = msg.h2t;
17. }
```

Ukázka 8: Výběr textu pro analýzu klíčových slov

V dalším kroku je prostřednictvím uzlů třetí strany Microsoft Text Analytics, Watson Natural Language Understanding a Amazon Comprehend provedena sémantická analýza textu. Všechny tři uzly vrací JavaScript objekt. Výstupy služeb a nadpisy ze stránky se spojí pomocí uzlu Join. Posledním krokem analýzy klíčových slov je uzel Function, který porovnává klíčová slova mezi sebou a klíčová slova vůči nadpisům. Výstupem funkce je průnik stejných frází.

```

1. /* if awsKeywords is equal to watsonKeywords */
2. if(watsonKeywords && awsKeywords) {
3.     for(var i=0;i<watsonKeywords.length;i++) {
4.         for(var j=0;j<awsKeywords.length;j++) {
5.             if( watsonKeywords[i].text.includes(awsKeywords[j].Text) ) {
6.                 tmpKeywords = {};
7.                 tmpKeywords.text = watsonKeywords[i].text;
8.                 /* set relevance to arithmetic mean */
9.                 tmpKeywords.relevance = (watsonKeywords[i].relevance + awsKeywords[
   j].Score) / 2;
10.                outputArr.push(tmpKeywords);
11.            }
12.        }
13.    }
14. }

```

Ukázka 9: Porovnání klíčových slov z různých služeb

4.3.7. Výpočet odhadovaného času čtení

Odhadovaný čas čtení může uživateli poskytnout přibližnou představu o tom, jak dlouhý je text na stránce. Výpočet se provádí v uzlu Function. Nejdříve se spočítá počet slov v textu, poté se dle vzorce vypočítá odhadovaný čas čtení.

```

1. // count number of words
2. wordsCount = text.split(' ').filter(String).length;
3.
4. // estimated reading time calculation
5. var estimatedReadingTime = Math.ceil((wordsCount/250).toFixed(2));

```

Ukázka 10: Výpočet odhadovaného času čtení

4.3.8. Klasifikace obrázků

Klasifikace obrázků je rozdělena do více uzlů. První je uzel Cheerio, který z kódu extrahuje elementy . Po něm se v uzlu Function nastavuje obrázkům absolutní URL adresa. Pak uzel Split rozdělí pole obrázků na sekvenci samostatných obrázků. Nejdůležitější je uzel Microsoft Computer Vision, do kterého se sériově posílají URL adresy obrázků a který vzdáleně provádí obsahovou analýzu obrazu. Výstupy ze služby chodí sériově. Prvně se z každého výstupu extrahuje atribut `captions.text` a nakonec se všechny popisky obrázků spojí uzlem Join do jednoho pole.



Obrázek 9: Klasifikace obrázků

4.3.9. Vytvoření výstupního popisu webové stránky

Výstupní popis stránky se tvoří v uzlu Function. V podstatě se jedná o zřetězení proměnných, které se v průběhu procesu vytvářely a ukládaly.

```
1. /* Build output text */
2.
3. if(title !== undefined) {
4.   outputText += "Title: " + title + ".\n";
5. }
6.
7. if(outputKeywords !== [] && outputKeywords !== undefined) {
8.   outputText += "Keywords: ";
9.   outputKeywords.forEach(function(element) {
10.    outputText = outputText + element.text + ", ";
11.  });
12.  /* Replace last comma with dot */
13.  outputText = outputText.slice(0, -2) + '.\n';
14. }
15.
16. if(summary !== undefined) {
17.   outputText += "Summary: " + summary + "\n";
18. }
19.
20. if(estimatedReadingTime !== undefined) {
21.   outputText += "Estimated reading time: " + estimatedReadingTime + " minutes.\n";
22. };
```

Ukázka 11: Vytvoření výstupního popisu webové stránky

4.3.10. Řečová syntéza

O řečovou syntézu se stará uzel Amazon Polly. Do uzlu se jen pošle připravený text. Rozhraní Amazon Polly text převede na zvukový signál a zpět vrací zvuk v binární podobě.



Obrázek 10: Node-RED uzly pro syntézu řeči

5. Testování

Jediným způsobem, jak spolehlivě ověřit návrh metodiky a odhalit její slabé stránky, je testování. Obecně lze říci, že pilotní aplikace je výborný nástroj k testování navržených řešení, protože se díky ní projeví všechna pozitiva i negativa návrhu. Následující podkapitola se zabývá funkčním testováním aplikace jakožto softwaru. Pozornost je věnovaná průběžnému testování funkcí, testovacím nástrojům a stádiím testování. Další podkapitola klade velký důraz na uživatelské testování se zrakově postiženými testery, jehož cílem je validace pilotní aplikace i navržené metodiky.

5.1. Funkční testování aplikace

Funkční testování aplikace spočívá v tzv. testování programátorem. Aplikaci průběžně testuje sám vývojář na úrovni zdrojového kódu. Editor Node-RED k tomuto účelu poskytuje ladící konzoli, do které se vypisují chybové hlášky a logy. Ta je součástí postranního panelu editoru Node-RED. Do ladící konzole lze vypsát celou stromovou strukturu zpráv, které mezi uzly putují. Spolu s obsahem zprávy se samozřejmě vypisuje také čas přijetí zprávy a identifikátor uzlu, který zprávu vypsál. Ke kontrole obsahu přenášených zpráv nejlépe slouží uzel debug. Uzel debug lze připojit na jakýkoli jiný uzel. Všechny zprávy, které do něj vstoupí, se vypíší do ladící konzole, díky čemuž má vývojář kompletní přehled o vstupech a výstupech jednotlivých uzlů.

Node-RED umožňuje definovat tři typy proměnných:

- globální proměnné viditelné pro všechny toky,
- proměnné viditelné jen v rámci toku,
- proměnné viditelné jen v rámci jednoho uzlu.

V postranním panelu editoru Node-RED je schovaná karta Context Data. Na této kartě lze průběžně sledovat všechny definované proměnné, ať už jsou kteréhokoli typu.

Aplikace byla implementovaná po částech. Díky tomu, že je tvořená s využitím flow-based programování, je možné ji poměrně jednoduše po částech i testovat. Na libovolné místo v toku lze vložit další uzly určené jen pro testovací účely. Tímto způsobem je možné posílat do uzlů testovací data a testovat např. jen práci s obrázky, aniž by byly ovlivněny ostatní uzly v toku. Během implementace nové funkce byla vždy tato funkce testovaná samostatně.

Jakmile bylo jasné, že je její chování vyladěno správně, přišlo na řadu otestování celého toku, tzn. otestování celé aplikace.

5.2. Uživatelské testování

Záměrem uživatelského testování bylo ověření reálné funkčnosti vytvořené metodiky a jejích výstupů. Pro validaci výstupů metodiky bylo využito dotazníkové šetření. Současně se kontrolovala také použitelnost a přístupnost pilotní aplikace.

Aplikace byla pro účely testování nasazena na platformu Microsoft Azure. Byla zprovozněna na virtuálním serveru s operačním systémem Linux Ubuntu a během testování byla přístupná na veřejné IP adrese. Díky tomu mohli všichni testovat aplikaci odkudkoli a na libovolném zařízení.

5.2.1. Profesionální testeři

Uživatelské testování pilotní aplikace bylo postaveno na testovacím scénáři. Hlavním bodem této fáze bylo ověření přístupnosti a použitelnosti aplikace. Aplikaci testovali tři profesionální, zrakově postižení testeři a jeden vidící tester. Zainteresovaní profesionální testeři jsou prakticky nevidomí a působí v různých firmách jako konzultanti a testeři přístupnosti aplikací. Oslovení profesionálních testerů probíhalo dvěma způsoby: osobně a elektronicky. Navázání osobních kontaktů se uskutečnila na konferenci pro osoby s těžkým postižením zraku Agora a na konferenci pro osoby se specifickými potřebami Inspo. Na těchto konferencích bylo osloveno pět potenciálních testerů, dva z nich poté aplikaci testovali. Další tři testeři byli osloveni e-mailem. Tímto způsobem se podařilo získat jednoho člověka.

Testeři měli za úkol otestovat aplikaci na třech předem daných webových stránkách, zkontrolovat, zda řečová syntéza přečetla údaje o stránkách, a zapsat do scénáře výsledek testování. Součástí scénáře bylo několik otázek, např. zda byl naplněn očekávaný výsledek, zda byl interpretován výstup analýzy u všech testovaných webů nebo jaké problémy během testování nastaly. Vyplněné scénáře jsou k dispozici v příloze 3: Testovací scénáře. Výsledky uživatelského testování jsou shrnuty v kapitole 6.1. Výsledky uživatelského testování.

5.2.2. Dotazníkové šetření

Pro validaci výstupů metodiky bylo využito dotazníkové šetření. Oslovení zrakově postižených respondentů probíhalo převážně elektronickou cestou. V první vlně bylo, prostřednictvím společnosti TyfloCentrum České Budějovice, o.p.s., osloveno e-mailem přibližně 50 těžce zrakově znevýhodněných uživatelů, kteří aktivně pracují s internetem. V další vlně byl dotazník společně s odkazem na aplikaci publikován na blogu poslepu.cz. Blog POSLEPU – Pomoc SLEPým Uživatelům – je zaměřen na přístupnost webu, aplikací a informací, asistivní technologie pro uživatele se specifickými potřebami a inkluzivní design. Ve třetí vlně odkazy sdílel Radek Pavlíček na svém osobním twitterovém účtu a na facebookové stránce Poslepu.cz. Mgr. Radek Pavlíček je autorem blogu POSLEPU a předním českým odborníkem na přístupnost. Na blogu poslepu.cz si článek o pilotní aplikaci během prvního týdne zobrazilo 122 lidí. Na Twitteru měl příspěvek dosah 9 753 zobrazení (oslovil 135 lidí) za týden, na Facebooku pak 32 zobrazení.

Dotazník byl vytvořen ve službě Google Formuláře, což zaručovalo možnost jednoduchého sdílení i přístupnost pro zrakově znevýhodněné osoby. Byl sestaven tak, aby bylo možné jej vyplnit i bez testování aplikace. Všechny potřebné informace byly obsaženy přímo v něm. Dotazník obsahoval sedm otázek a dva popisky webových stránek, které se vztahovaly k jednotlivým otázkám. Některé otázky byly výběrové, jiné stupnicové. Všechny otázky byly povinné. Respondenti hodnotili kvalitu popisků stránek, délku popisků, množství klíčových slov či délku popisků obrázků. V posledních dvou otázkách byly srovnávány výstupy vytvořené metodiky s výstupy jiných služeb. Respondenti vybírali, který z popisků jim více vyhovuje. Šestá otázka srovnávala výstup vytvořené metodiky s výstupem sumarizátoru smmry.com. Sedmá otázka srovnávala výstup vytvořené metodiky s výstupem funkce Skim Reading Summary z odečítače obrazovky JAWS. Dotazník je k nahlédnutí v příloze 4: Dotazník. Odpovědi respondentů jsou vyhodnoceny v podkapitole 6.1.2. Vyhodnocení dotazníkového šetření.

6. Vyhodnocení

Metodika pro obsahovou analýzu webových stránek byla hodnocena na základě jejího výstupu. Textový výstup metodiky byl srovnáván s výstupy jiných služeb a to na dvou úrovních: prostřednictvím dotazníkového šetření a komparací metrikami ROUGE. Poslední kapitola tedy shrnuje výsledky testování a posuzuje vytvořenou metodiku.

6.1. Výsledky uživatelského testování

6.1.1. Testování pomocí scénářů

Úkolem uživatelského testování pomocí scénáře bylo ověřit chování aplikace v reálném provozu. Testeři testovali na vlastních počítačích, zpravidla s operačními systémy Windows 10 a MacOS. Při testování měli aktivní očečítače obrazovky NVDA, Apple VoiceOver nebo Windows Narrator.

V některých případech testeři upozorňovali na chybu `Error: connect ECONNREFUSED 127.0.0.1:5051`. Důvodem chyby byl nestabilní pynodered server, který vyřizoval požadavky mezi platformou Node-RED a programem `bert_summarizer.py`. Celkový výstup aplikace tato chyba ovlivnila jen minimálně. Kvůli chybě se sice nepodařilo vygenerovat sumarizovaný souhrn, ostatní prvky výstupu (klíčová slova, popisky obrázků aj.) však byly v pořádku zpracovány. Chybě by šlo předejít nahrazením testovacího pynodered serveru za produkční WSGI server.

Tři ze čtyř testerů měli problém s nefunkční řečovou syntézou aplikace. Informace se vypsaly na obrazovku, díky čemuž si je mohli přečíst nativním očečítačem obrazovky, ale nebyly interpretovány integrovanou syntézou pilotní aplikace. Testeři kvůli tomu upozorňovali na absenci zpětné vazby. Tester Roman ve scénáři zmiňuje „*velmi dlouhé načítání bez jakékoli odezvy, která by uživatele informovala, že se něco děje*“. Během testování vyšlo najevo, že se jedná o problém s podporou Web Audio API na straně webového prohlížeče. Tento problém se vyskytuje pouze v některých webových prohlížečích.

I přes výše zmíněné potíže byla celková přístupnost aplikace hodnocena kladně. Jak napsal tester Karel: „*všechny prvky byly bez problémů dosažitelné očečítačem*“.

6.1.2. Vyhodnocení dotazníkového šetření

Dotazníkové šetření sloužilo k získání zpětné vazby na výstupy metodiky od samotných uživatelů. Celkem se shromáždilo 11 vyplněných dotazníků. Kompletní vyhodnocení v podobě grafů je v příloze 5: Vyhodnocení dotazníkového šetření. Zde následuje souhrn základních bodů.

Všichni respondenti byli s výstupním popisem webové stránky spokojeni. Většina (55 %) respondentů uvedla, že popis stránky je dobrý, ale vyžaduje změny. Zbýlých 45 % uvedlo, že jim popis stránky vyhovuje beze změn. Více než polovina (55 %) respondentů uvedla, že celková délka výstupního popisku stránky je optimální. Popisek stránky může obsahovat až deset klíčových slov, jež vystihují obsah stránky. Majorita (64 %) respondentů uvedla, že deset klíčových slov je mnoho, vedle toho 36 % to považuje za optimální množství. Celkový popisek stránky v sobě nese také popisky obrázků na stránce. Téměř všichni (91 %) uvedli, že popisky obrázků jsou optimální. V posledních dvou otázkách respondenti vybírali mezi výstupem vytvořené metodiky a výstupem konkurenční služby. U obou otázek většina (82 %) respondentů uvedla, že jim více vyhovuje výstup vytvořené metodiky. Pouze 18 % upřednostnilo výstup konkurenční služby.

Celkově lze konstatovat, že obsah i délka popisku webové stránky uživatelům spíše vyhovuje, ale popisek vyžaduje určité změny. Jedna změna byla provedena ihned po vyhodnocení. Na základě zpětné vazby od respondentů byl snížen počet klíčových slov ve výstupu metodiky z deseti na šest.

6.2. Hodnocení sumarizace

Výstup vytvořené metodiky by měl sloužit jako zkrácený souhrn webové stránky. Ve své podstatě tomu tak opravdu je, protože součástí metodiky je i extraktivní sumarizace textu z webové stránky. Respondenti v dotazníkové šetření hodnotili délku a celkovou podobu výstupního popisku metodiky. Neměli však možnost hodnotit kvalitu sumarizace. Jinak řečeno – nehodnotili, zda zkrácený souhrn webové stránky obsahuje relevantní informace. Z tohoto hlediska byl výstup metodiky hodnocen pomocí metrik ROUGE.

Žádný odečítač obrazovky neumí vytvořit textové shrnutí webové stránky nebo výtah klíčových bodů stránky. Výstup metodiky byl proto porovnáván s výstupy online sumarizátorů. Do porovnání byly zahrnuty tři sumarizátory:

- resoomer.com,
- smmry.com,
- splitbrain.org/services/ots.

Do porovnání byl navíc zahrnut výstup z funkce Skim Reading Summary, kterou nabízí odečítač obrazovky JAWS. Chování této funkce je blíže popsáno v podkapitole 1.3.1. Odečítač obrazovky JAWS. Díky tomu bude ilustrováno porovnání výstupu metodiky s výstupem odečítače obrazovky.

Porovnání textových výstupů bylo zpracováno v programu ROUGE 2.0. [32] Jedná se o open-source sadu nástrojů napsanou v jazyce Java, jež lze použít jako alternativu k originální sadě nástrojů od autora metrik ROUGE.

6.2.1. Metriky ROUGE

ROUGE, neboli Recall-Oriented Understudy for Gisting Evaluation, je soubor metrik pro hodnocení automatické sumarizace textů a strojového překladu textů. Základem je porovnání automaticky vytvořeného shrnutí se sadou referenčních shrnutí, obvykle vytvořených člověkem. [33] Soubor ROUGE obsahuje tyto metriky:

- ROUGE-N,
 - ROUGE-1,
 - ROUGE-2,
- ROUGE-L,
- ROUGE-W,
- ROUGE-S,
- ROUGE-SU.

Všechny metriky vrací tři hodnoty:

- recall – úplnost,
- precision – přesnost,
- F-measure – sloučení přesnosti a úplnosti do jedné hodnoty.

Původní výzkum [33] doporučuje pro porovnání krátkých shrnutí použít metriky ROUGE-1, ROUGE-L a ROUGE-W. Metrika ROUGE-1 porovnává překrytí automatického shrnutí a referenčního shrnutí pro všechna slova, tzv. unigramy. ROUGE-L porovnává překrytí

nejdelší shodující se sekvence slov. ROUGE-W porovnává překrytí nejdelší shodující se sekvence slov, navíc s přidáním váhových parametrů.

6.2.2. Výsledky komparace

Pro porovnání výstupu metodiky s výstupy jiných služeb byla vybrána sada deseti webových stránek různého zaměření. Každá webová stránka byla nejprve zpracována všemi službami. Tím vzniklo celkem 50 různých automatických souhrnů deseti webových stránek. Referenční souhrny vytvořil autor diplomové práce. Tyto souhrny byly následně porovnány metrikami ROUGE-1 a ROUGE-L. Při zpracování byla použita stematizace a odstranění tzv. stop slov. Výsledky metrik nemusí být zcela přesné, protože byl pro každou testovanou webovou stránku vytvořen jen jeden referenční souhrn. Výsledné hodnoty metrik byly zprůměrovány a jednotlivé služby byly porovnány na základě F-measure. Hodnota F-measure byla zvolena, jelikož v sobě kombinuje recall a precision. Výstupem komparace je finální tabulka níže.

Z tabulky lze vyčíst, že porovnání služeb metrikou ROUGE-L pouze potvrzuje výsledky, jež byly vyhodnoceny metrikou ROUGE-1. Fakticky by tedy pravděpodobně stačilo provést srovnání pouze metrikou ROUGE-1. Podle hodnoty F-measure dosahuje nejlepších výsledků sumarizace textu služba *summry.com*. V porovnání s ostatními službami si se shrnutím obsahu dobře poradila také vytvořená metodika. Další srovnávané služby dosahují podstatně horších výsledků.

	ROUGE-1			ROUGE-L		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Vytvořená metodika	0,40706	0,26871	0,31977	0,31750	0,23146	0,25201
Skim Reading Summary	0,57819	0,08791	0,14743	0,50111	0,11418	0,18128
resoomer.com	0,20521	0,31938	0,23487	0,24108	0,23683	0,19308
summry.com	0,34123	0,45646	0,38019	0,31684	0,42193	0,35003
splitbrain.org/services/ots	0,20391	0,09515	0,12645	0,15211	0,09554	0,11448

Tabulka 2: Výsledky komparace metrikami ROUGE

6.3. Porovnání popisu obrázku

Odečítač obrazovky JAWS obsahuje, od verze 2019, funkci pojmenovanou Picture Smart. Tato funkce by měla rozpoznat a klasifikovat obsah obrázku. Popis obrázku by měl být uživateli standardně interpretován řečovou syntézou. V této kapitole mělo být srovnání vygenerovaných popisků z odečítače JAWS s popisky obrázků z vytvořené metodiky. Praktické porovnání však, bohužel, nemohlo být provedeno. Testovací verze odečítače mají tuto funkci zablokovanou, a tudíž nelze obrázky klasifikovat. Během uživatelského testování neměl ani jeden z testerů k dispozici odečítač JAWS, z tohoto důvodu nebylo možné zapojit do problému testery. V každém případě se jedná o velký krok kupředu. Je vidět, že i tvůrci odečítačů jsou progresivní a snaží se nevidomým uživatelům nabídnout nejlepší možná řešení.

6.4. Shrnutí

Vytvořená metodika generuje vlastní textový výstup, jenž byl porovnán s výstupy jiných služeb a to jak z hlediska uživatelského, tak z hlediska obsahového. Z dotazníkového šetření vyplynulo, že obsah i délka výstupu vytvořené metodiky jsou optimální, jen by bylo vhodné provést drobné změny. Z hlediska sumarizace obsahu si lépe vedla služba [smmry.com](https://www.smmry.com). Sami uživatelé však v dotazníku preferovali textový souhrn vytvořený pilotní aplikací, před souhrnem vytvořeným službou [smmry.com](https://www.smmry.com). Vezme-li se v potaz celkové srovnání, lze vytvořenou metodiku a její výstupy hodnotit kladně.

6.5. Potenciální rozvoj metodiky

Náměty na budoucí rozvoj metodiky pro analýzu obsahu webových stránek vyvstaly prakticky hned po jejím vyhodnocení. Sumarizace textu je v metodice řešena pomocí knihovny Bert Extractive Summarizer. Potenciál pro zlepšení celé metodiky je ukrytý právě ve zdokonalení sumarizace obsahu. První možností je napojení na API služby [smmry.com](https://www.smmry.com), jež si v hodnocení vedla nejlépe. Druhou možností je výměna sumarizačního modelu BERT za novější algoritmus. Jak dokazují různé výzkumy a žebříčky, např.: [34], v současné době již existují modely, které v porozumění textu dosahují mnohem lepších výsledků, např. ALBERT. Je však potřeba si uvědomit, že vývoj algoritmů pro analýzu textu je neuvěřitelně rychlý a tímto způsobem by bylo možné metodiku vylepšovat donekonečna.

Další návrhy na zlepšení poskytlí sami testeři. Velice užitečné by bylo, implementovat metodiku jako rozšíření do webového prohlížeče. Pak by bylo možné vyžádat si souhrn libovolné stránky, např. z kontextové nabídky. Stejně tak by bylo možné vyžádat si přečtení popisku jednoho konkrétního obrázku.

Ačkoli proběhlo pár testů s českými, francouzskými, španělskými a německými webovými stránkami, metodika byla primárně vytvořena pro anglické texty. Zcela logicky se proto nabízí úprava metodiky tak, aby plnohodnotně podporovala více jazyků, včetně češtiny. Toho lze docílit jinou kombinací služeb pro analýzu klíčových slov a úpravou nebo výměnou modelu pro sumarizaci textu.

Závěr

Hlavním cílem diplomové práce byl návrh metodiky pro obsahovou analýzu webové stránky. Práce se zaměřila na osoby se specifickými potřebami. Konkrétně zcela nevidomé uživatele a uživatele se zbytky zraku, kteří si čtou webové stránky, ale při práci na počítači již nemohou využívat svého zraku. K dosažení stanoveného cíle vedly následující kroky.

Ze všeho nejdříve byla provedena rešerše odečítačů obrazovky a byly zmapovány jejich významné funkce. Vedle toho vzniklo také srovnání dostupných služeb pro analýzu textu, analýzu obrazu a syntézu řeči. Na základě rešerše byla navržena řešení problémů, s kterými se uživatelé těchto speciálních programů při procházení webových stránek setkávají.

V rámci klíčové fáze byla navržena a vypracována vlastní metodika pro zpracování obsahu webové stránky. Metodika definuje obecný postup, který zahrnuje: výběr relevantních elementů na webové stránce, použití služeb pro analýzu textu a obrazu k získání sémantických informací ze stránky, spojení získaných informací a interpretaci popisu stránky uživateli.

Za účelem testování a vyhodnocení metodiky byla vytvořena pilotní aplikace. Ta byla postavena na platformě Node-RED a nasazena na veřejný server. Prostřednictvím pilotní aplikace metodiku otestovali zrakově postižení uživatelé.

Vytvořená metodika byla hodnocena na dvou úrovních. Zaprvé: výstupy metodiky hodnotili sami uživatelé v dotazníkovém výzkumu. Zadruhé: textové výstupy metodiky byly porovnány s výstupy obdobných služeb pomocí metrik ROUGE. Komparace metrikami ROUGE ukázala, že metodika produkuje poměrně relevantní popisky webových stránek a její výstupy jsou lepší, než výstupy jiných sumarizátorů. V dotazníkovém šetření pak drtivá většina respondentů upřednostnila výstup vytvořené metodiky před výstupy jiných služeb.

Žádný z odečítačů obrazovky dosud neumí sumarizovat text. Nevidomí uživatelé internetu sice mohou využít některý z volně dostupných sumarizátorů textu, ty však nevykazují tak dobré výsledky a ne vždy jsou přístupné pro osoby se specifickými potřebami. Vypracovaná metodika tak přináší nový přístup ke zpracování obsahu webových stránek. Implementuje-li se metodika např. jako rozšíření do webového prohlížeče, může pomoci nejen cílové skupině, pro kterou byla původně zamýšlena, ale i zdravým lidem.

Seznam použité literatury

- [1] BOURNE, Rupert R A, Seth R FLAXMAN, Tasanee BRAITHWAITE, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health* [online]. 2017, 5(9), e888-e897 [cit. 2019-04-01]. DOI: 10.1016/S2214-109X(17)30293-0. ISSN 2214109X. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S2214109X17302930>
- [2] MICHALÍK, Jan a kol. *Zdravotní postižení a pomáhající profese*. Praha: Portál, 2011. ISBN 978-80-7367-859-3.
- [3] SCHINDLER, Radek a Milan PEŠÁK. Kdo je zrakově postižený?. *Sjednocená organizace nevidomých a slabozrakých ČR* [online]. Praha: SONS ČR, c2002-2015 [cit. 2019-04-01]. Dostupné z: <http://archiv.sons.cz/kdojezp.php>
- [4] ČESKÝ STATISTICKÝ ÚŘAD. 3. Výběrové šetření osob se zdravotním postižením VŠPO 13: 3.7.3. Míra zdravotního postižení. In: ČESKÝ STATISTICKÝ ÚŘAD. *Výběrové šetření zdravotně postižených osob - 2013* [online]. Praha, 2014, 30.04.2014, s. 29 [cit. 2019-04-01]. Dostupné z: https://www.czso.cz/documents/10180/20543019/k3_260006-14_1.pdf/4384f318-fcae-4a20-941c-33f10d5a6324?version=1.0
- [5] SCHRÖDER, Marc. Emotional Speech Synthesis: A Review. In: DALSGAARD, Paul, Børge LINDBERG, Henrik BENNER a Zheng-hua TAN. *EUROSPEECH 2001 Scandinavia* [online]. Aalborg, Denmark, 2001 [cit. 2019-04-17]. Dostupné z: https://www.isca-speech.org/archive/archive_papers/eurospeech_2001/e01_0561.pdf
- [6] BENGIO, Yoshua. Learning Deep Architectures for AI. In: *Foundations and Trends® in Machine Learning* [online]. University of California, Berkeley, 2009, 2(1) [cit. 2019-04-22]. DOI: 10.1561/22000000006. ISBN 1935-8237. ISSN 1935-8237. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.701.9550&rep=rep1&type=pdf>

- [7] NUANCE COMMUNICATIONS, INC. Nuance Advances Text-to-Speech Technology through Deep Learning. *Nuance Advances Text-to-Speech Technology* [online]. Massachusetts, USA, 2018, February 14, 2018 [cit. 2019-04-22]. Dostupné z: <https://www.nuance.com/about-us/newsroom/press-releases/nuance-advances-text-to-speech-technology.html>
- [8] VAN DEN OORD, Aaron, Sander DIELEMAN, Heiga ZEN, et al. *WaveNet: A Generative Model for Raw Audio* [online]. In: Google, London, UK, 19 September 2016, s. 6 [cit. 2019-04-22]. Dostupné z: <https://arxiv.org/pdf/1609.03499.pdf>
- [9] KABELKA, Roman. Tajemství fokusu. In: *POSLEPU: POMOČ SLEPÝM Uživatelům* [online]. 2010, 3.1.2010 [cit. 2019-04-24]. Dostupné z: <http://poslepu.blogspot.com/2010/01/tajemstvi-fokusu.html>
- [10] FREEDOM SCIENTIFIC INC. *Skim Reading* [online]. [cit. 2019-05-23]. Dostupné z: https://support.freedomscientific.com/Training/Surfs-Up/Skim_Reading.htm
- [11] LINGUAMATICS. What is Text Mining, Text Analytics and Natural Language Processing? *Linguamatics* [online]. 2019 [cit. 2019-12-09]. Dostupné z: <https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing>
- [12] KAO, Anne a Steve R. POTEET, ed. *Natural Language Processing and Text Mining: Overview*. 1. Switzerland: Springer-Verlag London, 2007. ISBN 978-1-84628-754-1.
- [13] MONKEYLEARN INC. Text Analysis: Text Classification. *Text Analysis: the only guide you'll ever need* [online]. MonkeyLearn, 2019 [cit. 2019-12-13]. Dostupné z: <https://monkeylearn.com/text-analysis/>
- [14] BELIGA, Slobodan, Ana MEŠTROVIĆ a Sanda MARTINČIĆ-IPŠIĆ. An Overview of Graph-Based Keyword Extraction Methods and Approaches: Introduction. In: *Journal of Information and Organizational Sciences* [online]. Croatia: Faculty of Organization and Informatics, 2015, 2015-06-30, s. 1 [cit. 2019-12-16]. DOI: <https://doi.org/10.31341/jios>. ISSN 1846-9418. Dostupné z: <https://jios.foi.hr/index.php/jios/article/view/938/724>

- [15] KRYŚCIŃSKI, Wojciech, Nitish SHIRISH KESKAR, Bryan MCCANN, Caiming XIONG a Richard SOCHER. *Neural Text Summarization: A Critical Evaluation: Introduction* [online]. 23 Aug 2019 [cit. 2019-12-19]. Dostupné z: <https://arxiv.org/pdf/1908.08960.pdf>
- [16] GHOLAMREZAZADEH, Saeedeh, Mohsen Amini SALEHI a Bahareh GHOLAMZADEH. A Comprehensive Survey on Text Summarization Systems. In: *2009 2nd International Conference on Computer Science and its Applications* [online]. IEEE, 2009, 10-12 Dec. 2009 [cit. 2019-12-19]. DOI: 10.1109/CSA.2009.5404226. ISBN 978-1-4244-4945-3. Dostupné z: <http://ieeexplore.ieee.org/document/5404226/>
- [17] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. Fourth edition. Stamford, CT, USA: Cengage Learning, [2015]. ISBN 978-113-3593-607.
- [18] ANBARJAFARI (SHAHAB), Gholamreza. Introduction to image processing. *Introduction to image processing / Digital Image Processing* [online]. University of Tartu [cit. 2020-01-09]. Dostupné z: <https://sisu.ut.ee/imageprocessing/book/1>
- [19] KAELI, David, Perhaad MISTRY, Dana SCHAA a Dong Ping ZHANG. Case study: Image clustering: Introduction. *Heterogeneous Computing with OpenCL 2.0: (Third Edition)* [online]. Elsevier, 2015, 2015, 213 [cit. 2020-01-25]. DOI: 10.1016/B978-0-12-801414-1.00009-0. ISBN 9780128014141. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/B9780128014141000090>
- [20] MICROSOFT. How to: Detect sentiment using the Text Analytics API: Concepts. *Azure Cognitive Services / Microsoft Docs* [online]. 2020, 2020-01-02 [cit. 2020-01-30]. Dostupné z: <https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis>
- [21] MICROSOFT. Language and region support for the Speech service. *Azure Cognitive Services / Microsoft Docs* [online]. 2020, 2020-01-31 [cit. 2020-02-03]. Dostupné z: <https://docs.microsoft.com/en-gb/azure/cognitive-services/speech-service/language-support>

- [22] IBM. About. *IBM Cloud Docs* [online]. 2020, 2020-02-04 [cit. 2020-02-07]. Dostupné z: <https://cloud.ibm.com/docs/services/speech-to-text?topic=speech-to-text-about>
- [23] GOOGLE. Supported voices and languages. *Cloud Text-to-Speech Documentation* [online]. [cit. 2020-02-11]. Dostupné z: <https://cloud.google.com/text-to-speech/docs/voices>
- [24] KASTURE, N. R., Neha YARGAL, Neha NITYANAND SINGH, Neha KULKARNI a Vijay MATHUR. A Survey on Methods of Abstractive Text Summarization. In: *IJREST. International Journal for Research in Emerging Science and Technology* [online]. Bhandara, India: IJREST, 2014, November 2014, s. 53-57 [cit. 2020-04-05]. ISSN 2349-7610. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.2132&rep=rep1&type=pdf>
- [25] ANANTHAKRISHNAN, Abhinaya. Data Scientist's Guide to Summarization: Auto-text summarization tutorial for beginners. *Medium* [online]. Austin, USA, 2019, Mar 16, 2019 [cit. 2020-04-05]. Dostupné z: <https://medium.com/@abhinaya08/data-scientists-guide-to-summarization-dde46b30b4c3>
- [26] CHIMERA0. Pysummarization 1.1.4: Automatic Summarization Library: pysummarization. *PyPI* [online]. 2019, Aug 4, 2019 [cit. 2020-04-05]. Dostupné z: <https://pypi.org/project/pysummarization/>
- [27] MILLER, Derek. Bert-extractive-summarizer 0.4.1: Bert Extractive Summarizer. *PyPI* [online]. 2020, Apr 4, 2020 [cit. 2020-04-10]. Dostupné z: <https://pypi.org/project/bert-extractive-summarizer/>
- [28] DEVLIN, Jacob, Ming-Wei CHANG, Kenton LEE a Kristina TOUTANOVA. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [online]. 24 May 2019 [cit. 2020-04-10]. Dostupné z: <https://arxiv.org/pdf/1810.04805.pdf>
- [29] MILLER, Derek. *Leveraging BERT for Extractive Text Summarization on Lectures* [online]. 7 Jun 2019 [cit. 2020-04-12]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1906/1906.04165.pdf>

- [30] KABIR, Rasel, Shaily KABIR a Shamiul AMIN. *Isolating Informative Blocks from Large Web Pages Using HTML Tag Priority Assignment Based Approach* [online]. 2015, 4(3), 61-72 [cit. 2020-03-05]. DOI: 10.14810/ecij.2015.4305. ISSN 22015957. Dostupné z: <http://www.wireilla.com/engg/ecij/papers/4315ecij05.pdf>
- [31] MORRISON, J. Paul. Flow-based Programming. *Flow-based Programming* [online]. 2019, 15 Aug 2019 [cit. 2020-03-08]. Dostupné z: <https://jpaulm.github.io/fbp/>
- [32] GANESAN, Kavita. ROUGE 2.0. *GitHub* [online]. San Francisco, 10 Apr 2020 [cit. 2020-05-08]. Dostupné z: <https://github.com/kavgan/ROUGE-2.0>
- [33] LIN, Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out* [online]. Barcelona, Spain: Association for Computational Linguistics, 2004, 74–81 [cit. 2020-05-08]. Dostupné z: <https://www.aclweb.org/anthology/W04-1013.pdf>
- [34] *SQuAD2.0: The Stanford Question Answering Dataset* [online]. Stanford, USA: The Stanford NLP Group, 2020 [cit. 2020-05-11]. Dostupné z: <https://rajpurkar.github.io/SQuAD-explorer/>

Seznam obrázků

<i>Obrázek 1: Vizuální schéma metodiky</i>	28
<i>Obrázek 2: Vývojové prostředí Node-RED</i>	36
<i>Obrázek 3: Podoba pilotní aplikace v Node-RED</i>	38
<i>Obrázek 4: Grafické uživatelské rozhraní aplikace a Node-RED Dashboard uzly</i>	38
<i>Obrázek 5: Node-RED uzel Cheerio a jeho nastavení</i>	39
<i>Obrázek 6: Node-RED uzel html-to-text</i>	39
<i>Obrázek 7: Node-RED uzly Cheerio a Function</i>	40
<i>Obrázek 8: Node-RED uzel bert_summarizer</i>	41
<i>Obrázek 10: Klasifikace obrázků</i>	42
<i>Obrázek 11: Node-RED uzly pro syntézu řeči</i>	43

Seznam tabulek

<i>Tabulka 1: Nastavení priority HTML elementům podle důležitosti.....</i>	<i>30</i>
<i>Tabulka 2: Výsledky komparace metrikami ROUGE</i>	<i>50</i>

Seznam ukázek zdrojového kódu

<i>Ukázka 1: Výstup extrakce klíčových slov Microsoft Text Analytics API</i>	22
<i>Ukázka 2: Výstup extrakce klíčových slov Watson NLU API</i>	23
<i>Ukázka 3: Výstup extrakce klíčových slov Amazon Comprehend</i>	23
<i>Ukázka 4: Zkrácený výstup Microsoft Computer Vision API</i>	26
<i>Ukázka 5: Příklad výstupu metodiky</i>	28
<i>Ukázka 6: Nastavení priority prvkům H2</i>	39
<i>Ukázka 7: Kód programu bert_summarizer.py</i>	40
<i>Ukázka 8: Výběr textu pro analýzu klíčových slov</i>	41
<i>Ukázka 9: Porovnání klíčových slov z různých služeb</i>	42
<i>Ukázka 10: Výpočet odhadovaného času čtení</i>	42
<i>Ukázka 11: Vytvoření výstupního popisu webové stránky</i>	43

Seznam příloh

Příloha 1: Rešerše odečítačů obrazovky	63
Příloha 2: Srovnání služeb	64
Příloha 3: Testovací scénáře	67
Příloha 4: Dotazník	79
Příloha 5: Vyhodnocení dotazníkového šetření	80
Příloha 6: Zdrojové kódy aplikace	83

Příloha 1: Rešerše odečítačů obrazovky

	Výrobce	Platforma	Režim čtení	Navigace mezi elementy	WAI-ARIA	Porozumění obrázkům	OCR	Kontrola textu	Výtah stránky	Podpora Brailského řádku	Licence
NVDA	NV Access	Windows	✓	✓	✓	alt	✓			✓	Free, open-source
JAWS	Freedom Scientific	Windows	✓	✓	✓	alt	✓	✓ ¹	✓ ²	✓	Komerční
VoiceOver	Apple	Mac OS X, iOS	✓	✓	✓	alt	—		✓ ³	✓	Free, Komerční
Microsoft Narrator	Microsoft	Windows	✓	✓	✓	alt	—			✓	Free, Komerční
ZoomText	Freedom Scientific	Windows	✓	✓	✓	alt	✓			✓	Komerční
System Access	Serotek	Windows	✓	✓	✓	alt	—			✓	Komerční
ReadSpeaker	HOYA Group	Web-based	✓	—	—	—	—			—	Free, Komerční
BrowseAloud	Texthelp, Ltd	Web-based	✓	—	—	—	—			—	Komerční
Orca	GNOME	Linux	✓	✓	✓	alt	—			✓	Free, open-source
Dolphin Computer Access	Dolphin Computer Access	Windows	✓	✓	✓	alt	✓			✓	Komerční
WebAnywhere	University of Washington	Web-based	✓	✓	✓	alt	—			—	Free open-source
Window-Eyes	GW Micro	Windows	✓	✓	✓	alt	—			—	Prodej byl ukončen

¹ Oprava chyb během psaní textu. Kontroluje chybějící závorky, zdvojené mezery, interpunkci a nechtěné formátování textu.

² Skim Reading Summary – výtah dokumentu obsahující první řádky nebo první věty jednotlivých odstavců.

³ Řekne, kolik je na webové stránce nadpisů, odkazů, tabulek, oblastí, regionů a dalších prvků.

Příloha 2: Srovnání služeb

	IBM Natural Language Understanding	Microsoft Text Analytics	Google Natural Language	Amazon Comprehend
cena	30 000 položek/měsíc zdarma	5 000 položek/měsíc zdarma	5 000 položek/měsíc zdarma	až 10 mil. položek \$0,0001/položka
jazyk	✓	✓	✓	✓
sémantika	✓	—	—	—
podmět	✓	—	—	—
přísudek	✓	—	—	—
předmět	✓	—	—	—
vztahy	✓	—	—	—
lokace	✓	—	—	—
čas	✓	—	—	—
příbuznost	✓	—	—	—
keywords	✓	✓	—	✓
sentiment	nad slovy	nad dokumentem	nad dokumentem i nad slovy	nad dokumentem
emoce	✓	—	—	—
entity	✓	—	✓	✓
osoba	✓	—	✓	✓
zařízení	✓	—	—	—
lokace	✓	—	✓	✓
množství	✓	—	—	✓
datum	?	—	?	✓
událost	?	—	✓	—
dílo	—	—	✓	—
jiné	?	—	✓	—
kategorie	✓	—	✓	—
keywords	920-years-old; ancient oak trees date; Blenheim Palace; single area; unique group of oaks;ancestral home of Sir Winston Churchill; world wars; Middle Ages; people	single area; ancient oak trees date; world wars; single type of soil; unique group of oaks; civil war; Normans; ancestral home of Sir Winston Churchill; Henry; years-old	—	the trees; a unique group; the trees; the estate; the world; another piece; soil; the ancestral home; Britain; the son

	IBM Visual Recognition	Microsoft Vision	Google Vision	Amazon Rekognition
cena	1 000 obrázků/měsíc zdarma	0-1 milion transakcí \$1,50/1 000 transakcí	1 000 obrázků/měsíc zdarma	5 000 obrázků/měsíc zdarma
		1-5 milionů transakcí \$1/1 000 transakcí	1001-5 milionů obrázků \$1,50	5001-1 milion obrázků/měsíc \$1
popisek	—	✓	—	—
keywords	✓	✓	✓	—
pravděpodobnost	✓	✓	✓	—
topicality	—	—	✓	—
kategorie	—	✓	—	—
explicitní obsah	—	✓	—	—
barvy	—	✓	—	—
typ obrázku	—	✓	—	—
obličej	✓	✓	✓	—
logo	—	—	✓	—
OCR	—	—	✓	—
popisek	—	a dog standing on top of a grass covered field	—	—
keywords	norwegian elkhound dog; hound dog; dog; domestic animal; animal; ash grey color	grass; dog; outdoor; carnivore; animal; field; canis; mammal; sled dog; sakhalin husky	Dog; Mammal; Vertebrate; Canidae; Czechoslovakian wolfdog; Dog breed; Saarloos wolfdog; Wolfdog; Carnivore; Northern inuit dog	—

	IBM Text to Speech	Microsoft Text to Speech	Google Text-to-Speech	Amazon Polly
počet jazyků	13	48	31	91
cena	10 000 znaků/měsíc zdarma	5 milionů znaků/měsíc zdarma	4 miliony znaků zdarma	5 milionů znaků/měsíc zdarma
	\$0,02/dalších tisíc znaků	\$4/každý další 1 milion znaků	\$4/každý další 1 milion znaků	\$4/každý další 1 milion znaků
		\$16/každý další 1 milion znaků při použití AI syntézy		\$16/každý další 1 milion znaků při použití AI syntézy

Příloha 3: Testovací scénáře

Tester: Roman

Na jakém zařízení testujete? Mobil? Počítač?

Počítač

Jaký máte operační systém?

Windows 10 1909 64Bit, MacOS 10.15.4

Pokud používáte odečítač obrazovky, napište, v jakém odečítači aplikaci testujete (např. JAWS, NVDA, Apple Voiceover aj.).

Windows Narrator, Apple VoiceOver

Testovací scénář

Zadání:

Cíl:

Ověření použitelnosti a přístupnosti aplikace.

Vstupní podmínka:

Ve webovém prohlížeči si nejprve otevřete web na adrese 13.66.186.54:1880/app

Scénář – Krok 1:

Chcete si zjistit základní informace o webové stránce prague-info.info. Proved'te analýzu webové stránky <https://prague-info.info>. Analýza stránky může trvat až 1 minutu.

Mějte prosím strpení. Zkontrolujte, zda hlasová syntéza řekla tyto údaje o stránce:

- Title
- Keywords
- Summary

- Estimated reading time
- Images count
- Images

Odpověď zapište k otázkám v části Výsledek testu. Část Výsledek testu naleznete na další straně tohoto dokumentu.

Scénář – Krok 2:

Nyní pomocí aplikace proveďte testování dalších dvou webových stránek. Adresy jsou uvedené za tímto textem. U každé stránky by aplikace měla přečíst tyto údaje:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpověď zapište k otázkám v části Výsledek testu.

Další webové stránky pro testování:

<https://www.historic-uk.com/HistoryUK/HistoryofBritain/RMS-Titanic-the-unsinkable-ship/>

<https://www.praguemorning.cz/eurostat-unemployment-in-the-czech-republic-in-january-2020-was-2/>

Výsledek testu

Byl naplněn očekávaný výsledek?

Spíše ne, chyběla hlasová odezva.

Byl interpretován výstup analýzy u všech webů?

Ano

Narazil jste na nějaké problémy s přístupností aplikace?

Velmi dlouhé načítání bez jakékoli odezvy, která by uživatele informovala, že se něco

děje nebo proces nějak selhal.

Došlo během práce k nějaké chybě nebo zamrznutí aplikace?

Napište odpověď:

- Deklarovaná hlasová odezva nefungovala ani na jedné testovací konfiguraci.
- Často se vyskytovala chyba "Error: connect ECONNREFUSED 127.0.0.1:5051"

Poznámky:

Kdyby aplikace byla distribuována jako doplněk do prohlížeče, bylo by možné uvažovat o využití takové funkce na stránce, na které by si ji uživatel vyžádal. Další podmínkou k tomu, aby taková funkce měla smysl, by byla rychlá odezva (do 10 vteřin), aby čekání na souhrn netrvalo déle, než prozkoumání obsahu stránky prostřednictvím konvenčního odečítače obrazovky.

KONEC SCÉNÁŘE

Tester: Jan

Na jakém zařízení testujete? Mobil? Počítač?

Desktop

Jaký máte operační systém?

Windows 10

Pokud používáte odečítač obrazovky, napište, v jakém odečítači aplikaci testujete (např. JAWS, NVDA, Apple Voiceover aj.). Pokud žádný odečítač obrazovky nepoužíváte, odpověď nevyplňujte.

NVDA

Testovací scénář

Zadání:

Cíl:

Ověření použitelnosti a přístupnosti aplikace.

Vstupní podmínka:

Ve webovém prohlížeči si nejprve otevřete web na adrese 13.66.186.54:1880/app

Scénář – Krok 1:

Chcete si zjistit základní informace o webové stránce prague-info.info. Proved'te analýzu webové stránky <https://prague-info.info>. Analýza stránky může trvat až 1 minutu. Mějte prosím strpení. Zkontrolujte, zda hlasová syntéza řekla tyto údaje o stránce:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpořed' zapiřite k otázkám v řásti Výsledek testu. řást Výsledek testu naleznete na další straně tohoto dokumentu.

Scénář – Krok 2:

Nyní pomocí aplikace proveřte testování dalších dvou webových stránek. Adresy jsou uvedené za tímto textem. U každé stránky by aplikace měla přeříst tyto údaje:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpořed' zapiřite k otázkám v řásti Výsledek testu.

Další webové stránky pro testování:

<https://www.historic-uk.com/HistoryUK/HistoryofBritain/RMS-Titanic-the-unsinkable-ship/>

<https://www.praguemorning.cz/eurostat-unemployment-in-the-czech-republic-in-january-2020-was-2/>

Výsledek testu

Byl naplněn očekávaný výsledek? Stačí odpořed': ano/ne.

U první stránky ano.

U druhé se vyskytl problém u pole Souhrn: „Summary: Error: socket hang up : <http://localhost:5051/>“

U třetí též problém u pole Souhrn: „Error: connect ECONNREFUSED 127.0.0.1:5051 : <http://localhost:5051/>“

Byl interpretován výstup analýzy u všech webů?

řádná hlasová syntéza nic neřetla ani po stisku tlačítka Znovu přeříst popis stránky.

Narazil jste na nějaké problémy s přístupností aplikace?

Vzhledem k tomu, že analýza trvá poměrně dlouho, měl by být uživatel odečítače nějak informován o tom, co se děje. Nyní, když se zahájí analýza, je odečítač potichu a uživatel neví, že analýza začala. Obdobně, když analýza skončila, nedozvěděl jsem se to od odečítače. Možná, že tyto informace by nějak měla sdělovat hlasová syntéza, ta mi ale nefungovala. Pokud by tyto informace měl říkat odečítač, stačilo by, kdyby někde na stránce byl prvek `Začala analýza` a posléze by se změnil na `Analýza dokončena`. Díky atributu `role="status"` by odečítač příslušnou informaci vždy automaticky ohlásil uživateli.

Došlo během práce k nějaké chybě nebo zamrznutí aplikace?

Stávalo se mi, že při opětovném otevření adresy <http://13.66.186.54:1880/app> bylo políčko předvyplněno minule zadaným URL a byly k němu zobrazeny výsledky. Očekával bych, že se stránka načte spíš ve výchozím prázdném stavu.

Poznámky:

Myslím, že by pomohlo uživatelům blíže vysvětlit účel aplikace.

KONEC SCÉNÁŘE

Tester: Karel

Na jakém zařízení testujete? Mobil? Počítač?

Počítač

Jaký máte operační systém?

MacOS Catalina. Preferován internetový prohlížeč Safari, když něco nejde, zkušeno v Google Chrome.

Pokud používáte odečítač obrazovky, napište, v jakém odečítači aplikaci testujete (např. JAWS, NVDA, Apple Voiceover aj.). Pokud žádný odečítač obrazovky nepoužíváte, odpověď nevyplňujte.

VoiceOver

Testovací scénář

Zadání:

Cíl:

Ověření použitelnosti a přístupnosti aplikace.

Vstupní podmínka:

Ve webovém prohlížeči si nejprve otevřete web na adrese 13.66.186.54:1880/app

Scénář – Krok 1:

Chcete si zjistit základní informace o webové stránce prague-info.info. Proved'te analýzu webové stránky <https://prague-info.info>. Analýza stránky může trvat až 1 minutu.

Mějte prosím strpení. Zkontrolujte, zda hlasová syntéza řekla tyto údaje o stránce:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count

- Images

Odpověď zapište k otázkám v části Výsledek testu. Část Výsledek testu naleznete na další straně tohoto dokumentu.

Scénář – Krok 2:

Nyní pomocí aplikace proveďte testování dalších dvou webových stránek. Adresy jsou uvedené za tímto textem. U každé stránky by aplikace měla přečíst tyto údaje:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpověď zapište k otázkám v části Výsledek testu.

Další webové stránky pro testování:

<https://www.historic-uk.com/HistoryUK/HistoryofBritain/RMS-Titanic-the-unsinkable-ship/>

<https://www.praguemorning.cz/eurostat-unemployment-in-the-czech-republic-in-january-2020-was-2/>

Výsledek testu

Byl naplněn očekávaný výsledek? **Stačí odpověď: ano/ne.**

Napište odpověď:

Zčásti ano. Text se vypsal, syntéza nic neříkala.

Byl interpretován výstup analýzy u všech webů?

Napište odpověď:

V Safari ne, hlasově to neříkalo vůbec nic. V Chrome hlasová syntéza funguje. Tam mi to přečetlo úplně všechno.

Narazil jste na nějaké problémy s přístupností aplikace?

Napište odpověď:

Přístupnost je naprosto v pohodě. Všechny prvky jsou bez problémů dosažitelné odečítačem.

Došlo během práce k nějaké chybě nebo zamrznutí aplikace?

Napište odpověď:

Zamrznutí vyloženě ne, ale připadlo mi že, trvalo dlouho, než to něco řeklo.

Poznámky:

Napište Vaše poznámky k průběhu testování, pokud chcete něco zmínit.

S odečítačem je to všechno ok. Všechno to vyčte hlasová syntéza mého odečítače.

Napadá mě, že by bylo super, kdyby to fungovalo jako čtečka obsahu. Že by to četlo články na vyžádání. Např. pro deník N, že by si to člověk zapnul jen tam, kde to chce. Ten hlas je krásný, dobře se poslouchá.

Nebo by bylo zajímavé, kdyby to fungovalo jako popis obrázků z kontextové nabídky. Že by člověk najel na obrázek a přečetlo by to popisek tam, kde by si to vyžádal.

Zajímavé by bylo, kdybych mohl do systému nahrát fotku, ve které je text a ta čtečka by mi to přečetla. Fungovalo by to jako OCR.

KONEC SCÉNÁŘE

Tester: Michal

Na jakém zařízení testujete? Mobil? Počítač?

Počítač

Jaký máte operační systém?

Windows

V jakém odečítači obrazovky aplikaci testujete (JAWS, NVDA, Apple Voiceover aj.)? Pokud žádný odečítač obrazovky nepoužíváte, odpověď nevyplňujte.

Testovací scénář

Zadání

Cíl:

Ověření použitelnosti a přístupnosti aplikace.

Vstupní podmínka:

Ve webovém prohlížeči si nejprve otevřete web na adrese 13.66.186.54:1880/app

Scénář – Krok 1:

Chcete si zjistit základní informace o webové stránce prague-info.info. Proved'te analýzu webové stránky <https://prague-info.info>. Analýza stránky může trvat až 1 minutu. Mějte prosím strpení. Zkontrolujte, zda hlasová syntéza řekla tyto údaje o stránce:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpověď zapište k otázkám v části Výsledek testu. Část Výsledek testu naleznete na další straně tohoto dokumentu.

Scénář – Krok 2:

Nyní pomocí aplikace analyzujte následující webové stránky. U každé stránky by aplikace měla přečíst tyto údaje:

- Title
- Keywords
- Summary
- Estimated reading time
- Images count
- Images

Odpověď zapište k otázkám v části Výsledek testu.

Další webové stránky pro testování:

<https://www.historic-uk.com/HistoryUK/HistoryofBritain/RMS-Titanic-the-unsinkable-ship/>

<https://www.praguemorning.cz/eurostat-unemployment-in-the-czech-republic-in-january-2020-was-2/>

Výsledek testu

Byl naplněn očekávaný výsledek? Stačí odpověď: ano/ne.

Ano, všechno proběhlo

Byl interpretován výstup analýzy u všech webů?

Ano

Narazil jste na nějaké problémy s přístupností aplikace?

Nevěděl jsem, jestli se stránka stále načítá, nebo se to seklo. Měl jsem tendenci to začít řešit. Potom to vypsalo informace.

Došlo během práce k nějaké chybě nebo zamrznutí aplikace?

Ne

Poznámky:

Aplikaci bych nevyužíval každý den, ale pro zkrácení nějakého hodně dlouhého článku by se to občas mohlo hodit.

KONEC SCÉNÁŘE

Příloha 4: Dotazník

Aplikace pro sumarizaci webových stránek - dotazník

Následující otázky se vztahují k tomuto popisu stránky. Zde je ukázka, jak může popis stránky vypadat.

Název: Nejlepší přítel člověka: mazlíček nebo zvíře? | OUPblog.

Klíčová slova: Psi, výzkum sekvence DNA, pravěcí lidé, výzkum chování, pravěcí psi, zvířata, osobní informace, ochrana soukromí, pravěký zájem, zvířecí chování

Shnutí: Chování psů, evoluce a poznání. Většina vědeckých otázek, odkazujících se na zvířata, zanedbává myšlenku individuality mezi zvířaty. Pravěcí psi možná pravěkému člověku neposkytovali žádný fyzický prospěch; existuje jen málo důkazů o jakémkoli dřívějším profitování z toho, že by měli psi kolem sebe. Protože psi reagují na komunikační akce lidí a jdou na příkázané místo, můžeme říci, že psi vykazují funkčně lidské chování.

Odhadovaný čas čtení: 4 minut.

Počet obrázků: 2 obrázky.

Obrázky: pes dívající se do kamery; foto psa zblízka.

Používal byste testovanou aplikaci pravidelně? *

Ano

Ne

Ohodnoťte, jak Vám vyhovuje popis webové stránky (obsah, podoba, délka apod.). *

Popis mi vyhovuje.

Popis je dobrý, ale vyžaduje změny.

Popis mi nevyhovuje.

Ohodnoťte délku celkového popisu stránky. Popis je podle Vás: *

Velmi krátký.

Krátký, ale dostačující.

Optimální.

Dlouhý, ale ještě snesitelný.

Zbytečně dlouhý.

Popis stránky obsahuje deset klíčových slov, která shrnují obsah stránky.

Ohodnoťte množství klíčových slov. Klíčových slov je podle Vás: *

Velmi málo.

Málo, ale stačí.

Optimální množství.

Mnoho, ale ještě je to snesitelné.

Přilíší mnoho.

Další otázka se vztahuje k popiskům obrázků. Zde je pár alternativních textů k obrázkům, které aplikace vygenerovala.

Lev sedící v poli.

Skupina fotbalistů pózující pro fotku.

Skupina lidí sedících u stolu před obchodem.

Hrad s lodí ve vodě.

Cedule zblízka.

Wolfgang Wahlster oblečený v saku a kravatě smějící se a dívající se do kamery.

Ohodnoťte, jak Vám vyhovují popisky jednotlivých obrázků. Popisky obrázků jsou podle Vás: *

Krátké

Optimální

Dlouhé

Niže jsou uvedené dva popisky webové stránky. Další otázka se bude ptát, který z nich Vám více vyhovuje.

První popisek:

Možná nikdy nezjistíme, co přímělo pravěcí lidé dovolit vlkům zasahovat do jejich životů, ale je jasné, že pes je stále považován za sociálního partnera; společníka poskytujícího emocionální a sociální podporu svým majitelům. Tvrdí se, že aby se psi stali členy společenství lidí, museli získat schopnost porozumět lidskému chování. Psi reagují na akce lidí a jdou na určené místo, můžeme říci, že psi vykazují funkčně lidské chování. Většina psů žijících v lidských domovech získává toto chování spontánně bez potřeby specifického výcviku a jejich chování je podobné chování u 1,5 letých kojenců. Ve srovnání s vlky, kteří by se mohli toto gesto také naučit v době, kdy jsou dospělí, se psi zdají být lépe připraveni na základě své specifické historie domestikace, což ukazuje na vyšší zájem o lidské chování a komunikační aktivity.

Druhý popisek:

Název: Nejlepší přítel člověka: mazlíček nebo zvíře? | OUPblog.

Klíčová slova: Psi, výzkum sekvence DNA, pravěcí lidé, výzkum chování, pravěcí psi, zvířata, osobní informace, ochrana soukromí, pravěký zájem, zvířecí chování

Shnutí: Chování psů, evoluce a poznání. Většina vědeckých otázek, odkazujících se na zvířata, zanedbává myšlenku individuality mezi zvířaty. Pravěcí psi možná pravěkému člověku neposkytovali žádný fyzický prospěch; existuje jen málo důkazů o jakémkoli dřívějším profitování z toho, že by měli psi kolem sebe. Protože psi reagují na komunikační akce lidí a jdou na příkázané místo, můžeme říci, že psi vykazují funkčně lidské chování.

Odhadovaný čas čtení: 4 minut.

Počet obrázků: 2 obrázky.

Obrázky: pes dívající se do kamery; foto psa zblízka.

Který z těchto dvou popisků Vám více vyhovuje? *

Více mi vyhovuje první popisek.

Více mi vyhovuje druhý popisek.

Niže jsou uvedené dva popisky jiné webové stránky. Další otázka se bude ptát, který z nich Vám více vyhovuje.

První popisek:

Praha Info: Online průvodce Prahou

Domů

Praha je jedním z nejnávštěvovanějších měst v Evropě.

Do Prahy přichází extrémně mnoho turistů a každý rok jich je čím dál tím víc.

I přesto Praha není drahá, i když ceny na Starém Městě se stále více přizpůsobují bohatým turistům ze zahraničí.

Proč bych měl jet do Prahy?

Existuje mnoho důvodů, proč jet do Prahy.

Mnoho turistů také přichází do Prahy za nákupy.

Mnoho lidí také přijíždí do města kvůli nočnímu životu.

Mladé turisty více přitahují tradiční kavárny a restaurace.

Pro většinu turistů v Praze je výlet do Prahy zaměřen na návštěvu velkých památek Zlatého města.

Jaké památky v Praze musím navštívit?

Významné památky jsou hlavně ve středu na obou stranách řeky Vltavy.

Mezi největší památky patří Karlův most a židovská čtvrť se synagogami a židovský hřbitov.

Praha Info

Staroměstské náměstí s radnicí

Staroměstské náměstí (centrum města)

Centrální náměstí Starého Města je považováno za jedno z nejkrásnějších náměstí v Evropě.

Okolo

Slavné hodiny na radnici

Pražský orloj, jak se jim také říká, byly postaveny v roce 1410 a jsou technologickým mistrovským dílem.

Karlův most

Karlův most

Pojád se na něm něco děje.

Pražský hrad

Pražský hrad je vedle Staroměstského náměstí a Karlova mostu třetí nejvýznamnější památkou v Praze.

Pražský hrad je největší hrad na světě.

Pražský hrad

Ačkoli byl už několikrát zásadně přestavěn.

Václavské náměstí

Toto náměstí je moderním srdcem Prahy.

Václavské náměstí

Na 800 metrů dlouhém náměstí v Praze najdete téměř všechno, co člověk může v centru moderního velkoměsta najít.

Prašná brána

Další dominantou a významnou památkou Prahy je Prašná brána ze 16. století na náměstí Republiky.

Starý židovský hřbitov

Starý

Nákupní centrum Palladium

Samozřejmě se můžeme dohadovat o tom, zda by se moderní nákupní centrum mělo zařadit mezi

prohlídky památek Prahy.

Praha Info

Druhý popisek:

Název: Praha Info: Online průvodce Prahou.

Klíčová slova: Pražský hrad, Karlův most, známé Václavské náměstí, Prašná brána.

Shnutí: Praha je jedním z nejnávštěvovanějších měst v Evropě. Některá místa v Novém městě jsou také velmi oblíbené turisty. Pražský hrad je vedle Staroměstského náměstí a Karlova mostu třetí

nejvýznamnější památkou v Praze. Nákupní centrum Palladium. Samozřejmě se můžeme dohadovat o tom, zda by se moderní nákupní centrum mělo zařadit mezi prohlídky památek Prahy. Jedna věc je jistá,

je to jedna z turistů nejnávštěvovanějších budov města.

Odhadovaný čas čtení: 5 minut.

Počet obrázků: 7 obrázků.

Obrázky: most přes vodní hladinu, letecký pohled na město; velké hodiny připevněné ke stěně budovy;

most přes vodní hladinu s městem v pozadí; zámek s hodinami v horní části budovy; skupina lidí

chodících před budovou; velká věž s hodinami tyčící se nad městem.

Který z těchto dvou popisků Vám více vyhovuje?

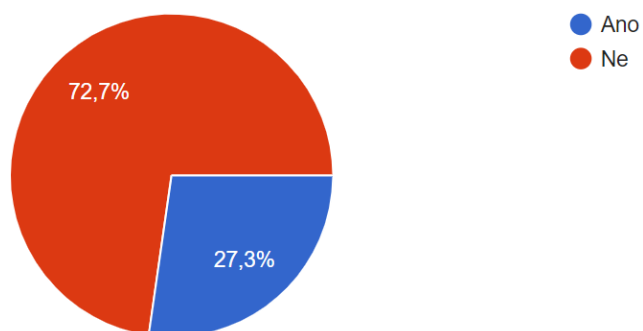
Více mi vyhovuje první popisek.

Více mi vyhovuje druhý popisek.

Příloha 5: Vyhodnocení dotazníkového šetření

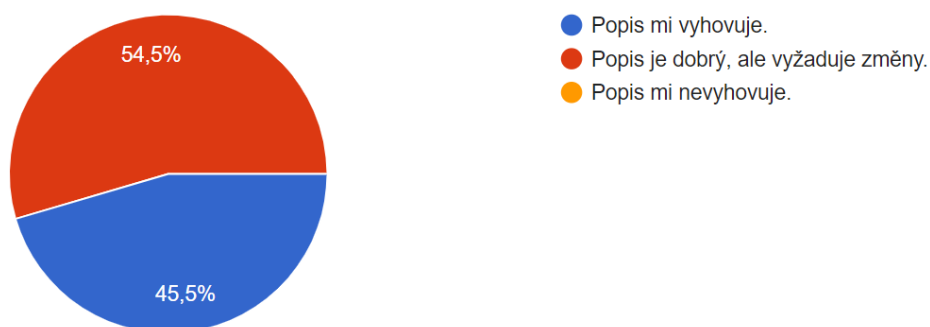
Používal byste testovanou aplikaci pravidelně?

11 odpovědí



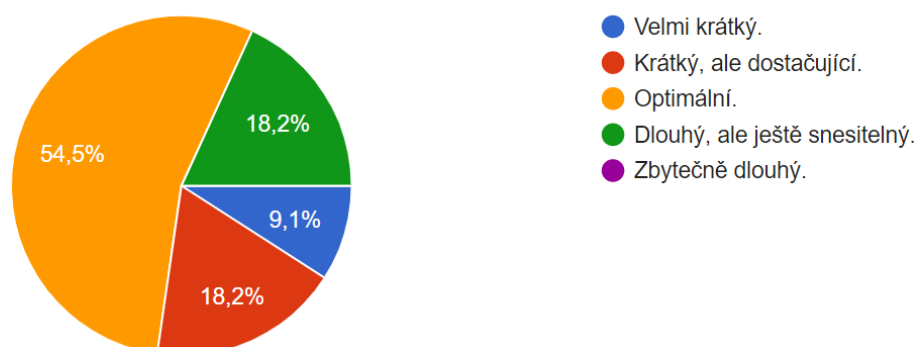
Ohodnoťte, jak Vám vyhovuje popis webové stránky (obsah, podoba, délka apod.).

11 odpovědí



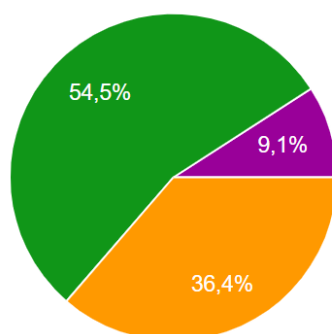
Ohodnoťte délku celkového popisu stránky. Popis je podle Vás:

11 odpovědí



Popis stránky obsahuje deset klíčových slov, která shrnují obsah stránky. Ohodnotte množství klíčových slov. Klíčových slov je podle Vás:

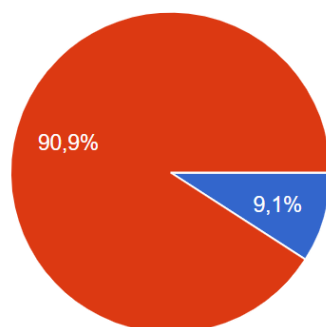
11 odpovědí



- Velmi málo.
- Málo, ale stačí.
- Optimální množství.
- Mnoho, ale ještě je to snesitelné.
- Příliš mnoho.

Ohodnotte, jak Vám vyhovují popisky jednotlivých obrázků. Popisky obrázků jsou podle Vás:

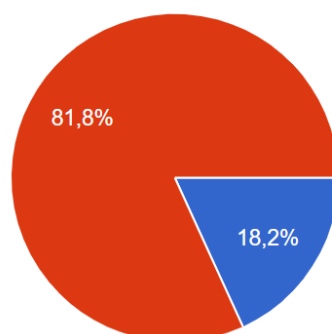
11 odpovědí



- Krátké
- Optimální
- Dlouhé

Který z těchto dvou popisek Vám více vyhovuje?

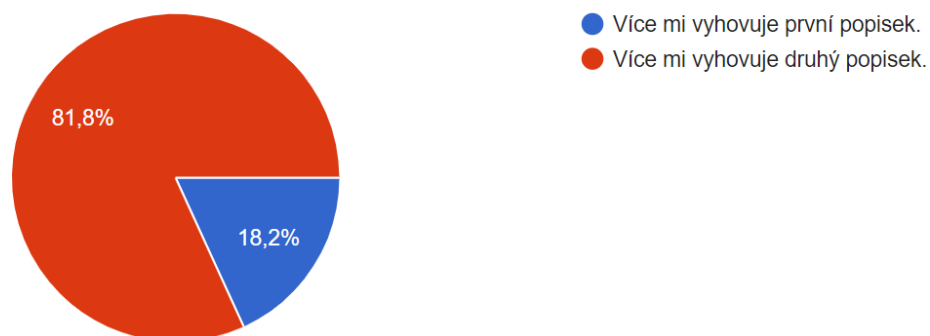
11 odpovědí



- Více mi vyhovuje první popisek.
- Více mi vyhovuje druhý popisek.

Který z těchto dvou popisů Vám více vyhovuje?

11 odpovědí



Příloha 6: Zdrojové kódy aplikace

Součástí diplomové práce jsou zdrojové kódy aplikace.

Adresářová struktura:

<code>./README.txt</code>	popis adresářové struktury
<code>./node-red</code>	adresář se zdrojovými soubory aplikace
<code>./bert</code>	adresář s programem <code>bert_summarizer.py</code>