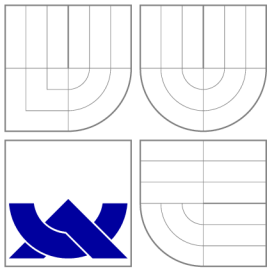
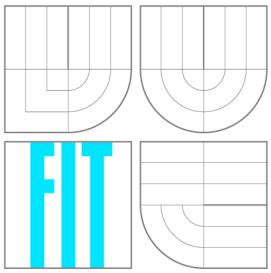


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÁ DETEKCE TÉMAT, SEGMENTACE A VIZUALIZACE ON-LINE KURZŮ

AUTOMATIC TOPIC DETECTION, SEGMENTATION AND VISUALIZATION OF ON-LINE COURSES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JOSEF ŘÍDKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. IGOR SZŐKE, Ph.D.

BRNO 2016

Zadání diplomové práce/18370/2015/xridky00

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Řídký Josef, Bc.**
Obor: Počítačová grafika a multimédia
Téma: **Automatická detekce témat, segmentace a vizualizace on-line kurzů**
Automatic Topic Detection, Segmentation and Visualization of On-Line Courses
Kategorie: Zpracování řeči a přirozeného jazyka

Pokyny:

1. Seznamte se se základy detekce témat z textových dokumentů (topic detection) a jejich segmentací.
2. Vytvořte množinu dokumentů (např. univerzitní přednášky) z veřejně dostupných zdrojů (YouTube). Tyto dokumenty převeďte do textu (přepis řeči na text) a aplikujte detekci a segmentaci témat.
3. Navrhněte a implementujte webovou službu, která vizualizuje podobné dokumenty nebo segmenty k právě přehrávanému záznamu.
4. Službu doplňte o možnost snadného přidávání dalších dokumentů a celý proces automatizujte.
5. Službu zveřejněte a otestujte na vhodné skupině uživatelů.
6. Diskutujte dosažené cíle a navrhněte směry dalšího vývoje.
7. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodu 3 a 4 ze zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této práce je vytvořit webovou aplikaci, která dokáže automaticky detekovat a segmentovat témata z videozáznamů on-line kurzů. Při následném přehrávání zpracovaných záznamů má uživateli vizualizovat záznamy z ostatních, tématicky shodných, on-line kurzů. Tento dokument obsahuje popis daného problému, představení použitých nástrojů, popis realizace, princip fungování a popis uživatelského prostředí výsledného systému.

Abstract

The aim of this work is to create a web application for automatic topic detection and segmentation of on-line courses. During playback of processed records, the application should be able to offer records from thematically consistent on-line courses. This document contains problem description, list of used instruments, description of implementation, the principle of operation and description of final user interface.

Klíčová slova

detekce témat, automatická detekce témat, Subspace Multinomial Model, PHP, Python, SpokenData.com, YouTube, on-line kurzy, cosine similarity

Keywords

topic detection, automatic topic detection, Subspace Multinomial Model, PHP, Python, SpokenData.com, YouTube, on-line courses, cosine similarity

Citace

ŘÍDKÝ, Josef. *Automatická detekce témat, segmentace a vizualizace on-line kurzů.* Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szőke Igor.

Automatická detekce témat, segmentace a vizualizace on-line kurzů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szőkeho, Ph.D. Další informace mi poskytli Santosh Kesiraju a Ing. Josef Žižka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Josef Řídký
23. května 2016

Poděkování

Chtěl bych poděkovat vedoucímu své diplomové práce panu Ing. Igoru Szőkemu, Ph.D. za jeho vedení a pomoc při tvorbě a zpracování této práce. Dále bych chtěl poděkovat panu Santoshi Kesirajuovi a panu Ing. Josefu Žižkovi za jejich trpělivou pomoc při řešení problémů, které se během tvorby této práce vyskytly. Také bych chtěl poděkovat panu Bc. Marku Sychrovi za jeho práci, na jejímž základě jsem mohl toto téma zpracovat a v neposlední řadě také paní Mgr. Marcele Marii Císařové za její ochotnou pomoc při korektuře textu této práce.

Obsah

Kapitola 1

Úvod

Žijeme v uspěchané době. Jsme denně doslova bombardováni neustálými požadavky na vysoký pracovní výkon, rychlost či přesnost, a to nejen při osobní komunikaci, ale, díky pokroku v oblasti výpočetní techniky, i při komunikaci probíhající převážně prostřednictvím elektronických prostředků. Dalo by se zjednodušeně říci, že chce-li být člověk v dnešní době úspěšný, musí buď oplývat velkým množstvím vědomostí, nebo musí vědět, kde má hledat informace k dosažení požadovaných výsledků.

Před celosvětovým rozmachem internetu byly hlavními zdroji informací knihy. V dnešní době je z knih samozřejmě také hojně čerpáno, ovšem ne každý člověk má tolik času, prostředků či elánu k tomu, aby si dohledal a přečetl knihy, ve kterých by mohl nalézt potřebné informace. S rozšířením internetu již člověk nehledá tak často v knihách, ale snaží se nalézat potřebné informace na internetu, a to jak v článcích či internetových encyklopediích, tak například i v dostupných videozáznamech. Dopátrat se textových informací na požadované téma z různých internetových zdrojů je v dnešní době vcelku bezproblémové. Problém však nastává v případě, kdy chceme na základě obsahového sdělení z videozáznamu nalézt další videozáznamy, ve kterých se hovoří na stejné či podobné téma. V takovém případě nám nezbývá než se probírat názvy a anotacemi jednotlivých videí a vyhledávat podobné záznamy.

Tato práce se zabývá automatickým vyhledáváním tématicky podobných částí ve videozáznamech z on-line kurzů, které jsou prezentovány v anglickém jazyce. Výsledná aplikace nabízí uživatelům při přehrávání jednotlivých videozáznamů další videozáznamy, které jsou obsahově podobné vzhledem k jednotlivým částem přehrávaného záznamu.

V tomto dokumentu je postupně v jednotlivých kapitolách představen cíl práce, nástin řešení a podmínky pro získávání audiovizuálních dat (kapitola 2). Následně je v kapitole 3 představeno schéma automatického procesu zpracování videozáznamu a jsou postupně popsány jeho jednotlivé části. Kapitola 4 obsahuje popis webové aplikace jak z vizuální, tak i programové stránky. V kapitole 5 jsou uvedeny provedené experimenty s daty. Kapitola 6 obsahuje výsledky testování. V závěru jsou zjištěné výsledky diskutovány a je uveden nástin dalšího vývoje.

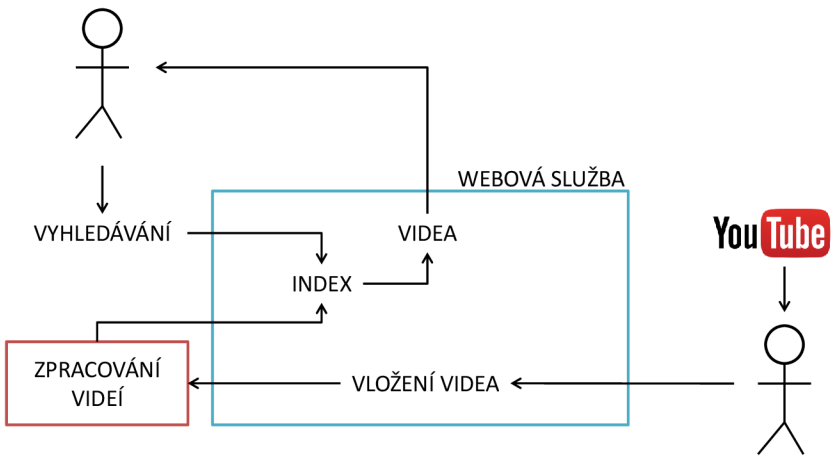
Kapitola 2

Popis problému

V této kapitole jsou představeny cíl práce, nástin jeho obecného řešení a stanovené podmínky pro získání audiovizuálních dat. Tyto informace poskytují ucelenou základní představu o této práci a jejich očekávaných výstupech.

2.1 Navržený koncept

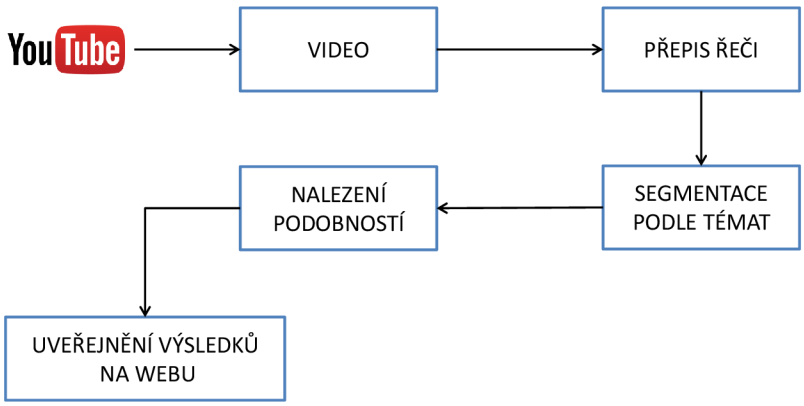
Cílem této práce je vytvořit webovou službu, která dokáže mezi videozáznamy vyhledávat tématicky podobné části a to na základě přepisů řeči z videozáznamů. Tato funkce by měla být co nejvíce automatizovaná. Další funkce webové služby, které by měly být uživatelům přístupné jsou nabídnutí tématicky podobných částí z jiných on-line kurzů při přehrávání videozáznamů ve vytvořené webové službě a také vyhledávání mezi videozáznamy a to nejen v názvech a abstraktech, ale i v samotném zvukovém záznamu. Ukázka zamýšleného použití služby uživateli je na obrázku 2.1.



Obrázek 2.1: Schéma případu užití

Ke splnění uvedeného návrhu je nutné se zamyslet nad postupem, jehož pomocí bude dosaženo požadovaných cílů. Na obrázku 2.2 je uvedeno obecné schéma zpracování videozáznamů a hledání podobností mezi nimi. Podobnost mezi videozáznamy by se poměřovala

na základě přepisu mluveného slova z jednotlivých videí a to následovně. Videozáznamy z on-line kurzů by byly získávány například ze služby YouTube. Pro jejich další zpracování by došlo k jejich stažení. Dále by z nich byl vytvořen přepis mluveného slova, který by byl rozdělen na tématicky konzistentní části. Na základě těchto tématicky konzistentních částí by pro každou z nich byl vytvořen příznakový vektor – histogram četností jednotlivých slov, která se v dané části vyskytují. Tyto vektory by byly následně porovnány s příznakovými vektory částí z ostatních videozáznamů ve webové službě. Pro každou detekovanou část ze všech videozáznamů ve webové službě by byla nalezena skupina nejpodobnějších částí na základě výsledků porovnání jejich příznakových vektorů. Takto nalezené podobné části by byly posléze vizualizované uživateli při přehrávání videozáznamů prostřednictvím webové aplikace a bylo by v nich možné i vyhledávat.



Obrázek 2.2: Koncept dosažení cíle práce

K dosažení uvedeného konceptu je uvažováno využití existujících technologií, které by měly usnadnit dosažení potřebného cíle. Konkrétně jimi bude zajištěno získání přepisu mluveného slova z videozáznamů (viz sekce 3.2.1), segmentace přepisů na jednotlivé tématicky konzistentní celky (viz sekce 3.2.2), redukce vektorů detekovaných částí (viz sekce 3.2.3) a vyhledávání ve zpracovaných datech (viz sekce 4.2.1).

Při tvorbě takovéto aplikace se nelze spoléhat na to, že komunita uživatelů aktivně dohledá záznamy z jednotlivých kurzů, které by měla vytvářená webová služba obsahovat. Musí to být právě autor, který se postará o úvodní naplnění potřebnými daty a dokáže, že jeho služba je pro ostatní užitečná. Ze zadání je určeno, že aplikace bude pracovat pouze se záznamy z kurzů, které byly přednášeny v anglickém jazyce.

Otázkou, kterou si je však zapotřebí zodpovědět, je, komu by měla být výsledná aplikace určená a co by měla těmto uživatelům poskytovat za informace. Odpověď na první část otázky je vcelku prostá. Aplikace by měla být určena všem, kteří si chtějí osvojit, prohloubit či osvěžit znalosti, v této chvíli zatím jen z oblasti informačních technologií, za pomoci videozáznamů z více volně dostupných on-line kurzů, které svým obsahem pojednávají o stejném, hledaném, tématu. Odpověď na druhou část otázky je poněkud obtížnější, a proto je jí věnována následující podkapitola 2.2.

2.2 Omezující podmínky pro data a srovnání služeb

Na počátku, při vzniku zadání této práce, bylo nutné určit, jaké typy videozáznamů mají být ve výsledné aplikaci k dispozici. Prvotním požadavkem bylo vytvořit aplikaci, která by sdružovala relevantní a důvěryhodné videozáznamy pojednávající o tématech z oblasti informačních technologií. Asi každého by hned napadlo shromáždit libovolné videozáznamy z internetu, pojednávající na téma informačních technologií. Mezi nejčastější typy záznamů by se tak řadily tzv. videoblogy, tutoriály a další obdobná, relativně krátká, videa. Problémů je zde však hned několik. Prvním je relevantnost informací v těchto záznamech. Nikdo nedokáže alespoň částečně zaručit, že všichni autoři takovýchto videoblogů a tutoriálů ve svých příspěvcích prezentují pravdivé, k tématu relevantní, informace. Hlavně z tohoto důvodu tedy nemohlo být o těchto typech videozáznamů při sběru dat uvažováno. Druhým problémem je délka samotného záznamu. Při krátké stopáži by nabídka videozáznamů pojednávajících o podobném tématu byla bezpředmětná. Cílem totiž není najít podobná videa k hlavnímu tématu ze záznamu, ale podobná videa k dílčím tématům ze záznamu a k těm nabídnout přehrání tématicky nejvíce podobných videozáznamů. Jelikož by takto krátký videozáznam s velkou pravděpodobností nepojednával o více jak jednom tématu, není žádoucí, aby se ve velké míře ve výsledné aplikaci objevoval, protože tato funkcionalita je dostatečně suplována za použití obyčejného textového vyhledávání v názvech a anotacích ostatních videozáznamů. V neposlední řadě je důležité určit, odkud budou jednotlivé videozáznamy přejímány. Existuje několik hlavních služeb, ze kterých je možné videozáznamy převzít. Zástupci, o kterých bylo pro získání videozáznamů uvažováno, jsou představeni v následujících podkapitolách.

2.2.1 Kaltura

Obrázek 2.3: Logo společnosti Kaltura

Prvním zástupcem ze služeb, odkud by se teoreticky mohly převzít potřebné videozáznamy, je Kaltura[1]. Jedná se o první světovou open source video platformu, která poskytuje jak komerční software a služby, které má ve správě právě Kaltura, tak i řešení pro publikování videa určené zdarma open source komunitě uživatelů[2]. Mezi zákazníky této služby patří např. americká univerzita ve Virginii. Tato služba však pro potřeby této práce není vhodná, protože sama o sobě neposkytuje zdarma záznamy z kurzů jednotlivých zákazníků, ale pouze jim prodává software a služby, které jsou jimi následně využívány. Není tudíž splněna podmínka volného přístupu k videozáznamu.

2.2.2 Vimeo

Další službou, odkud by bylo možné získat videozáznamy, je služba společnosti Vimeo[3]. V roce 2004 byla založena skupinou filmařů. Chtěli sdílet svoji kreativní práci a momenty ze svých osobních životů. V průběhu času tuto službu začalo využívat čím dál více stejně

[1] Stránky služby: `https://corp.kaltura.com`
Logo převzato z `http://corp.kaltura.com/sites/default/files/Homepage%20New%20Logo.png`.

[2] `http://corp.kaltura.com/About-Kaltura`

[3] Stránky služby: `https://vimeo.com`
Logo převzato z archivu `https://f.vimeocdn.com/downloads/vimeo_logos.zip`.

smýšlejících lidí, kteří pomohli vystavět podporující komunitu, jež měla široký rozsah zájmů. Dnes službu Vimeo využívají miliony lidí z celého světa[4].

Obrázek 2.4: Logo společnosti Vimeo

Tato služba se na první pohled může jevit jako vhodná. Problémem však zde není volný přístup k videozáznamům, ale absence potřebných typů videozáznamů. Tato služba obsahuje v oblasti informačních technologií převážně videozáznamy, které nemají za cíl vzdělávat formou kurzů, ale představit zajímavosti či postupy pro dosažení určitého cíle. Jedná se tedy spíš o tutoriály a specifickou formu videoblogů, což již bylo v sekci 2.2 uvedeno jako nežádoucí.

2.2.3 Khan Academy

Obrázek 2.5: Logo společnosti Khan Academy

Třetí službou je Khan Academy[5]. Tato služba má jako své poslání zprostředkovat bezplatné vzdělání každému a kdekoliv. Je cílena na uživatele každého věku. Na svých internetových stránkách nabízí praktická cvičení, instruktážní videa a zacílený učební plán, který má studenty nadchnout pro učení mimo školu, a to při respektování jejich vlastních studijních temp[6]. Tato služba obsahuje velké množství instruktážních videí, která jsou přístupná pomocí služby YouTube (viz 2.2.4). Jedinou nevýhodou, proč ani tato služba není zcela vyhovující pro účely této práce je fakt, že jednotlivá videa jsou krátká a většinou pojednávají jen o jednom tématu.

2.2.4 YouTube

Obrázek 2.6: Logo společnosti YouTube

Posledním zástupcem je služba YouTube[7]. Tato služba byla spuštěna v květnu 2005 a umožňuje miliardám lidí objevovat a sdílet originální videa. Funguje jako platforma pro distribuci originálního obsahu malých i velkých autorů a inzerentů. Služba patří společnosti Google[8]. Tato služba jako jediná vyhovuje požadavkům, které byly v úvodu představeny. Služba obsahuje volně dostupná videa, která jsou dostatečně dlouhá a ve svém obsahu pojednávají o více než jednom tématu.

Z uvedeného srovnání služeb vyplynulo, že hlavním, a v této chvíli jediným, zdrojem dat je služba YouTube (viz 2.2.4). Jako jediná totiž obsahuje právě takové druhy videozáznamů, které jsou pro tuto práci požadovány. Hlavními zdroji, odkud byly videozáznamy ze zvolené služby čerpány, byly účty (resp. jejich video kanály – dále jen kanály) světově uznávaných univerzit, které volně poskytují záznamy ze svých kurzů

[4] `https://vimeo.com/about`

[5] Stránky služby: `https://www.khanacademy.org`
Logo převzato z `https://www.khanacademy.org/images/khan-logo-vertical-transparent.png`.

[6] `https://www.khanacademy.org/about`

[7] Stránky služby: `https://www.youtube.com`
Logo převzato z `https://www.youtube.com/yt/brand/media/image/YouTube-logo-full_color.png`.

[8] `https://www.youtube.com/yt/about/cs/`

v anglickém jazyce. Hlavními zdroji záznamů byly kanály Kalifornské univerzity v Berkeley, dále kanál Massachusettského technologického institutu (dále jen MIT), konkrétně jeho projektu MIT OpenCourseWare, a kanály Stanfordské univerzity a Harvardské univerzity. Při získávání záznamů byly přebírány, dohledávány či určovány následující informace:

- název přednášky
- abstrakt
- URL adresa k videozáznamu na YouTube
- jméno přednášejícího
- místo působení přednášejícího
- kategorie, do které přednáška tématicky spadá

Videozáznamy byly rozděleny do čtrnácti kategorií (viz příloha B.1). Snahou bylo vytvořit pro dané kurzy takové kategorie, které by reflektovaly tématické okruhy předmětů, jež jsou vyučovány na zdejší fakultě.

Tato kapitola obsahuje obecné schéma řešení popsaného úkolu a představuje i omezující podmínky na data, se kterými má výsledná služba pracovat. V následující kapitole je obecné řešení představené v této kapitole přiblíženo a je popsána jeho implementace.

Kapitola 3

Proces automatického zpracování videozáznamů

Tato kapitola má za úkol přiblížit představený koncept dosažení cíle této práce (viz obrázek 2.2) a osvětlit, co se v jednotlivých částech vytvořeného procesu pro zpracování videozáznamů odehrává. Jsou zde popsány všechny části vytvořeného procesu, jimiž musí projít každý záznam z přednášky, aby byl správně zaindexovaný a také prostřednictvím webové části služby zpřístupněný veřejnosti. Před upřesněním jednotlivých částí procesu zpracování videozáznamů je však zapotřebí představit jednotlivé nástroje a struktury, ze kterých je tento proces vytvořen či se kterým-i přímo spolupracuje.

3.1 Použité nástroje

Úkolem tohoto procesu je převážně pracovat s textovými daty. Z toho důvodu bylo přistoupeno k implementaci jednotlivých částí automatického procesu zpracování videozáznamů pomocí skriptů napsaných v jazyce Python.

Python

Python je moderní, objektově orientovaný programovací jazyk, v němž je možné vytvářet samostatné aplikace, rychlé skripty i prototypy komplexních aplikací. Je zdarma, má volně šiřitelný zdrojový kód a snadno se používá. Skripty v Pythonu jsou přenositelné mezi většinou platforem. Jsou stejně funkční pod Windows, Linuxem, Unixem i systémem MacOS. [3] Python navrhl v roce 1991 Guido van Rossum. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmata, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního. Python je vyvíjen jako open source projekt. Ve většině distribucí systému Linux je Python součástí základní instalace. [10] V této práci je pracováno s Pythonem verze 2.7 a verze 3.4.

MySQL databáze

Pro uložení a následnou práci s metadaty, která jsou podstatnou součástí této práce, byla vybrána MySQL databáze. Automatický proces zpracování videozáznamů sice nejvíce pracuje s textovými daty, ovšem výsledky segmentace a vypočítaných podobností jsou zapotřebí zpřístupnit i webové části vytvářené služby. Z toho důvodu je tedy ve vytvářeném procesu nutné komunikovat i s databází. V databázi je používáno pět tabulek.

První tabulkou je tabulka `category` (viz tabulka 3.1), ve které je uložen seznam všech podporovaných kategorií, do kterých lze získané videozáznamy rozdělit. Primárním klíčem této tabulky je sloupec `id`.

Sloupec	Datový typ	Význam
id	int (11)	jednoznačný číselný identifikátor kategorie
name	varchar (255)	název kategorie

Tabulka 3.1: Popis jednotlivých sloupců databázové tabulky category

Druhou tabulkou je tabulka `status` (viz tabulka 3.2), ve které je uložen seznam všech možných fází automatického procesu, v nichž se může libovolný záznam nacházet. Seznam všech stavů je uveden v příloze B.2. Primárním klíčem této tabulky je sloupec `id`.

Sloupec	Datový typ	Význam
id	int (11)	jednoznačný číselný identifikátor fáze
name	varchar (255)	název fáze

Tabulka 3.2: Popis jednotlivých sloupců databázové tabulky status

Třetí tabulka má název `records` (viz tabulka 3.3) a je využívána pro uložení získaných informací o videozáznamech z kurzů. Primárním klíčem této tabulky je sloupec `id`.

Sloupec	Datový typ	Význam
id	int (11)	jednoznačný číselný identifikátor videozáznamu
name	varchar (500)	název videozáznamu
abstract	text	popis videozáznamu
url	varchar (255)	URL adresa k videozáznamu na serveru YouTube
presenter	varchar (255)	jméno přednášejícího
place	varchar (255)	firma/univerzita, kde přednášející působí
category	int (11)	číselný identifikátor kategorie, do které je videozáznam zařazen
status	smallint (6)	číselný identifikátor, který označuje, v jaké fázi automatického procesu se daný záznam nachází
spokenId	int (11)	číselný identifikátor záznamu ve službě SpokenData

Tabulka 3.3: Popis jednotlivých sloupců databázové tabulky records

Čtvrtá tabulka nese název `segmentation` (viz tabulka 3.4) a slouží k uložení jednotlivých segmentů, které byly ve videozáznamu detekovány, včetně jejich hranic. Primárním klíčem tabulky je sloupec `id`.

Poslední tabulka se jmenuje `linking` (viz tabulka 3.5) a slouží k uložení vypočítaných vazeb mezi jednotlivými detekovanými segmenty. V procesu automatického zpracování videozáznamů je převážně pracováno s tabulkami `segmentation` a `linking`.

Sloupec	Datový typ	Význam
id	int (11)	jednoznačný číselný identifikátor segmentu
record_id	int (11)	jednoznačný číselný identifikátor videozáznamu, ze kterého pochází daný segment
begin	int (11)	čas počátku segmentu v sekundách
end	int (11)	čas konce segmentu v sekundách
thumbnail	int (1)	příznak generování náhledového obrázku

Tabulka 3.4: Popis jednotlivých sloupců databázové tabulky segmentation

Sloupec	Datový typ	Význam
record1	int (11)	jednoznačný číselný identifikátor segmentu z prvního videozáznamu
record2	int (11)	jednoznačný číselný identifikátor segmentu z druhého videozáznamu
confidence	double	míra podobnosti obou segmentů

Tabulka 3.5: Popis jednotlivých sloupců databázové tabulky linking

FFmpeg

FFmpeg je předním multimediálním frameworkem, který je schopný dekódovat, zakódovat, překódovat, mixovat, streamovat, filtrovat a přehrávat téměř jakýkoli multimediální soubor. Podporuje zpracování od nejstarších až po aktuálně používané multimediální formáty bez ohledu na tvůrce daného multimediálního formátu. FFmpeg je multiplatformní framework nezávislý na architektuře či konfiguraci zvoleného operačního systému.[1] V této práci je FFmpeg využíván pro vytváření náhledů z videozáznamů (viz sekce 3.9).

3.2 Použité technologie

Pro usnadnění dosažení cíle vytvořeného procesu bylo pro některé jeho části využito existujících technologií. Pro získávání přepisů mluveného slova z videozáznamů bylo přistoupeno k outsourcování pomocí některé z dostupných cloudových služeb (Google Cloud Speech, Speechmatics, SpokenData apod.). Pro účely segmentace bylo využito výstupu bakalářské práce Marka Sychry [9], která se zabývala problémem automatického hledání vazeb mezi částmi audiovizuálních dokumentů (viz 3.2.2). Další použitou technologií v této části práce je Subspace Multinomial Model, jehož implementace je převzata od Santoshe Kesirajua [2] (viz 3.2.3).

3.2.1 Přepis mluveného slova

Jednou z podstatných částí při řešení této práce bylo získání automatických přepisů mluveného slova z videozáznamů. Řešením tohoto problému bylo propojení vytvářeného procesu s cloudovou službou, která daný problém řeší. V tomto případě bylo přistoupeno k využití služby SpokenData, která díky svému API poskytuje automatizované získání přepisu

[1] Převzato z `https://ffmpeg.org/about.html`

z požadovaných videozáznamů. Obecně je však možné využít libovolnou cloudovou službu zabývající se automatickým přepisem mluveného slova. Zvolená služba je dostupná na adrese `https://spokendata.com`. Komunikaci se službou SpokenData zajišťuje webová část aplikace.

O službě

Služba SpokenData je součástí portfolia služeb zabývajících se zpracováním audiovizuálních dat, které nabízí společnost ReplayWell, s.r.o. Loga obou společností jsou uvedena na obrázku 3.1.

(a) SpokenData

(b) ReplayWell, s.r.o.

Obrázek 3.1: Loga společností – převzata z webových stránek

Služba SpokenData se zabývá zajištěním přepisu mluveného slova či titulků z multimediálních dat. Služba dokáže v současné době vytvořit přepisy ze záznamů, kde řečník hovoří anglicky, rusky, čínsky, španělsky, česky nebo slovensky. Dále je kromě přepisů schopna detekovat hlasovou aktivitu v nahrávce a rozdělit nahrávku na segmenty podle toho, který řečník kdy mluvil.

Uživatel má možnost po registraci a přihlášení nahrát multimediální soubor[2], popřípadě jen zadat URL adresu k multimediálnímu souboru na jehož základě dojde k získání přepisu. Možnost získat přepis mluveného slova na základě URL adresy k multimediálnímu souboru (např. odkaz na videozáznam na YouTube) je pro účely této práce optimální.

API

Aplikačně programové rozhraní poskytované službou SpokenData umožňuje vývojářům integrovat tuto službu do desktopových, mobilních či webových aplikací. Pro komunikaci s API služby SpokenData je nejprve zapotřebí se do služby prostřednictvím webového portálu přihlásit a následně si nechat vygenerovat jednoznačný identifikátor uživatele (číslo) a token (řetězec), který opravňuje daného uživatele využívat API služby SpokenData.

Každé volání API funkce služby SpokenData je prováděno pomocí URL adresy. Ta se skládá z URL k API, identifikátoru uživatele, API tokenu a názvu požadované API funkce. Jednoznačný identifikátor uživatele a API token jsou hodnoty, které má každý uživatel jiné. URL k API je však pro všechny stejná a to `https://spokendata.com/api`. Každý požadavek je tedy reprezentován URL adresou ve formátu `https://spokendata.com/api/IDUživatele/APItokenUživatele/APIfunkce`. V tabulce 3.6 jsou uvedeny ty API funkce, které byly v této práci použity, a to včetně jejich popisu. Kompletní seznam všech API funkcí je dostupný na adrese `https://spokendata.com/api-for-developers`

Produkované výstupy

Po dokončení automatického přepisu mluveného slova je možné výsledný přepis získat v šesti různých formátech. Konkrétně ve formátech HTML, SRT, TRS, TXT, WebVTT a XML.

[2] podporované formáty wav, mp3, mp4, ac3, avi, mpg, wmv, flv a mkv

API funkce	Popis
recording/RID	vrátí data o nahrávce s identifikátorem RID, pro získání dat ze dvou a více nahrávek v jednom dotazu je zapotřebí zadat identifikátory nahrávek oddělené čárkou **recording/RID,RID,RID ...**
recording/RID/subtitles.xml	vrátí přepis mluvené řeči ve formátu XML
recording/add?url=...& language=...&annotatorId=	přidá novou nahrávku k účtu uživatele, vrátí zprávu o stavu s přiděleným ID záznamu **url** - URL adresa k záznamu **language** - jeden z english, russian, chinese-ma, spanish-us, czech, czech-medicine, czech-broadcastnews, slovak **annotatorId** - id přiřazeného anotátora (pokud není, ponechá se prázdné)

Tabulka 3.6: Seznam použitých API funkcí služby SpokenData

V této práci byl jako nejvhodnější pro automatické zpracování zvolen formát XML.

3.2.2 Automatické hledání vazeb mezi částmi audiovizuálních dokumentů

Tato diplomová práce vychází z výsledků bakalářské práce jejímž autorem je Marek Sychra [9]. Ten se ve své práci se dostatečně podrobně věnuje problému automatického hledání vazeb mezi částmi audiovizuálních dokumentů. Z toho důvodu je v následujících odstavcích uvedeno jen obecné shrnutí, které čerpá ze zmíněné práce.

Popis problému hledání tématu

Výzkum hledání tématu[3] začal velmi expandovat kolem roku 1997. V tomto období vznikl program Transligual Information Detection, Extraction and Summarization (dále jen TIDES), který se podílel velkou měrou na úspěchu v tomto odvětví. Iniciativa TIDES rozdělila TDT problém do pěti kategorií:

- **Story sementation** - Jedná se o první část TDT úkolu. Cílem je rozdělit delší text na menší celky, které obsahují právě jedno hlavní téma.
- **First story detection** - Druhá část postupně prochází výsledky segmentace a zjišťuje, které textové segmenty (články) pojednávají o tématu, jež se v předchozích segmentech nevyskytovalo. V této fázi se tedy vyhledávají unikátní témata.
- **Topic tracking** - Ve třetí části se na základě dostatečného počtu článků, které byly přiřazeny k danému tématu, vyhledávají další články spadající do stejného tématu. Toto přiřazování by se mělo dít nezávisle na tématu, tedy bez využití informace o přiřazení daného článku k jiným tématům.
- **Topic detection** - Ve čtvrté části dochází ke shlukování jednotlivých článků dle odhadnutých témat. Jednotlivé shluky budou reprezentovat různá témata.

[3]Topic Detection and Tracking - dále jen TDT

- **Link detection** - V páté části je cílem určit, zda jsou dva články tématicky stejné.

Z výše uvedeného se dá odvodit, že nejpodstatnější pro tuto práci jsou části *Story segmentation*, *First story detection* a *Link detection*. Pro účely této práce není závadou, že se nebude zjišťovat přesné téma dané části přepisu mluveného slova z videozáznamu. Stačí totiž k dané části textu nalézt další části z jiných přepisů, které jsou jí nejvíce podobné. Na základě této podobnosti pak bude uživatelům navrhováno shlédnutí dalších videozáznamů.

3.2.3 Subspace Multinomial Model

Subspace Multinomial Model (dále jen SMM) je logaritmicko-lineární model, který může být použitý pro učení spojitých nízkodimenzionálních reprezentací pro diskrétní data. SMM je spolu se svými variantami používán pro verifikaci řečníka, která je založena na rozpoznávání jazyka pomocí prozodických faktorů[4] a fonotaktiky[5]. V práci Santoshe Kesirajua, který je autorem SMM použitého v této práci, je představena nová varianta SMM, která zavádí řídkost. Ukazuje, že tato vlastnost může být použita pro učení modelu na takových dokumentech, které jsou užitečné při řešení problému identifikace nebo klasifikace a clusteringu témat. [2] Pan Kesiraju ve své práci čerpal ze článku [7]. Matematicky lze SMM popsat následovně.

Nechť D je počet dokumentů na vstupu a d je index reprezentující jednotlivé dokumenty, V je celkový počet slov ve slovníku a i je index daného slova ve slovníku. Na každé slovo v dokumentu může být nahlíženo jako na nezávislou událost (bag-of-words) generovanou specifickou multinomiální distribucí dokumentu. A dále nechť c_d odznačuje vektor vyskytujících se slov v dokumentu d a ϕ_{di} jsou parametry dokumentu specifikující multinomiální distribuci (pravděpodobnosti slov), pak korespondující logaritmická pravděpodobnost může být vyjádřena pomocí rovnice 3.1.

$$log(P(c_d|\phi_d)) = \sum_{i=1}^{V} c_{di} log(\phi_{di}), \tag{3.1}$$

kde $\sum_i \phi_{di} = 1, \phi_{di} \geq 0$ a c_{di} je počet slov i v dokumentu d. Funkci obecného SMM lze následně popsat pomocí rovnice 3.2.

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{i=1}^{V} c_{di} log(\phi_{di}) \tag{3.2}$$

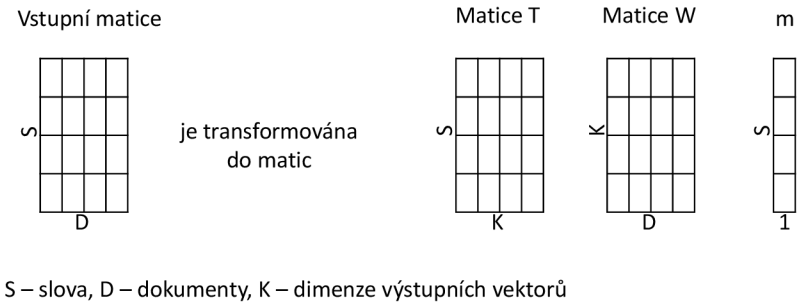
SMM od Santoshe Kesirajua zavádějící řídkost lze na základě rovnice 3.2 popsat pomocí rovnice 3.3. V této variantě jsou bázová matice T, která reprezentuje vektorový podprostor v oblasti logaritmické pravděpodobnosti a matice w, která reprezentuje i-vektory dokumentů regularizovány pomocí l_1 (pro matici T) a l_2 (pro matici w) norem. Tím jsou potlačeny případné problémy při trénování SMM a je tak zajištěna možnost zpracování řídkých vstupních vektorů.

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{i=1}^{V} \Big(c_{di} log(\phi_{di}) - \gamma \|t_i\|_1 - \frac{\lambda}{2} \|w_d\|_2 \Big), \tag{3.3}$$

[4] tempo mluvy, tón hlasu, melodie, pauzy, hlasitost, rytmus řeči [4]

[5] kombinatorika fonémů při tvoření slabičných vzorců [5]

kde γ je regularizační koeficient pro t_i, kde $t \in T$ a λ je regularizační koeficient pro w_d. Matematický popis a výpočty byly převzaty z [2]. Na obrázku 3.2 je znázorněno vytvoření SMM, který je reprezentován pomocí matic T, W a m ze vstupní matice, která svými sloupci reprezentuje jednotlivé detekované segmenty a řádky jednotlivá slova, která se v segmentech vyskytovala. Hodnotou je pak počet výskytů daného slova v daném segmentu.



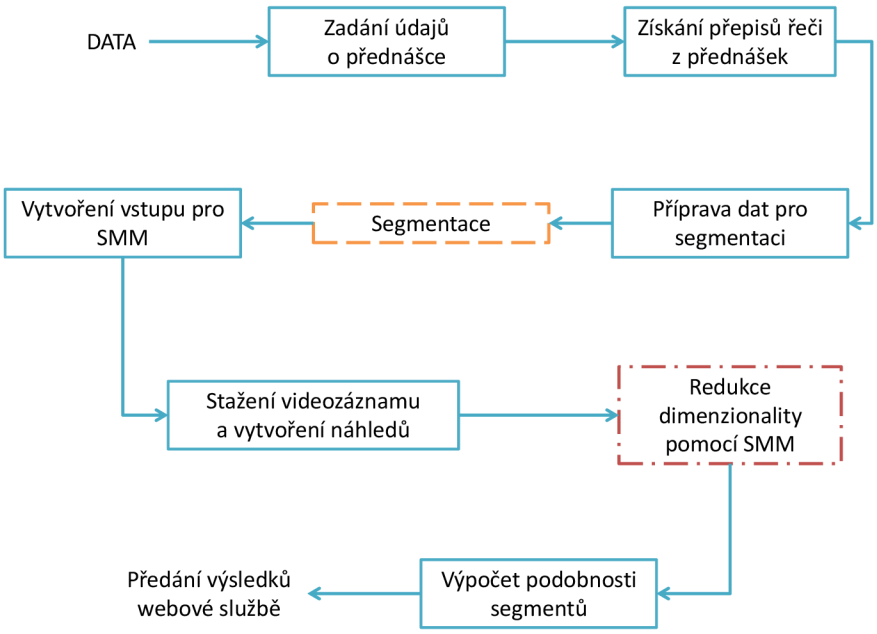
Obrázek 3.2: Schéma SMM. Matice T reprezentuje vektorový podprostor v oblasti logaritmické pravděpodobnosti, matice W reprezentuje i-vektory dokumentů a m je vektor posunu.

Důvodem použití SMM v této práci je jeho schopnost redukce dimenzionality vstupních dat, které jsou ve formě řídké matice. Je-li tedy na vstupu například matice vektorů s dimenzí 33 000, dokáže vytvořit výstupní matici redukovaných vektorů, kde každý má například dimenzi 300. Procesu trénování SMM je věnována sekce 5.2 a extrakci je věnována sekce 3.10.

3.3 Popis procesu automatického zpracování dat

Hlavním podkladem této kapitoly je níže uvedené schéma vytvořeného procesu (obrázek 3.3). Na schématu jsou přítomné tři druhy bloků. Blok, který je vyznačen oranžovým čárkovaným okrajem označuje tu část procesu, jehož provedení obstarává skript od Marka Sychry. Blok s červeným čerchovaným okrajem označuje tu část automatického procesu, jehož provedení zajišťuje třída od Santoshe Kesirajua. Všechny ostatní bloky jsou předmětem řešení této diplomové práce.

Hlavním cílem této práce je vytvořit automatický proces zpracování videozáznamů. Schéma uvedené na obrázku 3.3 však popisuje poloautomatický proces zpracování. Pro hladký průběh celého procesu je nutná interakce s uživatelem a to v prvních dvou uvedených blocích (popsány v sekci 3.4 a 3.5). Důvodem této odchylky je fakt, že videozáznamy zadané uživateli do systému musí být při první interakci nejdříve zkontrolovány, zda obsahují pravdivé a úplné informace o nově přidaném záznamu a zda daný videozáznam tématicky spadá do oblasti informačních technologií. Při druhé interakci je nutné zajistit vytvoření a stažení přepisu mluveného slova u požadovaných videozáznamů. Posledním úkonem, který musí uživatel před započetím plně automatické části procesu provést, je výběr těch záznamů, které mají dokončený přepis řeči a mají být po zpracování zpřístupněny ve vytvořené webové službě. Pro snazší orientaci, ve které fázi procesu se daný záznam nalézá, je každému záznamu přiřazen stav ze seznamu možných stavů, který je uložený v databázové tabulce `status`. Její obsah je uveden v příloze B.2.



Obrázek 3.3: Schéma procesu automatického zpracování přednášek

3.4 Zadání údajů o přednášce

Vstupním blokem celého procesu zpracování nemůže být nic jiného, než získání potřebných informací, na jejichž základě lze následně provést požadovanou akci. V sekci 5.1 je nastíněno, jakým způsobem byla data do systému dodávána v době, kdy ještě nebylo implementováno žádné webové rozhraní výsledné služby. Po jeho implementaci se toho mnoho nezměnilo. I nadále jsou údaje o přednáškách a přednášejících zadávány prostřednictvím webové aplikace (výsledné rozhraní viz kapitola 4). Libovolný uživatel má možnost přidat do systému libovolný záznam. Stejně jako před vytvořením webového rozhraní, kdy byly potřebné informace zadávány prostřednictvím podpůrné stránky (viz sekce 4.3.1), je nutné i nyní ručně zadat informace o videozáznamu (název, URL, abstrakt, kategorie) a o prezentujícím (jméno a pracoviště).

V průběhu vývoje této práce byly buď při semestrální obhajobě či při prezentaci této práce na studentské konferenci Excel@FIT (viz sekce 6.1) kladeny dotazy na možnost případného automatického či poloautomatického získávání informací požadovaných po uživateli například přímo ze serveru YouTube. Bohužel tato jinak velmi přínosná myšlenka není v tomto případě proveditelná. Informace, které jsou po uživateli požadovány totiž v drtivé většině případů nejsou ve službě YouTube zadány. Velmi často je název přednášky, její tématické zaměření či jméno samotného přednášejícího nutné dohledat. Buď tak, že je nutné si přehrát videozáznam a s trochou štěstí doufat, že budou potřebné informace při přehrávání vysloveny nebo alespoň vyobrazeny a nebo potřebné informace dohledávat přímo na stránkách organizátorů kurzů (převážně na stránkách univerzit). V případě, že by název videozáznamu a abstrakt vždy obsahovaly potřebné informace, bylo by možné získávat tyto informace bez další interakce uživatele (například jen zadáním URL adresy k videozáznamu).

3.5 Získání přepisů řeči z přednášek

Druhým blokem v poloautomatické části procesu je získání přepisů mluveného slova z videozáznamů uložených v databázi. K tomuto účelu je využito služby SpokenData, jejíž hlavní výhodou je, že dokáže získat přepis mluveného slova i jen za pomoci URL odkazu k multimediálnímu souboru. Výměnu informací mezi popisovaným procesem a službou SpokenData zajišťuje webová část aplikace.

Na server SpokenData.com je prostřednictvím jím poskytovaného API odeslána URL adresa videozáznamu na YouTube. Zároveň je v dané službě vytvořena nová úloha pro zajištění přepisu mluveného slova v poskytnutém videozáznamu. Jako potvrzení je navráceno číslo vytvořené úlohy, které jednoznačně identifikuje daný videozáznam ve službě SpokenData. Toto číslo je uloženo do databázové tabulky `records` k patřičnému videozáznamu a je klíčovou informací při další komunikaci se službou SpokenData.

Po obdržení emailu s potvrzením o dokončení přepisu u zvolených videozáznamů je možné za využití webové části vytvářené aplikace stáhnout a uložit dokončený přepis. Na server SpokenData.com je prostřednictvím jím poskytovaného API zasláno číslo, které jednoznačně identifikuje daný záznam ve službě SpokenData. Na základě tohoto čísla je následně vyžádán přepis mluveného slova a to ve formě XML souboru. Příklad XML dokumentu s přepisem ze služby SpokenData je uvedena v příloze C.

Každý XML soubor získaný ze serveru SpokenData.com následně prochází transformací, díky které je zajišťována následná jednodušší práce s přepisem při jeho automatickém zpracování. Transformaci zajišťuje stále webová část aplikace. Pro práci s XML soubory a řetězci v jazyce PHP je využíváno funkcí `simplexml_load_file()` a `simplexml_load_string()`.

Při zpracování XML souborů jsou extrahovány informace o času začátku a konce jednotlivých částí přepisu ve videozáznamu a samotný text přepisu dané části videozáznamu. Tyto informace jsou následně uloženy do souborů .ts. Ukázka souboru .ts je v příloze D. Soubory .ts obsahují celý přepis videozáznamu. Jednotlivé části z přepisu jsou uvozeny sekvencí `# NNNN`, kde `NNNN` je čas konce dané části v milisekundách. Po úvodním řádku jsou vypsána na samostatné řádky jednotlivá slova z dané části přepisu. Každá část je zakončena znakem tečka, který je také na samostatném řádku a odděluje tak od sebe jednotlivé části přepisu. Z důvodu případných problémů při hledání tématicky konzistentních částí při segmentaci se při ukládání přepisu do souboru .ts odstraňují interpunkční znaky.

3.6 Příprava dat pro segmentaci

Poslední blok, který je závislý na interakci od uživatele a zároveň první blok již plně automatického procesu zpracování videozáznamů má za úkol připravit z uživatelem vybraných záznamů vstup pro segmentační skript.

První část tohoto úkolu je ještě zpracována webovou částí aplikace. Ta na základě uživatelem definovaných záznamů, které mají být na vstupu automatického procesu, vytvoří soubor, jehož obsahem jsou relativní cesty[6] k souborům s přepisem. Po jeho vytvoření je webovou službou spuštěn hlavní obslužný skript `autorun.py`. Tento skript zajišťuje spouštění všech ostatních skriptů, které jsou součástí vytvořeného automatického procesu, v požadovaném pořadí s potřebnými parametry. Od této chvíle lze tedy hovořit o plně automatizovaném procesu zpracování dat.

[6] vzhledem k umístění skriptu webové služby

Skript `autorun.py`[7] očekává jako parametr cestu k souboru, který obsahuje relativní cesty k souborům s přepisy řeči z videozáznamů, jež mají být zpracovány. Volitelně ještě podporuje za názvem souboru se vstupními daty i přepínač `-t`, který udává, že má být při zpracování postupováno takovým způsobem, aby bylo možné ve výsledku spustit trénování SMM (více v sekci 5.2).

Prvním krokem v řídícím skriptu je spuštění transformačního skriptu `transform.py`. Ten jako parametr očekává cestu k souboru, jehož obsahem jsou cesty k přepisům zpracovávaných záznamů. Volitelně může být zadán i parametr `-t` určující lehce odlišný způsob práce s daty (v návaznosti na sekci 5.2). Tento skript postupně prochází záznamy v zadaném vstupním souboru a pro každý přepis uložený v souboru .ts vytvoří dva soubory. V prvním souboru, který má koncovku `.word`, jsou na jednotlivých řádcích uvedena slova z uzlů `<segment>` z .ts souborů[8] (ukázka v příloze E.1). Ve druhém souboru, který má koncovku `.meta` (ukázka v příloze E.2), jsou uložena metadata o souboru `.word`. Na každém řádku je uveden počet znaků od začátku souboru do daného řádku a čas, kdy se text z daného řádku má vyskytovat v původním videozáznamu.

Při tvorbě souborů `.word` je vytvářen či aktualizován slovník všech slov, která se objevila v libovolném přepisu. Indexem do slovníku je slovo, které se vyskytlo v přepisu a hodnotou je jeho pořadí, v jakém bylo do slovníku přidáno. Výsledný soubor se slovníkem je za využití knihovny *pickle* uložen do souboru `dict-train.pkl`. V případě zadání přepínače `-t` je vytvořený slovník uložen i do souboru `dict.pkl`, který je následně využíván ve zbytku automatického procesu. Pokud parametr `-t` nebyl zadán, pracuje se ve zbytku procesu s dříve vytvořenou verzí slovníku, která je uložena v souboru `dict.pkl`. Vytvoření tohoto slovníku je velmi podstatné při vytváření vstupu pro SMM (viz sekce 3.8).

3.7 Segmentace

Po vytvoření souborů .word a .meta pro všechny uživatelem zvolené záznamy je každý z nich postupně segmentován za použití skriptu `segmentation.py`, jehož autorem je Marek Sychra. Tento skript nejprve rozdělí text na spoustu drobných částí (např. 50 až 90 slov dlouhých). Následně jsou tyto části mezi sebou porovnávány za použití metody *cosine similarity* pomocí vzorce 3.4. Více informací o této metodě je uvedeno v sekci 5.3.2.

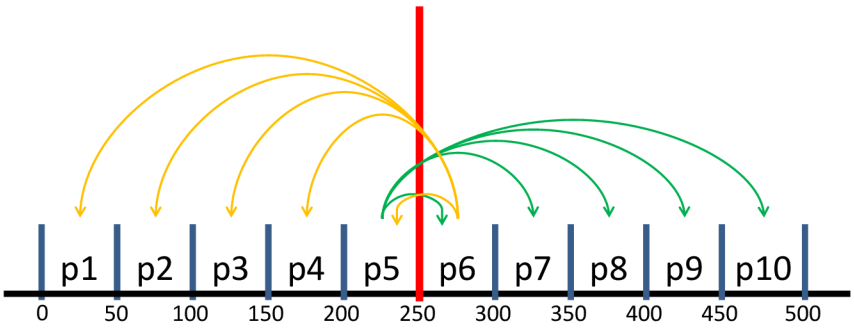
$$\cos\theta = \frac{A \cdot B}{|A| \cdot |B|} \tag{3.4}$$

Skript využívá při hledání hranic témat techniku posuvného okna, které zachovává větší délku základního bloku[9] a zároveň je tím zajištěna detekce případné hranice témat uvnitř bloku. Na obrázku 3.4 je uvedeno schéma hledání hranice témat, kde hranice (čára mezi bloky p5 a p6 se nevyhledává pouze porovnáváním dvou sousedních bloků. Blok *p5* se porovnává s až pěti dalšími částmi za hranicí, stejně tak i část *p6* akorát v opačném směru. Výsledný součet se zprůměruje a uloží i s indexem předělu. Výsledkem této části procesu jsou textové soubory, které obsahují vždy jeden segment přepisu odpovídající jednomu tématu. Výstupem tohoto skriptu jsou textové soubory, které obsahují všechny detekované tématické celky z přepisu.

[7] spouštěn interpretem ve verzi 2 z důvodu použití třídy SMM, která je psána pro Python verze 2. Ostatní skripty jsou spouštěny interpretem ve verzi 3.

[8] slova z jedné části přepisu jsou vždy na jednom řádku

[9] blok ve smyslu velikosti posuvného okna, které může být až 5x90 slov dlouhé v závislosti na zdroji dat



Obrázek 3.4: Schéma principu segmentace

Pro ukázku úspěšnosti segmentačního skriptu slouží graf na obrázku 3.5. Tento graf znázorňuje výsledek z testování segmentačního skriptu prováděné jeho autorem. V rámci testování segmentace si připravil soubory se zhruba 6 000 slovy[10], které měly imitovat asi hodinový záznam řeči. V těchto souborech si nejdříve zjistil hranice témat a poté se je snažil za využití svého segmentačního skriptu vyhledat.

Segmentační skript byl doplněn o následující funkcionalitu, která zajišťuje vytvoření relevantních a pro další běh procesu potřebných informací. První úpravou, která byla v segmentačním skriptu provedena, je vytvoření více souborů .part z jednoho vstupního souboru s přepisem (ukázka v příloze F). V původní verzi byly totiž všechny detekované tématicky konzistentní části videozáznamu uloženy v jednom souboru .part. To bylo upraveno takovým způsobem, aby každý detekovaný tématický celek z přepisu byl uložen do samostatného souboru.

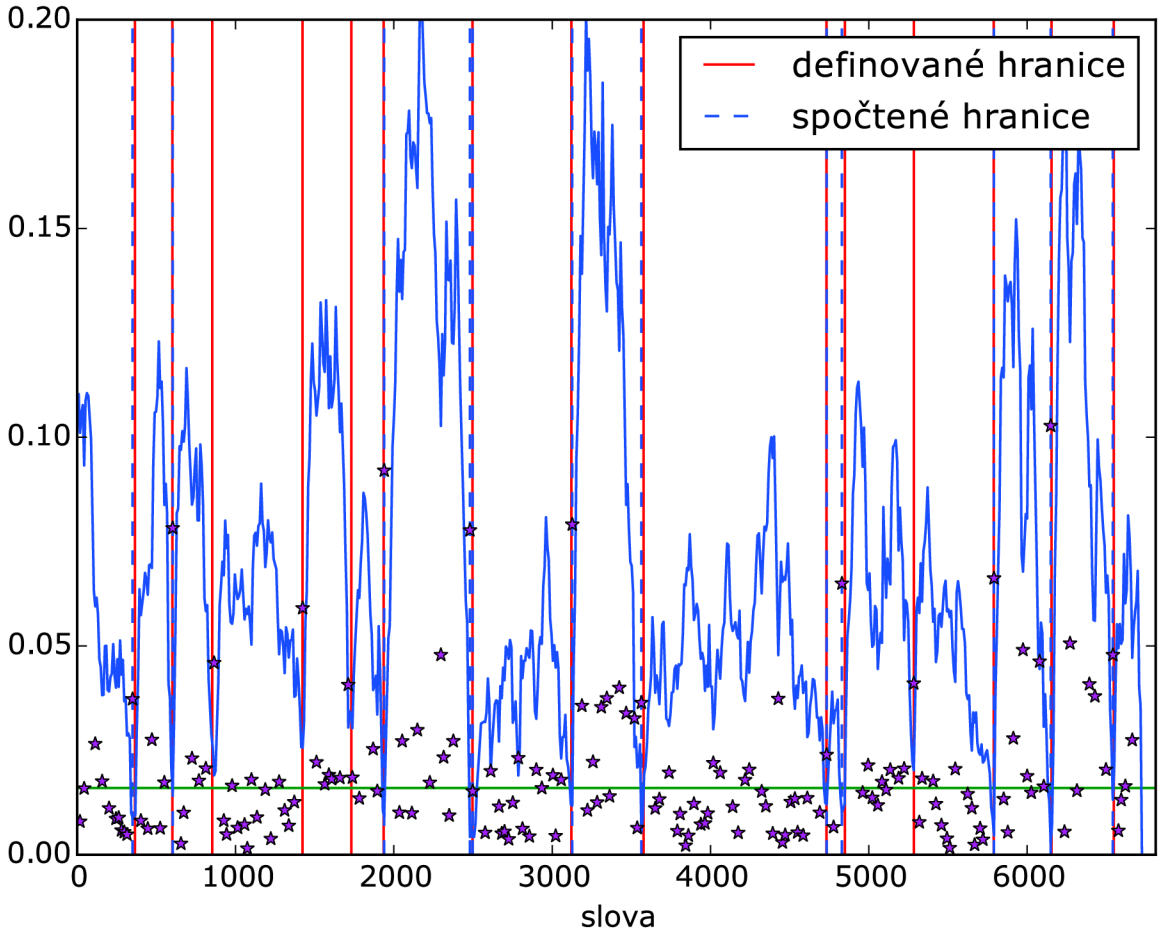
Druhou úpravou je ignorování detekovaných tématicky konzistentních částí, které ve videozáznamu nedosahují délky alespoň 60 sekund. Toto opatření bylo zavedeno z toho důvodu, aby se odfiltrovaly ty segmenty, které nepřinášejí uživatelům žádný užitek. Jedná se totiž převážně o úvodní části videozáznamů, kde je přehrávána informace o univerzitě, ze které záznam pochází, popřípadě informace o možnosti finanční podpory dané univerzity a nebo se jedná o několik posledních slov, se kterými se přednášející se studenty loučí.

Poslední úpravou je vytvoření slovníku nalezených tématicky konzistentních celků. Obdobně jako při vytváření slovníku vyskytujících se slov, je nyní indexem do slovníku název souboru .part, který se skládá z názvu přepisu a pořadového čísla detekovaného segmentu v rámci daného přepisu. Hodnotou je číslo, které určuje pořadí, ve kterém byl daný segment vložen do slovníku. Výsledný slovník detekovaných částí je za pomoci knihovny *pickle* uložen do souboru `parts.pkl`.

V určitých případech je možné, že segmentace skončí vyvoláním výjimky, a to v případě, že nebylo možné v přepisu detekovat předěl témat . V takovém případě je řídícím skriptem `autorun.py` uložen název souboru s přepisem, který výjimku vyvolal. Seznam všech přepisů, které při segmentaci skončily chybou, je ukládán do souboru `errorStates`. Tyto přepisy jsou v databázi posléze převedeny do stavu *error*.

Po dokončení segmentace jsou automaticky odstraněny soubory .word a .meta, které se vztahují k právě segmentovanému přepisu. K jejich odstranění dojde i v případě, že segmentace skončí vyvoláním výjimky.

[10] pospojované zprávy z TDT5 Multilingual News Text datasetu, který obsahuje souhrn asi 400 tisíc krátkých zpráv ze světových novin



Obrázek 3.5: Graf hodnot kosinů v předělech. Osa X znázorňuje počet slov v testovaném souboru. Osa Y vypočítanou hodnotu kosinu mezi dvěma bloky při hledání hranice předělu témat. Zelená horizontální čára označuje hranici akceptovatelnosti předělu témat. Hvězdičky označují míru tzv. hloubkového skóre, které udává o jak velké minimum se jedná (čím menší hodnota kosinu tím větší hodnota hloubkového skóre – v grafu místy narušeno drobným poklesem sousední hodnoty). Autor grafu Marek Sychra [9].

3.8 Vytvoření vstupu pro SMM

V následujícím bloku automatického procesu je vytvořena matice, která je použita jako vstup pro SMM. Tato matice je vytvářena za pomoci slovníku slov, která se v přepisech vyskytovala, slovníku tématicky konzistentních částí, na které byly přepisy rozděleny, a slov, která jsou přítomná v souborech .parts. O vytvoření vstupní matice vektorů se stará skript `sparse.py`.

Jednotlivé vektory detekovaných tématicky konzistentních segmentů, které odpovídají sloupcům vytvářené matice, jsou histogramem četností výskytů jednotlivých slov v detekovaném segmentu. Jelikož ale ne všechny segmenty obsahují všechna slova, která jsou uložená ve slovníku detekovaných slov, je výsledná vstupní matice tzv. řídkou maticí, tedy obsahuje větší množství položek s nulovou hodnotou. Jednotlivé řádky matice reprezentují svojí pozicí hodnoty ze slovníku detekovaných slov, který je přítomný v souboru `dict.pkl`.

Sloupce ve vytvářené matici pak svojí pozicí reprezentují hodnoty ze slovníku vytvořených segmentů, který je přítomný v souboru `parts.pkl`. Pro práci s řídkými maticemi v jazyce Python je využíváno knihovny *scipy.sparse*. Jako nejvhodnější z hlediska přístupu k jednotlivým prvkům matice byla pro práci s řídkou maticí z uvedené knihovny zvolena varianta *lil_matrix*[11].

Stejně jako jsou pro vytvoření vstupní matice nepostradatelné soubory `dict.pkl` a `parts.pkl`, jsou podstatné i soubory .parts, které obsahují všechna slova z detekovaného segmentu. Při vytváření vstupní matice je procházen slovník detekovaných segmentů. Pro každý jeho index je zjišťováno, zda existuje soubor .part, jehož název by korespondoval s kontrolovaným indexem. V případě že existuje, je vytvořen na základě jeho obsahu vektor popisující histogram četností jednotlivých slov v dokumentu. Zároveň je v takovém případě vytvořen v databázové tabulce `segmentation` nový záznam, který obsahuje identifikátor přepisu, ke kterému se segment vztahuje, čas začátku a konce segmentu v sekundách a samozřejmě identifikátor segmentu, který koresponduje s hodnotou ze slovníku detekovaných segmentů uloženou pod indexem, kterým je název detekovaného segmentu. Pokud daný soubor neexistuje, tak to znamená, že pro daný segment z přepisu již byl vektor dříve vypočítán. Pokud by se v přepisu vyskytlo slovo, které není ve slovníku detekovaných slov přítomné, je takové slovo ignorováno. Tento stav může nastat pouze v případě, že řídící skript `autorun.py` byl spuštěn bez parametru `-t`, který mimo jiné zajišťuje vytvoření či aktualizaci slovníku detekovaných slov. Výsledná matice je uložena do souboru `sparse_matrix.mtx`.

3.9 Stažení videozáznamu a vytvoření náhledů

Tento blok má jediný, ovšem důležitý úkol. Vytvořit snímky z videozáznamu v časech začátků jednotlivých segmentů. Splnění tohoto úkolu zajišťuje skript `youtube.py`.

Aby bylo možné zadaný úkol splnit, je zapotřebí si nejdříve obstarat videozáznam, ze kterého se budou potřebné snímky extrahovat. Jak je uvedeno v sekci 2.2.4 jediným zdrojem videozáznamů, který splňuje nastavené požadavky jsou videozáznamy z kanálů univerzit a konferencí ve službě YouTube. K identifikaci těch segmentů a tedy i záznamů, u kterých má dojít ke stažení a vytvoření snímků z videozáznamu slouží v databázové tabulce `segmentation` sloupec `thumbnail`.

V první části tohoto bloku automatického procesu jsou z databáze vybrány URL adresy k videozáznamům, a to z těch položek, které již byly úspěšně segmentovány, avšak doposud mají nastavený příznak `thumbnail` na hodnotu `0`, tedy že snímek pro daný segment nebyl vygenerován. V druhé části tohoto bloku jsou získané adresy postupně procházeny a dochází ke stažení jednotlivých záznamů. Ke stahování videozáznamů ze služby YouTube je v jazyce Python využívána knihovna *pythube* a z ní třída *YouTube*.

Pro úspěšné stažení videozáznamu za použití třídy YouTube je zapotřebí vytvořit instanci této třídy. Při instanciaci je jako parametr třídy zadána URL adresa k videozáznamu na YouTube. Dalším krokem už by mohlo být samotné stažení videozáznamu. Problémem by ale byl název videozáznamu, který by byl takto staženému videozáznamu nastaven podle jeho názvu na serveru YouTube.com. Tomuto je zabráněno pomocí metody `set_filename()`, která stahovanému videozáznamu nastaví zvolený název (v tomto případě *output*). Následně je ještě nutné vybrat, v jakém formátu a kvalitě se má daný videozáznam stáhnout. K tomu slouží metoda `get()`, která jako parametry má zvolený formát a rozlišení

[11] řídká matice založená na řádkově spojovaném seznamu

stahovaného videozáznamu a vrací instanci videozáznamu v požadované kvalitě.

Jelikož mají mít vytvářené náhledy pouze ilustrativní charakter, je pro všechna videa zvolen formát *mp4* při rozlišení *360p*. Samotné stahování je zahájeno voláním metody `download()` nad výsledkem metody `get()`. Metoda `download()` může mít pomocí parametru zadánu cestu ke složce, kde má být stahovaný záznam uložen. Pokud není cesta ke složce zadána, je stahovaný záznam uložen do aktuální složky, ze které byl skript spuštěn. Pro jednoduchost byla pro stahování videozáznamů nastavena jako cíl pro uložení složka `/tmp`.

Po stažení videozáznamu je proveden další databázový dotaz, který tentokrát vybere pro právě zpracovávaný záznam všechny identifikátory jeho segmentů včetně jejich času počátku ve videu. Výsledky tohoto dotazu jsou postupně procházeny a na jejich základě jsou za pomoci nástroje *ffmpeg* vytvářeny náhledy pro jednotlivé segmenty. Náhledy jsou vytvářeny z videozáznamu v časech začátků segmentů a jsou ukládány do složky `img/thumbnails/`, která je umístěna v kořenové složce vytvořené webové aplikace. Náhledy jsou automaticky zmenšovány tak, aby změnou jejich velikosti byl zachován původní poměr stran obrazu z videozáznamu a aby šířka náhledového obrázku byla 190 pixelů. Výstupem nástroje *ffmpeg* je obrázek ve formátu .jpg, jehož název je shodný s jednoznačným identifikátorem daného segmentu v databázi.

Po každém vygenerování náhledového obrázku je proveden aktualizační dotaz, který u záznamu právě zpracovaného segmentu v databázi nastaví příznak vygenerovaného náhledového obrázku na hodnotu `1`. Po dokončení generování náhledových obrázků pro všechny segmenty zpracovávaného záznamu je odstraněn soubor `output.mp4` a je pokračováno zpracováváním dalšího záznamu. Tato část automatického procesu se kvůli stahování videozáznamů z YouTube řadí mezi časově náročnější.

3.10 Redukce dimenzionality pomocí SMM

Jak je popsáno v sekci 3.2.3, SMM je v této práci používán pro redukci dimenze vektorů. Autorem třídy SMM, která je použita v této práci je Santosh Kesiraju. Konstruktor třídy SMM vypadá následovně: `SMM(ubm_init, reg_w, lw, reg_T, lT, ivec_dim, trn_iters, ext_iters, verbose, sge_T, out_dir)`, kde `ubm_init` je řetězec určující typ maximálního odhadu pravděpodobnosti (const či est), `reg_w` je řetězec určující typ regularizace pro váhy i-vektorů (jen řetězec l2), `lw` je číslo určující regularizační koeficient pro váhy i-vektorů, `reg_T` je řetězec určující typ regularizace pro bázovou matici (řetězce l1 či l2), `lT` je číslo, které udává regularizační koeficient pro bázovou matici, `ivec_dim` je celé číslo určující požadovanou dimenzi výstupních vektorů, `trn_iters` je číslo určující počet trénovacích cyklů SMM (dle doporučení autora stačí 10), `ext_iters` je číslo určující počet iterací pro extrahování výstupu (dle doporučení autora stačí 3 iterace), `verbose` je binární hodnota zajišťující výpis průběžných informací na standardní výstup, `sge_T` je binární hodnota zajišťující aktualizaci řádků báze (ve výchozím nastavení je zakázáno) a parametr `out_dir` je řetězec určující cestu ke složce, kde má být uložen výsledek extrakce.

V sekci 3.2.3 je uveden popis obecného i v této práci využívaného SMM. Proces trénování SMM je uveden v sekci 5.2. Dimenze jednotlivých vstupních vektorů je redukována jejich pouhým zanesením do N-dimenzonálního podprostoru vytvořeného SMM, kde N je požadovaná dimenze redukovaného vektoru. Dalším krokem je extrakce matice dimenzionálně redukovaných vektorů. Extrahované vektory jsou koordináty do N-dimenzionálního podprostoru vytvořeného díky SMM, ve kterém jsou dané dokumenty, jež jsou reprezentovány pomocí vstupní matice příznakových vektorů, zaneseny. Pro extrakci ze vstupní matice je

ve třídě SMM připravena metoda `extract()`, která jako parametr dostává cestu ke vstupní matici vektorů a výstupem je soubor s maticí dimenzionálně redukovaných vektorů. U extrakce je možné postupovat více způsoby. Jako vstup funkce `extract()` může být buď celá vstupní matice najednou, nebo může být tato metoda volána vícekrát pro jednotlivé části matice. Hlavní podmínkou však je, aby matice, které budou metodě `extract()` v parametru předány měly stejný počet řádků jako matice, která byla použita pro trénování SMM. Z tohoto důvodu jsou při absenci parametru pro trénování v části procesu, která je popsána v sekci 3.6, ignorována nová, doposud se v přepisech nevyskytující slova. Výstupní matice se v případě extrakce prováděné nad jednotlivými částmi vstupní matice skládá ze všech postupně generovaných výstupních matic. Jednotlivé výstupy jsou tedy postupně spojovány do jedné výsledné matice dimenzionálně redukovaných vektorů.

Extrakci lze provádět i bez nutnosti předchozího trénování SMM, pokud máme uložen již natrénovaný model. V takovém případě je dříve natrénovaný model pouze načten. Výstupem této části je soubor `finish.npy`, který obsahuje matici dimenzionálně redukovaných příznakových vektorů reprezentující detekované segmenty z videozáznamů a to na dimenzi s hodnotou 300.

3.11 Výpočet podobnosti segmentů

Předposlední a také nejpodstatnější částí automatického procesu zpracování videozáznamů je výpočet podobnosti mezi dvěma detekovanými segmenty. Výstupem tohoto bloku je pole, které pro každý segment určuje pět nejpodobnějších segmentů a jejich umístění v matici podobnosti vektorů, která je v této části vytvářena. Výpočet podobnosti zajišťuje skript `similarity.py`.

Základem pro výpočet podobnosti dvou segmentů je matice dimenzionálně redukovaných vektorů segmentů, která vznikla jako výstup předchozího bloku. Na začátku tohoto skriptu je načtena tato matice a dle počtu segmentů, který je stejný jako počet sloupců vstupní matice je vytvořena čtvercová matice, do které budou ukládány vypočítané hodnoty podobností.

Čtvercová matice je následně procházena buňku po buňce, kde index řádku označuje první segment k porovnání a index sloupce druhý porovnávaný segment. Ze vstupní matice je tedy vybrán sloupec s indexem shodným s indexem řádku čtvercové matice a stejně i sloupec s indexem shodným s indexem sloupce čtvercové matice. V případě že buňka, kam by měl být uložen výsledek, obsahuje hodnotu `0`, je mezi takto vybranými sloupci (vektory) vypočítána jejich vzájemná podobnost. K výpočtu podobnosti dvou vektorů je použito vzorce pro výpočet *cosine similarity* (viz vzorec 5.2), který je včetně principu popsán v sekci 5.3.2. Výsledek tohoto výpočtu je následně uložen na zvolenou pozici a stejně i na její diagonálně souměrnou pozici. Tím je zajištěno, že nebudou prováděny zbytečné výpočty podobností.

V praxi to znamená, že pokud je vypočítána podobnost dokumentu x a y a jejich výsledek je uložen do čtvercové matice na pozici `[x,y]`, tak již nebude probíhat výpočet podobnosti dokumentu y a x, který by se uložil do čtvercové matice na pozici `[y,x]`, protože takovýto výpočet povede ke stejnému výsledku, který je uložen na pozici `[x,y]`. Pokud jsou indexy řádku i sloupce čtvercové matice shodné (tzn. porovnával by se segment sám vůči sobě), je do dané buňky, bez nutnosti jakéhokoli výběru či výpočtu, uložena hodnota `NaN`.

Pro každý řádek je po dokončení výpočtů podobností se všemi ostatními segmenty vyhledáno pět největších vypočítaných hodnot. Tyto hodnoty, včetně jejich pozic, jsou uloženy do pomocného pole a ve čtvercové matici jsou na daných pozicích v daném řádku

hodnoty nahrazeny hodnotou `NaN`. Výstupem této části je tedy pole, které obsahuje pole pěti indexů a pole pěti hodnot pro všechny segmenty, reprezentující nejpodobnější segmenty pro každý segment.

3.12 Předání výsledků webové službě

Posledním částí vytvářeného procesu je uložení vypočítaných výsledků do databáze. Tento úkol je součástí skriptu `segmentation.py`. Vstupem je pole, které obsahuje informace o nejpodobnějších segmentech pro všechny zpracované segmenty. Nejdříve jsou z databázové tabulky `linking` odstraněny všechny záznamy, které mají ve sloupci `record1` uveden identifikátor právě zpracovávaného segmentu. Hodnoty z pole s výsledky jsou následně ukládány do této tabulky, kde jeden záznam obsahuje identifikátor prvního i druhého porovnávaného segmentu a vypočítanou hodnotu podobnosti těchto segmentů. Po vytvoření nového záznamu v tabulce `linking` je proveden aktualizační dotaz, jenž má za následek změnu stavu u záznamu, ze kterého daný segment pochází, v tabulce `records` na stav `done`. Od této chvíle je daný záznam zpřístupněn ke shlédnutí ve webové službě.

3.13 Shrnutí

Tato kapitola se zabývá popisem automatického procesu pro zpracování videozáznamů za účelem hledání podobností mezi nimi. V jednotlivých sekcích jsou popsány kroky, které byly pro vytvoření procesu uvedeného na obrázku 3.3 vykonány. V následující kapitole bude představena webová část aplikace, která je důležitou částí při prezentaci výsledků podobností uživateli.

Kapitola 4

Webová aplikace

V této kapitole jsou uvedeny všechny nástroje a skripty, které byly při tvorbě webové části diplomové práce použity. Výčet a struktury databázových tabulek, se kterými webová část pracuje, jsou uvedeny v sekci 3.1.

Vizualizace výsledků je v této práci realizována prostřednictvím webové prezentace. Jejím prostřednictvím mají uživatelé možnost jednak přehrávat uložené videozáznamy z online kurzů, dále je mohou do databáze přidávat a také v nich vyhledávat. Pro administraci těchto záznamů a tím i celé služby slouží vytvořená administrační sekce, do které má přístup pouze uživatel, který zná potřebné přihlašovací údaje. Než však je představeno finální řešení, je popsán postupný vývoj vizualizační části práce (viz sekce 4.3). Nejdříve je ale uveden výčet použitých nástrojů a technologií, které webová služba využívá či s nimi spolupracuje.

4.1 Použité nástroje

Při tvorbě webových aplikací je nejčastěji využíváno značkovacího jazyka HTML. Stejně je tomu i u vizualizační části této práce. Pro generování bylo využito PHP skriptů. K přívětivějšímu vizuálnímu výsledku celé aplikace byly využity kaskádové styly. Ovšem webové rozhraní není jedinou částí práce. Velmi důležité jsou skripty, které vykonávají požadovanou funkcionalitu. Tyto skripty jsou napsány v jazyce Python. Jejich činnost je popsána v kapitole 3, která se věnuje popisu automatického procesu zpracování dat.

Kaskádové styly

Vizuální část webové stránky lze vytvářet různými způsoby, například použitím specifických značek jazyka HTML, které jsou vytvořeny právě za účelem stylování dokumentu. Z hlediska tvorby webových stránek je však nejlepším a jediným správným způsobem tvorby stylu hypertextového dokumentu využití kaskádových stylů. Soubor `style.css`, který obsahuje definici stylů všech bloků zobrazených ve výsledné aplikaci, je uložen ve složce `css` webové části vytvářené aplikace. Při implementaci byla vybrána taková stylovací pravidla, která jsou podporována nejrozšířenějšími internetovými prohlížeči v nejnovější verzi jejich vydání[1]. Pro určení míry podpory jednotlivých pravidel kaskádových stylů prohlížeči byla využívána internetová stránka `http://caniuse.com`.

[1] Byly uvažovány prohlížeče Internet Explorer, Mozilla Firefox, Opera, Google Chrom a Safari (Safari na systému MAC OS X).

JavaScript

Když se v roce 1995 jazyk JavaScript poprvé objevil, bylo jeho hlavním účelem ověřování údajů zadávaných uživateli, o což se dříve staraly jazyky na straně serveru jako je kupříkladu Perl. Od té doby se JavaScript stal důležitou součástí všech hlavních webových prohlížečů na trhu. Kromě jednoduché validace dat se nyní stará o interakci s téměř všemi aspekty okna prohlížeče a jeho obsahu. ([11], str. 29, odst. 1 a 2).

V této práci je právě za pomoci JavaScriptu vytvořena interaktivní část webové aplikace. Vše, co se na stránce jakkoli mění po akci uživatele, je prováděno právě JavaScriptem s využitím knihovny jQuery.

jQuery

jQuery je volně šiřitelná JavaScriptová knihovna, která uživatelům při programování v JavaScriptu ulehčuje práci při manipulaci a změnách v HTML dokumentu, navazování akcí na nastalé události či použití animací a AJAXu. Tato knihovna je podporována napříč všemi nejvíce používanými webovými prohlížeči. Při práci s jQuery bylo využíváno on-line dokumentace[2] a knihy Mistrovství v jQuery [1]. Právě AJAX je nevíce využíván v administrační části webu.

4.2 Použité technologie

Stejně jako v části automatického zpracování videozáznamů tak i ve webové části aplikace jsou pro dosažení požadovaných cílů vytvářeného procesu využity existující technologie. Webová část využívá pouze jedinou technologii a to pro indexování informací o videozáznamech včetně jejich přepisů za účelem jejich vyhledávání.

4.2.1 Vyhledávání ve webové službě

Umožnit uživatelům vyhledávat ve webové službě, sice není v zadání práce výslovně uvedeno, ale podvědomě je s touto možností pro uživatele počítáno. Při implementaci webové služby bylo navíc potřeba zajistit možnost vyhledávání, a to nejen v informacích získaných od uživatele, ale také v automaticky generovaných přepisech. Při pravidelných konzultacích s vedoucím práce bylo pro zajištění vyhledávání uvažováno o dvou knihovnách, které by mohly indexování a následné vyhledávání v datech zajišťovat. Konkrétně se jednalo o knihovny *Sphinx* a *Lucene*.

Sphinx

Sphinx je open source server pro fulltextové vyhledávání, který byl od základu navržen s ohledem na výkon, kvalitu poskytovaných výsledků vyhledávání a co nejjednodušší integraci do systému. Server je napsán v jazyce C++ a je navržen pro operační systémy Linux, Windows, MacOS, Solaris, FreeBSD a další.

Dokáže indexovat a vyhledávat v datech, která jsou uložena v SQL databázích, NoSQL databázích či souborech. Sphinx je dostupný pod licencí GPLv2. [8]

[2] `https://api.jquery.com/`

Lucene

Lucene, celým názvem Apache Lucene, je knihovna v jazyce Java zaměřená na textové vyhledávání. Nabízí mnoho funkcí, které jsou dostupné prostřednictvím poskytovaného API. Díky implementaci v jazyce Java je multiplatformní vyhledávací knihovnou.

Po vytvoření indexů nad daty, která mají být předmětem vyhledávání, je schopna rychle vrátit výsledky vyhledávání seřazené podle relevance, či dokáže vyhledávat jen v určitých oblastech (názvy, anotace, autoři, přepisy apod.). Apache Lucene je open source knihovnou dostupnou zdarma i ke komerčním účelům a to pod Apache Licencí verze 2.0[3].

Jelikož na serveru SuperLectures.com již bylo implementováno vyhledávání pomocí knihovny Lucene, byl tento vyhledávač z této služby převzat. Výhodou tohoto vyhledávače je také možnost hledat danou frázi i v hypotézách jednotlivých slov, které musí být součástí generovaného přepisu. Tyto hypotézy jsou pak součástí nově vytvořeného indexu. Jedinou změnou tedy bylo vytvoření indexů z přednášek, které jsou uloženy ve výsledné službě.

4.3 Vizualizace

Tato sekce popisuje postupný vývoj vizualizační části diplomové práce od úplných počátků. Představený finální stav je pro předvedení principu fungování webové služby pro účely této práce dostačující ale jistě ne konečný. Nezanedbatelná část času byla totiž věnována experimentům s daty, a proto bylo přistoupeno k vytvoření plně funkčního ovšem graficky ne tolik oslnivého webového rozhraní. V dalším vývoji této práce bude na vizualizaci kladen větší důraz, aby grafická část prezentace dosahovala alespoň takové kvality jako webové prezentace například z portfolia služeb společnosti ReplayWell.

4.3.1 Sběr přednášek

V sekcích 5.1 a 3.4 je popsán princip a způsob, jakým byly do systému zadávány údaje o prvních videozáznamech z on-line kurzů. Ani jedna z uvedených sekcí se však nevěnuje vizuální stránce, která sice nebyla nikterak oslnivá, avšak stala se základem pro budoucí webovou službu.

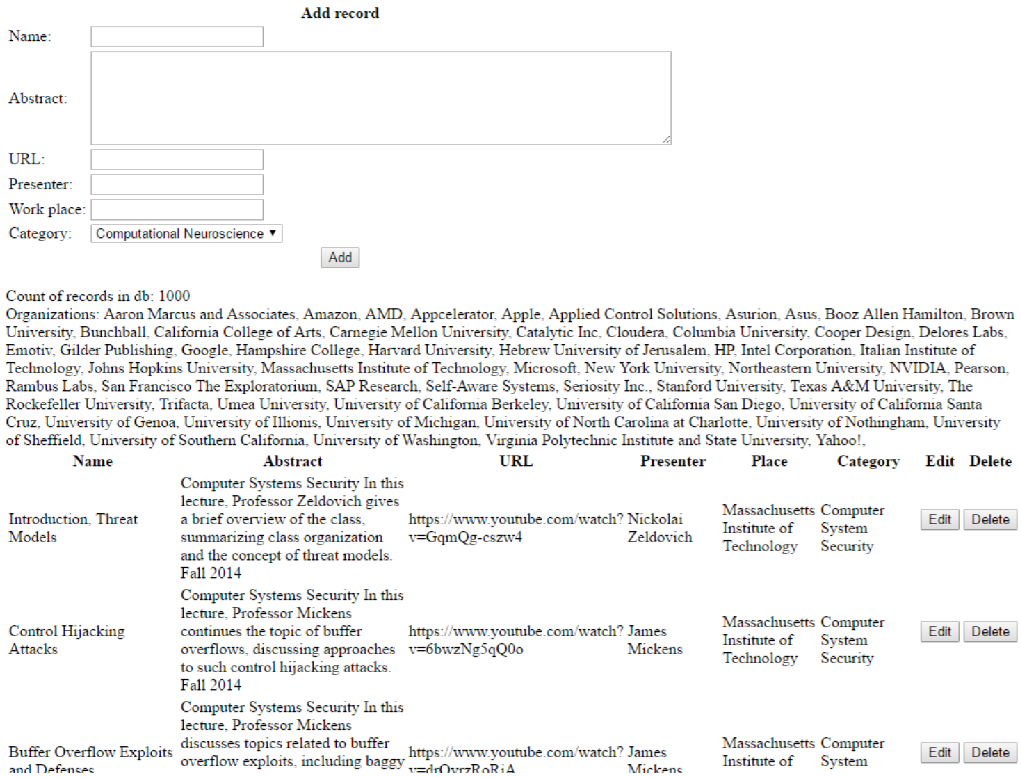
Na obrázku 4.1 je ukázka úplně první stránky z vizualizační části práce. Tato stránka byla využívána pouze pro sběr a úpravu informací o videozáznamech a dále pro generování CSV souboru, který byl předán členům výzkumné skupiny Speech@FIT ke zpracování.

Stránka obsahovala formulář, který sloužil k zadávání a upravování informací o přednáškách v databázi. Pod tímto formulářem byl kontrolně vypisován počet již nasbíraných záznamů, za kterým následoval výpis všech univerzit a společností, ze kterých pocházeli přednášející ve shromážděných videozáznamech. Pod těmito informacemi se nacházela tabulka, která obsahovala výpis všech informací o všech záznamech v databázi. Každý z nich bylo možné upravit nebo vymazat.

4.3.2 První vizualizace

Po obdržení prvních přepisů bylo již nutné vytvořit základní strukturu výsledné webové služby. Webová služba měla uživatelům umožňovat přehrát všechny záznamy uložené v databázi a při jejich přehrávání jim nabídnout pro všechny detekované segmenty návrhy po-

[3] dostupná z `http://www.apache.org/licenses/LICENSE-2.0.html`



Add record

Name:

Abstract:

URL:

Presenter:

Work place:

Category: Computational Neuroscience

Add

Count of records in db: 1000

Organizations: Aaron Marcus and Associates, Amazon, AMD, Appcelerator, Apple, Applied Control Solutions, Asurion, Asus, Booz Allen Hamilton, Brown University, Bunchball, California College of Arts, Carnegie Mellon University, Catalytic Inc, Cloudera, Columbia University, Cooper Design, Delores Labs, Emotiv, Gilder Publishing, Google, Hampshire College, Harvard University, Hebrew University of Jerusalem, HP, Intel Corporation, Italian Institute of Technology, Johns Hopkins University, Massachusetts Institute of Technology, Microsoft, New York University, Northeastern University, NVIDIA, Pearson, Rambus Labs, San Francisco The Exploratorium, SAP Research, Self-Aware Systems, Seriosity Inc., Stanford University, Texas A&M University, The Rockefeller University, Trifacta, Umea University, University of California Berkeley, University of California San Diego, University of California Santa Cruz, University of Genoa, University of Illinois, University of Michigan, University of North Carolina at Charlotte, University of Nothingham, University of Sheffield, University of Southern California, University of Washington, Virginia Polytechnic Institute and State University, Yahoo!,

Name	Abstract	URL	Presenter	Place	Category	Edit	Delete
Introduction, Threat Models	Computer Systems Security In this lecture, Professor Zeldovich gives a brief overview of the class, summarizing class organization and the concept of threat models. Fall 2014	https://www.youtube.com/watch?v=GqmQg-cszw4	Nickolai Zeldovich	Massachusetts Institute of Technology	Computer System Security	Edit	Delete
Control Hijacking Attacks	Computer Systems Security In this lecture, Professor Mickens continues the topic of buffer overflows, discussing approaches to such control hijacking attacks. Fall 2014	https://www.youtube.com/watch?v=6bwzNg5qQ0o	James Mickens	Massachusetts Institute of Technology	Computer System Security	Edit	Delete
Buffer Overflow Exploits and Defenses	Computer Systems Security In this lecture, Professor Mickens discusses topics related to buffer overflow exploits, including baggy	https://www.youtube.com/watch?v=dqOvrzRqRjA	James Mickens	Massachusetts Institute of	Computer System	Edit	Delete

Obrázek 4.1: Ukázka pomocné stránky pro sběr záznamů z přednášek

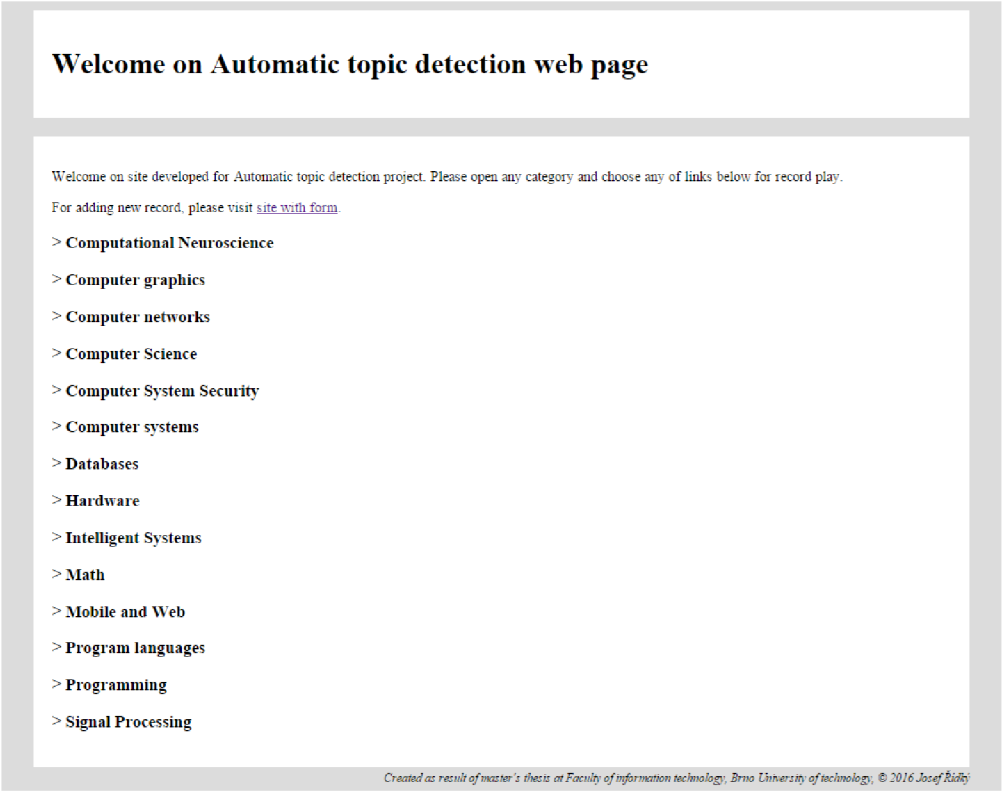
dobných segmentů z ostatních videozáznamů. Seznamy všech videozáznamů z databáze, které byly rozděleny do předem určených kategorií byly vypsány na úvodní stránce webové aplikace. Ukázka rozložení úvodní stránky je na obrázku 4.2.

Stránka obsahovala dva hlavní boxy, kde v prvním byl vypsán nadpis stránky a ve druhém byly seznamy videozáznamů rozdělené podle kategorií.

Druhá stránka byla určena pro přehrání zvoleného videozáznamu. Na stránce byl zobrazen název, samotný videozáznam, abstrakt, jméno přednášejícího a místo jeho působení a jako poslední kategorie, do které byl záznam zařazen. Pod videozáznamem byly pod sebou vypsány časy jednotlivých detekovaných segmentů ve videozáznamu a u každého z nich uvedeny odkazy na maximálně pět tématicky nejpodobnějších segmentů z ostatních videozáznamů. Ukázka rozložení stránky s přehrávačem je na obrázku 4.3.

Bylo také nutné uvést alespoň koncept stránky, která by sloužila k přidávání nových videozáznamů do databáze. Byla tedy vytvořena stránka s formulářem, který obsahově odpovídal formuláři z obrázku 4.1. Ukázka rozložení stránky pro přidání nového záznamu je na obrázku 4.4. Vytvořená stránka byla opravdu jen ilustrativní, neboť i když byly vyplněny všechny položky formuláře a ten byl následně odeslán, nebyly žádné nové záznamy do databáze přidány.

Další nepříjemností této verze webové aplikace byla absence jakéhokoli menu, pomocí kterého by se dalo mezi všemi třemi stránkami jednoduše přepínat. Tato funkcionalita přišla až s novou verzí webové služby. V této podobě byla práce představena při obhajobě semestrálního projektu.



Obrázek 4.2: Ukázka prvotního rozložení prvků na hlavní stránce

4.3.3 Název výsledné služby

Potřeba vytvoření názvu webové služby byla známá již na počátku zpracování práce. Do popředí se tato otázka dostala až v návaznosti na účast práce ve studentské konferenci Excel@FIT (více viz 6.1). Do té doby byla webová prezentace označována jako *Automatic topic detection web page*.

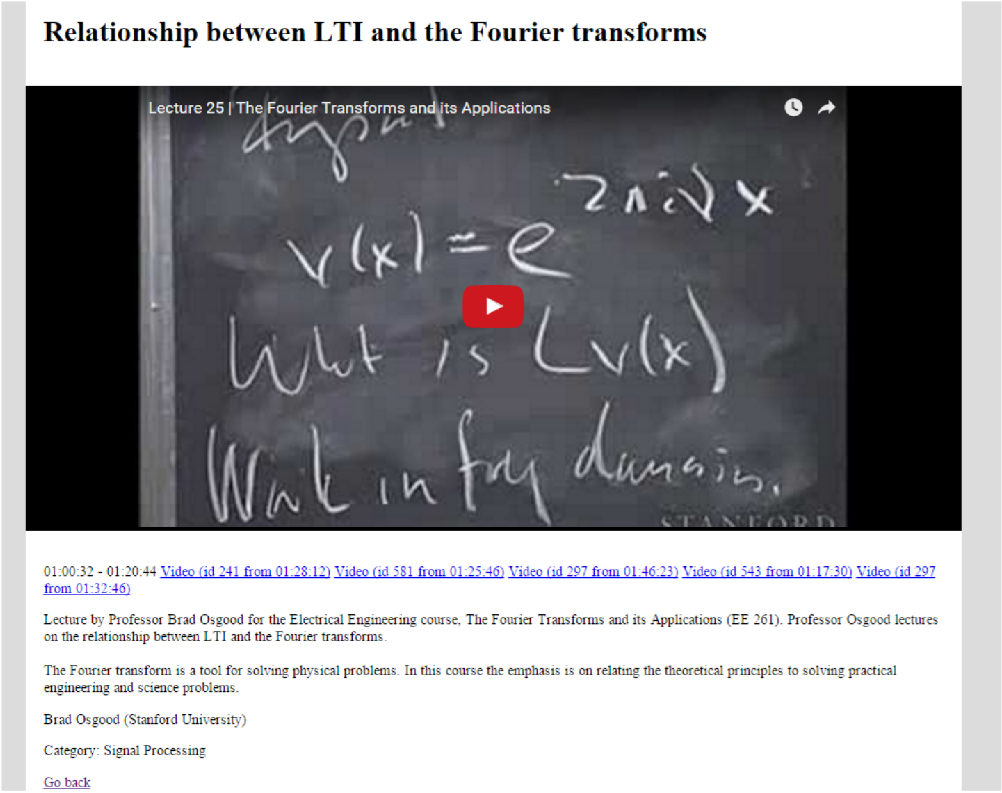
Při vytváření prvního návrhu názvu služby bylo uvažováno nad tím, jak co nejkratším a nejvýstižnějším způsobem tuto službu pojmenovat. Výsledkem byl název *SimTube*, který vznikl ze spojení slov *Similar* a *YouTube* (logo SimTube je na obrázku 4.5).

Tento návrh však názvem výsledné služby není, neboť bylo zapotřebí danou webovou službu uvést alespoň názvem do portfolia produktů, mezi které bude po svém dokončení umístěna. Od společnosti ReplayWell byl tedy propůjčen název *SuperLectures.net*, který odkazuje na projekt *SuperLectures.com* této společnosti (logo SuperLectures.net je na obrázku 4.6).

Krom názvu byla k užívání propůjčena i doména `http://superlectures.net`, která je nyní touto službou využívána.

Při tvorbě grafické části loga bylo využíváno služby *Logo Maker*[4], která umožňuje vytvářet loga on-line. Podmínkou jejich použití je uveřejnění autorů, kteří vytvořili podklady, z nichž je výsledné logo vytvořeno. Tyto informace jsou uvedeny ve výsledné webové službě vždy na konci každé stránky.

[4] `http://logomakr.com/`



Obrázek 4.3: Ukázka rozložení prvků na původní stránce pro přehrání záznamů

Obrázek 4.4: Ukázka rozložení prvků na původní stránce pro přidání nových záznamů

Obrázek 4.5: Ukázka loga prvního návrhu názvu webové aplikace

Obrázek 4.6: Ukázka výsledného loga webové aplikace

4.3.4 Model View Controller

Při vytváření webových stránek je obecně velice doporučováno využívat některý z dostupných a prověřených frameworků (např. Nette, Zend apod.), které mohou v mnohém velice ulehčovat jejich tvorbu. Někdy je ale využití těchto frameworků zbytečné, neboť pro danou prezentaci představují příliš složitý způsob řešení. Jedním z těchto případů je i tato práce. V ní nebylo využito žádného dostupného frameworku, ale byla vytvořena vlastní jednoduchá infrastruktura, která se snaží splňovat zásady Model View Controller architektury (dále jen MVC), která je využívána při vývoji softwaru.

MVC odděluje prezentaci a interakci od systémových dat. Systém je strukturován do třech logických komponent, které vzájemně interagují. Komponenta Model spravuje systémová data a přidružené operace s těmito daty. Komponenta View definuje a řídí, jak jsou data prezentována uživateli. Komponenta Controller obsluhuje interakci s uživatelem a předává tyto interakce komponentám Model a View. Výhodou tohoto modelu je, že umožňuje, aby se data měnila nezávisle na jejich reprezentaci a naopak. Podporuje také prezentaci stejných dat různými způsoby, kdy změny provedené v jedné reprezentaci se projeví i ve všech dalších. [6]

Model

Model je prostředníkem při komunikaci webové aplikace s MySQL databází. Je reprezentován souborem `MySql.php`, který obsahuje stejnojmennou třídu. Pro připojení a komunikaci s databází je zapotřebí získat přihlašovací údaje k databázi. Z praktických i bezpečnostních důvodů jsou přihlašovací údaje uloženy v souboru, který není přístupný z internetového prohlížeče.

Třída *MySql* poskytuje pro komunikaci s databází následující funkce:

- **select** – zajišťuje získání dat z databáze. Jako parametr jsou jí předávány tři proměnné. V první proměnné jsou uvedeny sloupce, které jsou obsahem dotazu. V druhé proměnné jsou uvedeny tabulky, ze kterých má být vybíráno a v poslední proměnné jsou uvedeny podmínky, pomocí kterých má být omezen výsledek dotazu. V případě že není požadováno omezení pomocí klauzule `where`, je proměnná pro podmínky ponechána prázdná. Výsledkem této funkce je pole výsledků, kde každý výsledek je polem jednotlivých hodnot.

- **insert** – zajišťuje vložení nových záznamů do databáze. Parametry této funkce jsou tři proměnné. První proměnná obsahuje název tabulky, do které má být nový záznam přidán. Druhá proměnná obsahuje názvy sloupců z dané tabulky, kterým budou tímto dotazem nastaveny hodnoty pro uložení. Poslední parametr obsahuje hodnoty, které mají být tímto dotazem uloženy.

- **update** – zajišťuje aktualizaci záznamů v databázi. Jako parametry jsou jí předávány tři proměnné. První proměnná určuje, jaká tabulka má být aktualizována. Druhá proměnná určuje aktualizované sloupce s jejich novými hodnotami a poslední parametr obsahuje omezující podmínku pro daný dotaz.

- **delete** – zajišťuje odstranění záznamů z databáze. Parametry této funkce jsou název tabulky, ze které mají být záznamy odstraněny a omezující podmínka určující rozsah platnosti daného dotazu.

- **query** – zajišťuje provedení SQL dotazu, který je funkci předán jako její jediný parametr, pomocí funkce `mysqli_query()`. Výsledkem této funkce je výstup funkce `result()`.

- **result** – zajišťuje vytvoření pole výsledků daného SQL dotazu a to na základě výstupu funkce `mysqli_fetch_row()`, které je předán jediný parametr funkce `result`, jímž je výstup z funkce `mysqli_query()`.

Při práci s objektem třídy *MySql* jsou využívány pouze funkce `select`, `insert`, `update` a `delete`. Funkce `query` a `result` jsou využívány pouze funkcemi z třídy *MySql*. Pro práci s databází v PHP je využíváno funkcí z třídy `mysqli`.

View

Vykreslování ve webovém prohlížeči obstarává v MVC část View. Ta je reprezentována souborem `Zobrazeni.php`, který obsahuje třídu *Zobraz*.

Třída *Zobraz* obsahuje následující funkce:

- **inicializace** – zajišťuje výpis úvodní části HTML dokumentu, kde jsou uvedeny všechny meta tagy, potřebné JavaScriptové knihovny a kaskádové styly. Parametry této funkce jsou: řetězec zajišťující korektní výpis relativní cesty ke knihovnám a kaskádovým stylům a název sekce, který je vypsán v tagu `title`. Součástí této metody je i vypsání přihlašovacího formuláře, ovšem jen v případě, že daný návštěvník webových stránek není do služby přihlášený. Přihlásit se může pouze administrátor.

- **text** – zajišťuje výpis obsahu dané stránky webové aplikace. Obsahy webových stránek[5] jsou uloženy v samostatných souborech, které zajišťují výpis požadovaného obsahu. Tato metoda má za úkol tyto soubory korektně načíst a přiložit při provádění skriptu do kódu. Parametry této funkce jsou pole, které obsahuje nadpis, jenž má být na generované stránce vypsán a název souboru, který zajišťuje vygenerování požadovaného obsahu. Dalším parametrem je proměnná reprezentující vytvořené spojení s MySQL databází. Třetím parametrem je řetězec pro korektní vypsání relativní cesty k potřebným souborům a posledním parametrem jsou data, která určité skripty využívají při zpracování. Tato metoda, kromě generování požadovaného obsahu, zajišťuje také výpis hlavičky s menu a bloku s vyhledávacím formulářem.

- **zakonceni** – zajišťuje zakončení všech tagů v HTML dokumentu a vypisuje informace o autorovi webu a autorech podkladů pro použité logo.

- **error** – zajišťuje výpis chybového oznámení uživateli. Parametrem této metody je řetězec či pole řetězců s chybovými oznámeními.

[5] předpisy pro jejich vygenerování

- **ok** – zajišťuje výpis potvrzení o korektním provedení akce uživateli. Parametrem této metody je řetězec či pole řetězců s oznámeními.
- **info** – zajišťuje výpis oznámení informačního charakteru. Parametrem této metody je řetězec či pole řetězců s oznámeními.

Controller

Controller je v MVC tou částí, která zajišťuje zpracování dat získaných pomocí části Model a vytvoření výsledné stránky za použití části View. Je tedy spojovacím a řídícím článkem celé architektury MVC. Controller je reprezentován souborem `Obsah.php`, který obsahuje třídu *Start*. Instance této třídy je vytvořena v souboru `index.php`, kde je volána následující metoda:

- **zobraz** – zajišťuje vytvoření požadované stránky, která je specifikována pomocí parametru GET v URL adrese dané stránky. Pro komunikaci s databází využívá vlastní instanci třídy *MySql* a pro vykreslování vlastní instanci třídy *Zobraz*.

Jelikož je webovou službou požadováno zobrazování pouze malého počtu typů stránek, jsou tyto stránky rozlišovány pouze parametrem v URL adrese. V závislosti na typu parametru je uživateli zobrazena patřičná stránka.

4.3.5 Výsledná webová aplikace

Výsledná webová aplikace obsahuje pět typů stránek, které se mohou uživateli zobrazit. Jedná se o úvodní stránku, stránku pro zobrazení výsledků vyhledávání, stránku pro přidání nového záznamu, administrační stránku a stránku pro přehrávání videozáznamů. V této sekci budou jednotlivé stránky postupně popsány.

Úvodní stránka

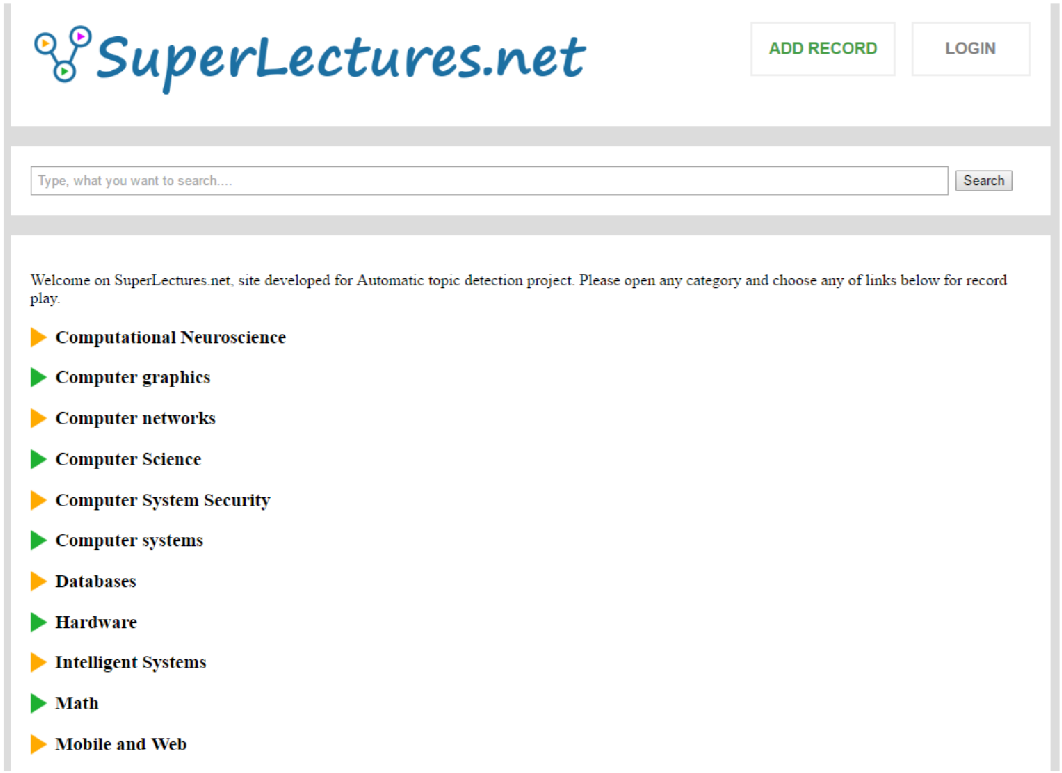
První představenou stránkou je úvodní stránka, která je generována při absenci parametru v URL adrese a to skriptem `enter.php`. Ukázka stránky je na obrázku 4.7.

V porovnání s dřívější implementací, která je na obrázku 4.2, se na první pohled mnoho nezměnilo. Na stránce došlo ke změně prvního boxu, který nově obsahuje aktivní logo[6] webové služby a menu, pomocí kterého může uživatel přejít na stránku pro přidání nového záznamu, popřípadě se přihlásit do administrační sekce. Pod úvodním boxem se nově nachází i box s vyhledávacím formulářem. Hlavní část stránky je věnována jednotlivým záznamům, které jsou rozděleny do předem připravených kategorií.

Vyhledávání

Webová služba umožňuje uživateli vyhledávání nejen v záznamech z databáze, ale i ve vygenerovaných přepisech. Na všech veřejných stránkách webové aplikace je umístěn vyhledávací formulář, jehož prostřednictvím je kontaktována knihovna Lucene (viz 4.2.1). Lucene běží na straně serveru jako služba, kterou je možné kontaktovat s dotazem i za použití webového prohlížeče. Lucene totiž funguje na nastaveném portu, prostřednictvím kterého očekává vyhledávací dotazy. Obecný vyhledávací dotaz pro Lucene prostřednictvím URL adresy je

[6] slouží zároveň jako odkaz na domovskou stránku



Obrázek 4.7: Ukázka hlavní stránky webové aplikace

následující: `http://domena:PORT/search/search?q=QUERY`, kde `PORT` je číslo portu, na kterém běží služba Lucene na straně serveru a `QUERY` je hledaný dotaz.

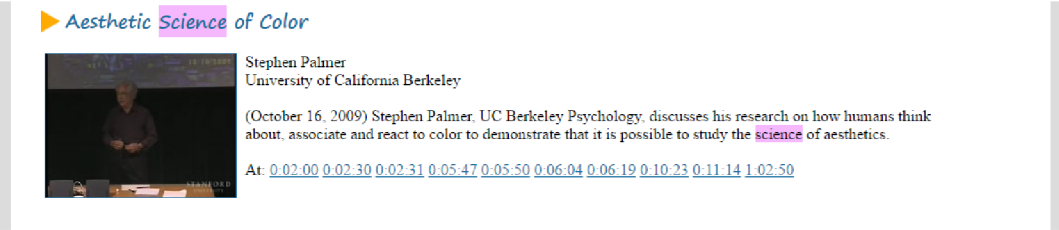
V URL adrese je možné i specifikovat, ve kterých polích má dané vyhledávání probíhat, ve kterých přepisech se má vyhledávat, jak mají být výsledky seřazené, že mají být vráceny výsledky jejichž ohodnocení přesahuje určitou hraniční hodnotu a mnohem více. Všechny tyto možnosti jsou specifikovány dle parametrů GET, které jsou přidávány na konec URL s dotazem. Jejich výčet závisí na konkrétním přizpůsobení vyhledávače. V této práci je využíváno pouhého zaslání dotazu pomocí parametru `q`. Odpovědí Lucene na vyhledávací dotaz je XML soubor obsahující nalezené výsledky. Ukázka XML dokumentu s odpovědí je v příloze G.

Takto získaná odpověď je nejdříve zpracována webovou službou pomocí skriptu `search.php`. Ten ze získané odpovědi za pomoci funkce `simplexml_load_string()` extrahuje informace o záznamech, ve kterých se hledaná fráze nachází a také ve kterých polích daného záznamu se hledaná fráze nachází. Poslední získanou informací jsou časy výskytu hledané fráze v přepisech videozáznamů. Všechny tyto informace jsou následně zobrazeny na stránce s výsledky vyhledávání. Ukázka generovaného výsledku je na obrázku 4.8.

Každý blok s výsledkem obsahuje název přednášky, jméno přednášejícího a jeho pracoviště, abstrakt a odkazy na videozáznam do časů, kdy se hledaná fráze vyskytla ve videozáznamu. Je-li hledaná fráze přítomná v nadpisu či abstraktu, je v nich barevně zvýrazněna.

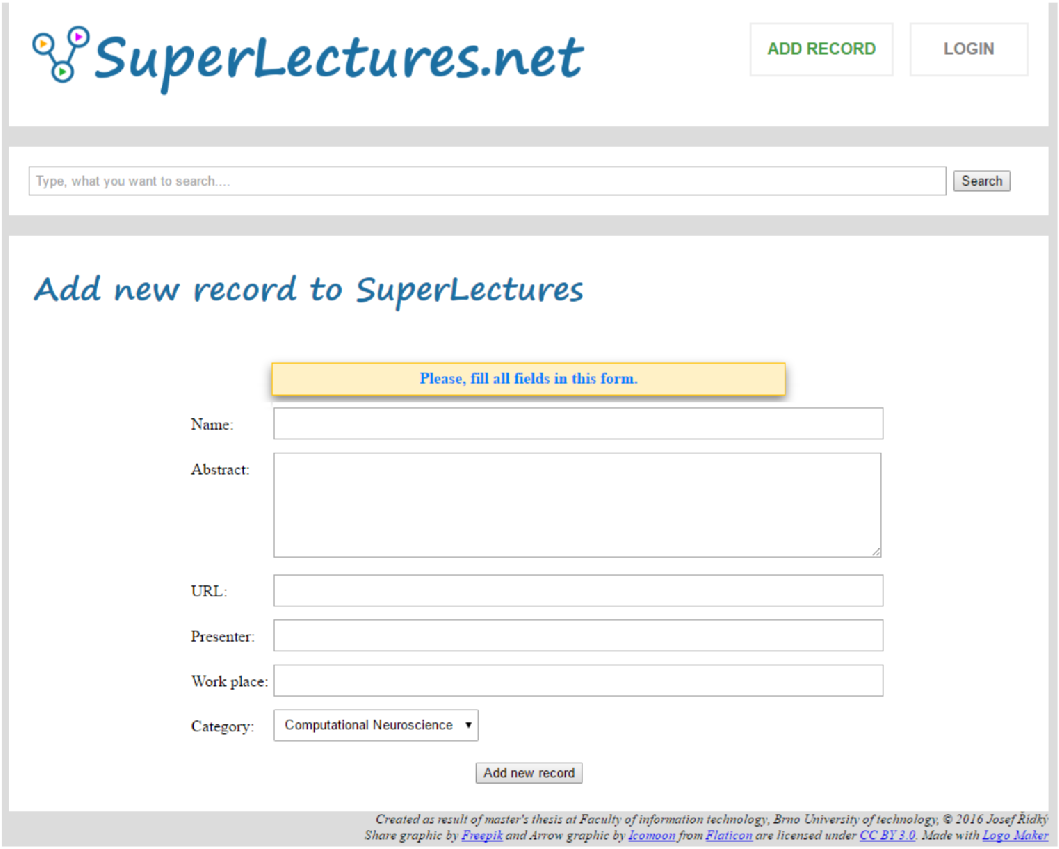
Přidání nových záznamů

Stránka pro přidání nových záznamů je generována v případě přítomnosti parametru `form` v URL adrese. Pro zobrazení této stránky uživatelům slouží tlačítko *ADD RECORD*, které



Obrázek 4.8: Ukázka bloku s výsledkem vyhledávání

se nachází v prvním boxu každé generované stránky. Ukázka stránky pro vložení nové přednášky do systému je na obrázku 4.9.

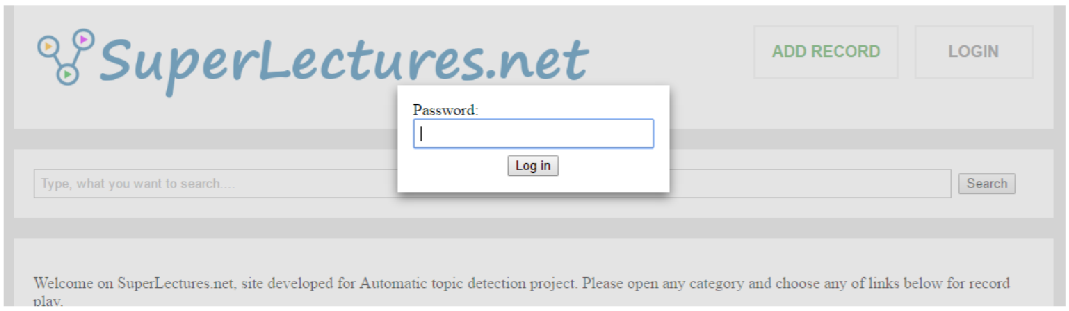


Obrázek 4.9: Ukázka stránky pro přidání záznamu z přednášky do webové aplikace

V porovnání s původní stránkou pro přidávání nových záznamů do webové služby (viz obrázek 4.4) došlo k zasazení dané stránky do nového konceptu webové aplikace. Přibyl tedy úvodní box s logem služby a tlačítky menu, dále přibyl box s vyhledáváním a oproti dřívější verzi byl přenesen nadpis stránky ze samostatného boxu do boxu s obsahem. Jednotlivé formulářové prvky byly následně pomocí kaskádových stylů zvětšeny a ošetřeny takovým způsobem, aby nebylo možné přidat nový záznam, který by neobsahoval všechny po uživateli požadované informace.

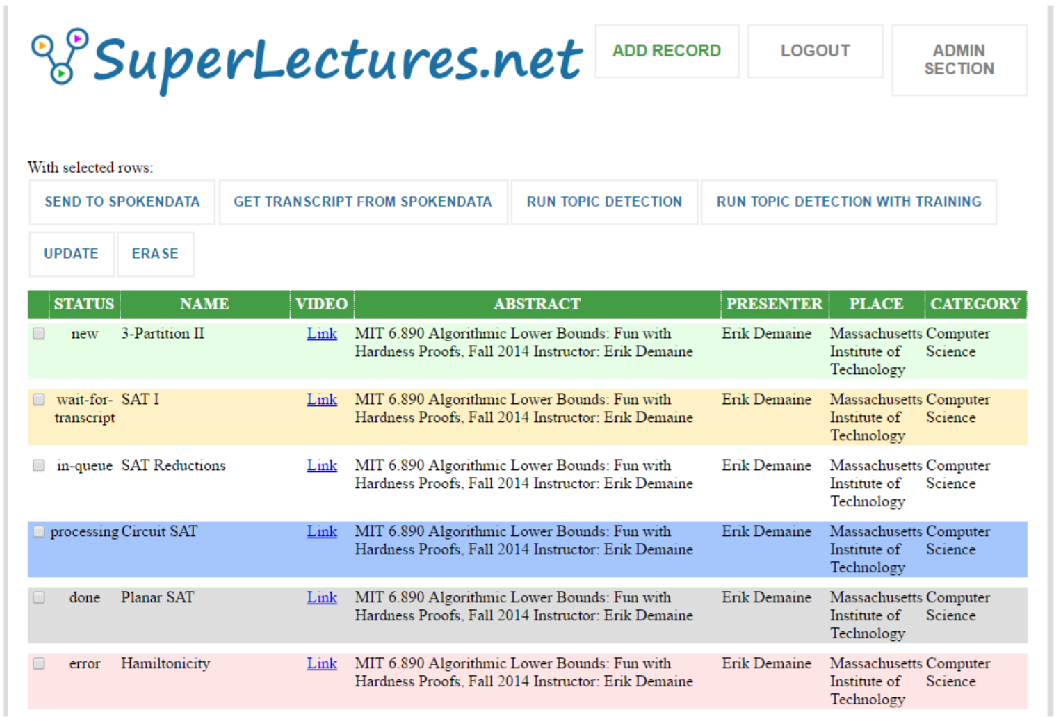
Administrace

Poslední vytvořenou částí aplikace byla administrační sekce. Ta je uživateli zpřístupněna po zadání hesla. K tomu je uživateli určen přihlašovací formulář (viz obrázek 4.10). O udělení či odmítnutí přístupu do administrační sekce rozhoduje skript `control.php`, ve kterém je možné určit přihlašovací heslo pro vstup do administrace. Při úspěšném přihlášení je uživatel přesměrován do administrační sekce, která se nachází ve složce `admin`. Administrační sekce je generována vlastním MVC Controllerem `Obsah.php`, který je uložený ve složce administrace. Tento Controller zajišťuje zobrazení hlavní administrační stránky, která je generována skriptem `enter.php` a zobrazení stránky zajišťující aktualizaci záznamů.



Obrázek 4.10: Ukázka přihlašovacího formuláře

Po úspěšném přihlášení je uživateli vygenerována administrační stránka, na které jsou zobrazeny všechny informace o přednáškách uložených v databázi. Ukázka této stránky je na obrázku 4.11.

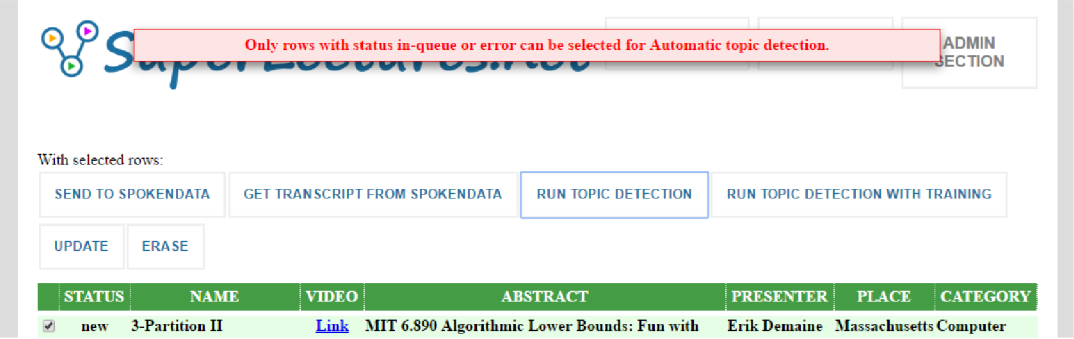


Obrázek 4.11: Ukázka administrační sekce

Záznamy z databáze jsou vypisovány do tabulky, kde každý řádek je v prvním sloupci

uvozen checkboxem. Ten slouží uživateli k označení záznamů, se kterými chce vykonat požadovanou akci. Na začátku této stránky jsou umístěna tlačítka, která jsou i výčtem jednotlivých akcí, jež lze s označenými řádky dělat. Pro snadnější orientaci jsou záznamy od sebe barevně odlišeny v návaznosti na stav, v němž se nacházejí. Záznamy ve stavu *new* jsou podbarveny světlezelenou barvou, *wait-for-transcript* světleoranžovou barvou, *in-queue* bílou barvou, *processing* modrou barvou, *error* světlečervenou barvou a *done* šedou barvou. Barevné označení jednotlivých stavů je znázorněno i na obrázku 4.11. Stav, ve kterém se daný záznam nachází, je uveden i ve druhém sloupci každého záznamu.

Níže jsou popsány jednotlivé akce, které je možné se záznamy provádět. Všechny akce, kromě možnosti *Update*, jsou zpracovány JavaScriptem a patřičné informace jsou na server odesílány pomocí AJAXu. U možnosti *Update* je také požadavek zpracováván JavaScriptem, ovšem v případě dodržení zavedených podmínek dojde k přesměrování na stránku pro úpravu záznamů. V případě že není některá z níže popsaných podmínek pro jednotlivé akce splněna, je uživateli vypsána chybová hláška popisující daný problém. Ukázka chybového hlášení na stránce je uvedena na obrázku 4.12.



Obrázek 4.12: Ukázka chybového hlášení v administraci

Send to SpokenData Ihned po odeslání požadavku na server jsou nejdřív všem zvoleným záznamům změněny stavy na *uploading-to-spokendata*. Tato akce zajistí automatické odeslání informací ze zvolených záznamů do služby SpokenData a uložení jejich identifikátorů ze služby SpokenData do databáze. Po dokončení těchto kroků jsou zvolené záznamy převedeny do stavu *wait-for-transcript*. Tuto akci je možné provádět pouze se záznamy ve stavu *new*. Zpracování tohoto požadavku na straně serveru zajišťuje skript `sendSpoken.php`.

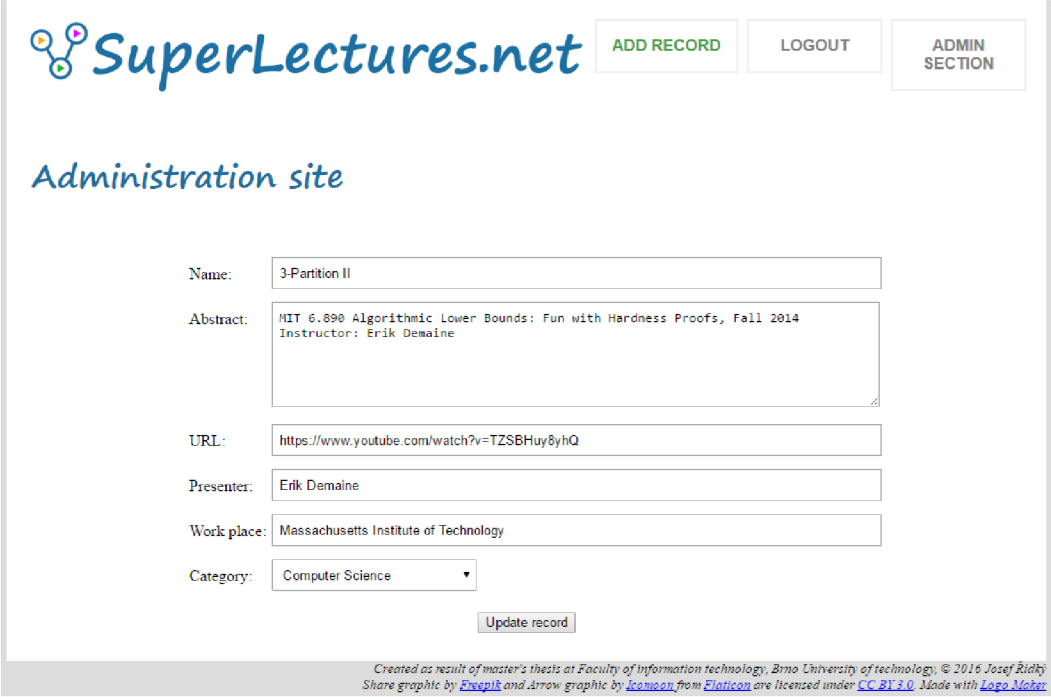
Get transcript from SpokenData Tato akce zajistí zkontrolování stavu, ve kterém se vybrané záznamy ve službě SpokenData nacházejí. Pokud zvolené záznamy mají dokončený přepis mluveného slova, dojde ke stažení přepisu ve formátu XML a jeho následné transformaci do souboru .ts. Po úspěšném stažení přepisu je daný záznam převeden do stavu *in-queue*. Tuto akci je možné provádět pouze se záznamy ve stavu *wait-for-transcript* a *error*. Zpracování požadavku na straně serveru zajišťuje skript `getSpoken.php`.

Run topic detection Tato akce má za následek spuštění automatického procesu detekce témat, segmentace a vizualizace nad zvolenými záznamy. Po odeslání požadavku na server jsou zvoleným záznamům změněny stavy na *processing*. Tuto akci je možné provádět nad záznamy ve stavu *in-queue* a *error*. Zpracování požadavku na straně serveru zajišťuje skript `process.php`.

Run topic detection with training Tato akce provádí stejný úkol jako je popsán u *Run topic detection*. Jediným rozdílem je, že proces automatické detekce témat, segmentace a vizualizace je spuštěn s parametrem `-t`, což má za následek spuštění trénování SMM a aktualizaci slovníku slov vyskytujících se v přepisech. Tuto akci je možné provádět nad záznamy ve stavu *in-queue* a *error*. Zpracování požadavku na straně serveru zajišťuje skript `process.php`.

Update Jedinou podmínkou pro provedení této akce je vybrání pouze jednoho záznamu k úpravě. Záznam se přitom může nacházet v libovolném stavu. Odesláním tohoto požadavku dojde k přesměrování na stránku s formulářem, jež je generována skriptem `update.php`. Ukázka vygenerované stránky je na obrázku 4.13. Jediným polem, které je možné upravit pouze u záznamů ve stavu *new* je URL adresa k videozáznamu. U záznamů, které jsou v libovolném jiném stavu je možnost úpravy URL adresy zakázána. Ostatní pole mohou být upravována dle libosti.

Erase Tato akce má za následek odstranění zvolených záznamů o přednáškách z databáze. Tuto akci je možné provést pouze nad záznamy ve stavu *new* nebo *error*. Zpracování požadavku na straně serveru zajišťuje skript `delete.php`.



Obrázek 4.13: Ukázka stránky pro upravení informací o záznamu z přednášky

Z administrační sekce je možné se odhlásit pomocí tlačítka *LOGOUT*, či se do ní navrátit pomocí tlačítka *ADMIN SECTION*, avšak jen v případě, že se před tím uživatel neodhlásil.

Přehrávání

Poslední a pro uživatele nejdůležitější stránkou celé webové aplikace je stránka zajišťující přehrání zvoleného videozáznamu. Tato stránka, jejíž ukázka je na obrázku 4.14, obsahuje

nejdůležitější část ve třetím bloku. Zde je pro uživatele připraven přehrávač se zvoleným záznamem z přednášky. Pod přehrávačem jsou v samostatných blocích uvedeny jednotlivé segmenty, které byly ve videozáznamu detekovány (oblast A).

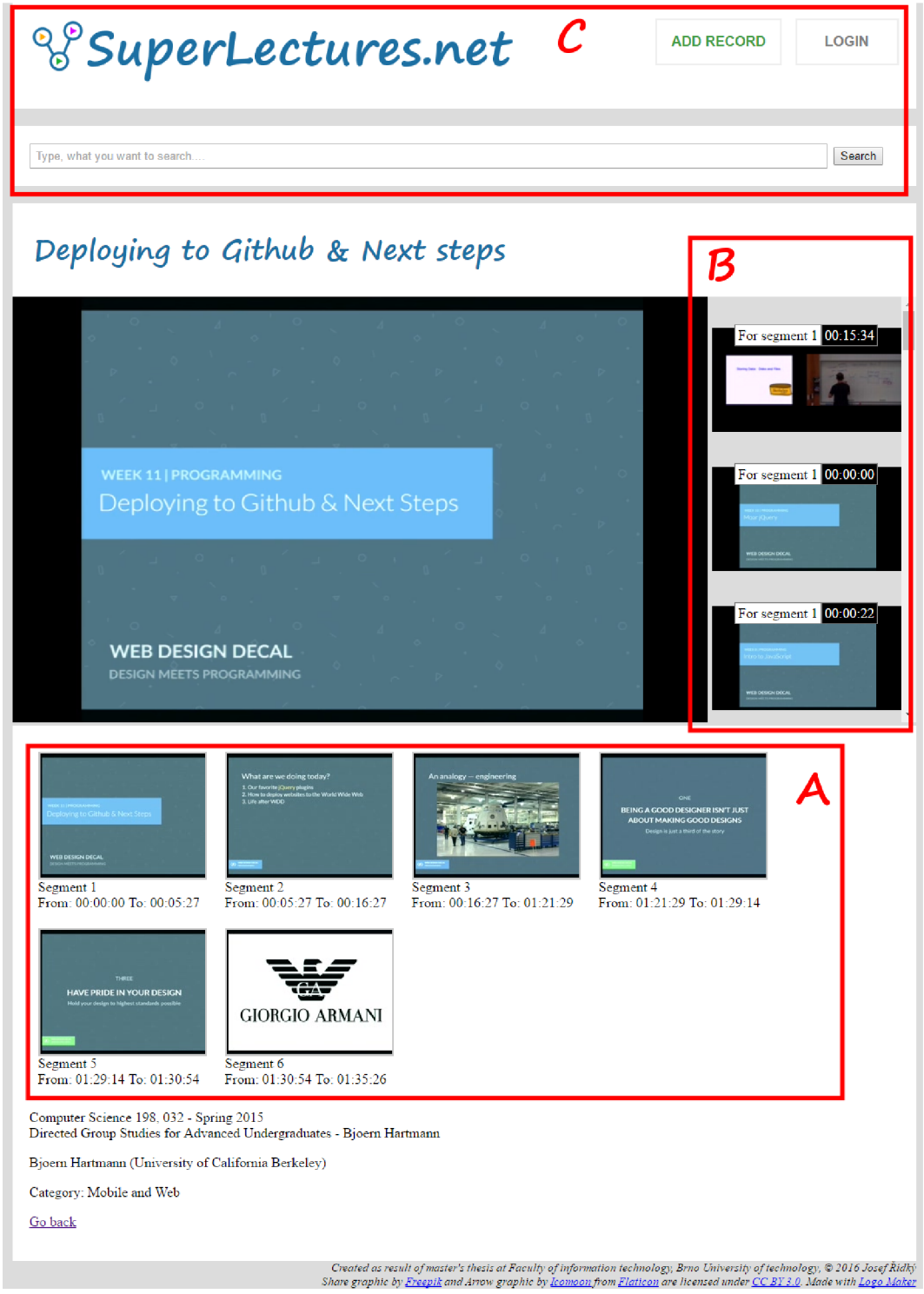
V těchto blocích se nachází náhledový obrázek segmentu, pořadové číslo segmentu ve videozáznamu a časy začátku a konce detekovaného segmentu ve videozáznamu. Na pravé straně od přehrávače (oblast B) je umístěn sloupec, který obsahuje nejpodobnější segmenty z ostatních videozáznamů, jež se vztahují k jednotlivým segmentům detekovaným v právě přehrávaném videozáznamu. Jednotlivé podobné segmenty jsou reprezentovány náhledovým obrázkem podobného segmentu, přes který je umístěn blok s informací, ke kterému segmentu z přehrávaného videa se vztahuje a s uvedením času začátku podobného segmentu ve videozáznamu, z něhož pochází. Po kliknutí na podobný segment je daná přednáška, ze které takový segment pochází, přehrána od toho času, který je u podobného segmentu vypsán. Oblast C zvýrazňuje hlavičku stránky s tlačítky menu a vyhledávací formulář.

Kromě výše popsaných částí stránky s přehrávačem obsahuje prostor pod přehrávačem ještě abstrakt právě přehrávaného záznamu, jméno a pracoviště přednášejícího a kategorii, do které byl daný záznam zařazen. Posledním prvkem v tomto bloku je odkaz, který uživatele přesměruje zpět na hlavní stránku webové aplikace.

Stránka s přehrávačem také využívá JavaScriptu pro zajištění interaktivních změn na stránce. Pro uživatele nejvýraznější změnou je posunutí videozáznamu na čas, kdy daný segment ve videozáznamu začíná, a to po kliknutí na libovolný detekovaný segment, který je umístěný pod videem. Zároveň dojde k posunu bloků s podobnými segmenty takovým způsobem, aby v horní části této oblasti byly umístěny odkazy na segmenty, které se vztahují k právě zvolenému detekovanému segmentu. Stejně dochází k posunu tohoto bloku s podobnými segmenty i v návaznosti na aktuální čas přehrávaného videozáznamu.

4.4 Shrnutí

V této kapitole jsou popsány použité nástroje, vývoj a aktuální stav vizualizační části vytvářené služby. Uvedený finální stav je pro předvedení principu fungování webové služby dostačující ale určitě ne konečný. Při dalším vývoji této práce bude na grafickou stránku služby kladen větší důraz.



Obrázek 4.14: Ukázka stránky pro přehrání záznamu z přednášky. Oblast A) Bloky reprezentující detekované tématické celky v přehrávaném videozáznamu. Oblast B) Nalezené tématicky podobné části z ostatních videozáznamů pro jednotlivé detekované segmenty. Oblast C) Ukázka hlavičky stránky s menu a vyhledávacím formulářem.

Kapitola 5

Experimenty a práce s daty

Tato kapitola popisuje získání a zpracování prvních dat, které tvoří základní sadu přednášek nabízených vytvořenou webovou službou (viz sekce 5.1). Dále je zde popisován průběh a experimenty během trénování SMM, které byly nejen časově velmi náročné (viz sekce 5.2). V neposlední řadě bylo také nutné zvolit vhodnou metodu pro výpočet podobnosti dvou segmentů za pomoci jejich příznakových vektorů (viz sekce 5.3). Nejdříve je však představen způsob, jakým byla základní sada přednášek získána.

5.1 Sběr dat

V sekci 2.2 byla představena omezení stanovená pro výběr dat a dále bylo uvedeno srovnání různých služeb, které jsou potenciálně vhodné pro získání potřebných videozáznamů. Jako jediný vhodný zdroj videozáznamů byla označena služba YouTube. Z ní bylo ve výsledku získáno 1000 záznamů z přednášek a konferencí, které odpřednášelo 186 přednášejících z celkem 57 společností či univerzit z celého světa.

Pro usnadnění sběru dat byla vytvořena jednoduchá webová služba (viz 4.1), která umožňovala vkládání, upravování a odstraňování posbíraných informací o jednotlivých videozáznamech. Tyto informace byly ukládány do MySQL databáze (více viz 3.1). Po úvodním nasbírání 1 000 odkazů na záznamy z přednášek a konferencí byly informace o těchto videozáznamech předány ve formě CSV souboru k dávkovému zpracování pro účely experimentování členům výzkumné skupiny Speech@FIT. Ti zajistili jednak stažení daných přednášek, ale i vytvoření přepisu řeči. Vzhledem k tomu, že tato práce má za úkol vytvořit automatizovaný proces pro zpracování vstupních dat, bylo nutné zajistit takovou alternativu, která by z tohoto procesu vyloučila zbytečnou součinnost s dalšími osobami, které by výsledný proces mohly zpomalit či ohrozit jeho úspěšné dokončení. Bylo přistoupeno k outsourcování řešení tohoto problému a to za využití některé z volně dostupných cloudových služeb, které se zabývají automatickým rozpoznáním řeči (Google Cloud Speech, Speechmatics či SpokenData).

5.1.1 Struktura prvních přepisů

Z každého získaného videozáznamu byl výzkumnou skupinou Speech@FIT vytvořen přepis řeči, který je uložen v souborech typu .mlf. V tomto souboru je vždy uveden název zvukového segmentu, ze kterého byl vytvořen přepis, a čas začátku a konce daného segmentu (ve stovkách milisekund). Následně jsou postupně na řádcích uvedena jednotlivá rozpoznaná slova společně s dalšími informacemi, které však pro tuto práci nejsou podstatné. Každý

```
#!MLF!#
"0000000audio_0000A_0001231_0001679.rec"
00000000 02600000 <s> 1
02600000 05500000 this 0.983403
05500000 12300000 presentation 0.992179
12300000 13600000 is 0.97592
44600000 44900000 </s> 1
.
"0000000audio_0000A_0019118_0019204.rec"
00000000 02900000 <s> 1
02900000 07900000 that 0.988764
07900000 08700000 </s> 1
.
"0000000audio_0000A_0023252_0023334.rec"
⋮
```

Obrázek 5.1: Ukázka struktury .mlf souboru

segment je ukončen znakem tečka, který je uveden na samostatném řádku za přepsaným segmentem. Ukázka přepisu je na obrázku 5.1.

Takto strukturované přepisy ve formě .mlf souborů jsou využívány pouze u přednášek, jejichž přepis byl vytvořený členy výzkumné skupiny Speech@FIT. Ve vytvořené webové aplikaci jsou přepisy získávány ze služby SpokenData a potřebné informace jsou ukládány ve formě souborů .ts (jejich popis viz sekce 3.5).

5.2 Trénování SMM

Jednou z časově nejnáročnějších částí tvorby této práce bylo zajistit optimální nastavení SMM. Jak je popsáno v sekci 3.2.3, SMM je v této práci používán k redukci dimenze příznakových vektorů, které reprezentují detekované tématické celky z videozáznamů přednášek. Pro práci se SMM je využívána třída SMM z knihovny *subspace_models*, jejímž autorem je Santosh Kesiraju. Konstruktor této třídy je již popsán v sekci 3.10. Dle doporučení autora byla většina atributů konstruktoru ponechána v jejich výchozích hodnotách. Jedinou měněnou hodnotou tedy byla hodnota regularizačního parametru `lT` bázové matice, jež reprezentuje vektorový podprostor v oblasti logaritmické pravděpodobnosti. Na základě tohoto parametru probíhalo testování nastavení SMM.

Aby bylo možné využít SMM pro redukci dimenze vektorů je nejprve zapotřebí daný model natrénovat. K tomu ve třídě SMM slouží metoda `train()`, která má jako parametr cestu ke vstupní matici vektorů, které mají být výsledným modelem redukovány. Cílem trénování je vytvořit N-dimenzionální prostor (kde N je velikost dimenze výstupních vektorů) do kterého budou zaneseny jednotlivé vstupní vektory. Koordináty pozic těchto vektorů v N-dimenzioálním prostoru jsou následně výstupem tohoto modelu. Výsledkem trénovacího procesu jsou soubory s maticemi reprezentující natrénovaný model (viz obrázek 3.2). Natrénovaný model pak dokáže pro vstupní data o stejné dimenzi, jako měla data při trénování, vytvořit vektor, který reprezentuje jejich pozici v N-dimenzionálním prostoru.

Trénování SMM však neprobíhá pokaždé. Pro spuštění trénovacího procesu je zapotřebí spustit hlavní řídící skript `autorun.py` s parametrem `-t`. Při vývoji automatického

procesu, který probíhal na soukromém serveru[1], se při práci se SMM vyskytlo několik problémů, které měly jednoho společného jmenovatele – nedostatek paměti pro výpočet. Tento nedostatek se vyskytl jak při trénování, tak i při extrakci dat. Trénování samo o sobě je jak časově, tak i paměťově náročné, jelikož trénování musí probíhat nad celou vstupní maticí příznakových vektorů. Například pro vstupní matici o 33 000 řádcích a 6 000 sloupcích je zapotřebí až 12 GB paměti a trénování zabere asi 40 minut. Z tohoto důvodu je počítáno s přetrénováním asi dvakrát za rok. Závisí však na počtu nových záznamů z přednášek, které budou do systému vloženy. Řešením nastalých komplikací bylo provedení trénování na školních serverech a u extrakce rozdělení vstupní matice na části po 1 000 dokumentech a získání výsledné matice spojením výstupů z extrakce jednotlivých částí vstupní matice příznakových vektorů.

Po vyřešení problémů s trénováním a extrakcí dat ze SMM bylo nutné zajistit jeho korektní nastavení a natrénování. Po konzultacích s autorem SMM bylo zjištěno, že jediná hodnota, která má zásadní vliv na výsledky produkované SMM je regularizační parametr `lT` bázové matice, která reprezentuje vektorový podprostor v oblasti logaritmické pravděpodobnosti. Obecně se dá říci, že čím řidší jsou vstupní data, tím vyšší by měla být hodnota parametru `lT`. Byla tedy provedena série trénování SMM a to za účelem nalezení optimálního nastavení modelu. K ověřování správnosti natrénovaného modelu z něj bylo zapotřebí získat výstupní matici vektorů a vypočítat podobnost vektorů v této matici. Pro výpočet podobnosti byly uvažovány metody euklidovské vzdálenosti a cosine similarity. Jejich srovnání je uvedeno v sekci 5.3.

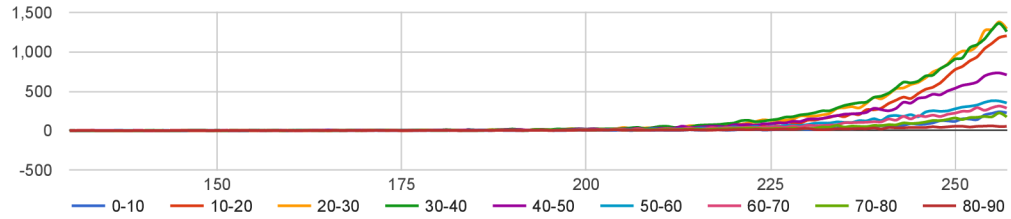
Při každém trénování byly podobnosti vektorů vypočítávány pomocí obou metod a výsledky byly následně uloženy do databáze. Výsledná matice s výsledky obsahovala více než 21 000 000 položek. Při vykreslování výsledků, které byly vykreslovány za použití PHP skriptu a JavaScriptové třídy GoogleCharts, ovšem nebylo časově ani výpočetně možné tato data plně vykreslit. Z toho důvodu se přistoupilo k vykreslení 120 000 segmentů s nejlepšími výsledky porovnání pro danou metodu. Jednotlivé grafy uvedené na obrázcích 5.2, 5.3 a 5.4 reprezentují histogramy četností hodnot podobností. Snahou bylo nalézt takové nastavení SMM, které by eliminovalo nedostatky, které se projevily při hledání podobností v práci Marka Sychry [9]. Zde byly časově delší segmenty při porovnávání zvýhodňovány. Na uvedených grafech by se takováto chyba projevila tak, že by střední hodnoty histogramů jednotlivých skupin segmentů byly vzájemně posunuté. Z toho důvodu bylo hledáno takové nastavení SMM, kde by histogramy jednotlivých kategorií v grafu měly stejnou střední hodnotu. Proběhlo tedy testování s hodnotami `lT = 100`, `lT = 1` a `lT = 1e-2`. Dle výsledků testování a jejich konzultaci s vedoucím byla jako nejlepší hodnota parametru `lT` pro SMM zvolena hodnota `1e-2`. Dimenze výsledných vektorů byla nastavena na hodnotu 300. Zvolená metoda výpočtu podobnosti vektorů je uvedena v sekci 5.3.2.

5.3 Srovnání výpočtů podobností vektorů

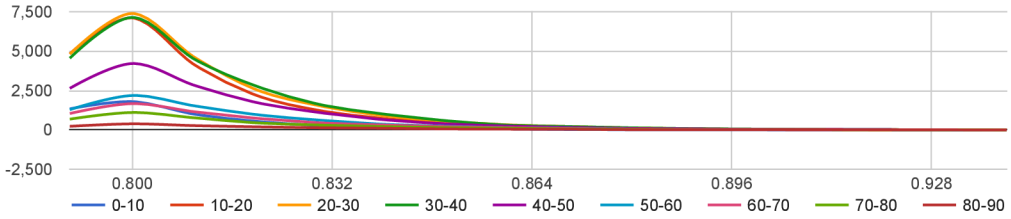
Pro měření podobnosti existuje velké množství koeficientů a metod. Některé z nich se doporučuje používat pro úlohy z oblasti zpracování řeči. Uvažovaní zástupci metod a koeficientů pro výpočet podobnosti dvou veličin (v tomto případě vektorů) tvoří Euklidovská vzdálenost a Cosine similarity. Na doporučení autora SMM bylo pro zjištění podobností uvažováno o použití euklidovské vzdálenosti. Naopak vedoucí práce se přikláněl ke cosine similarity.

Nezanedbatelná část času, která byla vymezena pro zpracování této práce, byla tedy

[1] 4GB RAM, CPU AMD FX-4130 (3.8GHz)

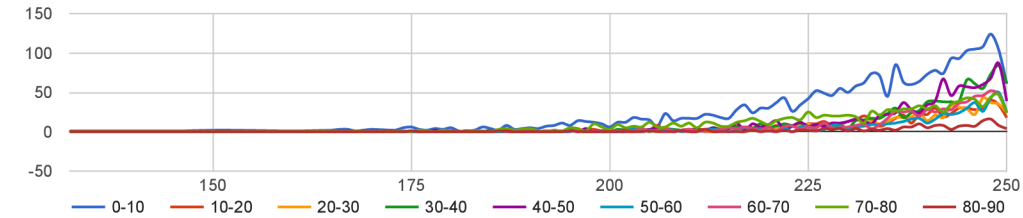


(a) Výsledky za použití euklidovské vzdálenosti

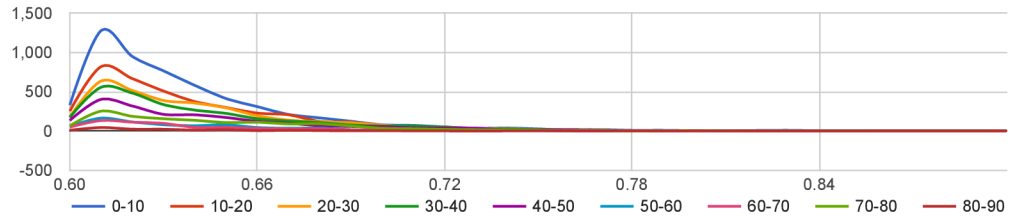


(b) Výsledky za použití cosine similarity

Obrázek 5.2: Výsledky trénování SMM pro hodnotu `lT = 100` pro 120 000 nejlepších výsledků u každé metody. Osa X označuje vypočtenou podobnost segmentu, osa Y určuje počet segmentů a každá barva reprezentuje skupinu segmentů dle jejich doby trvání v minutách.



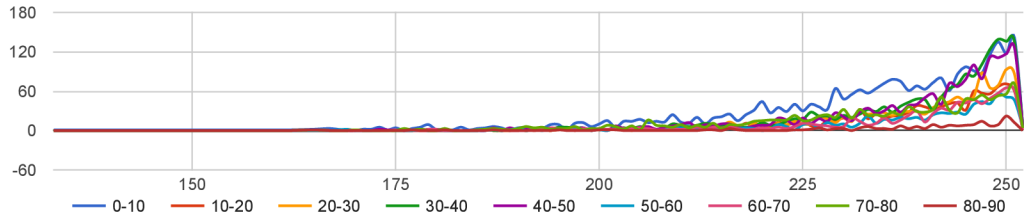
(a) Výsledky za použití euklidovské vzdálenosti



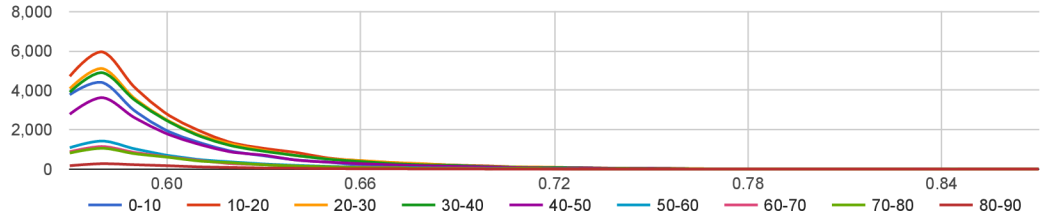
(b) Výsledky za použití cosine similarity

Obrázek 5.3: Výsledky trénování SMM pro hodnotu `lT = 1` pro 120 000 nejlepších výsledků u každé metody. Osa X označuje vypočtenou podobnost segmentu, osa Y určuje počet segmentů a každá barva reprezentuje skupinu segmentů dle jejich doby trvání v minutách.

věnována otázce, která z doporučených metod má být pro porovnání vektorů detekovaných segmentů použita. Bylo tedy nezbytné provést testování a následné vyhodnocení výsledků.



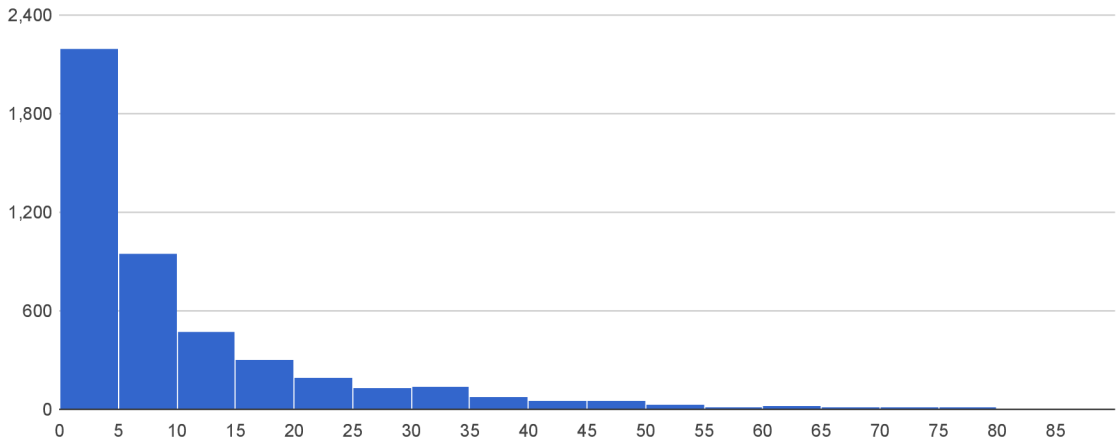
(a) Výsledky za použití euklidovské vzdálenosti



(b) Výsledky za použití cosine similarity

Obrázek 5.4: Výsledky trénování SMM pro hodnotu `1T = 1e-2` pro 120 000 nejlepších výsledků u každé metody. Osa X označuje vypočtenou podobnost segmentu, osa Y určuje počet segmentů a každá barva reprezentuje skupinu segmentů dle jejich doby trvání v minutách.

Pro obě porovnávané metody byly vstupem vektory reprezentující detekované segmenty. Na obrázku 5.5 je uveden histogram délek detekovaných segmentů v minutách. Těchto segmentů bylo celkem detekováno 4 596.



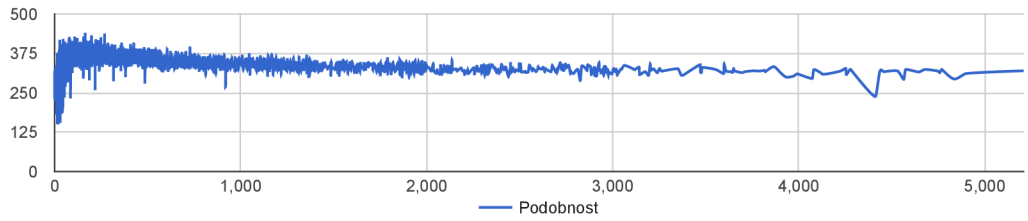
Obrázek 5.5: Histogram délek segmentů. Osa X určuje délky segmentů v minutách a osa Y určuje počet segmentů. Jeden sloupec v grafu znázorňuje 5 minutový bin.

5.3.1 Euklidovská vzdálenost

Míry podobnosti v ideálním případě nabývají hodnot od nuly – pro maximální rozdílnost, po jedničku – pro maximální totožnost. Tato metoda je však založena na míře nepodobnosti čili vzdálenosti. Pro vyjádření vztahu dvou objektů se používají míry vzdálenosti, které jsou založeny na prezentaci objektů v prostoru, jehož souřadnice představují jednotlivé proměnné. [12] Jedním z nejznámějších typů vzdálenosti je právě euklidovská vzdálenost, která byla původně užívána pro měření vzdálenosti ve dvourozměrných či trojrozměrných prostorech. Prostým zobecněním ji však lze rozšířit i do vícerozměrných prostorů. Vzorec pro euklidovskou vzdálenost je uveden v rovnici 5.1.

$$d_E(u, v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_n - v_n)^2} \tag{5.1}$$

K výpočtům pomocí této metody byla využita funkce *spatial.distance.euclidean* z knihovny *scipy*, které byly zadávány vektory k porovnání. Z grafu na obrázku 5.4(a), který reprezentuje výsledné nastavení SMM, vyplývá relativně chaotické zastoupení podobností v každém měřeném intervalu, které by se nedaly přirovnat k normálnímu rozložení reprezentovanému Gaussovou křivkou. To má za následek zvýhodnění některých segmentů z dané skupiny. Na obrázku 5.6 je uveden graf, který znázorňuje podobnost náhodného segmentu vůči všem ostatním segmentům. Náhodný segment byl vybrán z množiny 20 náhodně zvolených segmentů, pro které byly vytvořeny stejné grafy, jako je uveden na obrázku 5.6. U všech segmentů z dané množiny dopadl výsledek obdobně jako u uvedeného náhodného segmentu. Z grafů vyplývalo zvýhodnění kratších segmentů na úkor delších, neboť pro vizualizaci je bráno pět nejlepších výsledků (výsledky s nejnižší hodnotou) pro každý segment. Nevýhodou této metody je i absence jakékoli normalizace výsledku či vstupních dat.



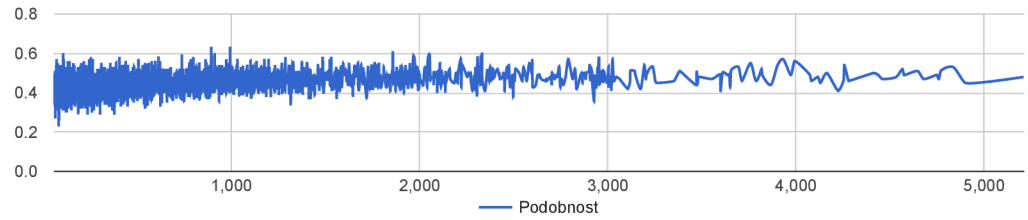
Obrázek 5.6: Graf podobností náhodného segmentu vůči ostatním za použití euklidovské vzdálenosti. Osa X určuje délku porovnávaných segmentů v sekundách a osa Y určuje vypočtenou podobnost segmentů.

5.3.2 Cosine similarity

Cosine similarity, kterou je možné v literatuře nalézt i pod slovním spojením *Kosinova míra*, vyjadřuje kosinus úhlu, který svírají vektory charakterizující zkoumané objekty. Čím více jsou si dané vektory podobné, tím menší je jimi svíraný úhel a tím větší je hodnota výsledného kosinu daného úhlu. [12] Výsledek této metody nabývá hodnot od nuly – pro naprosto odlišné vektory, po jedničku – pro naprosto identické hodnoty. Výpočet je prováděn pomocí rovnice 5.2.

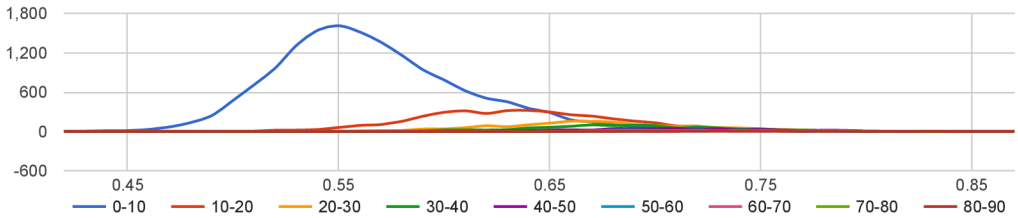
$$S_K(u,v) = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i^2 \cdot \sum_{i=1}^{n} v_i^2}} \tag{5.2}$$

Graf s výsledky této metody pro výsledné nastavení SMM je uvedeno na obrázku 5.4(b). Výhodou je, že střední hodnota je pro všechny měřené intervaly stejná a podobnosti v jednotlivých intervalech odpovídají normálnímu rozložení. Na obrázku 5.7 je uveden graf, který znázorňuje podobnost náhodného segmentu vůči všem ostatním segmentům. Náhodný segment byl vybrán z množiny 20 náhodně zvolených segmentů, pro které byly vytvořeny stejné grafy, jako je uveden na obrázku 5.7. U všech segmentů z dané množiny dopadl výsledek obdobně jako u uvedeného náhodného segmentu. Z grafů nevyplývalo zvýhodnění kratších segmentů, ale relativně rovnoměrné zastoupení napříč všemi délkami. Pro vizualizaci je totiž bráno pět nejlepších výsledků (výsledky s nejvyšší hodnotou) pro každý segment. Tato metoda normalizuje výsledky do intervalu od 0 do 1 a byla pro své výsledky vybrána ke zjišťování podobností jednotlivých segmentů.



Obrázek 5.7: Graf podobnosti náhodného segmentu vůči ostatním za použití cosine similarity. Osa X určuje délku porovnávaných segmentů v sekundách a osa Y určuje vypočtenou podobnost segmentů.

Na obrázku 5.8 je znázorněno rozložení podobností mezi jednotlivými segmenty daných délek. Graf je tvořen pěti nejlepšími hodnotami pro každý detekovaný segment. Z grafu vyplývá, že kratší segmenty dosahují obecně menší podobnosti s ostatními segmenty, což je v případě obsahové odlišnosti porovnávaných segmentů správné. Zároveň by to ale mohlo indikovat přetrvávání problému, který se vyskytl už v práci Marka Sychry [9]. Podstatné jsou ale výsledky testování od uživatelů, kteří mohou lépe odhalit případnou chybu.



Obrázek 5.8: Graf rozložení podobností mezi segmenty jednotlivých délek. Osa X určuje míru podobnosti, osa Y určuje počet segmentů s danou podobností a každá barva reprezentuje skupinu segmentů dle doby jejich trvání.

Kapitola 6

Testování

Velmi podstatnou částí při tvorbě jakékoli aplikace je otestování dosažených výsledků a jejich vyhodnocení. S určitou nadsázkou by se dalo hovořit o dvojím testování, kterým tato práce prošla. Hlavní význam však mají výsledky klasického testování reálnými uživateli.

6.1 Excel@FIT

První formou testování, kterou tato práce prošla, byla účast na studentské konferenci Excel@FIT. Důvodem této účasti byla jednak možnost představit dané téma a prezentovat je, ale také možnost získat zpětnou vazbu na tuto práci, a to nejen od odborníků z řad zástupců úspěšných firem z oblasti IT, ale také od studentů a akademických pracovníků fakulty. Na konferenci byla tato práce prezentována formou příspěvku v elektronickém sborníku[1] a plakátu[2]. V den konference poté bylo možné u stanoviště, které bylo dané práci přiřazeno, diskutovat s návštěvníky o představeném řešení a o jeho možném zlepšení. V době prezentace této práce na konferenci byla stále jako referenční metrika pro zjišťování podobností segmentů využívána euklidovská vzdálenost. Z uskutečněných rozhovorů vyplynulo, že uvedený proces řešení tohoto problému je teoreticky správný. Návštěvníky z řad akademických pracovníků, převážně z ústavu počítačové grafiky a multimédií, bylo zároveň doporučeno využití jiné metriky pro porovnávání vektorů detekovaných segmentů, a to konkrétně cosine similarity. Co se týče faktické správnosti řešení, její ověření bude záležet na výsledcích z testování mezi uživateli.

6.2 Hlasování uživatelů

Druhou a nejvíce vypovídající formou o faktické správnosti řešení bylo testování mezi reálnými uživateli. Cílem tohoto testování nebylo otestování uživatelské přívětivosti či intuitivnosti ovládání vytvořené webové aplikace. Cílem bylo zjištění, zda jsou k jednotlivým segmentům z videozáznamů přiřazeny jako podobné části z jiných videí takové segmenty, které jsou s vybraným segmentem tématicky podobné či nikoliv. Z grafu na obrázku 5.8 je přitom patrné, že například skoro všechny segmenty s délkou od 0 do 10 minut jsou se všemi ostatními podobné v průměru na 55%, tedy že jsou si asi z poloviny podobné. Z toho

[1] dostupné on-line z `http://excel.fit.vutbr.cz/submissions/2016/047/47.pdf`, verze s opravenou chybou citace je na přiloženém CD

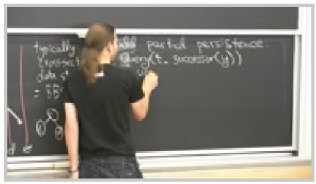
[2] dostupné on-line z `http://excel.fit.vutbr.cz/submissions/2016/047/47_poster.pdf` a na přiloženém CD

lze vyvozovat že například u segmentů z této kategorie se bude vyskytovat více názorů, které nalezené spojení nebudou označovat jako správné.

Pro účely tohoto testování byla webová aplikace mírně upravena a doplněna o hlasovací stránku, jejímž prostřednictvím mohli uživatelé provádět samotné testování. První potřebnou změnou, která se však ve vizuální části práce nikterak neprojevila, byla nutnost doplnění obsahu tabulky `linking`. Tato tabulka obsahovala pouze pět nejlepších výsledků ke každému detekovanému segmentu. Bylo tedy zapotřebí doplnit i zástupce, kteří by reprezentovali jednotlivé části vypočtených výsledků, a to od nejlepších po nejhorší. Z matice vypočítaných podobností bylo tedy ke každému segmentu spočítáno rozpětí spočtených podobností[3] a následně bylo toto rozpětí rozděleno na pětiny, kde z každé části by do databáze uložen jeden reprezentativní vzorek. Všechny záznamy v tabulce `linking` byly zároveň ohodnoceny číslem 1 až 5, kde 1 znamená nejpodobnější segmenty a 5 nejméně podobné segmenty. Na základě těchto ohodnocení bylo poté vytvářeno prostředí pro testování správnosti nalezených podobných segmentů. První změna, která byla ve webové části aplikace pro uživatele provedena, byla na stránce pro přehrávání.

Na obrázku 4.14 je znázorněna stránka pro přehrávání jednotlivých záznamů. Na této stránce bylo upraveno generování detekovaných segmentů, které se pod přehrávaným videozáznamem vyskytují, konkrétně byly doplněny o odkaz *Test of similarity,* který zajišťoval přechod na testovací stránku pro daný segment. Nová podoba těchto bloků je znázorněna na obrázku 6.1.

Segment 1
Test of similarity
From: 00:00:10 To: 00:09:25

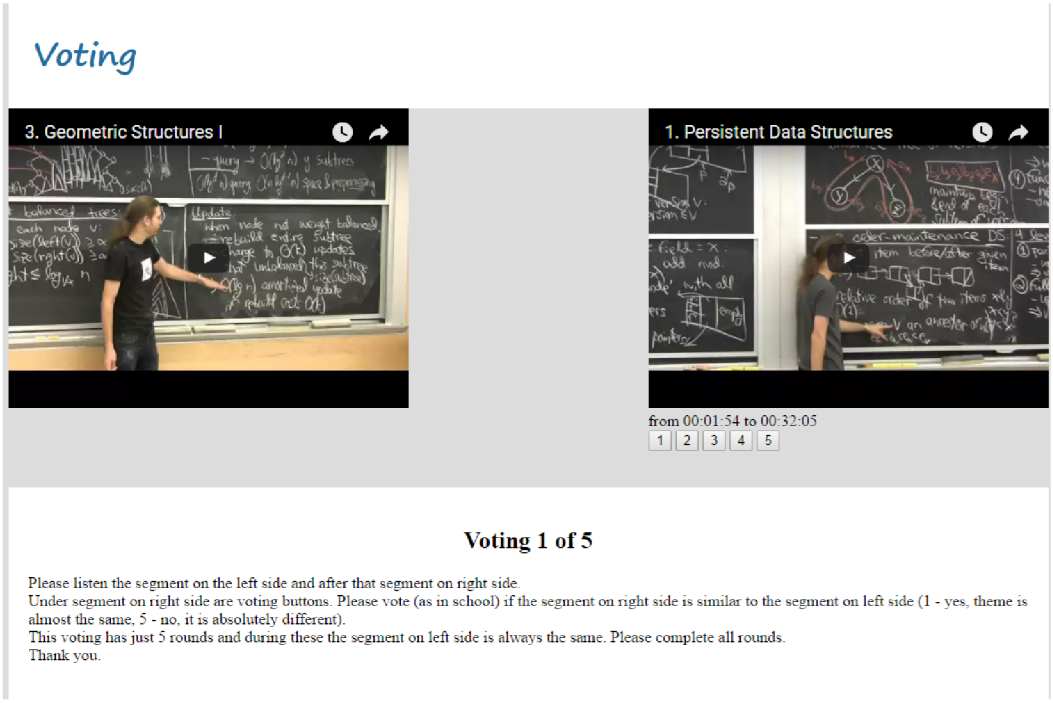

Segment 2
Test of similarity
From: 00:09:25 To: 00:17:12

Obrázek 6.1: Ukázka nově generovaných bloků s detekovanými segmenty z videozáznamu

Druhou změnou bylo vytvoření samotné stránky pro testování. Ukázka této stránky je uvedena na obrázku 6.2. Tato stránka zajišťovala přehrání videozáznamů dvou segmentů. První videozáznam, který byl umístěn na levé straně generované stránky, reprezentoval segment, u kterého bylo kliknuto na odkaz pro testování. Po celou dobu testování daného segmentu byl tento záznam stejný. Druhý videozáznam, který byl umístěn na pravé straně generované stránky, reprezentoval jeden ze segmentů, které byly ke zvolenému segmentu přiřazeny v tabulce `linking`. Pod videozáznamem na pravé straně byla navíc umístěna hlasovací tlačítka, která sloužila pro oznámkování podobnosti těchto představených segmentů. Po udělení známky byl uživatel přesměrován do následujícího hlasovacího kola. Uživatel byl takto požádán o absolvování pěti kol hlasování, při kterých měl za úkol vyslechnout oba segmenty a zvolit, jak moc se segment na pravé straně tématicky blíží segmentu na levé straně.

Algoritmus pro výběr segmentů, které se nabízely k porovnání pro referenční segment, byl následující. Z tabulky `linking` byl nejdříve vybrán takový záznam, který pro zvolený segment dosahoval teoretické známky 1. V každém dalším kole byly vybírány segmenty,

[3] bráno jako rozdíl nejlepšího výsledku a nejhoršího výsledku



Obrázek 6.2: Ukázka stránky pro testování

jejichž teoretická známka odpovídala pořadí daného kola v sérii hlasování. Takto uživatel určoval nejen, zda jsou jako podobné označeny správné výsledky, ale také zda jsou při výpočtech ignorovány takové výsledky, které s daným segmentem opravdu tématicky nesouvisí.

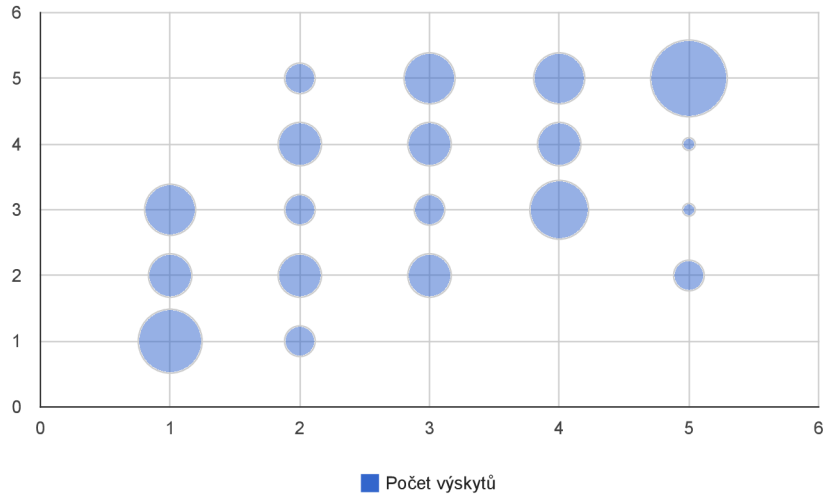
Pro testování byli osloveni jednak studenti, a to prostřednictvím facebokové stránky, dále také zaměstnanci fakulty a další osoby, které pracují v oboru informačních technologií.

6.3 Výsledky

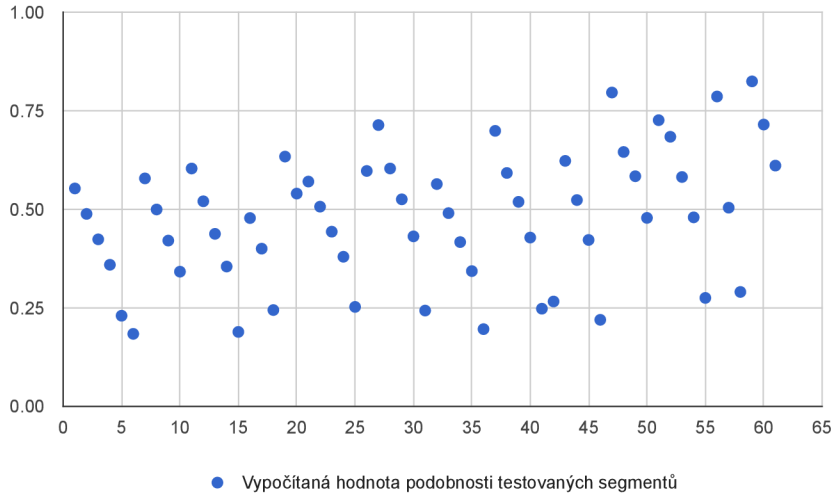
Na obrázku 6.3 jsou uvedeny výsledky z provedeného testování.

Testování proběhlo na náhodném vzorku segmentů. Celkem bylo provedeno 61 porovnání. Z nich vyplývá, že oproti předpokládaným hodnotám ohodnocení podobností došlo k odchylkám. Z uvedeného grafu vyplývá, že ve většině případů měli uživatelé na míru podobnosti v určitých případech odlišný názor, než jaký systém předpokládal. Při žádném porovnávání segmentů s nejlepším výsledkem se ovšem nestalo, že by odhadované ohodnocení bylo uživateli označeno jako naprosto opačné. Je důležité si také uvědomit, jakých hodnot vypočítaných podobností nabývaly testované segmenty. Na obrázku 6.4 jsou pro jednotlivé segmenty uvedeny i jejich vypočítané míry podobnosti.

Z tohoto grafu je patrné, že v případě nastavení ohodnocení podle absolutní hodnoty míry podobnosti daného segmentu (míra podobnosti vůči intervalu 0 až 1), by s velkou pravděpodobností k tolika odchylkám při testování nedošlo. U všech segmentů, které se mohly účastnit testování bylo totiž využito relativního ohodnocení vzhledem k intervalu, který byl ohraničen nejlepší a nejhorší vypočítanou mírou podobnosti každého segmentu. Například ohodnocení stupněm 1, tedy že testované segmenty jsou naprosto tématicky podobné, dosáhly jak segmenty, jejichž vypočítaná míra podobnosti dosahovala hodnoty kolem 0.58, tak i segmenty, které dosahovaly vypočítané míry podobnost okolo 0.82.



Obrázek 6.3: Graf ukazující podobnost referenčních ohodnocení a výsledky testování. Osa X znázorňuje výsledky, které byly od testujících uživatelů očekávány systémem. Osa Y určuje hodnoty, které uživatelé při testování skutečně zadali. Velikost bodu je přímo úměrná počtu odpovědí.



Obrázek 6.4: Graf vypočítaných podobností jednotlivých segmentů. Osa X označuje pořadí porovnávaného segmentu. Osa Y označuje vypočítanou míru podobnosti daného segmentu.

Při přihlédnutí k možnosti, že danou odchylku můžou z velké části ovlivnit jak uživa-

telé, jelikož každý má své subjektivní hodnocení podobnosti[4], tak i kvalita automatického přepisu mluveného slova, která nebyla vždy ideální, lze získané údaje z testování označit za přijatelné. Výsledný automatický proces pro zpracování videozáznamů lze považovat za funkční vůči požadavkům ze zadání práce, neboť aplikace dokáže smysluplně detekovat témata z videozáznamů, segmentovat je a při přehrávání k nim nabídnout relativně podobné části z ostatních videozáznamů. Výsledky testování jsou i přesto důležitým ukazatelem směru dalšího vývoje a zlepšení aplikace.

[4]pro některé už jen fakt, že úryvek k porovnání byl z jiného kurzu znamenalo uvedení hodnocení podobnosti daných segmentů jako odlišné

Kapitola 7

Závěr

Tento text popisuje, jakým způsobem, s jakými prostředky a s jakými výsledky byla tato diplomová práce, která se zabývá automatickou detekcí témat, segmentací a vizualizací online kurzů, vypracována. Cílem této práce bylo implementovat automatický proces, který by zajistil hledání tématicky podobných částí ve videozáznamech. Při implementaci automatického procesu zpracování videozáznamů z přednášek v jazyce Python bylo využito několika dostupných technologií. První je skript pro segmentaci textu dle témat, který vznikl jako výstup bakalářské práce, která se zabývala automatickým hledáním vazeb mezi částmi audiovizuálních dokumentů. [9] Autorem této práce je Bc. Marek Sychra. Druhá použitá technologie zajišťující redukci dimenze vektorů je od Santoshe Kesirajua, který je autorem třídy SMM pro práci se Subspace multinomial modelem. [2] Dalším nástrojem, který je v této práci využíván je volně dostupná knihovna Apache Lucene, která slouží pro indexování textových dat a jejich následné prohledávání.

Tato práce se krom detekce a segmentace zabývala také vizualizací výsledků uživatelům. Vizualizace je zajištěna webovou aplikací napsanou v PHP, která využívá JavaScriptu s podporou knihovny jQuery pro zajištění interaktivních změn na stránce a komunikaci se serverem pomocí AJAXu. Výsledná webová prezentace je dostupná na internetové adrese `http://superlectures.net`.

Práce byla také představena na studentské konferenci Excel@FIT a to formou příspěvku v elektronickém sborníku a prezentací plakátu na přehlídce přijatých prací. Pro výsledný výpočet podobnosti vektorů, které reprezentují jednotlivé detekované segmenty z videozáznamů, bylo dle dosažených výsledků z testování použitelných metod přistoupeno k použití metody cosine similarity.

Z výsledků testování na uživatelích se projevila odchylka od systémem očekávaných hodnot. Tato odchylka je akceptovatelná a mohla být způsobena některou z převzatých komponent, které byly v procesu použity a také způsobem hodnocení podobnosti testujícími uživateli. Velký vliv na výsledné vektory jednotlivých segmentů má nejen výsledek automatického přepisu řeči, ale i segmentační skript a také nastavení SMM, který zajišťuje redukci vektorů reprezentujících jednotlivé detekované segmenty. Je nutné brát na zřetel, že získané automatické přepisy nejsou vždy v pořádku. Vyskytly se případy, kdy kvůli rychlému řečovému tempu přednášejícího, neobsahoval výsledný přepis prakticky žádný přepsaný text. V těchto případech je tedy nemožné najít odpovídající podobné segmenty z jiných videozáznamů, ba ani není možné provést správnou segmentaci. Výsledný vytvořený proces i webová služba jsou funkční a splňují podmínky uvedené v zadání práce.

Dalším krokem ve vývoji této práce by měla být taková úprava komponent algoritmu, aby se problém projevený při testování odstranil či vykompenzoval. Řešením by mohlo

být přidání informace o míře podobnosti k jednotlivým segmentům. Druhou možností je kontrola korektní segmentace, která by v případě problémů byla znovu implementována, či zajištění přepisů člověkem. Při dalším vývoji je také nutné zlepšit grafickou část webové prezentace. Webová služba je z důvodu nedostatku času, který byl věnován experimentům se SMM, prezentována v graficky nedokonalém stavu. Funkční stránka aplikace je však plně dokončena. Při budoucím vývoji je také zapotřebí vytvořit automatický proces pro přidávání nových záznamů z přednášek do indexu pro vyhledávání.

Posledním bodem zadání této práce je vytvořit alespoň půlminutové video a A2 plakát prezentující dosažené výsledky (viz příloha H). Výsledný videozáznam i plakát jsou k dispozici na přiloženém CD.

Literatura

[1] CHAFFER, J.; SWEDBERG, K.; BAŠE, K.; aj.: *Mistrovství v jQuery: kompletní průvodce vývojáře.* vyd. 1, Brno: Computer Press, 2013, ISBN 978-80-251-4103-8.

[2] KESIRAJU, S.; BURGET, L.; SZŐKE, I.; aj.: Learning document representations using subspace multinomial model, 2016, odesláno do InterSpeech 2016.

[3] LUTZ, M.; ASCHER, D.: *Naučte se Python.* Vyd. 1, Praha: Grada publishing, 2003, ISBN 80-247-0367-X.

[4] MLČÁKOVÁ, R.: Komunikace s jedinci s narušenou expresivní či impresivní složkou řeči (v oblasti řečové produkce a rozumění) [online]. *Specifika komunikace s jedinci se speciálními vzdělávacími potřebami*, 2012: s. 57–62.
Dostupné z: http://kurzy-spp.upol.cz/CD/4/3-04.pdf

[5] Quizlet: *1. přednáška* [online]. [cit. 2016-05-02].
Dostupné z: https://quizlet.com/18355286/1-prednaska-flash-cards/

[6] SOMMERVILLE, I.: *Softwarové inženýrství.* Vyd. 1, Brno: Computer Press, 2013, ISBN 978-80-251-3826-7.

[7] SOUFIFAR, M.; KOCKMANN, M.; BURGET, L.; aj.: iVector Approach to Phonotactic Language Recognition. In *Proceedings of Interspeech 2011*, ročník 2011, International Speech Communication Association, 2011, ISBN 978-1-61839-270-1, ISSN 1990-9772, s. 2913–2916.
Dostupné z: http://www.fit.vutbr.cz/research/view_pub.php?id=9758

[8] Sphinx: *About | Sphinx* [online]. [cit. 2016-05-05].
Dostupné z: http://sphinxsearch.com/about/sphinx/

[9] SYCHRA, M.: *Automatické hledání vazeb mezi částmi audiovizuálních dokumentů.* Diplomová práce, Brno, FIT VUT v Brně, 2015.

[10] Wikipedia: *Python* [online]. [cit. 2015-12-14].
Dostupné z: https://cs.wikipedia.org/wiki/Python

[11] ZAKAS, N. C.: *JavaScript pro webové vývojáře: programujeme profesionálně.* Vyd. 1., Brno: Computer Press, 2009, ISBN 978-80-251-2509-0.

[12] ŘEZANKOVÁ, H.; HÚSEK, D.; SNÁŠEL, V.: *Shluková analýza.* Vyd. 2, Praha: Professional publishing, 2009, ISBN 978-80-86946-81-8.

Přílohy

Seznam příloh

Příloha A

Obsah CD

- /aplikace/ – zdrojové soubory aplikace
- /excel/ – příspěvek a plakát publikované na studentské konferenci Excel@FIT
- /media/ – videozáznam použití webové části aplikace a vytvořený plakát
- /zprava/ – zdrojové kódy písemné zprávy
- /xridky00-DP.pdf – elektronická verze písemné zprávy
- /sqlInit.sql – inicializační soubor databázových tabulek
- /readme.txt – návod k instalaci a použití

Příloha B

Seznamy v databázi

B.1 Kategorie vytvořené pro záznamy z on-line kurzů

- Computational Neuroscience
- Computer Graphics
- Computer Networks
- Computer Science
- Computer System Security
- Computer Systems
- Databases
- Hardware
- Intelligent Systems
- Math
- Mobile and Web
- Program Languages
- Programming
- Signal Processing

B.2 Seznam fází automatického procesu zpracování dat

- new
- uploading-to-spokendata
- wait-for-transcript
- in-queue
- processing
- error
- done

Příloha C

Výsledný XML soubor s přepisem řeči ze služby SpokenData

```
<?xml version="1.0"encoding="utf-8"?>
<data>
    <segment>
        <start>2.21</start>
        <end>19.43</end>
        <speaker>B</speaker>
        <text>Hi I'm Diane Dimond in the land. And this is my new book.
        Small talk big results should check out your way to success. Today.
        I'm commemorating the launch of my book and you can order it
        on Amazon dot com or my website. Small talk big results.</text>
    </segment>
    <segment>
        <start>19.6</start>
        <end>21.85</end>
        <speaker>AB</speaker>
        <text>But first I need to launch it.</text>
    </segment>
        ⋮
    <segment>
        <start>34.48</start>
        <end>45.49</end>
        <speaker>A</speaker>
        <text>My book didn't go that far. On this lunch. But I hope
        they you will go far and do you learn some tips and techniques
        to should check your way to success.</text>
    </segment>
</data>
```

Příloha D

Ukázka struktury soubor .ts

```
# 1943
Hi
I'm
Diane
Dimond
in
the
land
And
this
is
my
new
book
.
# 2185
But
first
I
need
to
launch
it
.
```

Příloha E

Ukázky struktur transformovaných souborů

E.1 Soubor .word

```
this presentation is delivered by the singer for professional development
that
all right
so it's time consuming to ramble
that stops now um
it's pretty generic
so
the easiest way of doing that is just to look at all of them and see if he's
one in turn is
but this
so
so here's
⋮
```

E.2 Soubor .meta

```
74 00:00:16
79 00:03:12
89 00:03:53
122 00:03:55
140 00:03:58
160 00:04:40
163 00:05:38
255 00:05:48
264 00:06:15
267 00:06:29
277 00:07:07
282 00:07:12
⋮
```

Příloha F

Výstup ze skriptu pro segmentaci textu

Obsah jednoho souboru:

talk000000001 00:00:37,00:06:03
om like of research ah she lecture on some other topics like what security later on a bike will decide later was going on exactly equal electricity bill have forty days this year uh Steven well ahah on and games and uh hopefully you'll need a and um office hours are over the year if you need a health
⋮
the protesters
okay
what it means to try to build a secure system um because th

Příloha G

XML soubor s výsledky vyhledávání

```
<searcher>
    <query>
        <user>trees</user>
        <words>
            <expandedword str="trees" >tree</expandedword>
        </words>
        <system>
            +conference:superlectures +(speech:tree title:tree
            author:tree abstract:tree slides:tree category:tree
            keywords:tree)
        </system>
    </query>
    <query_parsing_time>0.056</query_parsing_time>
    <searching_time>0.409</searching_time>
    <docpage size="10" >1</docpage>
    <occpage size="1000" >1</occpage>
    <documents found="723" >
        <doc name="talk000000747"doc_id="738"length="..."conference="..."score="..." >
            <field name="abstract"occurrences="1" >
                <occurrences>
                    <occ start="11"end="12"text="Static ^ trees$: least common"/>
                </occurrences>
            </field>
            ⋮
```

```
            ⋮
            <field name="title"occurrences="1" >
                <occurrences>
                    <occ start="1"end="2"text="Static ^ Trees$"/>
                </occurrences>
            </field>
            <field name="speech"occurrences="85" >
                <occurrences>
                    <occ start="27.0"end="28.0"score="0.983048"/>
                    ⋮
                    <occ start="4875.5"end="4876.0"score="0.994295"/>
                </occurrences>
            </field>
        </doc>
        <doc>
        ⋮
        </doc>
    </documents>
    <printing_results_time>0.318</printing_results_time>
    <total_time>0.82</total_time>
</searcher>
```

Příloha H

Plakát