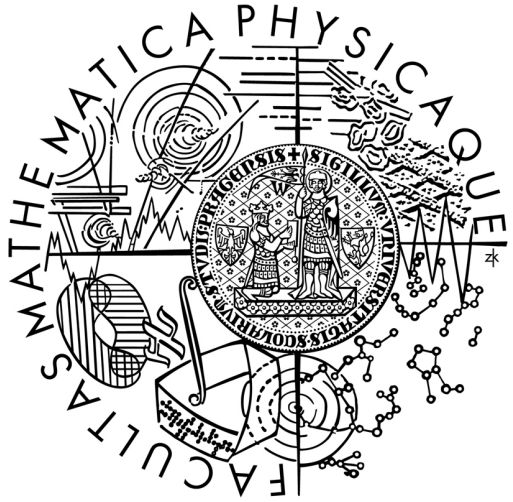
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Pavel Češka

Vyhledávání v nesegmentované mluvené řeči

Ústav formální a aplikované lingvistiky

Vedoucí práce: Mgr. Pavel Pecina
Studijní program: Informatika, Matematická lingvistika

2008

Děkuji vedoucímu své diplomové práce Mgr. Pavlu Pecinovi za podporu a cenné rady udělované v průběhu psaní práce.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

Ve Zlíně dne 4. srpna 2008 Pavel Češka

Obsah

1 Úvod **8**
1.1 Cíle diplomové práce 9
1.2 Shrnutí 10

2 Vyhledávání informací v textu **11**
2.1 Motivace 11
2.2 Formalizace problému 12
2.3 Vyhledávací modely 13
2.3.1 Základní boolský model 13
2.3.2 Vektorový model TF–IDF 14
2.3.3 Okapi BM25 15
2.3.4 Model INDRI 16
2.4 Zpracování dokumentů a témat 17
2.4.1 Normalizace slovních tvarů 18
2.4.2 Nevýznamová slova 18
2.5 Rozšiřování vyhledávacího dotazu 19
2.5.1 Zpětná vazba od uživatele 19
2.5.2 Slepá zpětná vazba 19
2.6 Evaluace 20
2.6.1 Přesnost a úplnost 20
2.6.2 F-measure 21
2.6.3 Průměrná přesnost 21

3 Popis dat **23**
3.1 Video záznamy 23
3.2 Automatické rozpoznání řeči 23
3.3 Témata pro vyhledávání 25
3.4 Hledání relevantních pasáží 27

4 Vyhledávání informací v mluvené řeči **29**
4.1 Reformulace problému . . . 30
4.2 Tvorba textových dokumentů . . . 30
4.3 Překryv relevantních pasáží označených různými anotátory . . . 31
4.4 Evaluace . . . 32
4.5 Wilcoxonův párový test . . . 34

5 Nástroje pro práci s kolekcí **36**
5.1 Zpracování dokumentů . . . 36
5.1.1 Příprava dat . . . 36
5.1.2 Morfologická analýza a lemmatizace . . . 38
5.1.3 Porovnání kvality dokumentů . . . 39
5.1.4 Skript pro vytvoření kolekce . . . 41
5.2 Zpracování témat . . . 44
5.3 Evaluační skript . . . 46

6 Experimenty **47**
6.1 Rozdělení témat . . . 47
6.2 Vyhledávací nástroje . . . 49
6.3 Základní experiment . . . 49
6.4 Experiment s lemmatizací . . . 50
6.5 Experimenty s různými vyhledávacími modely . . . 50
6.6 Experiment s rozšiřováním vyhledávacího dotazu . . . 52
6.7 Experiment s volbou různých ASR systémů a kanálů . . . 55
6.8 Experiment s použitím stoplistu . . . 57
6.9 Experiment s různou délkou generovaných dokumentů . . . 64
6.10 Experiment s podrobnějším popisem tématu . . . 68
6.11 Závěrečné experimenty na testovacích datech . . . 69

7 Závěr **74**

Literatura **76**

A Obsah přiloženého CD **79**

Seznam tabulek

4.1 Počet relevantních pasáží pro každé téma 33
4.2 Wilcoxonův párový test 35

5.1 Přehled počtu pásek v jednotlivých krocích 37
5.2 Kompletnost rozhovorů 37
5.3 Výstupní data po lemmatizaci 39
5.4 Srovnání kvality podle počtu rozpoznaných slov 40
5.5 Srovnání kvality podle počtu rozpoznaných písmen 40

6.1 Přehled postupu pro rozdělení témat na trénovací a testovací množinu. 48
6.2 mGAP skóre různých vyhledávacích modelů. 52
6.3 mGAP skóre pro různé druhy zpětné vazby. 55
6.4 mGAP skóre pro různé ASR systémy a různé kanály. 57
6.5 Tvorba stoplistu podle frekvence slov (originální tvary) 62
6.6 Tvorba stoplistu podle frekvence slov (lemmata) 63
6.7 mGAP skóre pro různé druhy stoplistu. 64
6.8 mGAP skóre pro různé délky dokumentů a různé přesahy. 64
6.9 mGAP skóre pro podrobnější popis tématu 69
6.10 Parametry systémů pro závěrečné experimenty 70
6.11 mGAP skóre pro závěrečné experimenty 71

Seznam obrázků

2.1 Schéma interferenční sítě 17

3.1 Anotátorský software: Informace o svědkovi holocaustu a přepis z ASR systému 28
3.2 Anotátorský software: Prohlížení relevantních pasáží 28

6.1 Přehled vlivu lemmatizace u jednotlivých témat 51
6.2 Srovnání různých vyhledávacích modelů pro originální tvary slov . . . 53
6.3 Srovnání různých vyhledávacích modelů pro lemmatizovaná data . . . 54
6.4 Přehled vlivu slepé zpětné vazby na vyhledávání 56
6.5 Přehled vlivu volby ASR systému pro levý kanál 58
6.6 Přehled vlivu volby ASR systému pro pravý kanál 59
6.7 Srovnání výsledků pro různé volby kanálu z přepisů ASR sytému z roku 2006 60
6.8 Srovnání různých druhů stoplistu 65
6.9 Vliv délky generovaných dokumentů na vyhledávání při pevném přesahu 0% 66
6.10 Vliv přesahu generovaných dokumentů na vyhledávání při pevné délce dokumentu 80 sekund 67
6.11 Srovnání experimentů s použitím polí témat TD a TDN 70
6.12 Přehled výsledků závěrečných experimentů na trénovacích datech . . 71
6.13 Přehled výsledků závěrečných experimentů na testovacích datech . . . 72

Název práce: Vyhledávání v nesegmentované mluvené řeči
Autor: Pavel Češka
Katedra (ústav): Ústav formální a aplikované lingvistiky
Vedoucí práce: Mgr. Pavel Pecina
e-mail vedoucího: pecina@ufal.mff.cuni.cz

Abstrakt: V této práci vyhledávám relevantní pasáže v nahrávkách českých svědků holocaustu z projektu MALACH. Zvukové záznamy těchto nahrávek jsou zpracovány systémem pro automatické rozpoznání řeči a přepisy z těchto systémů jsou lemmatizovány a opatřeny morfologickými tagy. V práci představuji skript, který z těchto dat generuje parametrizovatelné kolekce dokumentů. Problém vyhledávání informací v nesegmentované mluvené řeči poté přeformuluji na problém vyhledávání v těchto kolekcích dokumentů. V práci popisuji několik desítek experimentů zkoumajících vliv různých vyhledávacích technik na výsledky vyhledávání na těchto datech. Jedná se zejména o vliv normalizace slovních forem (lemmatizace), volby vyhledávacího modelu (TFIDF modelu, Okapi modelu a Indri modelu), obohacení dotazu o slepou zpětnou vazbu, odstranění nevýznamových slov podle frekvence či podle slovního druhu. Důraz je kladen také na různé hodnoty parametrů délky a přesahu generovaných dokumentů. Zjištěné poznatky jsou v závěru práce ověřeny na testovacích datech. Přepisy výpovědí ani témata pro vyhledávání nejsou z právních důvodů součástí této práce.
Klíčová slova: vyhledávání v mluvené řeči, vyhledávání informací, automatické rozpoznání mluvené řeči, kolekce dat

Title: Unsegmented Speech Retrieval
Author: Pavel Češka
Department: Institute of Formal and Applied Linguistics
Supervisor: Mgr. Pavel Pecina
Supervisor's e-mail address: pecina@ufal.mff.cuni.cz

Abstract: In this work I search through interviews of Czech witnesses of the holocaust from the MALACH project to find relevant parts of these testimonies. Audio records of these interviews are automatically recognized by a system for an automatic speech recognition. Automatically recognized texts are then lemmatized and tagged. In this work I present a script which generates parametrizable collections of documents from these preprocessed texts. The task of unsegmented speech retrieval is then reformulated to a task of information retrieval in this collections of documents. In this work, I describe many experiments which examine the influence of different retrieval techniques on retrieval results on this data collection. Mainly, I study an influence of a morphological normalization (lemmatization), different types of IR systems (TF-IDF model, Okapi model and Indri model), blind relevance feedback, stopword list based on frequencies of terms and part-of-speech categories. I also place emphasis on various values of length and overlap parameters of generated documents. The results of these experiments are verified on test data. Audio records, outputs from automatic speech recognition system and topics for information retrieval are not part of this work due to legal grounds.
Keywords: speech retrieval, information retrieval, automatic speech recognition, collection

Kapitola 1

Úvod

Vývoj lidstva dospěl do stádia informační exploze, kdy veškeré lidské vědění již nelze sepsat do jediné obrovské encyklopedie a množství nových informací se objevuje každý den.

Schopnost práce s informacemi se v dnešním světě stává velmi ceněnou. Každý den se na Internetu objevují nové informace o mnoha oborech lidské činnosti. Nikdo již není schopen toto obrovské množství informací číst a zpracovávat. Pro usnadnění orientace v tomto obrovském množství informací nám slouží internetové vyhledáváče jako je například Google či Seznam. Pro vyhledávání v textových dokumentech existuje též několik použitelných nástrojů často zabudovaných v operačních systémech. Vyhledávání se ale bude v dalších letech rozšiřovat i na jiné druhy nosičů informace než je prostý text či hypertext. Mnoho úsilí se v této době věnuje vyhledávání obrázků a začíná se pracovat i na vyhledávání ve zvukových záznamech.

Jedním z projektů, který se v posledních letech zabýval vyhledáváním ve zvukových záznamech je projekt MALACH (Multilingual Access to Large Spoken Archives)[1]. Na počátku tohoto projektu byl v roce 1993 úspěšný film Stephena Spielberga Schindlerův seznam. Mnoho lidí po jeho shlédnutí poslalo Spielbergovi žádost o vyslechnutí jejich svědectví holocaustu. Proto se Spielberg o rok později rozhodl založit nadaci Survivors of the Shoah Visual History Foundation (VHF)[2], která během 5 let shromáždila největší archiv ústních výpovědí od více než 52 000 svědků holocaustu a jejich osvoboditelů z 57 různých zemí.

Spielbergovou snahou bylo tyto výpovědi zkatalogizovat, poskytnout je vědcům různých oborů, široké veřejnosti i školám pro účely výchovy k boji proti antisemitismu a rasismu. Jeho další vizí bylo také připravit postupy pro shromažďování

[1] http://malach.umiacs.umd.edu/

[2] http://www.usc.edu/schools/college/vhi/

velkého množství ústních výpovědí, které by mohly být použity i pro další dějinné události.

Během let 1999 – 2000 nadace VHF ručně připravila indexaci, shrnutí a revizi u 10% z celkového počtu výpovědí za cenu 8 miliónů dolarů. Příprava výpovědi jednoho svědka zabrala v průměru 35 hodin. Kvůli této neefektivnosti a finanční náročnosti byl pro americkou státní agenturu National Science Foundation připraven projekt pro výrazné vylepšení přístupu k vícejazyčným mluveným archivům. U zrodu tohoto projektu stáli University of Maryland[3], Johns Hopkins University[4] a IBM[5]. Tento projekt (MALACH) byl přijat a bylo na něj poskytnuto 7,5 milionu dolarů v průběhu 5 let.

Cílem projektu MALACH bylo využít metod automatického rozpoznání řeči (Automatic Speech Recognition) a vyhledávání informací (Information Retrieval) k vylepšení přístupu k velkým vícejazyčným mluveným archivům. Tato práce bude na tento projekt navazovat a zaměří se na vyhledávání informací v nesegmentované mluvené řeči.

1.1 Cíle diplomové práce

K dispozici máme zvukové nahrávky rozhovorů se svědky holocaustu a také jejich textové přepisy automaticky získané pomocí systému pro rozpoznání řeči. Problém vyhledávání v mluvené řeči tak můžeme převést na vyhledávání v těchto textových přepisech. Cílem této diplomové práce je připravit sadu nástrojů pro vytváření kolekcí dokumentů[6] z automatických přepisů systémů pro rozpoznání řeči. Práce bude také obsahovat nástroje pro tvorbu vyhledávacích dotazů, vyhodnocení výsledků vyhledávacích systémů a porovnání jejich úspěšnosti.

Dalším cílem této práce je s pomocí těchto nástrojů vytvořit několik kolekcí dokumentů a na nich provést experimenty s nejpoužívanějšími modely pro vyhledávání informací. Práce bude zaměřena na vylepšení výsledků vyhledávání použitím technik jako je standardizace slovních tvarů, odstranění nevýznamových slov, apod. Důraz bude kladen také na různé možnosti rozdělení nesegmentovaných přepisů na doku-

[3] http://www.umd.edu/

[4] http://www.jhu.edu/

[5] http://www.ibm.com/

[6] V této práci budeme termínem **kolekce dokumentů** označovat množinu dokumentů, které automaticky vygenerujeme z přepisů systémů pro rozpoznání řeči. Termínem **kolekce dat** budeme označovat přepisy systémů pro rozpoznání řeči, témata pro vyhledávání a také označení relevance pasáží ke každému tématu.

menty.

1.2 Shrnutí

V kapitole 2 se seznámíme se základními přístupy používanými při vyhledávání informací v textu. Celý proces vyhledávání rozčleníme na několik fází: zpracování dokumentů a témat, popis vyhledávacích modelů, operace s vyhledávacími dotazy a vyhodnocení výsledků vyhledávání.

Ve třetí kapitole popíši, jak byly získány zvukové záznamy výpovědí českých svědků z projektu MALACH a jakým způsobem byly vytvořeny jejich automatické přepisy.

V kapitole 4 převedu problém vyhledávání v nesegmentované mluvené řeči na problém vyhledávání v textových dokumentech a zmíním se také o specifikách hodnocení relevance na těchto datech.

Podrobný popis nástrojů, které jsem použil pro vytvoření dat, ze kterých je možno vygenerovat parametrizovatelnou kolekci dokumentů popíši v 5. kapitole.

Předposlední kapitola podrobně popisuje exprimenty, které jsem provedl na několika vygenerovaných kolekcích, a diskutuje dosažené výsledky.

Kapitola 7 obsahuje shrnutí výsledků mé práce a nástin možností pro další práci na tomto tématu.

Kapitola 2

Vyhledávání informací v textu

Obor vyhledávání informací se zabývá reprezentací informací, jejich uložením, organizací a také přístupem k těmto informacím [1]. V dnešní době, kdy máme přístup k obrovským informačním zdrojům (zejména Internetu) je důležité, usnadnit uživateli přístup k informacím, které ho zajímají, nebo které hledá. K tomuto účelu nám slouží vyhledávací systémy[1].

2.1 Motivace

Představme si uživatele, který chce na Internetu nalézt všechny webové stránky, které by odpovídaly tomuto zadání:

- Hledaná stránka bude pojednávat o deskových hrách vydaných v roce 2007.
- Relevantní jsou pouze originální hry vydané českými autory u českého nakladatelství.
- Relevantní jsou pouze české stránky, které budou obsahovat stručné představení hry, informaci o délce hry, komentáře uživatelů a informaci o ceně hry.

Je jasné, že takto formulovaný požadavek nelze přímo použít pro vyhledávání (například na internetovém vyhledávači) a budeme jej muset upravit do vyhledávacího dotazu, než jej tímto systémem zpracujeme. Pro reprezentaci vyhledávacího dotazu se nejčastěji používají klíčová slova, která co nejlépe vystihují požadavek uživatele. Úkolem vyhledávacího systému je pro daný dotaz nalézt informace, které

[1] Někteří čeští autoři používají termínu **Dokumentografické informační systémy** [2].

jsou užitečné pro uživatele. Je důležité si uvědomit, že hledáme užitečné (relevantní) informace a ne pouze odpovídající text[2].

2.2 Formalizace problému

Proces vyhledávání informací se nyní pokusíme formalizovat. Nejdříve zavedeme několik pojmů:

Požadavek uživatele (Information Need) musí být pro účely vyhledávání informací jasně definovaný, aby bylo možné ohodnotit užitečnost nalezených informací. Požadavek musí být také neměnný, uživatel musí vědět, co hledá již před začátkem vyhledávání a svůj požadavek neměnit.

Informace vyhledáváme v množině textů, u kterých předpokládáme, že jsou obsahově koherentní. Těmto textům říkáme **dokumenty**. Za nalezení informace uspokojující požadavek uživatele považujeme nalezení dokumentu, který tuto informaci obsahuje. Výstupem vyhledávání nejsou přímo informace, ale dokumenty, u kterých se domníváme, že požadovanou informaci obsahují.

Všem dokumentům, mezi kterými budeme vyhledávat, říkame **kolekce dokumentů**.

Pro informace obsažené v dokumentu musíme zvolit nějakou logickou reprezentaci. Za nejjednodušší reprezentaci můžeme považovat přímo záznam plného textu (Fulltext Representation). Pro zrychlení vyhledávání a úsporu místa se také často informace obsažené v dokumentu reprezentují pomocí **termů**. Termy mohou být manuálně přiřazená klíčová slova (čehož se používá zejména v informačních vědách a knihovnictví), nebo automaticky získané části dokumentu (nejčastěji slova), které tento dokument popisují. Dokument v tomto případě představuje množinu či posloupnost termů a jejich vah. Váha termu obvykle odpovídá frekvenci (případně důležitosti) termu v dokumentu.

Vyhledávacím dotazem (Query) rozumíme reprezentaci požadavku uživatele vhodnou pro daný vyhledávací systém. Dotazy i dokumenty většinou reprezentujeme podobným způsobem.

[2]Je důležité si uvědomit rozdíl mezi **vyhledáváním informací** (Information Retrieval) a **vyhledáváním v textu** (Data Retrieval či Text Retrieval) pomocí striktně zadaných podmínek, např. regulárních výrazů.

Vyhledávací model popisuje, jakým způsobem budeme určovat, že dokument odpovídá dotazu uživatele. Někdy do tohoto pojmu zahrnujeme i popis zvolené reprezentace dokumentů a dotazů.

Většina vyhledávacích modelů pro každý dotaz uživatele vrací **uspořádaný seznam výstupních dokumentů** (Ranked List). Dokumenty jsou uspořádány podle předpokládané užitečnosti (relevance) pro uživatele: od nejvíce užitečného po nejméně užitečný.

Při procesu vyhledávání informací postupujeme následovně. Na základě obecných požadavků uživatelů, zvolíme vyhledávací model, který bude pro vyhledávání jejich požadavků výhodný. Jiný vyhledávací model použijeme pro hledání knih v knihovně a jiný pro hledání informací na Internetu. Předzpracujeme všechny dokumenty a uložíme je v reprezentaci odpovídající zvolenému modelu. Nyní může uživatel vznést svůj požadavek na hledanou informaci. Tento požadavek automaticky nebo manuálně přeformulujeme na vyhledávací dotaz a položíme jej vyhledávacímu systému. Ten nám vrátí uspořádaný seznam výstupních dokumentů. V tomto okamžiku můžeme ohodnotit užitečnost dokumentů v tomto seznamu a tím změřit úspěšnost vyhledávacího systému. V následujících sekcích se seznámíme s postupy používanými v jednotlivých částech vyhledávacího procesu u zpracování textových dokumentů.

2.3 Vyhledávací modely

Vyhledávací modely můžeme podle použitého matematického aparátu rozdělit do tří základních skupin:

- Boolské modely využívající množinových operacích
- Vektorové modely reprezentující dokumenty a dotazy jako vektory
- Pravděpodobnostní modely počítající pravděpodobnost, že daný dokument odpovídá dotazu

2.3.1 Základní boolský model

Základní boolský model se zrodil v 50. letech 20. století a sloužil zejména k automatizaci postupů v knihovnictví. Každý dokument z kolekce je popsán pomocí termů (klíčových slov) obsažených v dokumentu. Dotazování se provádí pomocí logických

operátorů `AND`, `OR` a `NOT` a vrácena je množina dokumentů, která odpovídá tomuto logickému výrazu. Mezi nevýhody tohoto systému patří:

- nelze ohodnotit relevanci vystupujících dokumentů,
- nelze ovlivnit velikost výstupu (výstup vždy obsahuje všechny logicky odpovídající dokumenty),
- všechny termy v dotazu i v dokumentech jsou chápány jako stejně důležité,
- je velice težké správně formulovat dotaz.

Tyto nevýhody byly později odstraněny rozšířením boolským modelů [3].

2.3.2 Vektorový model TF–IDF

Tento vektorový model bývá často používán jako základní model pro porovnávání výsledků s novými vyhledávacími modely. Princip pro výpočet podobnosti dokumentu a dotazu, je pro tento model následující. Předpokládejme, že každý dokument v kolekci a každý dotaz je reprezentován jako vektor frekvencí termů (např. slov) $d = (x_1, x_2, ..., x_n)$ nebo $q = (y_1, y_2, ..., y_n)$, kde n označuje celkový počet termů ve všech dokumentech a dotazech nebo počet termů celého slovníku. Členy x_i a y_i jsou frekvence termu t_i v dokumentu d nebo dotazu q. Dále mějme kolekci dokumentů K. Potom inverzní frekvenci dokumentů (IDF – Inverted Document Frequency) spočítáme jako $log(N/n_t)$, kde N značí celkový počet dokumentů v kolekci K a n_t je počet dokumentů, které obsahují term t. Všechny termy v dokumentu nebo dotazu jsou zváženy podle následující vážící formule:

$$d = (tf_d(x_1)idf(t_1), tf_d(x_2)idf(t_2), ..., tf_d(x_n)idf(t_n))$$
$$q = (tf_q(y_1)idf(t_1), tf_q(y_2)idf(t_2), ..., tf_q(y_n)idf(t_n))$$

Skóre podobnosti dotazu a dokumentu je vypočítáno podle následujícího vztahu:

$$s(d, q) = \sum_{i=1}^{n} tf_d(x_i)tf_q(y_i)idf(t_i)^2$$

Různé TF-IDF modely se liší použitou formulí pro výpočet frekvence termů v dokumentu (TF – Term Frequency). V této práci uvedeme dva možné přístupy, které obsahuje volně dostupný vyhledávací nástroj LEMUR[3] a INDRI[4]. Prvním z

[3] http://www.lemurproject.org
[4] http://www.lemurproject.org/indri/

nich je model nazvaný Raw TF–IDF. Tento model za TF komponentu dosadí pouze frekvenci termu t v daném dokumentu. Druhý model je inspirován modelem Okapi BM25 (více o tomto modelu v další sekci) a pro výpočet TF komponenty používá následující formule (jinou formuli pro dokumenty a jinou pro dotazy):

$$tf_d(x) = \frac{k_1 x}{x + k_1(1 - b + b\frac{l_d}{l_C})}$$

$$tf_q(y) = \frac{k_1 y}{k_1 + y}$$

Proměnná l_d označuje délku dokumentu a proměnná l_C průměrnou délku dokumentu v kolekci. k_1 a b jsou parametry modelu, které musejí být určeny. V manuálu [4] nástroje LEMUR, ze kterého je převzat popis tohoto modelu, je doporučeno používat výchozí hodnoty $k_1 = 1,2$ a $b = 0,75$ pro funkci TF u dokumentů a $k_1 = 1$ u dotazů.

2.3.3 Okapi BM25

Vyhledávácí model Okapi BM25 vznikal na London City University v 80. a 90. letech a byl poprvé představen v roce 1995 na konferenci TREC-3 týmem kolem Karen Sparck Jonesové a Stephena E. Robertsona [5]. Okapi BM25 stejně jako všechny dosud představené modely nebere ohledy na vzájemné vztahy slov v dokumentu a považuje je za nezávislé veličiny. Formule pro výpočet podobnosti dotazu a dokumentu je tato:

$$\sum_{T \in Q} w^{(1)} \frac{(k_1 + 1)tf}{K + tf} \frac{(k_3 + 1)qtf}{k_3 + qtf}$$

kde:

- Q je dotaz obsahující termy T
- $w^{(1)}$ je Robertsonova/Sparck Jones váha termu T v dotazu Q
- $w^{(1)} = log\frac{(r + 0,5)/(R - r + 0,5)}{(n - r + 0,5)/(N - n - R + r + 0,5)}$
- N označuje počet dokumentů v kolekci

- n udává počet dokumentů, které obsahují daný term
- R označuje počet dokumentů, které jsou relevantní pro dané téma
- r udává počet relevantních dokumentů obsahujících daný term
- $K = k_1((1-b) + \frac{b.dl}{avdl})$
- k_1, b a k_3 jsou parametry, které závisí na typu dotazů a také kolekci, ve které se vyhledává; základní nastavení těchto parametrů je: $k_1 = 1,2$, $b = 0,75$, ale občas je lepší použít i menší hodnoty parametru b; pro dlouhé dotazy je parametr k_3 nastaven na hodnotu 7 nebo 1000, což prakticky nahrazuje nekonečno
- tf je frekvence výskytu termu v daném dokumentu
- qtf je frekvence výskytu termu v tématu, ze kterého byl vytvořen dotaz Q
- dl označuje délku dokumentu a $avdl$ průměrnou délku dokumentu

Zdroj: Popis tohoto modelu je převzat z díla [6].

2.3.4 Model INDRI

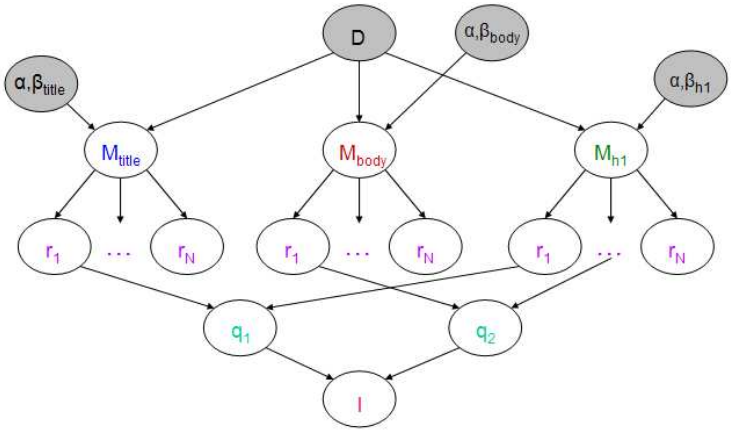
Vyhledávací model INDRI[5] zastupuje třetí kategorii vyhledávacích modelů. Je založen na kombinaci jazykového modelu a vyhledávaní pomocí interferenční sítě.

Jazykový model popisuje každý dokument v kolekci jako posloupnost vektorů. Každé slovo (nebo jiná základní jednotka textu) je totiž reprezentováno jako vektor vlastností tohoto slova obsahující pouze binární hodnoty. V tradičním pojetí reprezentace dokumentů založené pouze na frekvenci slov má tento vektor délku n, kde n udává velikost slovníku. U tohoto pojetí je tedy každé slovo reprezentováno jako vektor obsahující pouze jedinou nenulovou hodnotu, která značí, že se jedná o toto slovo. U komplexnějších modelů každá pozice vektoru reprezentuje určitou vlastnost

[5] http://www.lemurproject.org/indri/

textu (např. slovo je napsáno s velkým písmenem, je na konci věty, apod.) a jeden vektor může obsahovat i více nenulových hodnot. Více detailů o tomto přístupu je možné nalézt v [7].

Interferenční síť (též známá jako Bayesovská síť) se skládá z několika uzlů, jak můžete vidět na obrázku 2.1, který je převzat z díla [8].



Obrázek 2.1: Schéma interferenční sítě

Požadavek uživatele, který chceme modelovat pomocí této sítě, reprezentujeme několika koncepty (na obrázku je reprezentují uzly q_1 a q_2). Dále předpokládáme, že relevantní dokumenty obsahují stejné koncepty jako požadavek uživatele. Každý dokument v kolekci (uzel D) reprezentujeme pomocí jazykových modelů (různé uzly M), které závisí na vlastnostech dokumentu D a na parametrech α a β spočtených ze všech dokumentů v kolekci. Na obrázku vidíme několik jazykových modelů vytvořených z částí HTML stránek (názvů, těl stránek a nadpisů). Uzly r_i reprezentují vlastnosti dokumentu formulované jako evidence konceptů požadavků uživatele. Uzel I je poté vyjádřením toho, že dokument D odpovídá dotazu. Poprvé tento přístup použil Turtle [9]. Detailní popis implementace tohoto modelu lze nalézt v [7].

2.4 Zpracování dokumentů a témat

Vyhledávací systémy obvykle obsahují alespoň základní předzpracování dokumentů a dotazů, které vylepší vlastnosti vyhledávacího systému. Mezi tato předzpracování patří jednak normalizace slovních tvarů, která nám zajistí určení správného významu slova v dokumentu či dotazu, a také odstranění nevýznamových slov (Stopwords), kterým vylepšíme reprezentaci dokumentů pomocí významových slov. Z ostatních

technik spadajících do této oblasti zmin9m alespoň označování víceslovných spojení (kolokací) jako jediného termu a nahrazování zájmen podstatnými jmény, které ve větě zastupují (rozeznávání koreferencí[6]). V dalším textu se budu věnovat pouze prvním dvěma zmíněným základním technikám.

2.4.1 Normalizace slovních tvarů

Normalizací slovních tvarů rozumíme převedení slov odvozených od stejného základu na tento základ. U anglického jazyka touto normalizací obvykle bývá odstranění standardních předpon a přípon slova, až ze slova zůstane jen jeho kořen (tzv. stemming). Pro tento účel je známo několik algoritmů, uveďu alespoň ten nejznámější – Porterův stemmer [10]. Pro morfologicky bohatší flexivní jazyky, jakým je například čeština, není snadné automaticky odstranit předpony, přípony a koncovku, protože u mnoha slov dochází ke změně hlásek v kořenu slova (např. břeh x březích). Proto se u těchto jazyků často používá lemmatizace, což je nahrazení slova jeho základním slovním tvarem. Lemmatizace na rozdíl od stemmingu přináší také rozlišení významů synonym (např. slovo „ženu“ bude lemmatizováno jako „žena“ či „hnát“ podle svého významu) a mnohdy nám také určí morfologické kategorie slova, které můžeme využít při odstraňování nevýznamových slov, jak bude uvedeno v další sekci.

2.4.2 Nevýznamová slova

Příliš obecná slova nejsou pro identifikaci dokumentů vhodná, protože se vyskytují téměř v každém dokumentu. Pokud bychom vyhledávacímu systému položili dotaz obsahující několik nevýznamových slov a jedno slovo významové, vyhledané dokumenty by s velkou pravděpodobností neodpovídaly našim požadavkům, protože téměř každý dokument z kolekce by obsahoval většinu hledaných slov. Přední vyhledávací systémy proto nevýznamová slova z dokumentů a dotazů odstraňují.

K vytvoření seznamu nevýznamových slov (Stoplist) můžeme přistoupit dvěma rozdílnými cestami. První z nich je založena na četnosti daného slova v prohledávané kolekci dokumentů nebo přímo na četnosti slova v daném přirozeném jazyce (četnost můžeme odhadnout z korpusu daného jazyka). Vypuštěním nejčetnějších slov také zmenšíme celkovou velikost dat, ve kterých budeme vyhledávat.

Druhou cestou, jak vytvořit stoplist, je zahrnout do něj všechna slova z těch morfologických kategorií, které jsou pro jazyk nevýznamové (nebo málo významné).

[6] V zahraniční terminologii se občas rozlišuje význam pojmů anafora a koreference. Zde **koreferencí** míníme odkazování dvou výrazů v textu k téže osobě, předmětu či skutečnosti.

K tomuto účelu můžeme pro český jazyk využít morfologické kategorie slovního druhu a z dokumentů i dotazů odstranit tyto slovní druhy: zájména, předložky, spojky, částice a citoslovce.

2.5 Rozšiřování vyhledávacího dotazu

Vyhledávací dotazy prochází stejným předzpracováním jako dokumenty (normalizace slovních tvarů, odstranění nevýznamových slov). Po položení dotazu vyhledávacímu systému, získáme uspořádaný seznam výstupních dokumentů, které tomuto dotazu odpovídají. Mnohé vyhledávací systémy využívají tohoto výstupního seznamu, aby vylepšily výsledky svého vyhledávání. V následujících sekcích si představíme dva rozdílné přístupy, které mohou vést k vylepšení.

2.5.1 Zpětná vazba od uživatele

U prvního z těchto přístupů potřebujeme od uživatele informaci o relevanci jednotlivých dokumentů, které vyhledávací systém vrátí po prvním vyhledávání. Předpokládejme, že pro vyhledávání využíváme vektorový vyhledávací systém. Uživatel ohodnotí relevanci (nejčastěji pomocí binární hodnotící funkce) několika prvních dokumentů z výstupního seznamu a náš systém podle jeho ohodnocení upraví vyhledávací dotaz tak, aby se přiblížil vektorům reprezentujícím kladně ohodnocené dokumenty a vzdálil se dokumentům ohodnoceným záporně. Tento cyklus (položení dotazu, vrácení výstupních dokumentů, ohodnocení dokumentů) lze provést i několikrát, i když nejlepší zlepšení lze pozorovat zejména po proběhnutí prvního cyklu.

2.5.2 Slepá zpětná vazba

Druhý přístup nevyžaduje ohodnocení relevance výstupních dokumentů uživatelem a slepě předpokládá, že n nejlépe ohodnocených dokumentů je relevantních, a proto se vyplatí obohatit vyhledávací dotaz o m nejčetnějších specifických termů z těchto dokumentů. Tento postup, známý jako slepá zpětná vazba (Blind Relevance Feedback) je také možno zkombinovat se snahou o vzdálení se nejhůře hodnoceným dokumentům a penalizovat tak termy, které nejhůře hodnocené dokumenty obsahují. Tato možnost se však téměř nepoužívá.

2.6 Evaluace

Pro vyhodnocení úspěšnosti (evaluaci) vyhledávacího systému, potřebujeme vědět, které dokumenty z kolekce odpovídají vyhledávané informaci. Toto ohodnocení relevance dokumentů nazýváme anotace a je nutné jej provést manuálně. Pro vyhodnocení také potřebujeme míru, kterou budeme výsledky poměřovat. Pro hodnocení výsledků různých vyhledávacích systémů je nutné jejich úspěšnost poměřovat se stejnou kolekcí dokumentů, stejnými tématy pro vyhledávání a stejným ohodnocením relevance. Veličiny, o kterých se zmíníme v následujících sekcích nelze nikdy poměřovat pro výsledky vyhledávacích systémů dosažených na různých kolekcích dat.

2.6.1 Přesnost a úplnost

Pro vyhledávácí systémy, které vrací seznam výsledků hledání (ať již uspořádaný či neuspořádaný), se obvykle používá dvourozměrné hodnotící funkce. První rozměr odpovídá přesnosti (Precision), která se zvyšuje pokud výstupní dokumenty odpovídají dotazu, který jsme systému položili. Druhý rozměr popisuje úplnost (Recall), která je nejvyšší pokud seznam výsledků hledání obsahuje všechny relevantní dokumenty z kolekce.[7] U vyhledávacích systémů, které vrací uspořádaný seznam výstupních dokumentů, musíme při výpočtu přesnosti a úplnosti zvolit prahovou hodnotu (Treshold), po kterou budeme výstupní dokumenty považovat za součást odpovědi vyhledávacího systému. Pro výpočet přesnosti a úplnosti definujeme následující proměnné:

Out – počet dokumentů v odpovědi vyhledávacího systému
Rel – počet relevantních dokumentů v celé kolekci
$OutRel$ – počet relevantních dokumentů v odpovědi vyhledávacího systému

Přesnost udává, jaká část dokumentů zařazených v seznamu výsledků vyhledávání je skutečně relevantní.

$$P = \frac{OutRel}{Out}$$

Úplnost udává, jaká část skutečně relevantních dokumentů je zařazen v seznamu výsledků vyhledávání.

$$R = \frac{OutRel}{Rel}$$

[7] České názvosloví jsem převzal ze slajdů Michala Kopeckého k přednášce Dokumentografické informační systémy.

2.6.2 F-measure

U vyhledávacích modelů, které vrací uspořádaný seznam výstupních dokumentů, můžeme různými volbami prahové hodnoty získávat různé hodnoty přesnosti a úplnosti. Pokud bychom do odpovědi zařadili všechny dokumenty z kolekce, pak budeme mít stoprocentní úplnost, ale přesnost bude velmi nízká. Pokud do odpovědi zahrneme pouze nejlépe hodnocený dokument, u kterého jsme si téměř jistí, že odpovídá požadavku uživatele, budeme mít stoprocentní přesnost, ale nízkou úplnost. Přesnost a úplnost se od určité prahové hodnoty vzájemně ovlivňují. Pokud zvyšujeme přesnost systému, snižujeme jeho úplnost a naopak. Z tohoto důvodu zavádíme novou míru, která výsledky přesnosti a úplnosti kombinuje a převádí tak do jednorozměrného prostoru. Tato míra se nazýva F-measure (a její hodnota F-skóre) a jde vlastně o vážený harmonický průměr přesnosti a úplnosti. Základní formule pro výpočet F-skóre je tato:

$$F = \frac{2PR}{P + R}$$

První formule pro výpočet F-skóre přiřazuje stejnou váhu oběma veličinám. Pokud ale víme, že je pro naše účely je některá veličina důležitější, můžeme také použít tuto obecnější formuli:

$$F_\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R}$$

Parametr β (nezáporné reálné číslo) určuje, že hodnota úplnosti je pro nás β-krát důležitější než hodnota přesnosti.

2.6.3 Průměrná přesnost

Další mírou úspěšnosti vyhledávacího systému je průměrná přesnost (Average Precision). Abychom mohli tuto míru využít, musí vyhledávací systém vracet uspořádaný seznam výstupních dokumentů. Tato míra upřednostňuje systémy, které vrací relevantní dokumenty na předních pozicích, před těmi systémy, které je vrací až na konci výstupního seznamu (což je praktické nejen při vyhledávání na Internetu). Průměrná přesnost je vypočítávána jako průměr přesnosti systému pro odpověď obsahující postupně $1, 2, 3, ..., Out$ dokumentů, kde Out označuje počet všech vrácených dokumentů.

$$AP = \frac{\sum_{r=1}^{Out}(P(r) * rel(r))}{Rel}$$

Proměnná r odpovídá pořadí dokumentu v seznamu výstupních dokumentů, proměnná Out označuje počet dokumentů v tomto seznamu (stejně jako dříve u výpočtu přesnosti a úplnosti), $rel(r)$ je binární funkce určující relevanci dokumentu na pozici r a $P(r)$ je přesnost vyhledávacího systému na prvních r výstupních dokumentech.

Kapitola 3

Popis dat

Tato kapitola pojednává o datech, které jsem měl k dispozici před zahájením samotného procesu vyhledávání informací.

3.1 Video záznamy

Nahrávky rozhovorů získané z projektu MALACH byly pro další zpracování uloženy jako video záznam ve formátu MPEG-1. Každý rozhovor byl nahráván na dva mikrofony, které jsou ve video záznamu uloženy na dvou různých kanálech. Jeden mikrofon byl namířen na svědka holocaustu, druhý na moderátora rozhovoru. Kvůli nedodržování metodiky k nahrávání rozhovorů, nelze s jistotou určit, který z těchto mikrofonů byl namířen na svědka holocaustu a který na moderátora. Pro každý rozhovor tak existují dva zvukové záznamy, které budeme dále označovat jako levý a pravý kanál (left channel, right channel). Páska, na kterou se rozhovory nahrávaly, měla délku přibližně 30 minut. Většina nahraných rozhovorů se proto skládá z několika pásek, které na sebe sice chronologicky navazují, ale plynulost rozhovoru je narušena, protože pásky byly vyměňovány ručně a protože na začátku každé pásky moderátor opakuje identifikační informace o svědkovi holocaustu, se kterým je rozhovor veden.

3.2 Automatické rozpoznání řeči

Systémy pro automatické rozpoznání řeči nám umožňují převést audio záznam na textovou podobu. Video záznamy z rozhovorů proto byly převedeny na audio záznam ve kvalitě 128 kb/s v 16-bitovém stereu a při samplovací frekvenci 44 kHz. Tato data byla dále zpracována systémem pro automatické rozpoznání řeči vyvinutým

Západočeskou univerzitou v Plzni[1]. Při konstrukci rozpoznávače řeči musel tým ZČU rozhodnout, jakým způsobem se budou přepisovat hovorová slova. Bylo rozhodnuto, že přepisy budou obsahovat ortografickou podobu hovorových slov. Důvodem pro toto rozhodnutí bylo zejména zjednodušení práce pro anotátory, kteří i bez hlubších lingvistických znalostí mohli anotovat data potřebná k natrénování rozpoznávače řeči, jak je uvedeno v této práci [11]. Druhým důvodem bylo též to, že algoritmus pro automatický fonetický přepis slov potřebuje znát jejich ortografickou podobu, jinak nefunguje správně. Kvůli tomtuto rozhodnutí musel být i jazykový model obohacen o všechny možnosti výslovnosti daného slova.

Příklad různých ortografických podob jednoho slova:

```
oběd  [o b j e d]      Osvětim [o s v j e t i m]
      [o b j e t]              [v o s v j e t i m]
      [v o b j e d]            [o s v j e n č i m]
      [v o b j e t]            [o z v j e t i m]
```

Dalšími komplikacemi při tvorbě rozpoznávače byl vysoký věk vyslýchaných svědků holocaustu (průměrně 75 let) a obrovské emoční zapojení svědků v prožitých událostech holocaustu, které znesnadňovalo porozumění slov a obohatilo projev o velké množství neřečových událostí (pláč, smích, kašel...), jak uvádí ve své práci Psutka a kol. [11]. Vytvořený rozpoznávač řeči umožňuje vypisovat výstupní text v českém jazyce s hovorovými výrazy nebo ve standardizované formě češtiny[2], což je pro účely vyhledávání informací užitečnější. Více informací o zvoleném řešení stavby automatického rozpoznávače řeči lze nalézt v práci [11].

Pro účely vyhledávání informací v českých výpovědích byly vytvořeny dva přepisy, které budou dále sloužit jako jediný zdroj pro automatické vyhledávání informací. První z nich, rozpoznávací systém ASR 2004 ve svém výstupu obsahuje i hovorová slova. Druhý z nich, systém ASR 2006 obsahuje již pouze slova spisovná. Na následujících řádcích najdete ukázkový výstup textu z rozpoznávače řeči:

```
cell 1 0.00 12.80 <s>
cell 1 12.80 0.14 A
```

[1] http://www.zcu.cz/

[2] Převod do této standardizované formy zahrnoval jednak nahrazení hovorových slov spisovnými výrazy a také standardizaci názvů, což lze chápat jako zvolení klíčových výrazů pro slova popisující stejný objekt (např. pro Osvětim a Auswitz).

```
cell 1 13.13 0.35 A
cell 1 13.85 0.53 BYLA
cell 1 17.13 0.69 DÍRA
cell 1 18.56 0.14 A
cell 1 21.30 0.46 ŽE
cell 1 22.05 0.28 SE
cell 1 23.48 0.83 A
cell 1 25.00 0.75 POVÍDAL
cell 1 25.89 0.29 ŽE
cell 1 26.18 0.37 JO
cell 1 26.55 0.01 </s>
cell 1 26.58 1.50 <s>
cell 1 28.08 0.80 POVÍDAL
cell 1 28.94 1.16 NOVÝHO
cell 1 30.41 0.76 NE
cell 1 31.17 0.01 </s>
cell 1 31.20 0.67 <s>
cell 1 31.87 0.37 A
cell 1 32.40 0.29 A
cell 1 42.17 0.31 JÁ
cell 1 68.25 0.48 SEM
cell 1 68.73 0.32 SE
cell 1 69.19 0.64 ALEXEJEM
cell 1 69.83 0.50 HRONEM
cell 1 70.33 0.01 </s>
```

První a druhé pole v každém řádku obsahuje nevýznamné hodnoty. Třetí pole v sekundách označuje počátek rozpoznávání daného slova (počítáno od začátku dané pásky, ne výpovědi) a číslo na čtvrté pozici říká, jak dlouhý časový úsek slovo na pásce zabýrá. Poslední pole obsahuje rozpoznané slovo (napsané velkými písmeny) nebo řetězec `<s>` pro počátek věty či řetězec `</s>` pro její konec.

3.3 Témata pro vyhledávání

Témata pro vyhledávání v kolekci byla vybírána ve spolupráci s odborníky z oboru historie, sociologie, medicíny a dalších. Jedná se tedy o popis reálných požadavků

uživatelů, kteří by mohly vyhledávat informace v této kolekci nahrávek. Pro českou kolekci bylo celkem vytvořeno 115 vyhledávacích témat[3]. U 105 z nich se jednalo o překlad témat z anglické kolekce (jak jsme již zmínili kolekce MALACH obsahuje rozhovory z celkem 32 jazyků), u zbývajících 10 témat byla z originálních témat odebrána geografická omezení, aby se zvýšil předpokládaný počet relevantních pasáží v české kolekci, která je několikrát menší než anglická. Každé téma pro dotazování obsahuje čtyři XML elementy. První element obsahuje identifikační číslo tématu (`<num>`), zbylé tři popisují téma:

- element (`<title>`) obsahuje název tématu,
- element (`<desc>`) obsahuje popis tématu,
- element (`<narr>`) obsahuje obšírnějším popis tématu.

Jednotlivé elementy tématu mohou být využity pro tvorbu dotazů pro vyhledávací systémy. Nejčastěji se pro vyhledávání vytváří dotazy z polí `<title>` a `<desc>`, nebo se používá obsahu všech tří elementů.

Ukázkové téma v anglickém jazyce:

```
<top>
<num>1185</num>
<title>Doctors and Nurses in the Holocaust</title>
<desc>The Good Doctors and Nurses in the Holocaust. Ethical dilemmas
that confronted health-care professionals seeking to do good in the
midst of evil.</desc>
<narr>These could be Jewish or non-Jewish doctors providing care in
different environments: prisons, concentration camps, resistance,
hiding places. Care could be formal or informal.</narr>
</top>
```

Stejné téma v českém jazyce:

```
<top>
<num>1185</num>
<title>Lékaři a zdravotní sestry v holokaustu</title>
```

[3] **Téma** je strukturovaný popis požadavku uživatele pro vyhledání informace.

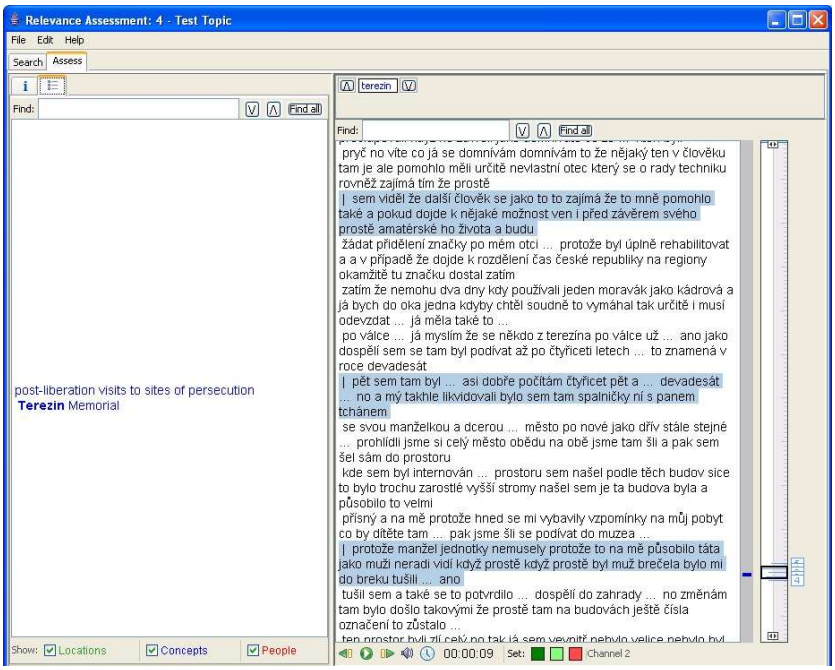
```
<desc>Ti "dobří" lékaři a zdravotní sestry v holokaustu. Etická
dilemata, před kterými stáli zdravotníci snažící se konat dobro
obklopeni zlem.</desc>
<narr>Může se jednat o židovské i nežidovské lékaře, kteří
poskytovali péči v různých prostředích např. věznicích,
koncentračních táborech, v odboji a úkrytech. Péče mohla být jak
formální, tak neformální.</narr>
</top>
```

3.4 Hledání relevantních pasáží

České rozhovory z projektu MALACH byly anotovány na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze. U každého tématu použili anotátoři dva přístupy anotace. U prvního přístupu střídavě vyhledávali informace o tématu v externích zdrojích (zejména na Internetu) a prohledávali kolekci výpovědí svědků holocaustu. Pro tyto účely Ryen White z University of Maryland vytvořil anotátorský software, který umožňuje interaktivní zapojení během vyhledávání. Z každé pásky rozhovorů byl vybrán nejkvalitnější (co do počtu slov) přepis z ASR systému a byly k němu automaticky přiřazeny hesla z anglického tezauru. Vyhledávání v tomto systému probíhalo přes dotazy obsahující slova z přepisu záznamů a/nebo z hesel tezauru. Když anotátor nalezl slibný rozhovor, program v automatickém přepisu rozhovoru graficky znázornil relevantní pasáže, které odpovídaly dotazu, který systému položil. Anotátor si také mohl od libovolného okamžiku poslechnout originální audio záznam kanálu, a po ověření, že daná pasáž je relevantní, začátek a konec pasáže označil v přepisu textu. Tento začátek a konec byl automaticky převeden na nejbližší bod s 15 sekundovou přesností (granularitou).

Když bylo vyhledávání zaměřené na hledání relevantních pasáží ukončeno, anotátoři dostali ke kontrole místa, které našli automatické vyhledávací systémy účastníků soutěže CLEF 2006[12] a 2007[13] v běhu Cross-Language Speech Retrieval. Anotátoři prošli 50 nejlépe hodnocených pasáží z každého soutěžního běhu a relevantní pasáže objevené touto metodou byly přidány k vyhledaným pasážím z prvního kroku. Relevantní pasáže byly takto vyhledány pro 98 témat (pro 17 témat se nenalezly žádné relevantní pasáže). Anotátoři dohromady celkem označili 5436 relevantních pasáží (některé z těchto pasáží se vzájemně překrývají, protože většinu témat anotovalo nezávisle více anotátorů).

Obrázek 3.1: Anotátorský software: Informace o svědkovi holocaustu a přepis z ASR systému



Obrázek 3.2: Anotátorský software: Prohlížení relevantních pasáží

Kapitola 4

Vyhledávání informací v mluvené řeči

Vyhledávání informací v nesegmentovaných nahrávkách přináší několik problémů. Nahrávky jsou velmi dlouhé a tematicky nekoherentní. Pokud bychom za dokument považovali celou nahrávku, bude dohledání informace v audio záznamu vyžadovat neúměrně mnoho času (u textových dokumentů je dohledání snažší, protože můžeme zapojit fulltextové vyhledávání nebo také dokument relativně rychle přečíst). Jednou z možností, jak vyhledávat v audio nahrávkách je převést tyto nahrávky na texty a poté vyhledávat informace v těchto textech. Právě touto možností se budeme zabývat v této práci.

Jedním z cílů této práce je provést vyhledávací experimenty na výpovědích českých svědků holocaustu. Každá kolekce dat pro vyhledávání informací musí obsahovat tyto součásti:

- Dokumenty, ve kterých budeme vyhledávat
- Témata, ze kterých budeme tvořit dotazy pro vyhledávání
- Seznam relevantních dokumentů pro každé téma
- Hodnotící funkci pro vyhodnocení úspěšnosti

Proto i při převodu našeho problému na vyhledávání v textových dokumentech musíme uspokojivě vyřešit podobu každé z těchto částí.

4.1 Reformulace problému

Uživatel, který chce vyhledat určitou informaci v audio záznamech, by chtěl od vyhledávacího systému nejen nalézt audio záznam, ve kterém se nachází hledané informace, ale na rozdíl od textových dokumentů by chtěl znát i co nejpřesnější místo (čas), kde se informace nachází, aby nemusel poslouchat celý záznam. V textovém dokumentu můžeme přesné místo rychle dohledat pomocí nástrojů na vyhledávání textu, v audio záznamu tuto možnost ale nemáme a poslech celého záznamu je značně zdlouhavý. Při výzkumu chování uživatelů se zjistilo, že pro uživatele je mnohem důležitější znát počátek relevantního dokumentu než jeho konec, jak uvádí [14]. Za vyhledávání informací v nesegmentované řeči proto budeme považovat identifikaci počátků relevantních pasáží audio záznamů, což odpovídá požadavkům uživatelů.

4.2 Tvorba textových dokumentů

Myšlenka vytvoření textových dokumentů z přepisů řeči je zcela intuitivní. Jelikož přepisy rozhovorů nejsou rozděleny do tematických částí a obsahují pouze časovou osu rozpoznaných slov, rozdělíme je do dokumentů právě podle času. Při disjunktním rozdělení celé časové osy na intervaly o stejné délce vytvoříme kolekci dokumentů, která svými vlastnostmi zcela odpovídá textovým dokumentům – každé rozpoznané slovo se v kolekci objevuje právě v jednom dokumentu. Jelikož se snažíme nalézt počátky relevantních výpovědí dokument budeme identifikovat podle času prvního slova, které obsahuje. Jak začátek relevantní pasáže označené anotátorem, tak identifikace dokumentu tak odkazují na obsah, který po nich následuje.

Disjunktní rozdělení časové osy se ale ukázalo jako nevyhovující, a proto bylo navrženo, aby se z přepisů utvořily dokumenty, které se budou navzájem přesahovat. Vytvořená kolekce tak bude obsahovat některé části přepisů vícekrát, ale bude více odpovídat tomu, že nevíme, jak přesně rozdělit souvislou promluvu na části. Tím, že vytvoříme dokumenty, které se překrývají spíše identifikujeme relevantní dokument. Tímto postupem byla vytvořena také kolekce dokumentů pro běh CL-SR soutěže CLEF 2006 a 2007.

4.3 Překryv relevantních pasáží označených různými anotátory

Jelikož u většiny témat hledalo relevantní pasáže více anotátorů, musíme stanovit, jak naložíme s jejich rozdílným pohledem na relevantní části dokumentů. Pro přehlednost připomeňme, že každý anotátor u každé relevantní pasáže označil její počátek a konec. Na rozdíl od textových dat, kde k neshodě anotátorů může dojít jen tak, že stejný dokument je jedním anotátorem označen jako relevantní a druhým jako nerelevantní, může u nesegmentované řeči dojít k překryvu intervalů relevantních pasáží nalezených různými anotátory.

Tyto překryvy můžeme rozdělit do dvou kategorií. První z nich jsou odchylky vzniklé pouze chybným nastavením anotátorského software, který se během anotačních prací upravoval. Touto technickou chybou došlo ke změně označování časů relevatních pasáží a tím vznikla malá časová odchylka u označených počátků i konců pasáží. Pokud je rozdíl mezi počátky i konci dvou relevantních pasáží menší než 7,5 sekundy (toto číslo je odvozeno od maximální odchylky zaznamenání času v anotátorském software, která činí 15 sekund), považujeme označenou pasáž za identickou.

Druhým typem překryvů jsou všechny ostatní překryvy – tedy překryvy kde došlo i k rozdílnému vnímání relevance informační hodnoty výpovědi. Pro vyřešení problému s těmito překryvy můžeme přemýšlet nad několika řešeními. Implementačně nejsnažším může být ponechání všech překryvů v kolekci, čímž bychom nepřišli o žádnou ohodnocenou pasáž, ale výsledek by nebyl vhodný pro evaluaci výsledků.

Podobně přímočarým řešením by bylo slití všech překrývajících anotovaných intervalů do jediného. Ve výsledku bychom tak měli méně relevantních pasáží než před slitím a tyto pasáže by byly delší a pravděpodobně i méně informačně hodnotné – do výsledné relevantní pasáže by byly spojeny pasáže relevantní pro všechny anotátory i pasáž relevantní jen pro jediného z nich.

Po vyřešení tohoto problému jsem nakonec zvolil střední cestu a intervaly, které se vzájemně překrývali alespoň o 50% jsem sloučil do jediného intervalu následovně:

- za počátek nově vytvořeného intervalu jsem zvolil medián hodnot počátků překrývajících se relevantních pasáží
- za konec nově vytvořeného intervalu jsem zvolil medián hodnot konců překrývajících se relevantních pasáží

Výsledné řešení v sobě skrývá nevýhodu, že vytvořená relevantní pasáž nemusela být označena žádným anotátorem. Výhodou tohoto přístupu je shoda alespoň

poloviny anotátorů na relevanci úseku mezi takto označeným počátkem a koncem výsledné relevantní pasáže.

Po tomto sloučení zůstalo celkem 4668 nepřekrývajících se relevantních pasáží. U 15 témat byl počet nalezených relevantních pasáží menší než 5 a nejsou proto do kolekce zahrnuty, protože by při tomto počtu nadměrně zkreslovaly výsledky vyhledávání. Pro zbylých 83 témat anotátoři celkem nalezli 4640 relevantních pasáží, což činí průměrně 56 pasáží na jedno téma (minimálně 5 pasáží, maximálně 349 pasáží u jednoho tématu). Tabulka 4.3 přehledně zobrazuje počet relevantních pasáží pro každé téma.

4.4 Evaluace

Pro vyhledávání v segmentovaných kolekcích se s úspěchem používá hodnotítí míry zvané průměrná přesnost (Average Precision), kterou jsme představili v kapitole 2.6.3. Vyhledávání v automatických přepisech z mluvené řeči je ale specifické v tom, že i přesto, že známe dokument, ve kterém se nachází hledaná informace, není snadné tuto informaci dohledat, což stěžuje zejména nekvalitní výstup přepisů z rozpoznávače řeči, čímž jsme odkázáni na poslech originálního audio záznamu a až poté můžeme zjistit, jestli záznam vypovídá o tom, co jsme hledali. K vyhledávání v nesegmentované mluvené řeči by nám pomohlo znát místa, kde bychom měli začít audio záznamy poslouchat. S ohledem na to, že nás bude zajímat počáteční bod relevantní výpovědi, od kterého máme audio záznam poslouchat, vytvořili Liu a Oard pro tyto účely jednostrannou míru (v úvahu bere pouze začátek relevantní výpovědi), kterou nazvali zobecněná průměrná přesnost (Generalized Average Precision) [14].

Ideální vyhledávací systém objeví počáteční časy velmi blízko počátečních časů relevantních pasáží. U textových kolekcí je míra relevance většinou binární funkce, která každý dokument v kolekci pro každé vyhledávané téma ohodnotí jako relevantní či nerelevantní. U generovaných dokumentů z přepisů textu tento způsob použít nelze. Jak jsme již dříve zmínili, snažíme se o identifikaci místa, kde by měl uživatel začít audio záznam poslouchat. Proto budeme relevanci dokumentu poměřovat podle vzdálenosti, o kterou se liší začátek dokumentu a začátek nejbližšího relevantního úseku. Liu a Oard zvolili, že pokud se tyto body liší o méně než 15 sekund (kupodivu je jedno kterým směrem i když při poslechu záznamu se pohybujeme pouze vpřed), bude zisk tohoto systému maximální a za každých dalších započatých 15 sekund rozdílu se zisk sníží o 10%.

číslo tématu	počet pasáží	číslo tématu	počet pasáží	číslo tématu	počet pasáží
24313	349	3023	66	1181	24
3028	173	1620	61	1630	22
3009	144	2224	59	15602	22
1508	139	4012	58	3022	21
4006	133	4005	55	2367	20
2253	126	3008	53	2198	20
2000	113	1185	53	1192	18
4011	110	2006	51	3010	17
2358	107	4007	50	14313	17
3020	104	1663	50	1166	17
3007	102	3016	49	3021	16
15601	102	2384	48	1979	16
2265	100	3011	44	3019	13
3005	94	3004	44	1330	13
3001	89	1897	43	4010	12
3024	88	3000	41	1345	12
4000	87	2264	41	4004	11
3017	87	14312	39	4002	11
3002	87	1554	38	1337	11
3014	86	1311	38	1225	11
1286	86	1424	37	1605	10
1173	82	1321	34	3032	9
3033	80	1187	33	4009	8
1843	79	4001	32	1288	8
3015	78	3026	27	2404	6
3027	75	3018	27	1829	6
3025	74	2185	25	2055	5
2012	69	1310	25		

Tabulka 4.1: Počet relevantních pasáží pro každé téma

Průměrná zobecněná přesnost je tedy definována takto:

$$GAP = \frac{\sum_{R_k \neq 0} p_k}{N}$$

kde N označuje počet začátků relevantních pasáží, R_k označuje skóre hodnotící funkce (tak jak jsem ji popsal výše) pro čas na k-té pozici a člen p_k představuje přesnost systému pro prvních k vrácených dokumentů. Skóre hodnotící funkce se vypočítavají v pořadí od nejrelevantnějšího času po nejméně relevantní bez nahrazování. Pokud byl počátek relevantního úseku již jednou použit pro ohodnocení, další čas v blízkosti bodu již nevynese žádné bodové ohodnocení.

4.5 Wilcoxonův párový test

Wilcoxonův párový test (Wilcoxon Matched-Pairs Signed-Ranks Test) [15] se používá pro porovnání rozdílů výsledků mezi dvěma souvisejícími měřeními, kde rozdíl těchto měření nabývá hodnoty z určitého intervalu (nelze jej tedy použít pro porovnání kategorických měření). Tento test je neparametrický, což znamená, že neklade žádné požadavky na vlastnosti rozdělení pravděpodobnosti rozdílů mezi měřeními.

Předpokládejme, že máme $2n$ měření, tj. dvě měření ke každému z n objektů (např. hodnoty míry GAP k n tématům od dvou různých vyhledávacích systémů). Označme tyto objekty proměnnou $i = 1, 2, ..., n$. Hodnoty prvního měření tak budeme značit X_i a hodnoty druhého měření Y_i. Nechť $Z_i = Y_i - X_i$ pro každé $i = 1, 2, ..., n$. Předpokládejme, že jednotlivé rozdíly Z_i jsou na sobě nezávislé a že každý tento rozdíl pochází ze spojité populace a je symetrický k mediánu θ.

Nulovou hypotézou, kterou budeme testovat, je tvrzení, že $\theta = 0$ a tedy, že obě měření jsou statisticky nerozlišitelná (což by znamenalo, že nemůžeme říci, že jeden z vyhledávacích systémů dosahuje lepších výsledků).

Pro výpočet Wilcoxonova testu nejdříve spočítáme absolutní hodnoty $|Z_1|$ až $|Z_n|$, setřídíme je od nejmenšího po největší (nulové hodnoty $|Z_i|$ vyřadíme). Takto setříděným hodnotám přiřadíme hodnotu odpovídající jejich pořadí (pokud je více $|Z_i|$ shodných přiřadíme jim průměrnou hodnotu jejich pořadí) a označíme ji R_i. Na závěr spočítáme hodnoty W^+ a W^- podle následujících vztahů:

$$W^+ = \sum_{i=1}^{n} R_i, \forall i \in \{i; Z_i > 0\}$$

$$W^- = \sum_{i=1}^{n} R_i, \forall i \in \{i; Z_i < 0\}$$

Příklad výpočtu:

i	X_i	Y_i	Z_i	$\lvert Z_i \rvert$	R_i	$sgn(Z_i)$ $*R_i$	výpočet W^-	výpočet W^+
1	1	1	0	0				
2	1	9	-8	8	1	-1	-1	
3	179	202	-23	23	2	-2	-2	
4	653	617	36	36	3	3		3
5	154	115	39	39	4	4		4
6	7	47	-40	40	5,5	-5,5	-5,5	
7	165	125	40	40	5,5	5,5		5,5
8	28	79	-51	51	7	-7	-7	
9	5	148	-143	143	8	-8	-8	
10	1214	695	519	519	9	9		9
							-23,5	21,5

Tabulka 4.2: Wilcoxonův párový test

Menší ze součtů W^- a W^+ poté srovnáme s tabulkou všech možných distribucí pořadí. Získáme tím pravděpodobnost p, která vyjadřuje, že rozdíly našich měření pochází ze spojité populace, která je symetrická k mediánu θ. Při zvyšujícím se n, se distribuce všech možných pořadí blíží normální distribuci a pro $n > 20$, se pro její výpočet používá aproximace normálním rozdělením. Rozdíly mezi měřeními uvedenými v příkladové tabulce jsou na hladině pravděpodobnosti $p < 0,05$ statisticky nevýznamné.

Kapitola 5

Nástroje pro práci s kolekcí

Při programování skriptů pro tvorbu kolekce pro účely vyhledávání informací byl kladen důraz na následující vlastnosti:

- do kolekce bude možné snadno přidat výstupy z nových ASR systémů
- každé slovo z přepisů ASR systémů 2004 a 2006 bude obohaceno o své lemma a morfologický tag
- ke každému slovu lze jednoduše přidat další tvary a značky
- výstupní data budou v XML formátu
- skript pomocí volby parametrů umožňuje vytvoření různých kolekcí

Vytvoření nástrojů pro generování kolekcí dokumentů můžeme rozdělit do několika kroků. Nejdříve jsem musel zkompletovat data ze všech pracovišť, které se na zpracování českých rozhovorů z projektu MALACH podílela, a zkontrolovat jejich vzájemnou shodu. V dalším kroku jsem ke všem rozpoznaným slovům z obou ASR systémů přidal lemmata a tagy a byla zvolena míra, která porovnává kvalitu rozpoznaných textů. Posledním krokem bylo vytvoření generujícího skriptu a zvolení parametrů, kterými se určují vlastnosti vygenerované kolekce. Více o jednotlivých krocích je napsáno v následujících sekcích.

5.1 Zpracování dokumentů

5.1.1 Příprava dat

Pro projekt Visual History Foundation bylo natočeno celkem 575 českých rozhovorů, které byly nahrány na 2174 páskách. Pro potřeby projektu MALACH bylo do roku

2006 z tohoto množství poskytnuto jen 1294 pásek. Předané pásky nebyly vytvořeny výběrem z natočených rozhovorů. U některých rozhovorů chybí jedna páska z pěti, jiné jsou zcela kompletní.

ASR systémem z roku 2004 bylo zpracováno 1256 pásek z tohoto množství. Stejné pásky a k tomu nových 24 pásek bylo do kolekce přidáno pro rozpoznávač z roku 2006. Ke zbývajícím 14 páskám existuje pouze audio záznam, který ale nebyl použit ani pro jeden rozpoznávač řeči. Anotátorský software neobsahoval časové údaje o rozpoznaných slovech u rozhovoru číslo 36164 (celkem 4 pásky), a proto je z kolekce vyřazen. Dále anotátorský software neobsahoval audio záznamy u 23 pásek. Tyto pásky ale byly do kolekce přidány, protože anotátoři relevantní pasáže vyhledávali v textových přepisech z ASR systému z roku 2006, které se v anotátorském software zobrazovaly (anotátoři pouze nemohli pro kontrolu relevance nalezených pasáží využít originální audio záznam).

	počet pásek
pásky pro ASR 2004	1256
pásky pro ASR 2006	1280
audio záznamy pásek	1271
anotátorský software	1276
celkem různých pásek	1294

Tabulka 5.1: Přehled počtu pásek v jednotlivých krocích

Do kolekce tedy byly zařazeny všechny pásky, které měli k dispozici anotátoři. U každé z těchto pásek máme její přepis z rozpoznávače z roku 2006. Přepisy z rozpoznávače z roku 2004 chybí u 24 pásek z této kolekce. Následující tabulka přehledně ukazuje kompletnost jednotlivých rozhovorů.

kompletnost	počet rozhovorů	počet poskytnutých pásek
kompletní rozhovor	310	1137
chybí 1 páska	32	113
chybí 2 pásky	10	24
chybí 3 pásky	1	2
Celkem	353	1276

Tabulka 5.2: Kompletnost rozhovorů

5.1.2 Morfologická analýza a lemmatizace

Špičkové vyhledávací systémy obvykle obsahují alespoň základní lingvisticky motivované předzpracování dokumentů a dotazů, které by vylepšilo vlastnosti vyhledávacího systému. Pro anglicky jazyk tímto předzpracováním většinou bývá hledání kořene slova a jeho přípony, jako poskytuje například dobře známý Porterův stemmer [10]. Čeština je ale na rozdíl od angličtiny morfologicky bohatý jazyk a neexistuje pro ni tak jednoduchý způsob, jak rozdělit slovo na předpony, kořen a přípony. Proto jsem se rozhodl pro slova v této kolekci použít český morfologický analyzátor a tagger vytvořený na Univerzitě Karlově v Praze [16], [17], který každému vstupnímu slovu přiřadí základní slovní tvar (lemma) a morfologickou značku. Přesnost tohoto nástroje je přibližně 95%. Příklad výstupu pro jedno slovo je:

```
<f>koncentračních<MDl>koncentrační<MDt>AAIP6----1A----
```

Značka `<f>` je následován originální podobou slova, po značce `<MDl>` je uvedeno lemma a značka `<MDt>` odděluje 15 místný řetězec s morfologickými kategoriemi (na první pozici je uveden slovní druh; A reprezentuje adjektivum), více informací lze nalézt v [16].

Hajičův morfologický analyzátor a tagger může být volitelně spuštěn s hádací jednotkou (guesser), která je částí morfologické analýzy a která generuje výstup i pro slova, která nejsou uvedena v morfologickém slovníku. Tato jednotka se snaží na základě podoby neznámého slova odhadnout, jaké lemma a morfologická značka je pro toto slovo nejpravděpodobnější. Je-li hádací jednotka vypnuta, neznámé slovo je označeno speciálním taggem začínajícím písmenem `X`. Analyzátor a tagger očekává na vstupu standardně napsaný text, ale výstupní text z rozpoznávačů řeči je celý napsán velkými písmeny. Abychom mohli správně určit lemma a morfologická značku, bylo nutné každé slovo převést na všechny tři možné tvary zápisu (příklad: „tvar“, „Tvar“, „TVAR“). Takto přepsané texty byly poté dvakrát zpracovány analyzátorem a taggerem (se zapnutou a bez zapnuté hádací jednotky). Z šesti výsledných lemmat a tagů jsme se podle následujícího klíče snažili vybrat správné lemma a morfologickou značku.

Pokud byl bez zapojení hádací jednotky rozpoznán pouze jeden tvar slova, pak je tento tvar slova hledané slovo. V ostatních případech je při shodě lemmat a tagů u dvou tvarů ze tří upřednostněn tvar slova psaný malými písmeny (nebo tvar s velkým počátečním písmenem u shody „Tvaru“ a „TVARU“). Pokud není shoda ani u dvou tvarů, hledá se shoda u tvarů ve výstupních datech se zapnutou hádací jednotkou,

dle stejného klíče, a pokud i poté nelze uplatnit žádné pravidlo, slovo se napíše ve tvaru s malými písmeny.

Do výsledných souborů, ze kterých skript generuje kolekci, jsou kromě originální podoby slova uloženy také odpovídající lemma, plná morfologická značka (15 pozic) a zkrácená morfologická značka tvořená pouze 1., 4., 10. a 11. pozicí plné morfologické značky (tyto pozice morfologické značky mění význam slova pro účely vyhledávání informací). Pro strukturní značky `<s>` a `</s>` nebyly vloženy žádná lemmata ani jiné značky. Příklad věty po morfologické analýze a lemmatizaci ukazuje tabulka 5.1.2.

původní tvar	lemma	morfologická značka plná	morfologická značka zkrácená
<s>			
KTERÝ	který	P4YS1----------	PY--
MĚL	mít	VpYS---XR-AA---	VY-A
VELKOU	velký	AAFS4----1A----	AF1A
SMŮLU	smůla	NNFS4-----A----	NF-A
V	v-1	RR--6----------	R---
NĚMECKU	Německo	NNNS6-----A----	NN-A
PROTOŽE	protože	J,------------	J---
KDEKOLI	kdekoliv	Db----------1	D---
FOTOGRAFOVAL	fotografovat	VpYS---XR-AA---	VY-A
TAK	tak-3	Db------------	D---
TA	ten	PDFS1----------	PF--
BUDOVA	budova	NNFS1-----A----	NF-A
POZDĚ	pozdě	Dg-------1A----	D-1A
VYLETĚL	vyletět	VpYS---XR-AA---	VY-A
DO	do-1	RR--2----------	R---
POVĚTŘÍ	povětří	NNNS2-----A----	NN-A
</s>			

Tabulka 5.3: Výstupní data po lemmatizaci

5.1.3 Porovnání kvality dokumentů

Kolekce automatických přepisů záznamů, která je přiložena k této práci, obsahuje výstupní texty ze dvou systémů pro automatické rozpoznávání řeči a výstupní texty dalšího ASR systému lze do kolekce snadno přidat. Pro účely vyhledávání informací v těchto přepisech, můžeme požadovat přepisy pouze z určitého systému, nebo také

můžeme chtít použít pouze nejlepší přepis. Pro vybrání nejkvalitnějšího rozpoznávacího systému, ale potřebujeme stanovit metodu, kterou budeme určovat míru kvality dokumentu.

V kolekci výpovědí, ve kterých vyhledávali anotátoři relevantní pasáže, byl mírou kvality počet slov rozpoznaný daným ASR systémem. Rozpoznané texty však obsahovaly několik tokenů, které tento výpočet negativně ovlivnily. Jmenovitě se jedná o tokeny `<s>` a `</s>`, které označují dlouhou pauzu mezi slovy, tedy začátek a konec věty, jak jej chápeme u mluvené řeči. A také se jedná o tokeny `ES`, `SPI`, `WO` a `ER` a další,které odpovídají pouze rozpoznanému šumu a neřečovým událesotem. Z tohoto důvodu byl skript na výpočet kvality upraven o seznam slov, které se nemají započítávat mezi významová slova. Výsledky lze shrnout do následující tabulky:

ASR systém	kanál	počet slov	nejkvalitnější
2004	left	2 668 807	648
2004	right	2 111 929	295
2006	left	2 753 606	249
2006	right	2 297 151	77

Tabulka 5.4: Srovnání kvality podle počtu rozpoznaných slov

Z tabulky vyplývá, že při srovnání kvality podle počtu rozpoznaných slov je kvalitnějším ASR systémem ten z roku 2004, i když výstupní přepis tohoto systému obsahuje oproti ASR systému 2006 hovorová slova, která nejsou příliš vhodná pro vyhledávání. Při bližším zkoumání jsem zjistil, že ASR systém z roku 2004 často dlouhá česká slova rozpozná jako několik krátkých slov, čímž se v mnou zvolené metrice stává kvalitnějším. Z tohoto důvodu jsem navrhl novou míru, která kvalitu automatického přepisu určuje podle počtu rozpoznaných písmen u slov, která nejsou ve stoplistu (byl použit stejný stoplist jako u kvality dle počtu slov). Výsledky shrnuje následující tabulka:

ASR systém	kanál	počet písmen	nejkvalitnější
2004	left	10 982 070	220
2004	right	8 478 265	103
2006	left	11 992 209	675
2006	right	10 128 261	268

Tabulka 5.5: Srovnání kvality podle počtu rozpoznaných písmen

Při rozlišování kvality podle počtu rozpoznaných písmen se jako kvalitnější ukazuje ASR systém z roku 2006, což odpovídá našim předpokladům z velikosti WER

na testovacích datech. Rozdílný kanál ASR systému byl oproti první míře kvality vybrán v 736 případech z 1262 dokumentů (58,3%).

5.1.4 Skript pro vytvoření kolekce

K vytvoření kolekce slouží skript `CreateCollection.pl` napsaný v jazyce Perl. Kromě něj je dále potřeba mít ve stejné složce nahrané soubory s těmito informacemi:

- `Interview.data` – v souboru jsou uložena jména svědků holocaustu (vetšinou pouze jméno a počáteční písmeno příjmení)
- `Keywords.data` – soubor s klíčovými slovy, které moderátoři vypisovali při rozhovoru na zvláštní list; soubor celkem obsahuje 38 925 položek
- `TapesLengths.data` – soubor s počátečním časem každé pásky v sekundách
- `QualityLetters.data` a/nebo `QualityWords.data` – soubory obsahují kvality rozpoznaných dokumentů pro každý ASR systém a každý kanál
- `Params.xml` – soubor s parametry, podle kterých se vytváří kolekce (více níže)
- `ASR/` – adresář obsahující přepisy z rozpoznávačů řeči; rozpoznaná páska musí být uložena v cestě `ASR/name/channel/interview.tape`, kde `name` označuje jméno ASR systému, `channel` rozpoznávaný kanál, `interview` je číslo rozhovoru a přípona `tape` označuje pořadové číslo pásky

Vlastnosti vytvářené kolekce lze ovlivnit těmito parametry perlovského skriptu. Parametry je třeba zadat přesně v tomto pořadí:

- `params_file` – cesta k parametrovému souboru
- `window_size` – velikost dokumentu
- `overlap` – velikost přesahu dvou dokumentů, musí být menší než velikost dokumentu
- `length_type` – velikost generovaných dokumentů a jejich přesahů můžeme udávat buď v sekundách (`seconds`) nebo slovech (`words`)
- `ommit_token_s` – má-li parametr hodnotu `ommit_s`, nebudou se strukturní značky `<s>` a `</s>` vypisovat do výstupního souboru

- `fallback_option` – má-li parametr hodnotu `no_fallback`, nemůže být použita jiná páska než ta, která je zadána v parametrovém souboru, má-li jinou hodnotu, skript u chybějící pásky vybere nejlepší z ostatních
- `word_output_separator` – zadaný textový řetězec bude oddělovat jednotlivá slova na výstupu
- `output_file` – cesta k výstupnímu souboru

Vlastnosti kolekce se dále nastavují v souboru `Params.xml`. Tento soubor obsahuje elementy `<lengths>`, `<interviews>`, `<keywords>` a `<quality>`, které obsahují cesty k výše zmíněným souborům. Element `<datafield>` obsahuje v elementech `<column>` názvy slovních forem, které obsahuje každé rozpoznané slovo (výjimku tvoří pouze strukturní značky `<s>` a `</s>`, které tyto formy obsahovat nemusí). Element `<textfield>`, obsahuje element `<name>` se jménem značky ve výstupních dokumentech, která bude obsahovat všechny slovní formy uvedené v elementech `<word_form>`, které pochází z přepisů rozpoznávače v elementu `<asr>` a z kanálu v elementu `<channel>`.

Následující volání skriptu s uvedeným parametrickým souborem, tak vytvoří dokumenty dlouhé 180 sekund s překryvem 60 sekund, které budou obsahovat 2 elementy. Element `<ORIG>` obsahuje pouze originální slovní formy nejlepších přepisů z rozpoznávače z roku 2006 a element `<LEMMA+TAG>` tvoří lemma, plná morfologická značka a zkrácená morfologická značka nejlepších přepisů pravého kanálu.

```
> CreateCollection.pl Params.xml 180 60 seconds ommit_s no_fallback
"/n" output.xml
```

Obsah souboru Params.xml:

```
<parameters>
  <lengths>TapesLengths.data</lengths>
  <interviews>Interview.data</interviews>
  <keywords>Keywords.data</keywords>
  <quality>QualityLetters.data</quality>
  <datafield>
    <column>lemma</column>
    <column>full_tag</column>
    <column>short_tag</column>
  </datafield>
```

```
  <textfield>
    <name>ORIG</name>
    <asr_system>2006</asr_system>
    <channel>best</channel>
    <word_form>orig_form</word_form>
  </textfield>
  <textfield>
    <name>LEMMA+TAG</name>
    <asr_system>best</asr_system>
    <channel>right</channel>
    <word_form>lemma</word_form>
    <word_form>full_tag</word_form>
    <word_form>short_tag</word_form>
  </textfield>
</parameters>
```

Ukázkový výstup:

```
<DOCNO>04106.00120</DOCNO>
<TEXT>
<INTERVIEWDATA>Tommy K...-K...</INTERVIEWDATA>
<KEYWORDS></KEYWORDS>
<ASRSYSTEM>2006</ASRSYSTEM>
<CHANNEL>right</CHANNEL>
<LEMMA+TAG>
takový       PDXP6----------   PX--
STOČLENÝCH   NNFXX-----A---8   NF-A
skupina      NNFP6-----A----   NF-A
prostě       Dg-------1A----   D-1A
jít          VpMP---XR-AA---   VM-A
pěšky        Db-------------   D---
překvapit    VsYS---XX-AP---   VY-A
tři          ClXP4----------   CX--
kilometr     NNIP4-----A----   NI-A
do-1         RR--2----------   R---
ten          PDZS2----------   PZ--
...
```

```
</LEMMA+TAG>
<ASRSYSTEM>2006</ASRSYSTEM>
<CHANNEL>right</CHANNEL>
<ORIG>
TAKOVÝCH
STOČLENÝCH
SKUPINÁCH
PROSTĚ
ŠLI
PĚŠKY
PŘEKVAPEN
TŘI
KILOMETRY
DO
TOHO
...
</ORIG>
</TEXT>
</DOC>
```

5.2 Zpracování témat

Témata pro vyhledávání byla zpracována podobně jako dokumenty. Protože témata jsou popsána ve větách, bylo nutné tato data před zpracováním Hajičovým morfologickým analyzátorem a taggerem tokenizovat a segmentovat (více o tomto nástroji v [16], [17]). Pro tyto účely jsem použil nástroj, který jsem vytvořil pro svou bakalářskou práci [18]. Výstupní data z témat vypadají podobně jako u dokumentů, navíc jsou ale rozčleněna na pole `<title>`, `<desc>` a `<narrative>`.

Ukázkové téma:

```
<top number="1166">
<title>
Chasidismus    chasidizmus  NNIS1-----A----  NS-A
</title>
<desc>
```

Chasidové	chasid	NNMP1-----A----	NP-A
a	a-1	J^------------	J---
jejich	jeho	PSXXXXP3-------	PX--
nezlomná	nezlomný	AAFS1----1A----	AS1A
víra	víra	NNFS1-----A----	NS-A
</desc>			
<narr>			
Relevantní	relevantní	AAIS1----1A----	AS1A
materiál	materiál	NNIS1-----A----	NS-A
by	být	Vc------------	V---
měl	mít	VpYS---XR-AA---	VS-A
vypovídat	vypovídat	Vf--------A----	V--A
o	o-1	RR--6----------	R---
Chasidismu	chasidizmus	NNIS6-----A----	NS-A
v	v-1	RR--6----------	R---
období	období	NNNS6-----A----	NS-A
před	před-1	RR--7----------	R---
holokaustem	holokaust	NNIS7-----A----	NS-A
v	v-1	RR--6----------	R---
průběhu	průběh	NNIS6-----A----	NS-A
holokaustu	holokaust	NNIS2-----A----	NS-A
a	a-1	J^------------	J---
po	po-1	RR--6----------	R---
něm	on-1	P5ZS6--3-------	PS--
Informace	informace	NNFS1-----A----	NS-A
o	o-1	RR--6----------	R---
chasidských	chasidský	AAFP6----1A----	AP1A
dynastiích	dynastie	NNFP6-----A----	NP-A
a	a-1	J^------------	J---
založených	založený	AAFP6----1A----	AP1A
a	a-1	J^------------	J---
zničených	zničený	AAFP6----1A----	AP1A
geografických	geografický	AAFP6----1A----	AP1A
lokalitách	lokalita	NNFP6-----A----	NP-A
</narr>			
</top>			

5.3 Evaluační skript

Pro vyhodnocení evaluace slouží skript `mGAP.pl`, který vypočítává průměrnou hodnotu míry GAP ze všech vyhledávacích témat. Tento skript pro své spuštění potřebuje 4 základní parametry:

- `window_size` – tento parametr v sekundách určuje maximální odchylku rozpoznaného počátku od počátku označeného anotátorem pro započítání výsledků
- `granularity` – tento parametr v sekundách určuje, jak rychle bude klesat ohodnocení vyhledané pasáže. Za každých celých `granularity` sekund klesne ohodnocení o `granularity`/`window_size` sekund. Více o výpočtu míry GAP je uvedeno v kapitole 4.5.
- `assessments` – cesta k souboru, který obsahuje relevantní pasáže vyhledané anotátory ve formátu: identifikační číslo tématu, identifikační číslo rozhovoru, počáteční a koncový čas relevantní pasáže
- `ranked_list` – cesta k souboru, který obsahuje setříděný seznam výstupních dokumentů vyhledávacího systému ve standardním TREC formátu (identifikační číslo tématu, dummy číslo, identifikační číslo dokumentu, pořadové číslo ve výstupu, hodnota pro vyhledávací systém a název pokusu). Identifikační číslo dokumentu obsahuje kromě čísla rozhovoru i počáteční čas prvního slova tohoto dokumentu

Základní nastavení skriptu je s parametrem `window_size` 150 sekund a parametrem `granularity` 15 sekund (tj. klesání hodnotící funkce o 0,1 bodu za každých celých 15 sekund rozdílu mezi časy).

Kapitola 6

Experimenty

V následující kapitole nejdříve rozdělíme témata na dvě skupiny – množinu trénovacích a testovacích témat. S nástroji, které jsem představil v této práci vytvořím několik kolekcí, u kterých použiji různé vyhledávací modely a další techniky vylepšující výsledky vyhledávání jako je například lemmatizace, odstranění nevýznamových slov či rozšiřování vyhledávacího dotazu. Budu také zkoumat vliv délky a přesahu generovaných dokumentů, což jeproblém specifickýpro tuto kolekci dat. Podle výsledků dosažených na trénovacích tématech, zvolíme nastavení 3 závěrečných experimentů, které provedeme s testovacími tématy.

6.1 Rozdělení témat

Pro kolekci jsem použil pouze 83 témat, u kterých anotátoři označili alespoň 5 relevantních pasáží. Dvě z těchto témat (1173 a 24313) jsem automaticky zařadil mezi trénovací témata, protože se jedná o témata, na kterých se anotátoři učili správně data anotovat. Téma 2055 s nejméně relevantními pasážemi jsem také zařadil mezi trénovací témata.

Zbylých 80 témat jsem setřídil podle klesajícího počtu relevantních pasáží a rozdělil je do čtveřic. Každou tuto čtveřici jsem zvlášť setřídil podle klesajícího počtu slov v názvu a popisu tématu (elementy `<title>` a `<desc>`). Za trénovací témata jsem poté označil sudá témata z lichých čtveřic a lichá témata v sudých čtveřicích. Podrobný postup rozdělení můžete také vidět v tabulce 6.1.

téma	# pasáží	# slov	kolekce	téma	# pasáží	# slov	kolekce
4006	133	21	test	2264	41	29	test
3028	173	14	train	14312	39	11	train
3009	144	13	test	3000	41	10	test
1508	139	8	train	1897	43	10	train
2253	126	19	train	1311	38	20	train
2358	107	16	test	1554	38	16	test
2000	113	15	train	1321	34	15	train
4011	110	6	test	1424	37	10	test
3020	104	26	test	3018	27	47	test
15601	102	17	train	3026	27	23	train
3007	102	13	test	4001	32	20	test
2265	100	9	train	1187	33	14	train
3005	94	19	train	1630	22	19	train
3024	88	15	test	1181	24	17	test
3002	87	14	train	2185	25	15	train
3001	89	14	test	1310	25	8	test
4000	87	21	test	2198	20	31	test
1286	86	20	train	2367	20	23	train
3017	87	19	test	15602	22	23	test
3014	86	13	train	3022	21	22	train
3015	78	20	train	3010	17	24	train
1843	79	17	test	1192	18	18	test
3027	75	12	train	14313	17	10	train
3033	80	10	test	1166	17	6	test
1620	61	20	test	3019	13	27	test
2012	69	18	train	3021	16	18	train
3025	74	16	test	1979	16	16	test
3023	66	13	train	1330	13	15	train
1185	53	26	train	1345	12	25	train
2224	59	26	test	1337	11	19	test
4012	58	14	train	1225	11	12	train
4005	55	6	test	4010	12	10	test
4007	50	16	test	4004	11	27	test
2006	51	16	train	3032	9	23	train
1663	50	10	test	1605	10	19	test
3008	53	10	train	4002	11	7	train
3011	44	21	train	4009	8	24	train
3004	44	21	test	1829	6	21	test
2384	48	20	train	2404	6	15	train
3016	49	10	test	1288	8	8	test

Tabulka 6.1: Přehled postupu pro rozdělení témat na trénovací a testovací množinu.

6.2 Vyhledávací nástroje

Pro všechny experimenty jsem použil vyhledávacích nástrojů LEMUR[1] a INDRI[2]. Tyto nástroje obsahují několik vyhledávacích modelů (TFIDF model, Okapi BM25), které dokáží zpracovat pouze dokumenty v anglickém jazyce[3]. Z tohoto důvodu jsem nejdříve převedl všechna písmena v termech a dokumentech na malá a poté jsem zobrazil množinu takto upravených termů na množinu přirozených čísel a dále pracoval pouze s touto reprezentací dat.

6.3 Základní experiment

Abychom mohli snadno srovnávat zlepšení vyhledávacího systému u různých experimentů, provedeme nyní základní experiment (Baseline Experiment), který stanoví výchozí hodnoty pro porovnávání s dalšími experimenty. Tento experiment na závěr také provedeme na testovacích datech, abychom viděli, jaké zlepšení na nezávislých datech přinesou experimenty představené dále v tomto textu. Délku generovaných dokumentů pro tento experiment jsem zvolil 180 sekund, což přibližně odpovídá průměrné délce relevatní pasáže, jak ji označili anotátoři[4]. Přehled všech parametrů pro experiment naleznete níže.

Parametry základního pokusu:
Vyhledávací model: Raw TFIDF
ASR systém a kanál: nejlepší kanál ze všech systémů[5]
Termy: originální slovní tvary
Nevýznamová slova: žádná
Velikost a překryv dokumentů: 180/60 sekund
Dotazy: `<title>` + `<description>`
Rozšiřování dotazu: žádné

[1] http://www.lemurproject.org/

[2] http://www.lemurproject.org/indri/

[3] Term (slovo) je pro tyto systémy definován jako posloupnost písmen a číslic anglické abecedy. Každý český znak s diakritikou tak rozdělí term na dvě části.

[4] Toto nastavení bylo zvoleno i u kolekce Quickstart, která byla poskytnuta účastníkům soutěže CLEF CL-SR Track v letech 2006 a 2007. Kvůli chybě v programu, který generoval kolekci, byly dokumenty nerovnoměrně dlouhé v intervalu 150 až 210 sekund, ale původním záměrem organizátorů bylo generovat dokumenty dlouhé přesně 180 sekund.

[5] Pro porovnání kvality jednotlivých výstupů jsme použili porovnání podle počtu rozpoznaných písmen, více podrobností v kapitole 5.1.3.

Pro evaluaci výsledků byl použit skript `mGAP.pl` s parametry `window_size` 150 sekund a `granularity` 15 sekund[6]. Toto nastavení evaluačního skriptu budeme používat u všech experimentů uvedených v této práci.

Výsledná hodnota mGAP[7] na trénovacích datech je **0,0247**.

6.4 Experiment s lemmatizací

U dalšího experimentu se pokusíme prokázat pozitivní vliv lemmatizace na výsledky vyhledávání. Všechna slova v originálních přepisech z ASR systémů i ve všech vyhledávacích dotazech jsme nahradili jejich základní slovním tvarem – lemmatem. Ostatní parametry jsme pro tento experiment zvolili stejné jako u baseline (základního experimentu).

Výsledná průměrná hodnota GAP je **0,0363**. Přehled hodnot GAP u jednotlivých témat a jejich srovnání s baseline můžete vidět na obrázku 6.1.

Pro porovnání výsledků s baseline jsme použili Wilcoxonův párový test. Na základě výsledků tohoto testu s pravděpodobností $p < 0,05$ zamítáme hypotézu, že mezi výsledky není signifikantní rozdíl. Pozitivní vliv lemmatizace na výsledky vyhledávání jsem ověřil také při účasti na soutěži CLEF [19].

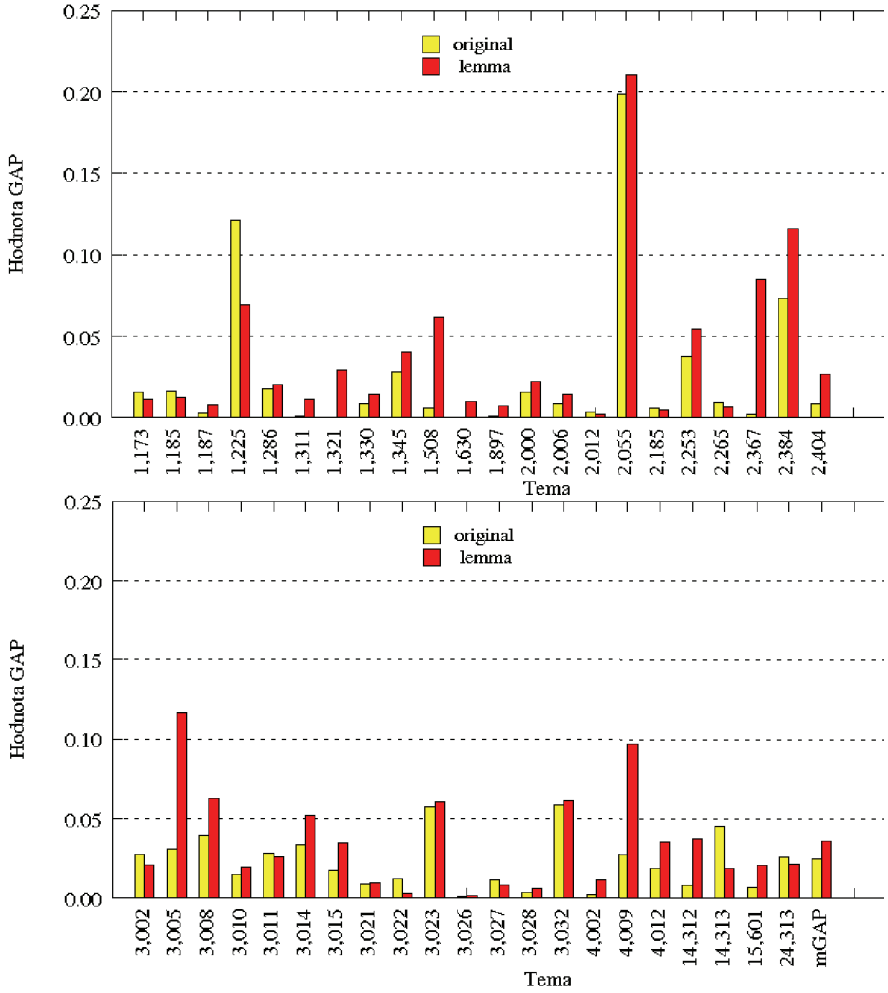
6.5 Experimenty s různými vyhledávacími modely

V těchto experimentech srovnáme výsledky čtyř vyhledávacích modelů implementovaných v nástrojích LEMUR a INDRI:

- modelu TFIDF se základní formulí pro výpočet TF (Raw TFIDF)
- modelu TFIDF s formulí BM25 pro výpočet TF (BM25 TFIDF) a hodnotami parametrů: $k_1 = 1$, $b = 0,3$
- modelu Okapi BM25 (Okapi) a hodnotami parametrů: $k_1 = 1,2$, $b = 0,75$, $k_3 = 7$
- modelu INDRI při použití Dirichletovy metody výběru dokumentů a parametru $mu = 2500$

[6]Toto nastavení bylo použito rovněž u soutěže CLEF 2006 a 2007 CL-SR Track.

[7]Zkratkou **mGAP** budeme v dalším textu označovat mean GAP – průměrnou hodnotu zobecněné průměrné přesnosti.



Obrázek 6.1: Přehled vlivu lemmatizace u jednotlivých témat

Tyto modely jsme použili pro experimenty s originálními přepisy ASR systémů a také s lemmatizovanými přepisy. Hodnoty ostatních parametrů (ASR systém a kanál, nevýznamová slova, velikost a překryv dokumentů, rozšiřování dotazu) jsme nastavili stejně jako u základního experimentu. Výsledky shrnuje tabulka 6.2 a obrázky 6.2 a 6.3.

	originál	lemmata
Raw TFIDF	**0,0247**	**0,0363**
BM25 TFIDF	**0,0282**	**0,0379**
Okapi	0,0190	0,0272
INDRI	**0,0260**	**0,0385**

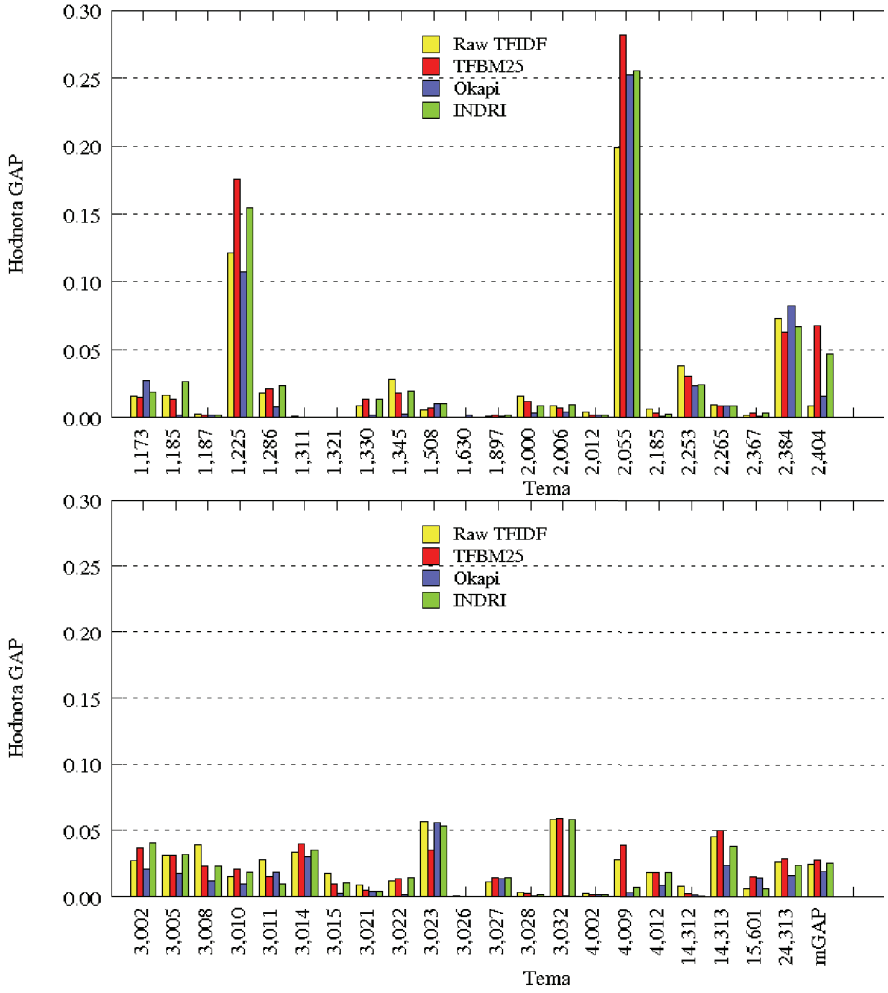
Tabulka 6.2: mGAP skóre různých vyhledávacích modelů.

Z výsledků můžeme vidět, že každý z vyhledávacích systémů poskytuje lepší výsledky na lemmatizovaných datech, což potvrzuje naši domněnku z minulého experimentu. Model Okapi je statisticky rozeznatelně horší než všechny ostatní modely. Přesné důvody, proč tomu tak je nelze jednoduše určit, ale těchto výsledků dosáhl i Ircing [20], [21]. Ircing se domnívá, že výsledky jsou horší možná kvůli tomu, že model Okapi těží zejména z různé délky dokumentů v kolekci (v testované kolekci se délka dokumentů v počtu termů liší jen nepatrně). Pokud by tomu tak bylo, měl by být také výsledek u modelu BM25 TFIDF horší. Proto se domnívám, že za špatným výsledkem modelu Okapi budou jiné důvody – špatná volba parametrů či jeho chybná implementace v nástroji LEMUR.

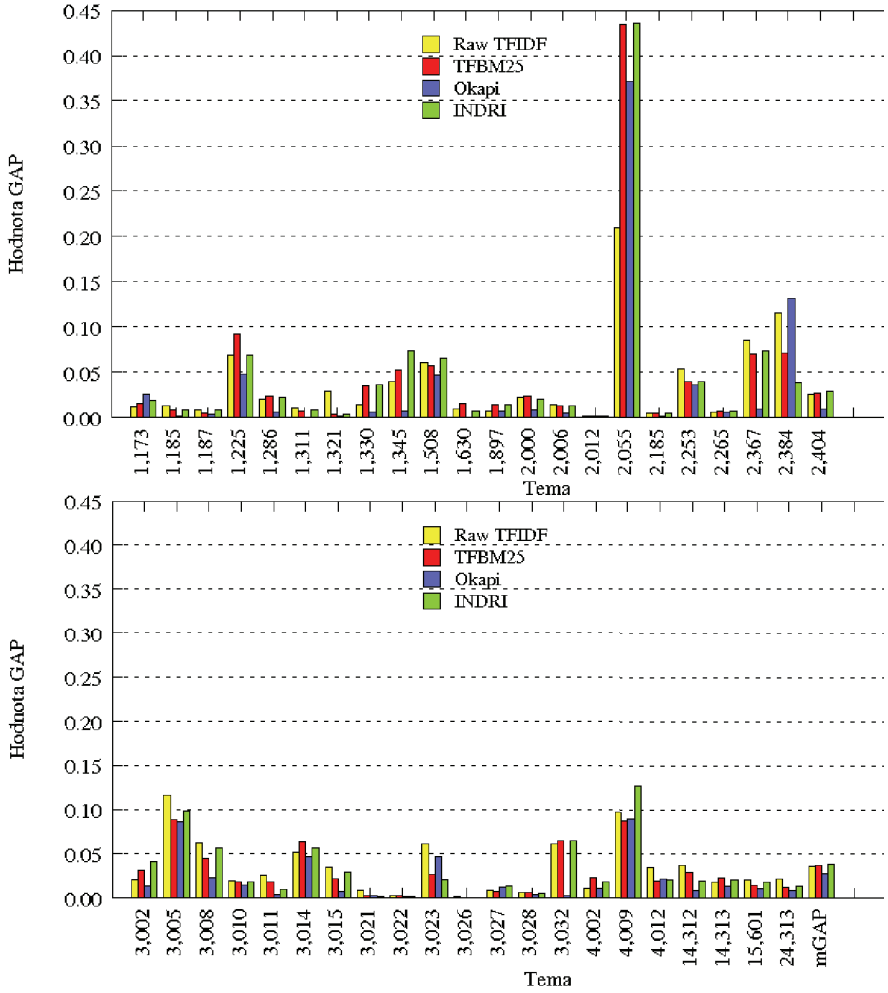
Protože modely Raw TFIDF, BM25 TFIDF a INDRI dosáhli statisticky nerozeznatelných výsledků, budeme další experimenty provádět pouze s jedním s těchto modelů – modelem Raw TFIDF, který jsme zvolili kvůli jeho jednoduchosti použití.

6.6 Experiment s rozšiřováním vyhledávacího dotazu

Pro rozšíření vyhledávacího dotazu jsme použili metodu nazvanou slepá zpětná vazba. Vyhledávací systém nejdříve vrátí uspořádaný seznam výstupních dokumentů a z n nejlépe hodnocených dokumentů vybere m termů, které nejlépe popisují tyto dokumenty a původní dotaz jimi obohatí. Tento nově vytvořený dotaz položíme znovu vyhledávacímu systému a obdržíme nové výsledky. Cílem tohoto experimentu je zjistit, jestli metoda slepé zpětné vazby přinese zlepšení vyhledávání.



Obrázek 6.2: Srovnání různých vyhledávacích modelů pro originální tvary slov



Obrázek 6.3: Srovnání různých vyhledávacích modelů pro lemmatizovaná data

Experimenty budeme provádět se stejným nastavením parametrů, jaké jsme použili u základního pokusu a také se základním pokusem s lemmatizovanými daty. Pro experimenty jsme zvolili tři nastavení pro použití slepé zpětné vazby:

- nepoužijeme žádnou zpětnou vazbu
- do vyhledávacího dotazu přidáme 20 termů z 5 nejlépe hodnocených dokumentů
- do vyhledávacího dotazu přidáme 20 termů z 50 nejlépe hodnocených dokumentů

K tomuto nastavení nás vedla domněnka, že kvůli dosavadním výsledkům pohybujících se na hladine mGAP kolem 0,02 až 0,03 bude lepší pro výběr rozšiřujících termů použít více nejlépe hodnocených dokumentů.

	originál	lemmata
bez zpětné vazby	0,0247	0,0363
slepá zpětná vazba, $m = 20, n = 5$	0,0273	0,0367
slepá zpětná vazba, $m = 20, n = 50$	0,0285	**0,0412**

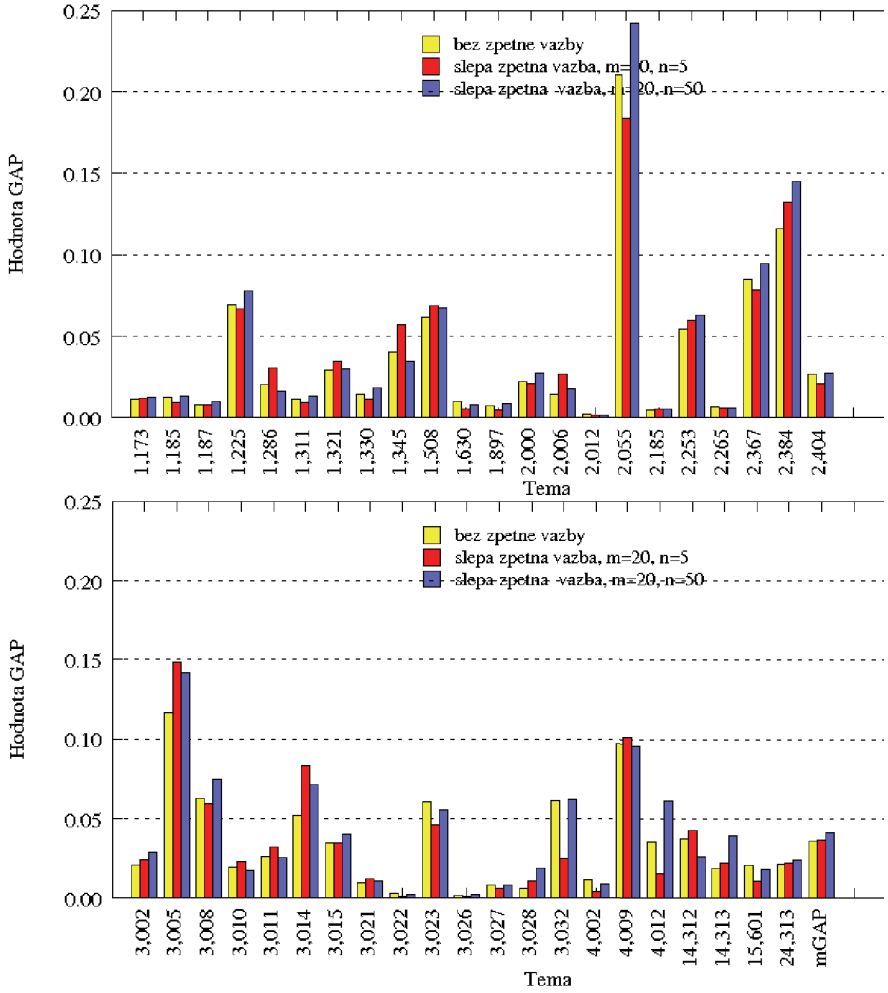
Tabulka 6.3: mGAP skóre pro různé druhy zpětné vazby.

Výsledky v tabulce 6.3 a obrázek 6.4 ukazují, že signifikantní zlepšení výsledků přináší pouze použití druhého nastavení slepé zpětné vazby na lemmatech. Rozšíření počtu dokumentů pro přidávání termů do nového dotazu z 5 na 50 přineslo signifikantní zlepšení na hladině pravděpodobnosti pouze $p < 0,1$. I přesto předpokládám, že má domněnka o zvýšení počtu dokumentů byla správná a použiji ji pro závěrečný pokus na testovacích tématech.

6.7 Experiment s volbou různých ASR systémů a kanálů

V tomto experimentu porovnáme výsledky vyhledávacích systémů vyhledávajících v přepisech z různých kanálů různých ASR systémů. Pokusíme se potvrdit naši domněnku, že ASR systém z roku 2006 je pro vyhledávání vhodnější než ASR systém z roku 2004. V experimentech také srovnáme čtyři různé možnosti volby kanálu:

- použijeme pouze přepisy z levého kanálu
- použijeme pouze přepisy z pravého kanálu



Obrázek 6.4: Přehled vlivu slepé zpětné vazby na vyhledávání

- u každé pásky použijeme ten kanál, který je kvalitnější[8]
- dokumenty pro vyhledávání budou obsahovat termy z obou kanálů (jelikož u modelu TF–IDF nezáleží na pořadí termů v dokumentu – důležité jsou pouze frekvence termů, lze použití oboukanálů snadno implementovat)

Ostatní parametry pokusu jsou stejné jako u základního pokusu s lemmatizovanými daty.

Výsledky experimentů (viz tabulka 6.4 a obrázky 6.5, 6.6 a 6.7) ukazují, že přepisy z ASR systému z roku 2006 jsou signifikantně lepší než přepisy z ASR systému z roku 2004. Experimenty s ASR systémem z roku 2006 zvyšují hodnotu mGAP až téměř o 30% oproti základnímu experimentu s lemmatizovanými daty (pokus uveden v kapitole 6.3).

kanál / ASR systém	2004	2006
levý	**0,0305**	**0,0411**
pravý	0,0196	0,0286
nejlepší	**0,0351**	**0,0475**
levý + pravý	**0,0331**	**0,0462**

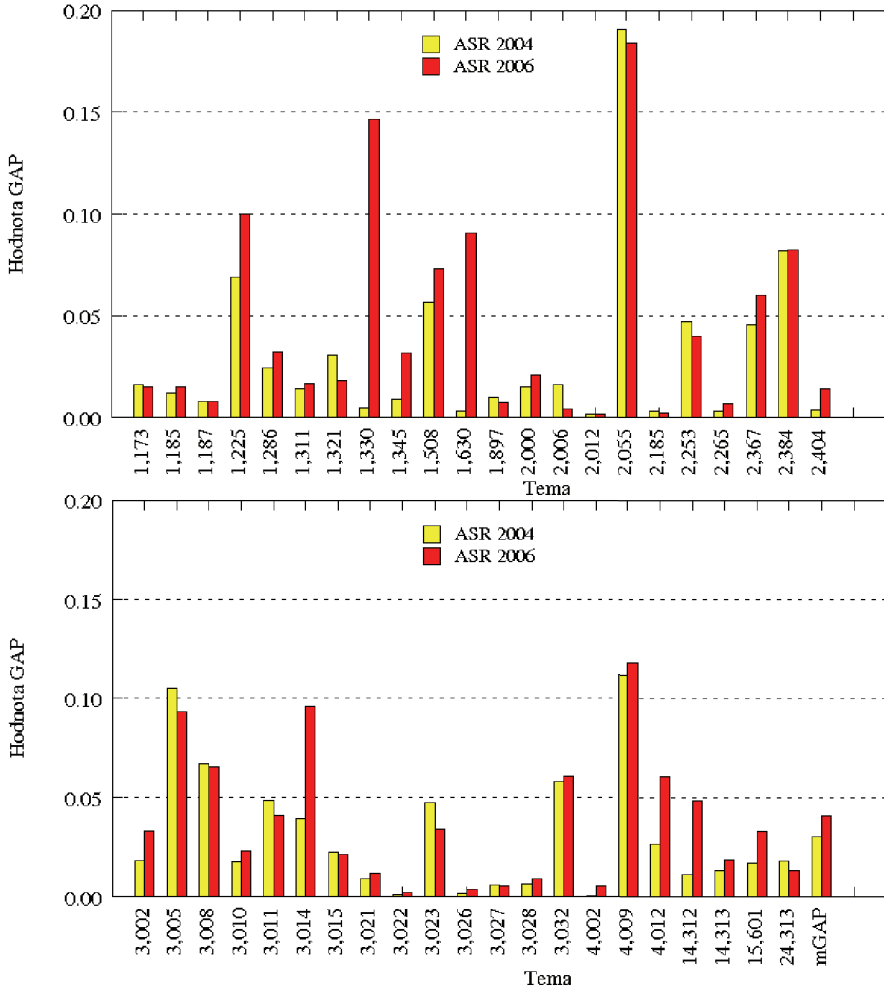
Tabulka 6.4: mGAP skóre pro různé ASR systémy a různé kanály.

Volba kanálu se ukázala také jako důležitá – pro oba systémy bylo použití pouze pravého kanálu signifikantně horší než použití ostatních voleb kanálu. U ASR systému z roku 2004 bylo použití nejkvalitnějšího kanálu signifikantně lepší než použití pouze levého kanálu, ostatní vztahy mezi nejlepšími třemi kanály jsou ale statisticky nevýznamné. U ASR systému z roku 2006 jsou výsledky naprosto shodné. Všechny testy signifikance byly opět provedeny Wilcoxonovým párovým testem s hladinou pravděpodobnosti $p < 0,05$. Pro další experimenty uvedené v této práci jsem již používal pouze přepisy nejkvalitnějšího kanálu ASR systému z roku 2006.

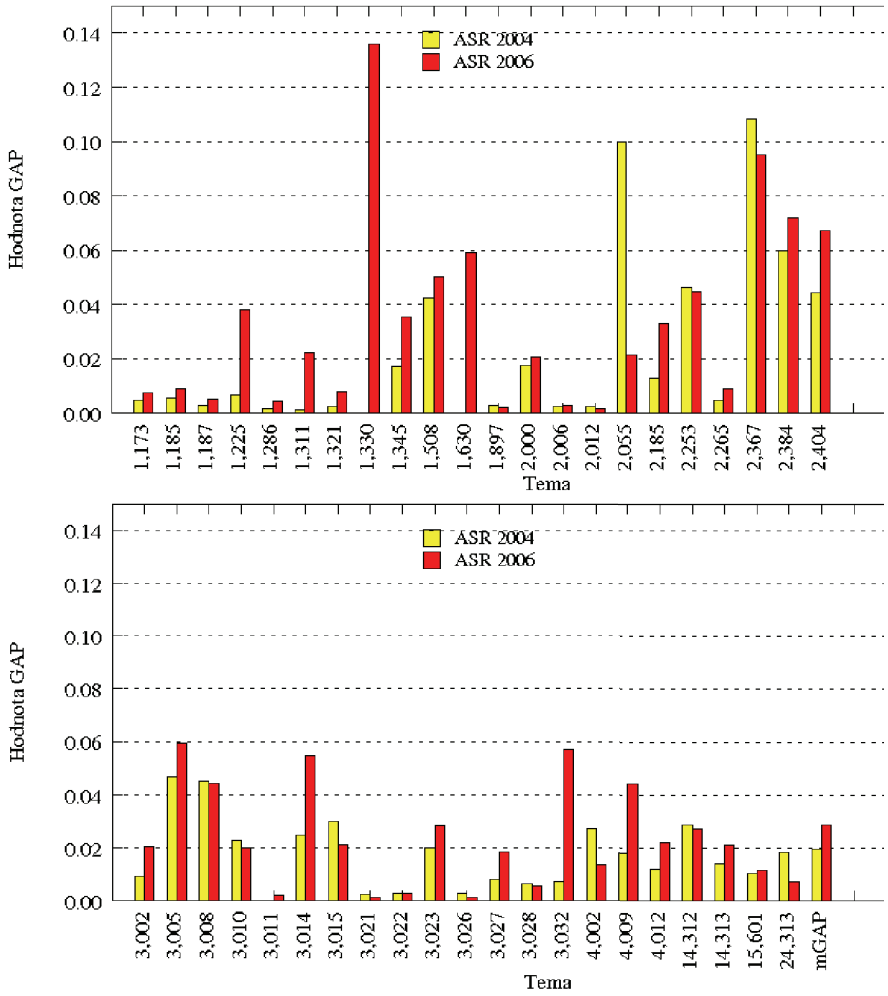
6.8 Experiment s použitím stoplistu

Příliš obecná slova nejsou pro identifikaci dokumentů vhodná, protože se vyskytují téměř v každém dokumentu. Pro vytvoření seznamu těchto slov jsem použil dva přístupy. První seznam jsem vytvořil podle frekvence slov v kolekci – 40 nejčastějších

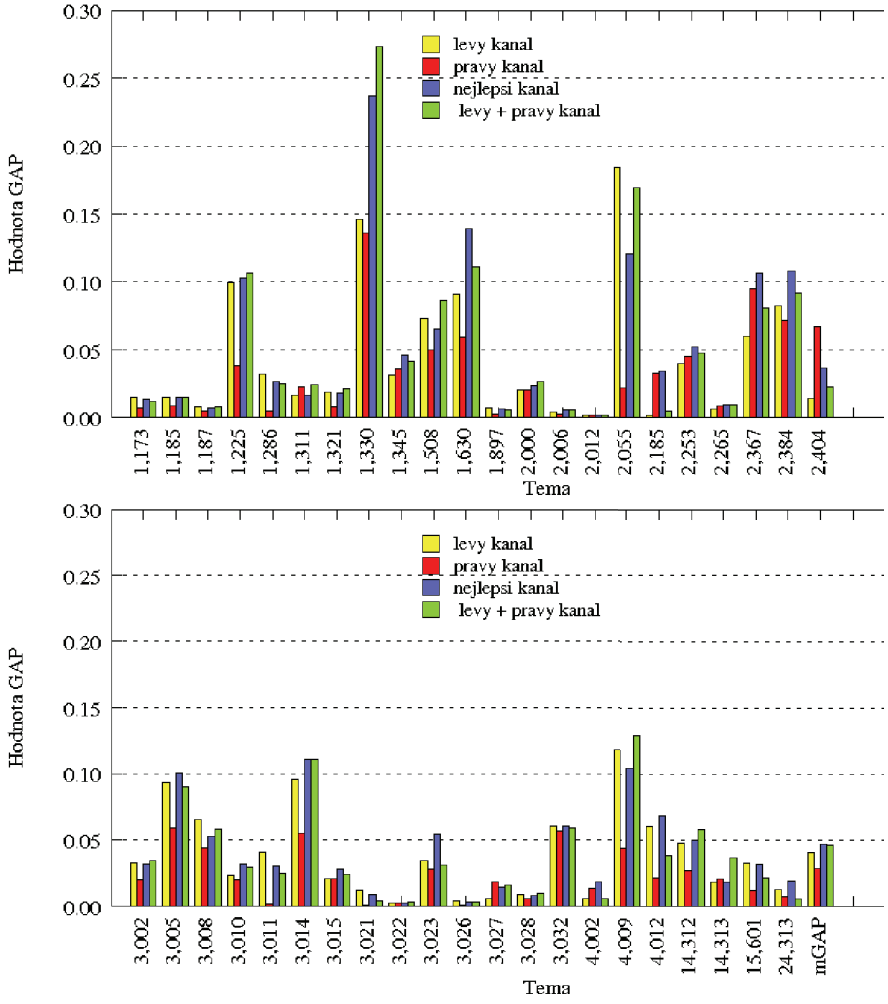
[8]Pro porovnání kvality jednotlivých výstupů jsme použili porovnání podle počtu rozpoznaných písmen, více podrobností v kapitole 5.1.3.



Obrázek 6.5: Přehled vlivu volby ASR systému pro levý kanál



Obrázek 6.6: Přehled vlivu volby ASR systému pro pravý kanál



Obrázek 6.7: Srovnání výsledků pro různé volby kanálu z přepisů ASR sytému z roku 2006

slov bylo ze všech dokumentů i dotazů odstraněno. V tabulkách 6.5 a 6.6 můžeme vidět, že celková kumulovaná frekvence 40 nejčetnějších originálních slov je téměř 40% a u lemmat dosahuje dokonce 50%.

Druhý způsob tvorby stoplistu využívá znalosti slovního druhu (první znak morfologické značky) a mezi nevýznamová slova řadí celé kategorie slovních druhů. Pro tento experiment jsem do stoplistu zařadil všechny zájmena, předložky, spojky, částice, citoslovce a nerozpoznaná slovní druhy. Takto vytvořený stoplist použil také Ircing [20], [21].

Z hlediska použitého stoplistu jsme celkově srovnávali čtyři přístupy k odstranění nevýznamových slov z dokumentů a dotazů:

- žádná lemmata nebyla odstraněna
- bylo odstraněno 40 nejfrekventovanějších lemmat
- byly odstraněny všechny zájmena, předložky, spojky, částice, citoslovce a nerozpoznaná slova
- byly zkombinovány oba výše zmíněné přístupy

Parametry pokusu:
Vyhledávací model: Raw TFIDF
ASR systém a kanál: nejlepší kanál z ASR systému z roku 2006
Termy: lemmata
Nevýznamová slova: proměnný parametr, viz výše
Velikost a překryv dokumentů: 180/60 sekund
Dotazy: `<title>` + `<description>`
Rozšiřování dotazu: žádné

Nejlepšího výsledku podle míry GAP dosáhl experiment bez odstraňování nevýznamových slov, který byl na hladině $p < 0,05$ signifikantně lepší než experiment s frekvenčním stoplistem. Lepší než frekvenční stoplist byl také slovnědruhový stoplist (i když tomu hodnoty mGAP zdánlivě neodpovídají). Ostatní vztahy mezi experimenty nejsou signifikantně významné. Z uvedených výsledků vyplývá, že použití frekvenčního stoplistu signifikantně zhoršuje výsledky vyhledávání. Stoplist tvořený na základě odstranění slovních druhů dosahuje podobných výsledků jako model s plným textem. Výhoda ve využití tohoto stoplistu tedy spočívá ve zmenšení objemu dat pro indexaci o téměř 50% se zachováním kvality vyhledávání. Protože však

Pořadové číslo	slovo	četnost	frekvence	kumulovaná frekvence
1	a	177 232	3,52%	3,52%
2	to	162 947	3,23%	6,75%
3	se	140 259	2,78%	9,54%
4	sem	102 666	2,04%	11,58%
5	že	93 562	1,86%	13,43%
6	jsme	88 622	1,76%	15,19%
7	tam	79 145	1,57%	16,76%
8	na	78 951	1,57%	18,33%
9	tak	68 702	1,36%	19,69%
10	do	64 941	1,29%	20,98%
11	já	54 431	1,08%	22,06%
12	no	53 854	1,07%	23,13%
13	v	50 667	1,01%	24,14%
14	jako	46 787	0,93%	25,07%
15	bylo	46 256	0,92%	25,99%
16	ty	41 715	0,83%	26,81%
17	ale	41 141	0,82%	27,63%
18	byla	38 391	0,76%	28,39%
19	si	38 145	0,76%	29,15%
20	byl	34 554	0,69%	29,84%
21	je	33 794	0,67%	30,51%
22	co	30 531	0,61%	31,11%
23	byli	29 477	0,59%	31,70%
24	ten	28 928	0,57%	32,27%
25	už	28 099	0,56%	32,83%
26	jste	27 522	0,55%	33,38%
27	také	26 361	0,52%	33,90%
28	takže	24 602	0,49%	34,39%
29	ta	24 355	0,48%	34,87%
30	s	23 382	0,46%	35,34%
31	mě	23 314	0,46%	35,80%
32	ještě	23 045	0,46%	36,26%
33	nebo	22 790	0,45%	36,71%
34	toho	22 569	0,45%	37,16%
35	z	21 473	0,43%	37,58%
36	potom	21 453	0,43%	38,01%
37	jak	20 900	0,41%	38,42%
38	prostě	18 877	0,37%	38,80%
39	tom	18 527	0,37%	39,17%
40	nás	18 320	0,36%	39,53%

Tabulka 6.5: Tvorba stoplistu podle frekvence slov (originální tvary)

Pořadové číslo	lemma	četnost	frekvence	kumulovaná frekvence
1	ten	353 006	7,01%	7,01%
2	být	352 530	7,00%	14,01%
3	se	180 727	3,59%	17,59%
4	a-1	176 450	3,50%	21,10%
5	já	138 716	2,75%	23,85%
6	sem	102 454	2,03%	25,88%
7	že	93 610	1,86%	27,74%
8	na-1	78 951	1,57%	29,31%
9	tam	78 850	1,57%	30,87%
10	tak-3	67 411	1,34%	32,21%
11	do-1	64 941	1,29%	33,50%
12	mít	59 133	1,17%	34,68%
13	v-1	58 627	1,16%	35,84%
14	on-1	57 856	1,15%	36,99%
15	no	53 854	1,07%	38,06%
16	jako	46 787	0,93%	38,99%
17	ale	41 141	0,82%	39,80%
18	vědět	33 480	0,66%	40,47%
19	co-1	30 442	0,60%	41,07%
20	už	28 099	0,56%	41,63%
21	s-1	27 395	0,54%	42,17%
22	také	27 158	0,54%	42,71%
23	z-1	26 050	0,52%	43,23%
24	nějaký	25 411	0,50%	43,73%
25	který	25 204	0,50%	44,23%
26	takže	24 602	0,49%	44,72%
27	můj	24 594	0,49%	45,21%
28	jít	23 885	0,47%	45,68%
29	nebo	23 863	0,47%	46,16%
30	ještě	23 045	0,46%	46,62%
31	takový	21 467	0,43%	47,04%
32	potom	21 453	0,43%	47,47%
33	ty	19 636	0,39%	47,86%
34	prostě	18 877	0,37%	48,23%
35	člověk	18 804	0,37%	48,61%
36	jak-3	18 521	0,37%	48,97%
37	rok	18 498	0,37%	49,34%
38	když	17 931	0,36%	49,70%
39	všechen	17 747	0,35%	50,05%
40	za-1	17 542	0,35%	50,40%

Tabulka 6.6: Tvorba stoplistu podle frekvence slov (lemmata)

tato kolekce není nijak významně velká, pro další experimenty nevyužiji žádný ze stoplistů.

stoplist	mGAP
žádný	**0,0475**
frekvenční	0,0458
slovnědruhový	**0,0441**
kombinace	**0,0440**

Tabulka 6.7: mGAP skóre pro různé druhy stoplistu.

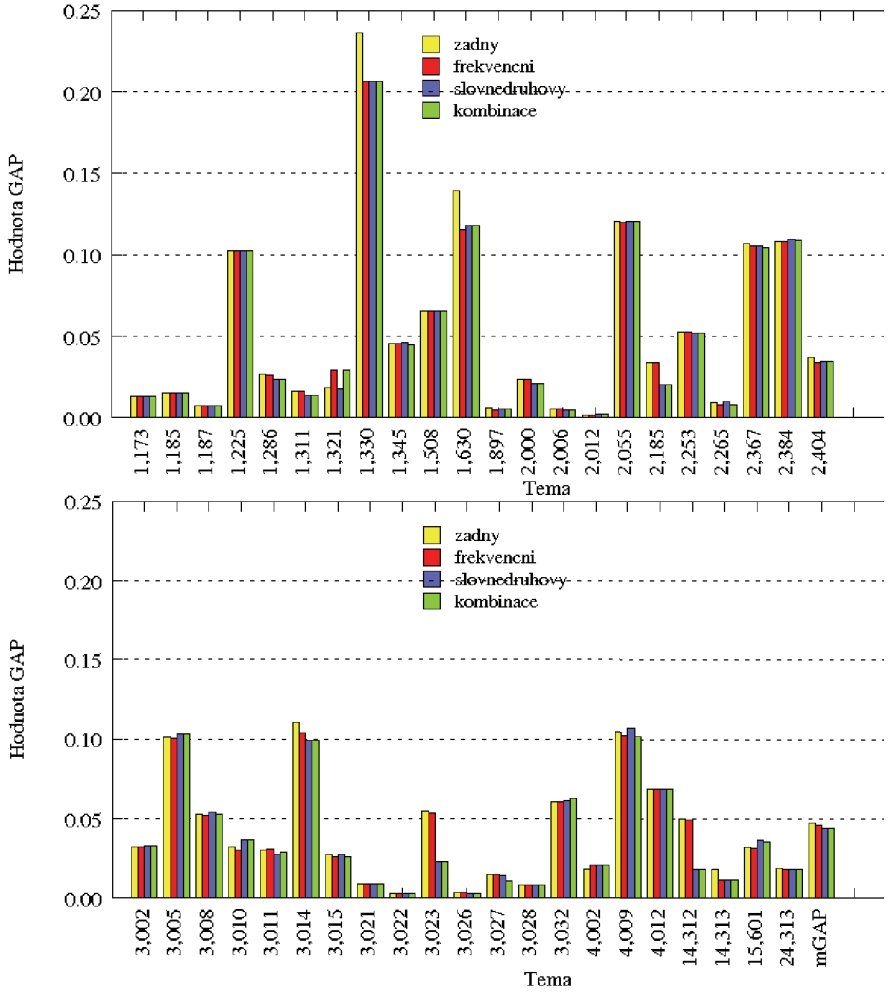
6.9 Experiment s různou délkou generovaných dokumentů

V tomto experimentu budeme zkoumat vliv dvou parametrů (délka dokumentu a přesah dokumentů) na výsledky vyhledávání. Jako první tento vliv zkoumal Ircing v roce 2007 [21]. Zjistil, že dvouminutové dokumenty jsou signifikantně lepší než tříminutové. V našich experimentech se zaměříme také na vliv přesahu, který by pro model počítající frekvence slov (TFI–DF) mohl mít špatný vliv. Pro experimenty použijeme stejný vyhledávací model jako v předchozí části (Raw TF–IDF, lemmata z nejlepšího kanálu z roku ASR systému z roku 2006).

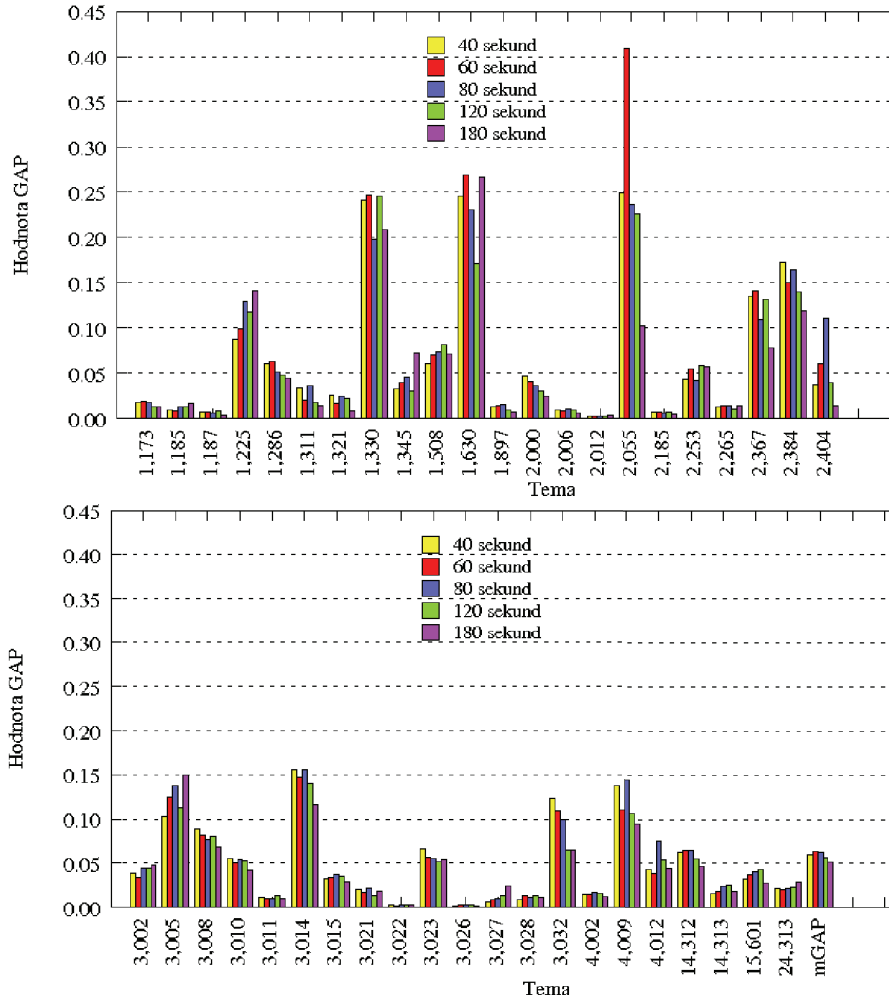
délka / přesah	0%	16%	33%	50%
40 sekund	**0,0604**	0,0529	0,0503	0,0473
60 sekund	**0,0641**	0,0579	0,0561	0,0491
80 sekund	**0,0624**	**0,0649**	**0,0544**	**0,0484**
120 sekund	**0,0559**	0,0618	0,0555	0,0450
180 sekund	**0,0513**	0,0473	0,0475	0,0444

Tabulka 6.8: mGAP skóre pro různé délky dokumentů a různé přesahy.

Výsledky experimentů můžeme vidět v tabulce 6.8 a na obrázcích 6.9 a 6.10. Přesah udává jakou částí délky dokumentu se přesahují dokumenty vedle sebe. Z výsledků je zřejmé, že systémy s kratšími dokumenty dosahují lepších výsledků (u příliš krátkých dokumentů – 40 sekund se výsledky již zhoršují). Pro vysvětlení tohoto jevu jsem provedl podrobnější analýzu délky relevantních pasáží u trénovacích témat. Průměrná hodnota délky relevantní pasáže je sice 160 sekund, ale medián je pouze 107 sekund. Při měření výsledků pomocí míry GAP s nastavením maximální

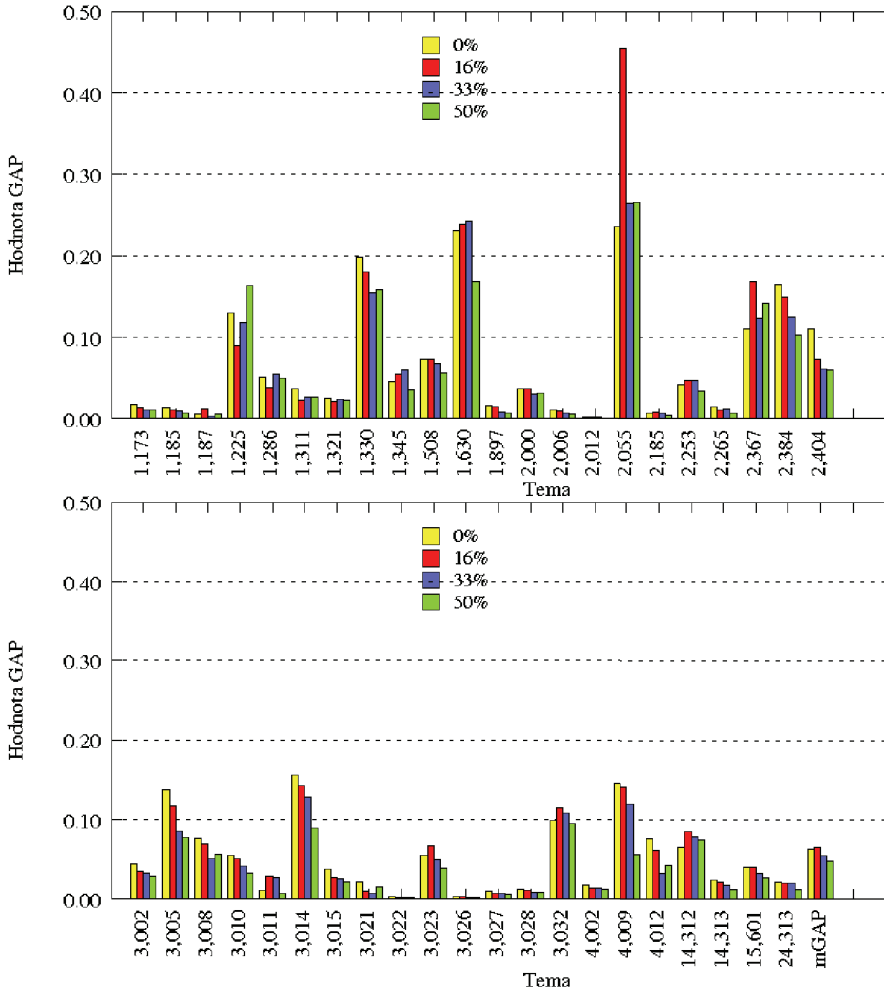


Obrázek 6.8: Srovnání různých druhů stoplistu



Obrázek 6.9: Vliv délky generovaných dokumentů na vyhledávání při pevném přesahu 0%

odchylky pro skórování 150 sekund, nemůžeme efektivně najít dlouhé relevantní pasáže. Pokud bychom se zaměřili pouze na hledání krátkých relevantních pasáží (např. kratších než 180 sekund), hodnota jejich mediánu se sníží na 90 sekund. Domnívám se, že zkrácení délky dokumentu dosahuje lepších výsledků z tohoto důvodu.



Obrázek 6.10: Vliv přesahu generovaných dokumentů na vyhledávání při pevné délce dokumentu 80 sekund

Dále je také zřejmé, že je výhodné generovat dokumenty bez přesahu nebo jen s malým přesahem. Pokud si uvědomíme, že model TF–IDF je založen na počítání frekvencí termů v dokumentech, je zřejmé, že volba přesahu 16% (nebo 33%) naruší vlastnosti kolekce, protože 33% (nebo 66%)[9] termů bude v kolekci dvakrát

[9]Dokumenty se přesahují na svém počátku i konci, proto tato hodnota odpovídá dvojnásobku přesahu.

(odhlédneme-li od počatečního a koncového dokumentu každého rozhovoru, který má přesah pouze z jedné strany). Při volbě přesahu 50% bude každý term v kolekci dvakrát. Špatné výsledky na takto generované kolekci můžeme vysvětlit vlastnostmi míry GAP, která penalizuje systémy vracející dokumenty z malými odstupy – ohodnocen je pouze první dokument poblíž začátku relevantní pasáže, další označené začátky v jeho blízkosti tak snižují přesnost. Při použití přesahu 50% bychom proto pro dobré výsledky museli seznam výstupních dokumentů upravit tak, aby obsahoval pouze dokumenty, které jsou od sebe dostatečně vzdáleny.
Pokud nezávisle srovnáme vliv délky a přesahu, zjistíme, že:

- vyhledávací systém s délkou dokumentů 80 sekund je signifikantně lepší než ostatní systémy,
- vyhledávací systém generující dokumenty bez přesahu je signifikantně lepší než systémy s přesahem dokumentů.

Proto pro závěrečné pokusy použijeme 80 sekundové dokumenty generované bez přesahu.

6.10 Experiment s podrobnějším popisem tématu

V posledním experimentu na trénovacích datech vytvoříme vyhledávací dotazy dvojím způsobem.

Nejdříve standardně použijeme pouze elementů `<title>` a `<desc>` a ve druhém experimentu přidáme i element `<narr>` s podrobnějším popisem. Tento podrobnější popis může obsahovat i podmínku, jaká data se nemají vyhledávat. Viz podrobný popis tématu 2384 (Červený kříž v holocaustu) níže. Poněvadž nedokážeme jednoduše rozlišit, kdy nám přidání obsahu elementu `<narr>` pomůže, negativní vymezení hledaných pasáží zanedbáme a vyhledávací dotaz vždy obohatíme i o termy v podrobnějším popisu.
Příklad podrobného popisu tématu:

```
<narr> Zprávy o tom, jak Červený kříž pomáhal lidem přeživším
holocaust s nalezením příbuzných nebo s vypátráním osudu
příbuzných nás nezajímají. </narr>
```

Parametry pokusů:
Vyhledávací model: Raw TFIDF

ASR systém a kanál: nejlepší kanál z ASR systému z roku 2006
Termy: lemmata
Nevýznamová slova: žádná
Velikost a překryv dokumentů: 60/0; 80/0; 120/0 sekund
Dotazy: `<title>` + `<desc>`; `<title>` + `<desc>` + `<narr>`
Rozšiřování dotazu: žádné

Rozšíření dotazu o element `<narr>` u všech experimentů zvýšilo výslednou hodnotu mGAP. Toto zvýšení je ale statisticky významné až na hladině pravděpodobnosti $p < 0,12$. Výsledné hodnoty obsahuje tabulka 6.9.

dokumenty / dotazy	TD	TDN
60 sekund	0,0641	0,0670
80 sekund	0,0624	0,0676
120 sekund	0,0559	0,0604

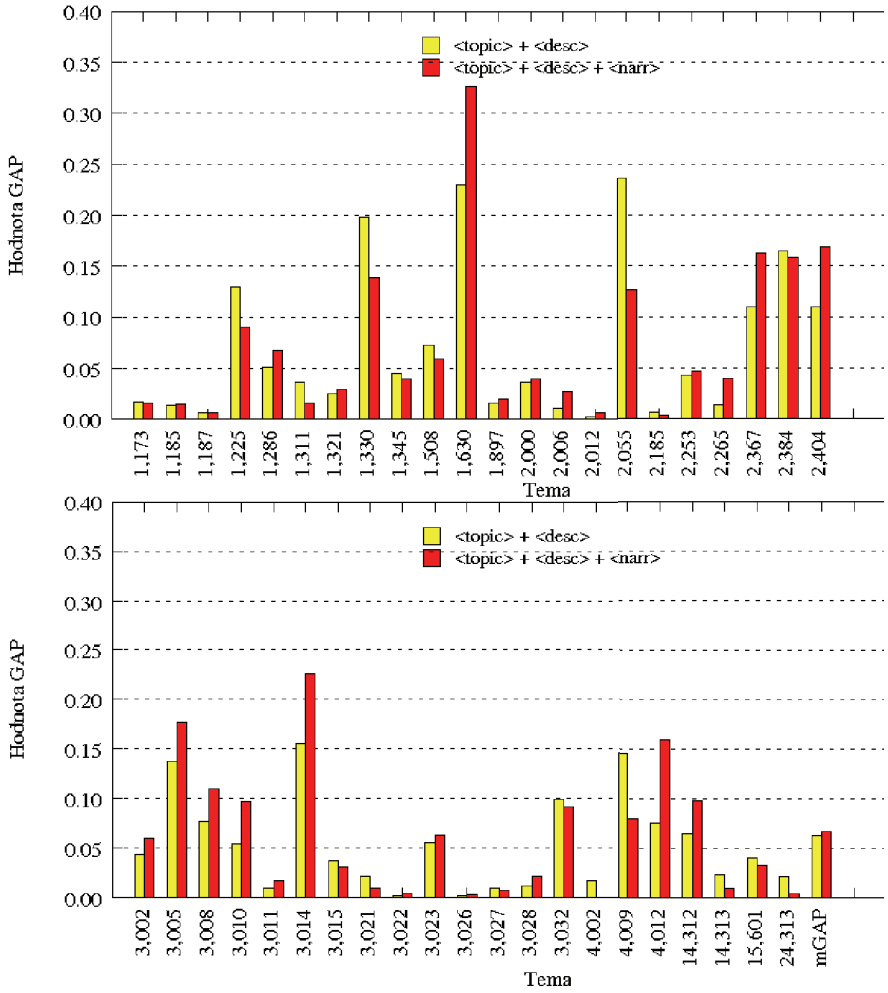
Tabulka 6.9: mGAP skóre pro podrobnější popis tématu

6.11 Závěrečné experimenty na testovacích datech

Pro závěrečné experimenty jsem vybral tři vyhledávací systémy. První z nich je systém, který jsem již použil při základním pokusu uvedém v kapitole 6.3. Ostatní dva systémy mají parametry zvoleny tak, aby dosahovali co nejlepších výsledků na trénovacích datech, tedy tak, jak naznačovaly experimenty popsané v kapitolách 6.4 až 6.9. Liší se pouze tím, které elementy témat využívají pro vyhledávání (viz kapitola 6.10). Podrobný popis všech experimentů obsahuje tabulka 6.10.

Výsledné hodnoty experimentů v tabulce 6.11 a na obrázcích 6.12 a 6.13 ukazují, že experimenty Exp_1 a Exp_2 dosahují podstatně lepších výsledků než základní experiment Exp_0. Tento rozdíl je i statisticky významný. Rozdíl mezi experimenty Exp_1 a Exp_2 je nesignifikantní.

To, že výsledky všech experimentů dopadly hůře na testovacích datech než na trénovacích je obvyklé. I přesto se ale domnívám, že horší výsledek je způsoben i rozdělením témat do trénovací a testovací množiny, které je nerovnoměrné vzhledem k počtu tématů, u kterých nedokáže skórovat žádný vyhledávací systém (jedná se zejména o příliš obecná témata či o témata obsahující klíčová slova, která byla špatně rozpoznána ASR systémy).



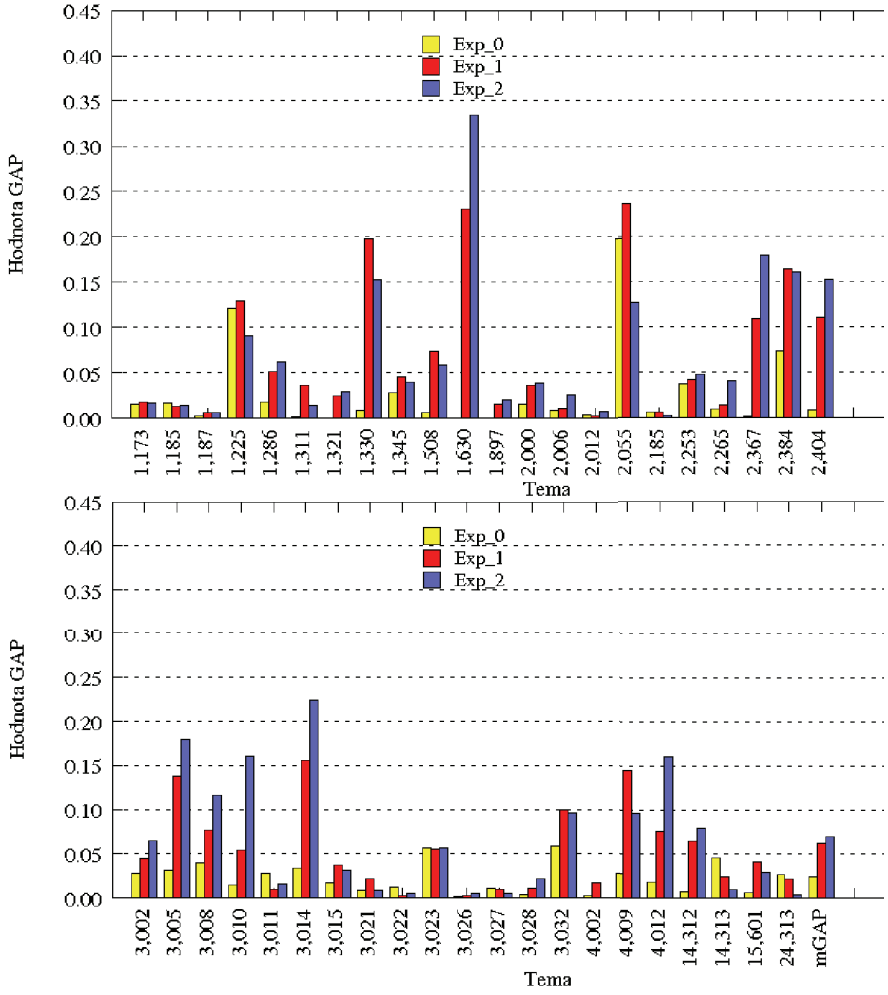
Obrázek 6.11: Srovnání experimentů s použitím polí témat TD a TDN

parametry / experiment	Exp_0	Exp_1	Exp_2
vyhledávací model	Raw TFIDF	Raw TFIDF	Raw TFIDF
ASR systém	nejlepší	2006	2006
kanál	nejlepší	nejlepší	nejlepší
termy	originál	lemmata	lemmata
stoplist	žádný	žádný	žádný
délka dokumentů	180 s	80 s	80 s
překryv dokumentů	33%	0%	0%
dotazy	TD	TD	TDN
rozšiřování dotazů	žádné	BRF(50, 5)	BRF(50, 5)

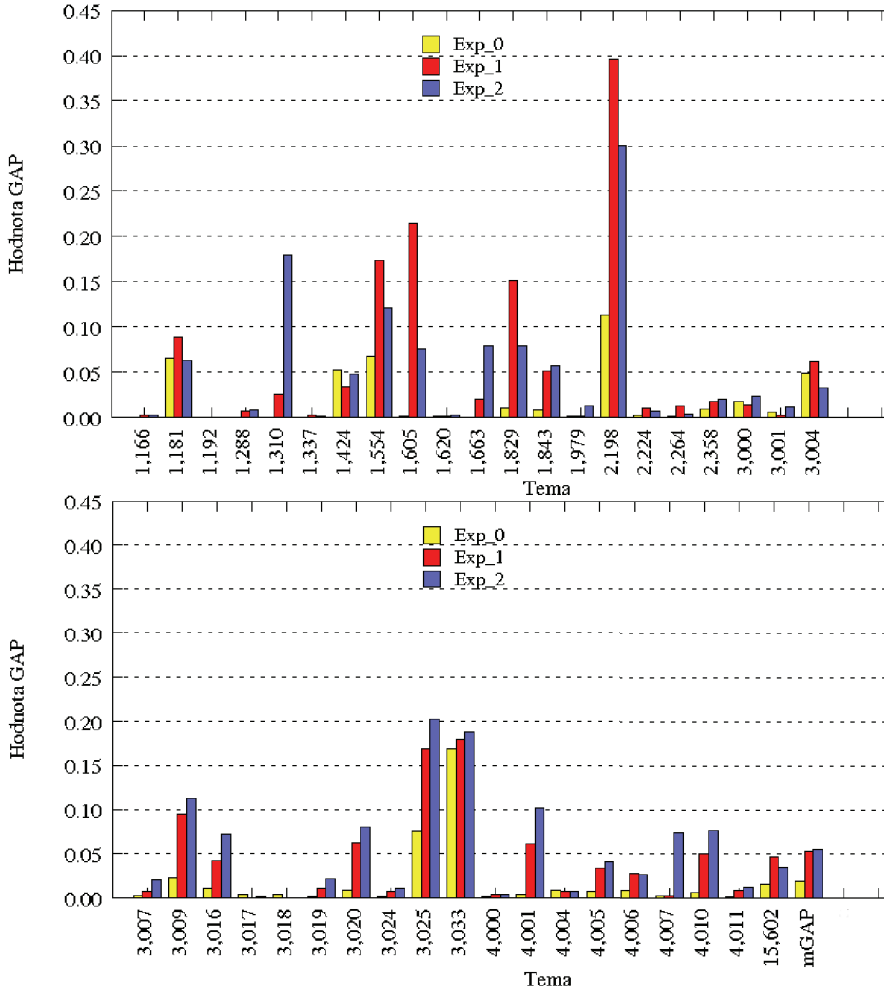
Tabulka 6.10: Parametry systémů pro závěrečné experimenty

	témat	Exp_0	Exp_1	Exp_2
trénovací data	43	0,0247	0,0624	0,0696
testovací data	40	0,0192	0,0527	0,0555

Tabulka 6.11: mGAP skóre pro závěrečné experimenty



Obrázek 6.12: Přehled výsledků závěrečných experimentů na trénovacích datech



Obrázek 6.13: Přehled výsledků závěrečných experimentů na testovacích datech

U trénovacích témat bylo dosáhnuto skóre GAP horšího než 0,02 u 10 témat a z toho měla 4 témata skóre pod 0,01. U testovacích témat bylo skóre GAP horší než 0,02 u 16 témat, z toho 9 témat se skórem pod 0,01.

Kapitola 7

Závěr

Cíl práce. Cílem práce bylo vytvořit nástroje, které by z automatických přepisů nesegmentované mluvené řeči připravily kolekci dat pro účely vyhledávaní informací. Dalším cílem práce bylo pomocí těchto nástrojů vytvořit testovací kolekce dokumentů, provést s nimi vyhledávací experimenty a vyhodnotit je.

Výsledky práce. Pro účely vyhledávání informací v českých rozhovorech z projektu MALACH jsem vytvořil komplexní sadu nástrojů pro generování kolekce dokumentů a vytváření vyhledávacích dotazů, které jsem rozdělil na trénovací a testovací množinu. Zpracoval jsem také relevantní pasáže označené anotátory a připravil skript pro evaluaci výsledků vyhledávacích systémů.

Pomocí těchto nástrojů jsem s trénovacími tématy provedl několik desítek experimentů, pomocí kterých jsem stanovil výhodnost použití různých vyhledávacích technik a nastavil hodnoty parametrů pro testovací experimenty. Pro vyhledávání informací na automatických přepisech nesegmentované mluvené řeči je výhodné zejména použití lemmatizace, použití ASR systému upravujícího hovorová slova na jejich spisovnou podobu a je také výhodné pečlivě zvolit délku generovaných dokumentů v závislosti na mediánu délky relevantních pasáží. Není výhodné generovat dokumenty s přesahem a odstraňovat z nich nevýznamová slova.

Pro ověření svých zjištění jsem provedl tři experimenty na testovacích datech. Základní experiment na 40 testovacích tématech dosáhl mGAP skóre 0,0192. Můj závěrečný experiment používající texty témat z polí `<title>` a `<desc>` dosáhl mGAP skóre 0,0527. Při použití plných textů témat jsem dosáhl mGAP skóre 0,0555. Tato práce se jako jedna z prvních zabývala problémem vyhledávání v nesegmentované mluvené řeči. Žádná práce se srovnatelnými výsledky nebyla dříve představena.

Diskuse výsledků. Představené přeformulování problému vyhledávání v nesegmen-

tované mluvené řeči na vyhledávání v textových dokumentech se zdá být úspěšnou cestou. Dosažené výsledky vyhledávání jsou mnohonásobně horší než úspěšnost vyhledávání v textových datech, ale to je zapříčiněno zejména relativně špatnou úspěšností ASR systémů a také stářím poslouchaných svědků holocaustu. Závěry vyvozené v této práci by měly být ověřeny na další kolekci nesegmentované mluvené řeči, aby byla ověřena jejich platnost. Trénovací a testovací témata tvoří sice disjunktní množiny, ale byla anotována na stejných datech stejnými anotátory. Výsledky jsou také zkresleny velkými rozdíly ve vhodnosti témat pro vyhledávání, u kterých kolísá skóre GAP u nejlepšího systému od hodnoty 0,0002 až k hodnotě 0,3960.

Práce do budoucna. Jednou z možných cest pro další zlepšení výsledků by bylo větší propojení systému pro automatické rozpoznání řeči s vyhledávacím systémem. Ať již použitím více možností rozpoznaných výstupních vět z ASR systému, nebo jeho upravením, aby lépe rozpoznával méně častá slova, která jsou pro úspěch vyhledávacích systému důležitá.

Další možností je také vytvořit novou míru pro ohodnocení výsledků, která by lépe řešila problém s velkým rozdílem v délce relevantních pasáží.

Literatura

[1] Baeza-Yates R., Ribeiro-Neto B.: *Modern Information Retrieval*, ACM Press, New York, 1999.

[2] Pokorný J., Snášel V., Húsek D.: *Dokumentografické informační systémy.* Skripta UK, Praha 1999.

[3] Salton G., Fox E. A., Wu H.: *Extended Boolean Information Retrieval*, Cornell University, Ithaca, 1982.

[4] Zhai C.: *Notes on the Lemur TFIDF model*, School of Computer Science, CMU, Pittsburgh, 2001.

[5] Robertson S. E., Walker S., Jones S., Hancock-Beaulieu M. M., Gatford M.: *Okapi at TREC-3*, In: The Third Text REtrieval Conference (TREC-3), NIST Special Publication, (1995) 109 –127.

[6] Robertson S. E., Walker S.: *Okapi/Keenbow at TREC-8*, In: The Eight Text REtrieval Conference (TREC-8), NIST Special Publication, (2000) 151–161.

[7] Strohman T., Metzler D., Turtle H., Croft W. B.: *INDRI: A Language-model Based Search Engine for Complex Queries (Extended Version).* Technical report IR-407, CIIR, UMass (2005).

[8] Metzler D.: *Indri Retrieval Model Overview*, updated 2005-07-16, [cit. 2008-07-31]. Dostupný z www: <http://ciir.cs.umass.edu/ metzler/indriretmodel.html>.

[9] Turtle H. R.: *Inference networks for document retrieval.* Doktorská práce, University of Massachusetts, Amherst, 1990.

[10] Porter M.: *An algorithm for suffix stripping*, Program, 14(3), (1980) 130–137.

[11] Psutka J., Ircing P., Hajič J., Radová V., Psutka J. V., Byrne W. J.: *Issues in annotation of the Czech spontaneous speech corpus in the MALACH project*, 2004.

[12] Oard D., Wang J., Jones G., White R., Pecina P., Soergel D., Huang X., Shafran I.: *Overview of the CLEF-2006 Cross-Language Speech Retrieval Track*. In Peters C., Clough P., Gey F., Karlgren J., Magnini B., Oard D., de Rijke M., Stempfhuber M., eds.: Evaluation of Multilingual and Multi-modal Information Retrieval – 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. Lecture Notes in Computer Science, Springer, Berlín (2007).

[13] Pecina P., Hoffmannová P., Jones G. J. F., Zhang Y., Oard D. W.: *Overview of the CLEF-2007 Cross-Language Speech Retrieval Track* In CLEF 2007. C. Peters et al., eds., Lecture Notes in Computer Science, LNCS 5152, Springer, Berlín (2008).

[14] Liu B., Oard D. W.: *One-sided measures for evaluating ranked retrieval efectiveness with spontaneous conversational speech*. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (2006) 673–674.

[15] Wilcoxon F.: *Individual comparison by Ranking Methods*, Biometrics Bulletin, Vol. 1, No. 6. (1945), 80–83.

[16] Hajič J., Vidová-Hladká B.: *Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset*. In Proceedings of thr Conference COLING – ACL 1998, Montreal (1998).

[17] Hajič J.: *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Nakladatelství Karolinum, Praha (2004).

[18] Češka P.: *Segmentace textu*, Bakalářská práce, Matematicko-fyzikální fakulta, Praha (2006).

[19] Češka P., Pecina P.: *Charles University at CLEF 2007 CL-SR Track*, 2008.

[20] Ircing P., Muller L.: *Benefit of Proper Language Processing for Czech Speech Retrieval in the CL-SR Task at CLEF 2006*. In Peters C., Clough P., Gey F., Karlgren J., Magnini B., Oard D., de Rijke M., Stempfhuber M., eds.: Evaluation of Multilingual and Multi-modal Information Retrieval – 7th Workshop of the

Cross-Language Evaluation Forum, CLEF 2006. Lecture Notes in Computer Science, Springer, Berlín (2007).

[21] Ircing P., Psutka J., Vavruška J.: *What Can and Cannot Be Found in Czech Spontaneous Speech Using Document-Oriented IR Methods – UWB at CLEF 2007 CL-SR Track*, 2008.

Příloha A

Obsah přiloženého CD

K této práci je přiloženo CD, které obsahuje následující soubory a složky:

Dummy_Collection/ – složka obsahuje ukázkovou kolekci dat, kvůli právním důvodům mohu k práci přiložit pouze velmi malý výřez originální kolekce

- Assessments/
  - Example_Result – ukázkový soubor s výstupem experimentu Exp_1 pro téma 1185
  - Train_Assessments.data – seznam relevantních pasáží pro téma 1185
  - mGAP.pl – skript pro výpočet GAP skóre
- Documents/
  - ASR/ – složka s ukázkovým výstupem ASR systémů
  - Interview.data – informace o poslouchaných svědcích holocaustu
  - Keywords.data – soubor s obsahující klíčová slova automaticky přiřazená rozhovorům
  - QualityLetters.data – soubor s tabulkou kvalit dokumentů podle počtu písmen
  - QualityWords.data – soubor s tabulkou kvalit dokumentů podle počtu slov
  - TapesLengths.data – soubor s počátečními časy pásek
  - CreateCollection.pl – skript pro tvorbu kolekcí dokumentů

  - Params.xml – ukázkový parametrový soubor
- Topics/
  - Topics.data – ukázka lemmatizovaného tématu 1185
- README.txt – nápověda k používání této kolekce