Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň, 2015 **Robin Popelka**

P R O H L Á Š E N Í

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni, dne ……

Anotace

Tato práce se zabývá dnešními možnostmi automatického rozpoznávání na mobilních zařízeních. Cílem je zanalyzovat možné přístupy pro rozpoznávání na mobilních zařízeních a vytvořit webovou aplikaci v HTML 5 a aplikaci pro Android. Dále jsou aplikace otestovány pomocí připravených testovacích nahrávek. Nahrávky jsou pořízeny jednou v nazarušeném domácím prostředí a podruhé ve venkovním zarušeném prostředí. Nahrávky jsou namluveny celkem od třech lidí. V závěru práce se zabýváme porovnáním výsledků jednotlivých přístupů pomocí programu HTK HResults.

Klíčová slova

Google ASR, HTML 5, automatické rozpoznávání řeči, Android, mobilní zařízení, Web Speech API

Abstract

This thesis deals with possibilities of today's automatic speech recognition on mobile devices. The aim is to analyze the possible approaches for recognition on mobile devices and create web application in HTML 5 and application for Android. The recordings are recorded once in a quiet home environment and for the second time in a noisy outdoor environment. Three people had their voices recorded for the purpose of the project. This project, in summation, compares the results of different approaches using HTK HResults.

Keywords

Google ASR, HTML 5, automatic speech recognition, Android, mobile device, Web Speech API

Obsah

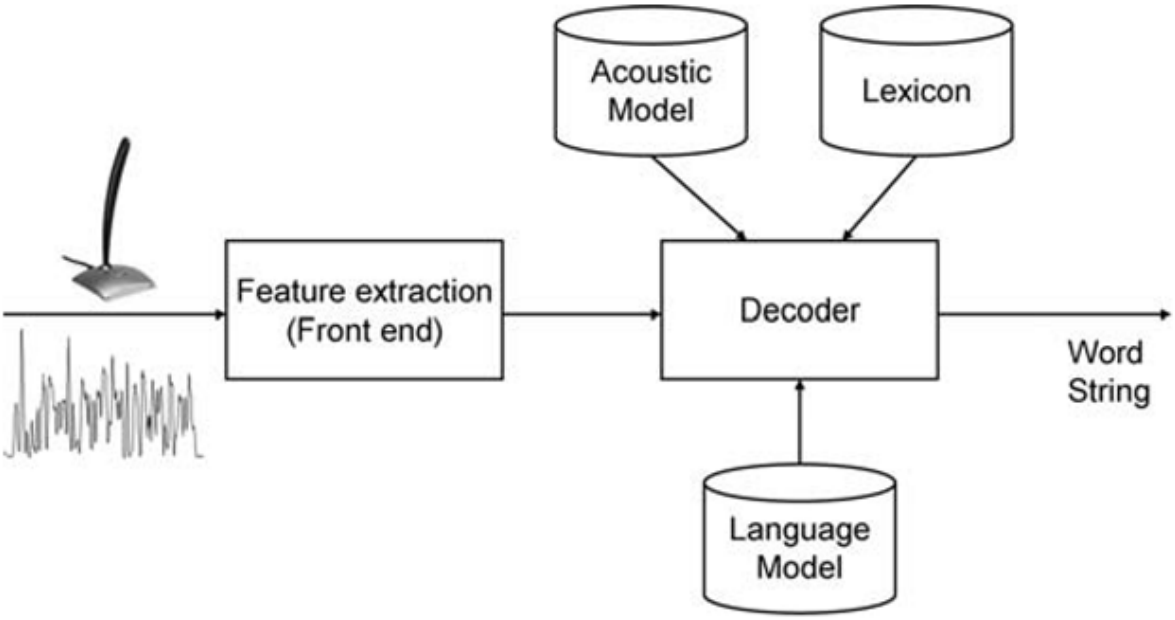
1. Úvod

Tato práce se zabývá problematikou automatického rozpoznávání řeči konkrétně na mobilních zařízeních. Jsou prozkoumány aktuální možné přístupy v této oblasti. Má práce bude pojednávat zejména o možnostech automatického rozpoznávání pomocí Google nástrojů. Rozpoznávání pomocí Google nástrojů je vybráno z důvodu, že sám Google je velkým propagátorem této problematiky a na dnešní poměry vykazuje kvalitní výsledky rozpoznávání. Neméně důležitým faktem je, že Google umožňuje veřejnosti jeho nástroje využívat. Vybrané přístupy jsou mezi sebou otestovány. Dále jsou vytvořeny dvě aplikace pro mobilní zařízení, kdy v prvním případě se jedná o aplikaci určenou pro zařízení Android a ve druhém případě se jedná o HTML 5 aplikaci.

Pro testování rozpoznávání jsou připraveny sady textových příkladů různého zaměření. Jedním z cílů práce je otestovat rozpoznávače na širokém spektru příkladů, aby vypovídající hodnota výsledků rozpoznávání byla co nejvyšší. Textové podklady můžeme rozdělit na obecné věty a věty specifického charakteru jako jsou například názvy ulic v Plzni. Nahrávky pro testování byly zaznamenány celkem dvěma ženskými hlasy a jedním mužským. Protože mobilní telefony či jiná zařízení nosíme běžně pořád u sebe, tak byly nahrávky zaznamenány celkem dvakrát. Jednou v domácím ničím nerušeném prostředí a podruhé v zarušeném venkovním prostředí. Vyhodnocení výsledků rozpoznávání je provedeno v programu HTK HResults, kde bylo nutné nahrávky nejprve anotovat. Dále jsou pořízené nahrávky otestovány jednou pro Google rozpoznávač a podruhé jsou otestovány v rozpoznávači ZČU LVSCR. Jednotlivé výsledky rozpoznávání a přístupy jsou celkově zhodnoceny v samotném závěru práce.

2. Automatické rozpoznávání řeči

Automatické rozpoznávání řeči (ASR – Automatic Speech Recognition) je systém, který umožňuje spojitě rozpoznávat mluvená slova či celé věty a poté zapsat výsledek do textové podoby. ASR pracuje se statistickými metodami (HMM – Skryté Markovovy modely). Rozpoznávání slov je možné jen takových, které systém již zná. Cílem automatického rozpoznávání řeči je 100% přesnost v rozpoznávání, nezávislost na řečníkovi a použitých slovech, použitém akcentu a hlasitostí či okolním šumem. V dnešní době v případě, že je systém natrénován na konkrétního řečníka a je použit velký slovník, je přesnost velmi vysoká.



Obr. 1: Schéma automatického rozpoznávání řeči [1].

2.1. Úlohy automatického zpracování řečového signálu

2.1.1. Dialogový systém

Tento systém funguje na principu komunikace s uživatelem přirozeným jazykem pomocí dialogu. Uplatnění nachází v telefonních automatech, které jsou propojeny s databázovým serverem. Tyto systémy používají TTS (text to speech) a ASR (Automatic Speech Recognition) [9].

2.1.2. Verifikace řečníka

Tento systém má za úkol rozpoznat daného řečníka a podle toho se rozhodnout o přijetí nebo odmítnutí dané osoby. Tento systém nachází uplatnění v bankovním sektoru nebo v síťových systémech. Systém řeší problém přiřazení správné osoby. Snaží se najít pokud možno co nejpřesnější shodu s daným vzorkem [9].

2.1.3. Rozpoznávání jazyka

Systém se snaží rozpoznat daný jazyk buď pomocí fonémického/fonetického významu, nebo uplatní všechny rozpoznávače na danou řeč najednou a podle toho, který jazyk dosáhnul nejvyššího počtu rozpoznaných slov pro daný jazyk a rozhodne o tom, jaký byl použit jazyk [9].

2.1.4. Rozpoznávání izolovaných slov

Tyto systémy jsou vytvořeny pro rozpoznávání pouze izolovaných slov a s tím souvisí i vyšší úspešnosti rozpoznávání. Takovéto rozpoznávače najdeme například v automobilovém průmyslu, kdy řidič může jednoduchými pokyny ovládat např. navigaci či handsfree a může se tak věnovat bezpečně řízení [9].

2.1.5. Rozpoznávání spojité řeči

Zatím nejsložitější způsob rozpoznávání řeči je rozpoznávání kontinuální rozpoznávání, kterým se v této práci budeme nejvíce zabývat. V praxi toto rozpoznávání může být uplatněno např. titulkování „živých“ pořadů nebo u spojitého diktátu [3]. Schopností těchto systému je např. nezávislost na řečníkovi a robustnost rozpoznávání v případě rušivých elementů jako je šum. Společnost Google je v posledních letech významným průkopníkem v této oblasti a nabízí pro vývojáře rozhraní pro testování rozpoznávače.

2.2. Závislost na řečníkovi

Systémy automatického rozpoznávání můžeme rozdělit do dvou základních skupin. První skupinou jsou systémy, které jsou takzvaně závislé na řečníkovi. Tyto systémy analyzují specifický hlas konkrétního řečníka a tyto poznatky poté využívá při rozpoznávání mluvené řeči. Výsledky těchto systémů jsou zpravidla velmi přesné, ale je nutné dostatečně, zpravidla použitím několika hodin nahrávek, natrénovat hlas konkrétního řečníka. Druhou skupinou jsou systémy nezávislé na řečníkovi. Takovéto systémy mají modely natrénované od velkého počtu řečníků. Nedosahují tak dobrých výsledků jako systémy závislé na řečníkovi. Existují tedy metody, které se dokážou přizpůsobit na jednotlivého řečníka [4].

2.3. Metody rozpoznávání

2.3.1. Metoda DTW

Metoda DTW (dynamické borcení času) se využívá při porovnávání krátké promluvy, nebo izolovaných slov. Je nutné mít databázi vzorů pro danou promluvu a s touto probíhá porovnávání. Tato metoda se nejvíce hodí pro rozpoznávání izolovaných slov. Problém při rozpoznávání slova je při této metodě u stejného řečníka takový, že se neliší ve frekvenční

oblasti, nýbrž v časování a metoda se tak snaží zkracovat či prodlužovat jednotlivé úseky signálu tak, aby byl rozdíl v porovnávání signálů pokud možno co nejmenší [4].

2.3.2. Statistický přístup

Nejčastěji momentálně používanou metodou pro rozpoznávání řeči jsou tzv. HMM (skryté Markovské modely), které obsahují skryté stavy. Tato metoda vychází ze statistického přístupu s konečným počtem stavů. Signál je charakterizován jako soubor vektorů s příznaky. Řečový signál je možné rozumět jako po částech stacionární signál. Tato metoda se nejvíce využívá při rozpoznávání kontinuální řeči. Při rozpoznávání izolovaných slov se řečník více soustředí na výslovnost a nedochází např. ke spodobě slov [4].

2.4. Vyhodnocení systémů ASR

Pro vyhodnocení výsledků rozpoznávání byl použit program HTK Speech Recognition Toolkit. Tento program je licencován Cambridgeskou univerzitou a je volně ke stažení na internetu [5]. Konkrétně se jedná o balík menších programů pro rozpoznávání řeči a zpracování výsledků. Pro naše účely tak byl použit program HTK HResults. Při vypočítání celkového procenta správně rozpoznaných slov program používá následující vzorce:

$$Percent\ Correct\ (procento\ správnosti) = \frac{N - D - S}{N} * 100\ \%$$

$$Percent\ Accuracy\ (procento\ přesnosti) = \frac{N - D - S - I}{N} * 100\ \%$$

N – značí celkový počet slov v referenčním dokumentu

D – vyjadřuje chybu celkového počtu slov, které nejsou vůbec rozpoznány

S – vyjadřuje chybu celkového počtu substitucí, tedy slov lišících se oproti slovům v referenčním dokumentu

I – vyjadřuje celkový počet slov, které byly přidány navíc oproti referenčnímu dokumentu

Za povšimnutí stojí fakt, že první vzorec vůbec neuvažuje počet slov, které byly přidány v rozpoznaném textu navíc. Toto je vyřešeno ve druhém uvedeném vzorci, který vyjadřuje procento přesnosti v rozpoznávání. Výsledky druhého vzorce jsou tak celkově relevantnější a vypovídají více informacemi o schopnostech rozpoznávače. Na následujících řádcích si ukážeme příklad, jak program vyhodnocuje výsledky rozpoznávání.

Příklad vyhodnocení věty:

Ref.	ŠEDESÁTÁ	LÉTA		PŘEDSTAVENÍ	ANDY	WARHOL	V	PRAZE
Rec.	ŠEDESÁTÁ	LÉTA	V	PŘEDSTAVENÍ	ANDYHO	WARHOLA		PRÁCE

První řádek tabulky představuje referenční tedy správný text, se kterým je rozpoznaný text na druhé řádce porovnáván. Žlutě je označen počet substitucí (S), modře je označen počet insercí (I) a zeleně počet smazaných slov (D). Po dosazení do výše uvedených vzorců nám vyjdou následující výsledky:

$$Percent\ Correct\ (procento\ správnosti) = \frac{7-3-1}{7} * 100\ = 42.86\%$$

$$Percent\ Accuracy\ (procento\ přesnosti) = \frac{7-3-1-1}{7} * 100 = 28.57\ \%$$

3. Řečový signál

Řeč člověka vzniká ve hlasovém ústrojí a používá se při komunikaci mezi lidmi. Toto ústrojí můžeme dělit na hlasovou, dechovou a artikulační část. Dechová část je spojená s dechem z plic, kdy vzduch jde při výdechu z plic přes hlasové ústrojí, kde rozechvěje hlasové orgány a poté jde přes rty směrem ven a tím vzniká řeč [6].

Hlasovou část tvoří hlasový orgán zvaný hlasivky. Ty jsou uložené v hrtanu. Mezi hlasivkami se nachází prostor, který se nazývá hlasová štěrbina. Právě tato štěrbina se různě roztahuje při mluvení, případně se maximálně otevře při pouhém dýchání [6].

Když v hlasovém ústrojí tedy v hrtanu vzniká zvuk, tak tomuto procesu říkáme fonace. Při tomto procesu se díky vzduchu vydechovaného z plic hlasová štěrbina zužována a dochází tak k rozkmitání hlasivek. S artikulační částí se pojí několik orgánů, jako jsou nosní dutina, ústní dutina, jazyk, dásně rty či zuby. Všechny tyto součásti spolu vytváří různé kombinace řečových signálů [6].

Samotná řeč je složená z jednotlivých hlásek. Souvislá řeč se dále skládá i z tzv. vyplňujících elementů, které nepředstavují žádné slova a je nutné je rozlišovat jako nežádoucí prvky zvukového signálu. Jsou jimi například zvuky vzniklé při dýchání, kašlání nebo i třeba mlaskání atp. [6].

3.1. Reprezentace řečového signálu v PC

Řečový signál je v PC reprezentován souborem digitálních dat. Zvuk je pomocí mikrofonu zaznamenáván do analogového signálu a ten je dále zpracován do digitální podoby pomocí vzorkování. V daný časový interval se zaznamenává aktuální vzorek. Čím je vzorkovací frekvence vyšší, tím je výsledný soubor kvalitnější. Dalším kritériem na kvalitu výsledného souboru určuje počet použitých úrovní ve vzorku. Běžné jsou následující vzorkovací frekvence: 8 kHz a 11 kHz (telefonní kvalita), 22 kHz (rádio kvalita), 44 kHz (CD kvalita), 48 kHz a 96 kHz. Počet bitů na vzorek je většinou 8, 16 nebo 24 [7].

3.2. Historie rozpoznávání a porozumění řeči

Již v padesátých letech minulého století měly tehdejší počítače schopnost rozpoznávat mluvenou řeč. Jednalo se však pouze o izolovaná slova a rozpoznávání probíhalo pomocí primitivního porovnávání se souborem nahrávek metodou DTW (dynamické borcení času). Tento systém fungoval dobře pouze s jedním řečníkem. V případě jiného řečníka úspěšnost správného rozpoznávání velice klesla. S postupem času se zvyšováním výkonu a kapacity

počítačů bylo možné vyvinout mnohem výkonnější systémy pro rozpoznávání, které nemusejí být závislé na řečníkovi, ale dokonce mají schopnost se na daného řečníka adaptovat. Dalším velkým pokrokem je schopnost rozpoznávat nejen izolovaná slova, ale také souvislou spontánní řeč [8].

Z výše uvedeného by se dalo usoudit, že jsou již dnes počítače schopné dokonale rozpoznávat mluvenou řeč, kde nezáleží, kdo je řečníkem a kdy nezáleží na pořadí použitých slov z daného slovníku, ale problém rozpoznávání je velmi složitý proces a je s ním spojeno hodně faktorů, které výrazně ovlivňují správné výsledky rozpoznávání. Problémy jsou spojené se schopností rozpoznávat spojitou řeč a se samotným správným porozuměním sémantiky dané promluvy.

3.3. Problémy spojené s rozpoznáváním řeči

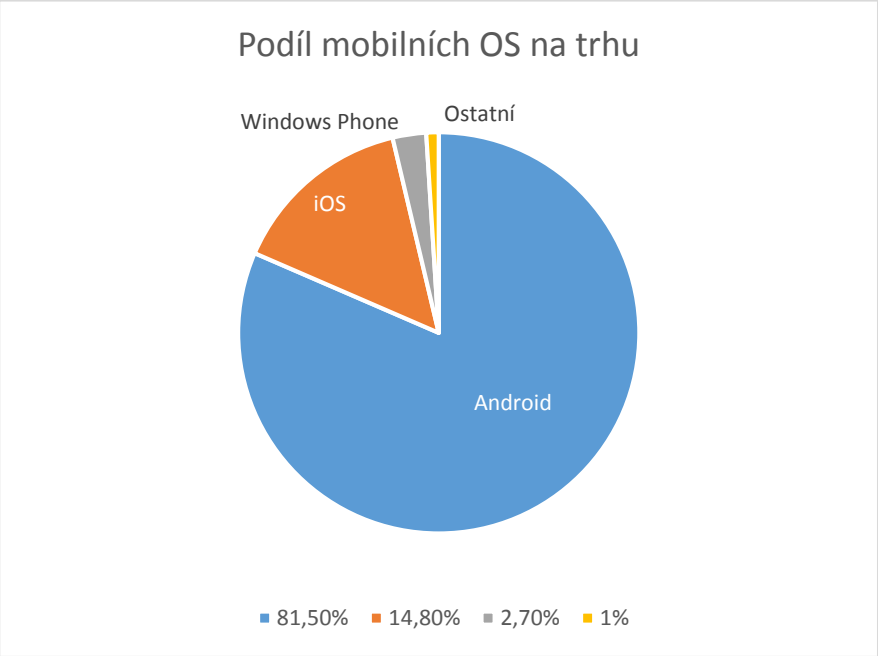
1) Každý hlas jednotlivce je unikátní a je určen konečnou stavbou hlasového ústrojí.
2) Hlas samotného jednotlivce je ve své podstatě také vždy jiný. Prakticky nelze zcela 100% vyslovit dvakrát naprosto shodně nějakou promluvu, ať už se jedná o rychlost promluvy, hlasitost, nebo její intonace.
3) V souvislé větě je těžké určit začátek a konec jednotlivých slov. Problém je v tzv. spodobě slov, kdy jsou slova ovlivněna poslední hláskou slova hláskou následujícího slova.
4) Schopnost rozpoznávání je také silně ovlivněna velikostí okolního šumu a ruchu.
5) V neposlední řadě výsledky ovlivňuje také samotná nahrávací aparatura, tedy mikrofon a jeho umístění [9].

4. Přístupy ASR na mobilních zařízeních

V dnešní době velkého rozvoje mobilních technologií, kdy podíl mobilních zařízení převažuje nad stolními počítači, je důležité se zaměřit na vývoj softwaru právě pro mobilní zařízení, protože tento trend se zdá půjde nadále tímto směrem, nabývá tak na důležitosti.

4.1. OS na mobilních zařízeních

Existuje několik operačních systému pro mobilní zařízení. Nejvíce rozšířeným operačním systémem na mobilní zařízení je bezesporu Android a to s většinovým zastoupením na trhu. Další systémy, které se významně podílejí a rozdělují si tak zbylou část zastoupení jsou iOS, Windows Phone a další. V této práci se budeme zabývat ASR pro operační systém Android, která jak již bylo řečeno se zdá jako nejdůležitější platforma pro vývoj aplikací [10].



Obr. 2: Podíl jednotlivých OS na mobilních zařízeních [10].

4.2. HTML 5

Dalším přístupem neméně důležitým je přístup pomocí webového prostředí. HTML je značkovací jazyk, který byl vyvinut jako standard pro webové stránky. S tímto jazykem je spojeno několik dalších technologií, bez kterých by nebylo dnes možné tvořit kvalitní interaktivní webové stránky. Jako nástroje, které tvoří webové obsah, můžeme uvést např. CSS kaskádové styly, které pomáhají vytvářet výsledný vzhled stránek, dalším velmi důležitým nástrojem je programovací jazyk Javascript, díky kterému můžeme tvořit složité funkce a pracovat s daty na straně klienta. Pro vývoj webových stránek se používá mnoho dalších

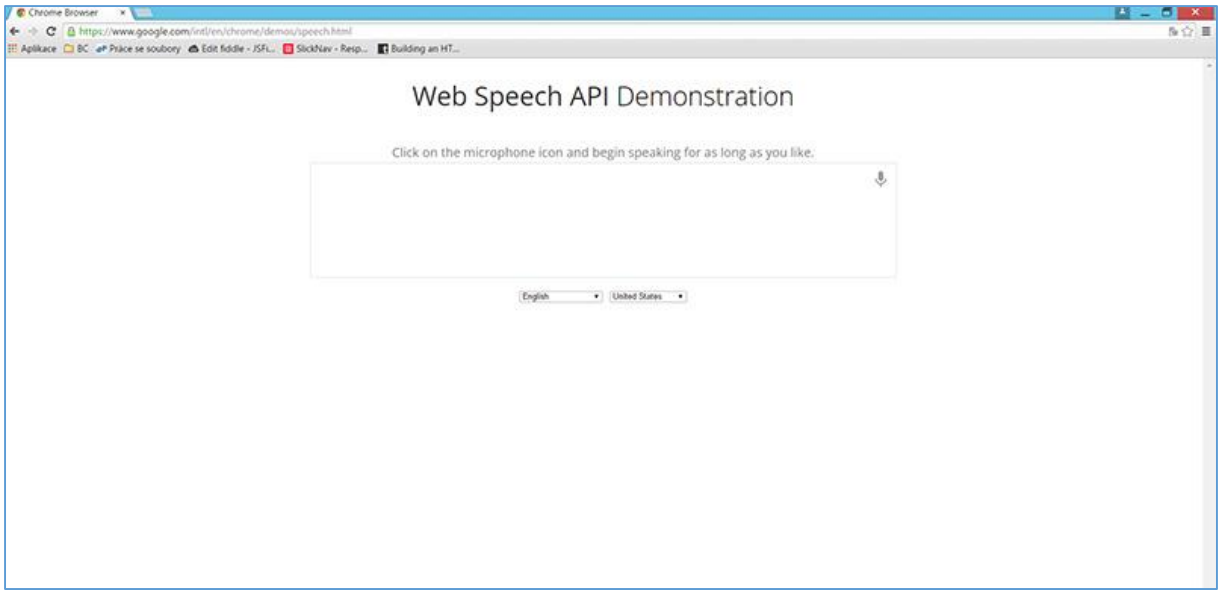
programovacích jazyků a nástrojů např. (PHP, MySQL, JQuery knihovna …). Aktuální verzí HTML je verze HTML 5. Tento standard rozšiřuje původní specifikaci HTML o nové prvky především pro snadnější práci s médii a s tím spojený přístup k nim bez použití externích nástrojů jako je Flash. Velká výhoda HTML 5 je postupná standardizace nástrojů pro práci s hardwarem nějakého ať už mobilního či běžného zařízení bez nutnosti instalovat externí aplikace pro jeho využívání. Nově je možnost využívat mikrofon, webovou kameru a jiné periférie zařízení.

4.3. Výhody HTML 5

Nejdůležitějším důvodem proč zvolit přístup pomocí webového rozhraní HTML 5 je schopnost mobilních zařízení, ale i desktopu fungovat na různých platformách. Nezáleží tedy na tom, jestli mobilní zařízení jede pod Androidem, iOS nebo Blackberry. Toto je nesporná výhoda oproti jinému řešení, které je šité na míru pro jednu platformu. Zajisté je na místě zmínit, že se trend webových technologií bude neustále vyvíjet kupředu a je tedy určitě dobré věnovat mu dostatečnou pozornost. Další nespornou výhodou nové HTML 5 specifikace je možnost běhu webové aplikace v off-line režimu. Není tak nutné být neustále připojen k internetu, ale v případě použití nějakého vyhodnocení přes vzdálený server je stejně nutné být online a to platí obecně i pro aplikace vyvíjené pro Android.

4.4. Google ASR

Firma Google vytvořila nové rozhraní Web Speech API a je tak možné ve webovém prostředí automaticky rozpoznávat řeč. Podpora různých jazyků je rozmanitá a nechybí pro nás důležitý český jazyk. Web Speech API podporuje dosud pouze webový prohlížč Chrome od verze 25 a výše (desktop i mobilní prohlížeč). Nutno podoktnout , že v mobilní verzi prohlížeče Chrome je možné toto API využít u mobilních zařízeních, které běží na Android platformě. Otestoval jsem i prohlížeč Chrome pro iOS od Applu, ale dosud nejsou tato zařízení podporována. Toto API rozhraní je specifikováno dle standardu Web Speech API Specification, které navrhnul a vytvořil sám Google. Vzhledem k již řečenému faktu, že toto Web Speech API funguje zatím pouze v prohlížeči Chrome, tak je zatím rozpoznávání ve webovém prostředí v praxi reálně nepoužitelné, ale pro naše účely toto omezení není nijak zvlášť důležité. Otázku, proč bylo zvoleno právě Google ASR, můžeme odpovědět poměrně jednoduše. V současné době se jedná o technicky vyspělé řešení, kdy Google má před konkurencí díky investování do této problematiky napřed a je přístupné veřejnosti pro testování [11].



Obr. 2: Ukázka rozhraní Web Speech API ve webovém prohlížeči

Existují dvě možnosti jak pracovat ve webovém prostředí s rozpoznávačem od Googlu. První možností je využít Web Speech API zmíněné výše a druhou možností je Google Speech API Full Duplex V1, kdy je nejprve nutné vygenerovat unikátní klíč API key a poté je možné poslat zvukový záznam ve formátu FLAC na vzdálené servery Googlu. Zpět po rozpoznávání obdržíme soubor ve formátu JSON s výsledkem v textové podobě. Tento způsob rozpoznávání má jisté omezení, které spočívá v malém počtu rozpoznaných výsledků, konkrétně je možné odeslat na servery Googlu 50 dotazů (zvukových záznamů)/den. Toto malé číslo je velmi omezující a opět těžko použitelné v praxi. Toto API od Googlu nemá žádnou oficiální dokumentaci a je tak nutné hledat po diskusních fórech.

5. Volba API pro webové rozhraní

Jedním z hlavních cílů této práce je schopnost nejprve nahrát zvukový záznam a ten uložit do zařízení a až poté tento soubor odeslat na server Googlu, aby bylo možné použít zvukový záznam pro další účely rozpoznávání. V první zmíněné možnosti přístupu (Web Speech API) není možné nejprve nahrát zvukový záznam, ale nahrávání probíhá bezprostředně po stisku tlačítka mikrofonu pro započetí rozpoznávání. V druhém přístupu pomocí Google Speech API V1 je možné respektive nutné nejprve pořídit zvukový záznam a ten odeslat na servery Googlu, ale je nutné záznam pořídit ve formátu FLAC, který není možné pomocí běžně dostupných nástrojů zkonvertovat z WAV formátu. Je to možné pomocí externího nástroje, který umí pracovat se zvukovými nahrávkami a tento nástroj se jmenuje SoX (Sound eXchange). Tento nástroj musí být umístěn na nějakém serveru (nejlépe s aplikací) a přidává tak další komplikaci v případě řešení našeho probému nahrávání zvukového záznamu.

Vzhledem k již zmíněným omezením jsem vybral pro HTML 5 aplikaci přístup Web Speech API. K druhému přístupu se budeme věnovat dále v aplikaci pro Android. Web Speech API má obrovskou výhodu a sice je možné neomezeně dlouho rozpoznávat řeč a výsledek je ihned k dispozici.

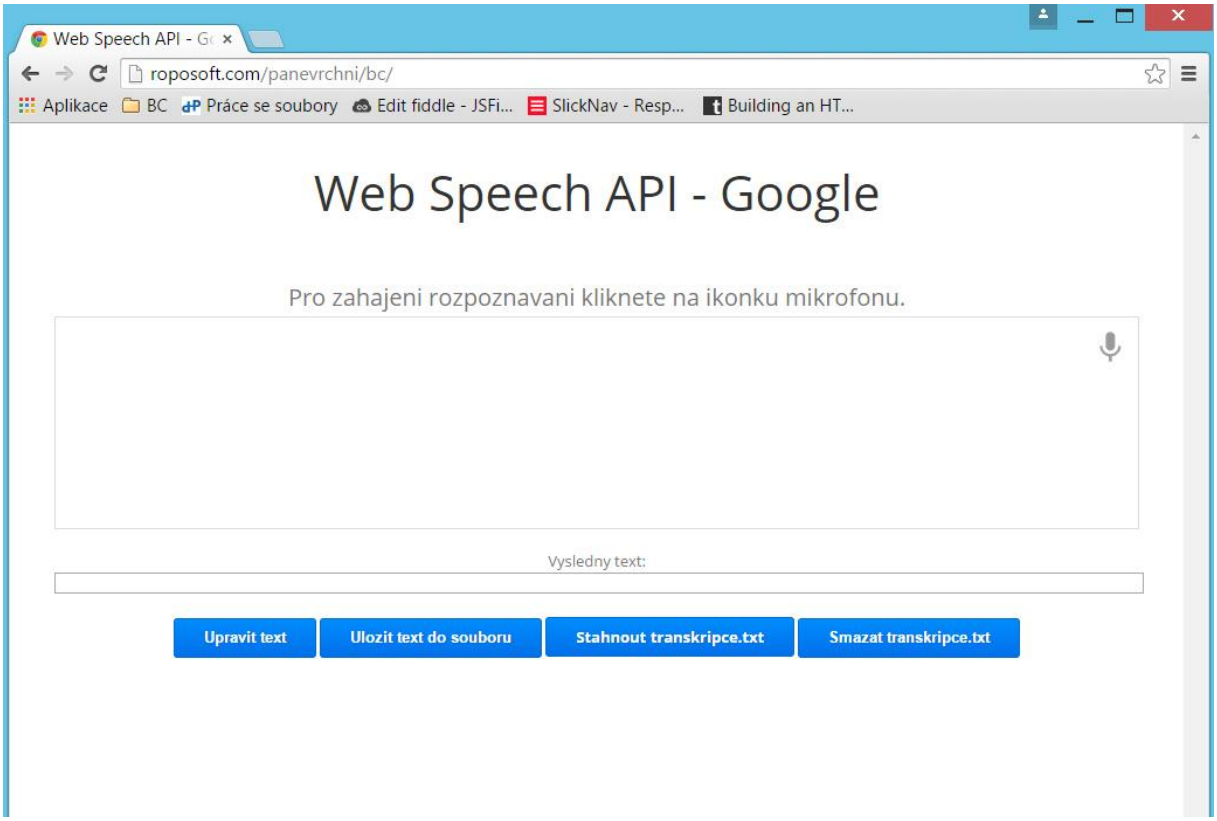
5.1. Web Speech API

Důležitým úkolem bylo se vypořádat nějakým způsobem s pořízením zvukového záznamu. Ve webovém prostředí je poměrně jednoduché pořídit zvukový záznam ve formátu WAV, který díky svému nekomprimovanému formátu je výpočetně nenáročný a vhodný pro další účely rozpoznávání, nejen proto, že je zaznamenáván v nejvyšší bezztrátové kvalitě.

V tomto API není možné nejprve uložit zvukový záznam a ten pak odeslat na server, protože rozpoznávání probíhá okamžitě, je tedy nutné zvuk zaznamenávat zároveň, kdy již běží jeden proces, který využívá mikrofon pro Google. Pořízení zvukového záznamu do formátu WAV probíhá paralelně. Toto řešení skýtá jeden problém a to takový, že v mobilní verzi webového prohlížeče Chrome v zařízení, které jede na Androidu, není možné tyto dvě paralelní záznamy pořídit (desktop funguje bez problému), ale vzhledem k tomu, že rozpoznávání od Googlu přes webový prohlížeč zatím není nijak výrazně podporováno v mobilních zařízeních, tak tento problém vyřešíme dále v této práci použitím druhého přístupu API od Googlu v aplikaci určené pro Android.

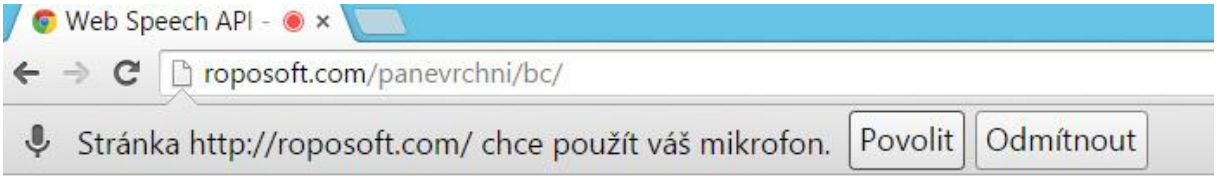
5.2. Aplikace HTML 5 – Google ASR

Ukázkovou aplikaci Web Speech API bylo nutné upravit pro naše účely aplikace. Nejprve bylo nutné pořídit, jak již bylo řečeno, zvukový záznam ve formátu WAV. Tento skript napsaný v Javascriptu je dostupný volně na internetu, je tedy možné ho použít i komerčně. Bylo ho nutné poupravit, aby výsledná nahrávka bylo ve formátu MONO, protože je tím zmenšena výsledná velikost záznamu a nejsou tím ztracena důležitá data pro rozpoznávání.



Obr. 3: Ukázka aplikace HTML 5 - Google

Defaultně je v aplikaci nastaveno rozpoznávání českého jazyka. Z bezpečnostních důvodů je nutné povolit použití mikrofonu. Toto povolení je nutné zprvu dvakrát odsouhlasit, jednou pro Google rozpoznávač a jednou pro záznam do WAV. Pokud přistupujete na server, kde je aplikace pomocí HTTP protokolu, tak musíte při každém rozpoznávání povolit použití mikrofonu. Obejít toto stereotypní neustálé povolení užívat mikrofon lze jen tehdy, když je server zabezpečený pomocí protokolu HTTPS, pak je třeba povolit použití mikrofonu pouze jednou.



Obr. 4: Povolení užívání mikrofonu prohlížečem

Nyní se podíváme, jak samotné API aplikace funguje. Nejprve je nutné ověřit, jestli prohlížeč podporuje Web Speech API přítomností webkitSpeechRecognition objektu. Pakliže objekt existuje, tak jej inicializujeme.

Ukázka kódu inicializace Web Speech API:

```
if (!('webkitSpeechRecognition' in window)) {
  upgrade();
} else {
  start_button.style.display = 'inline-block';
  var recognition = new webkitSpeechRecognition();
  recognition.continuous = true;
  recognition.interimResults = true;
}
```

Atribut recognition.continuous nastavíme na hodnotu true. Tento atribut nám určuje co udělat v případě, když řečník přestane v průběhu promluvy mluvit. Pakliže je nastaveno na true, tak rozpoznávač čeká na další promluvy, v opačném případě rozpoznávání ukončí. Dalším atributem je recognition.interrimResults, který také nastavíme na hodnotu true, která znamená, že výsledky se v průběhu rozpoznávání můžou v závislosti na dalším kontextu ještě měnit. Text, který se již měnit nebude je označen černým fontem a naopak šedým, který se ještě změnit může [11].

V proměnné langs je uložen kód jazyka, který bude rozpoznávač používat. Lze samozřejmě přenastavit na jiný jazyk podle BCP-47 specifikace. Např. en-US pro americkou angličtinu. Po stisku tlačítka mikrofonu se aktivuje funkce rozpoznávání dle zvoleného jazyka. Zároveň se spustí nahrávání zvukového záznamu do formátu WAV.

Funkce pro zahájení rozpoznávání:

```
function startButton(event) { …  }
```

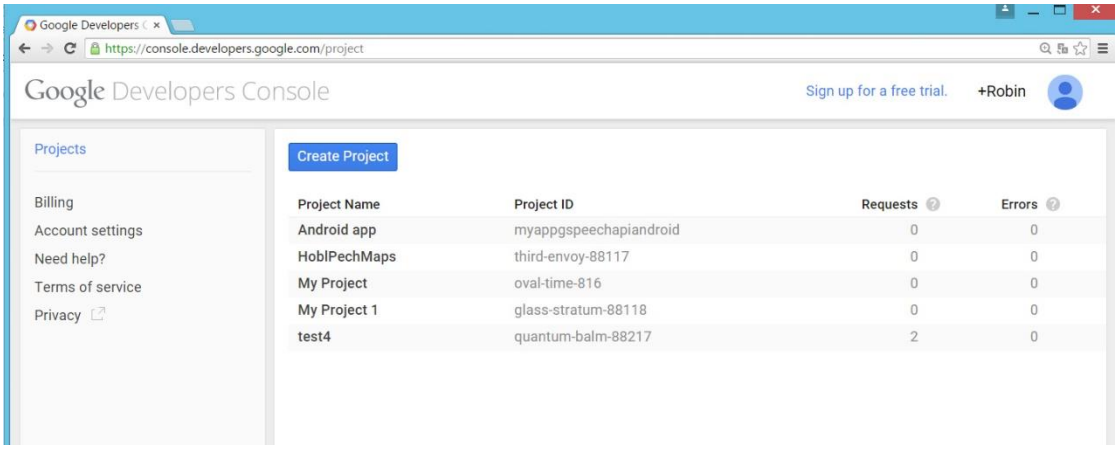
Kliknutím opět na ikonku mikrofonu nahrávání přerušíme a výsledný text se nám zobrazí do textového pole a je uložen v proměnné final_transcript. S touto proměnnou dále pracujeme a je možné dodatečně upravit text stisknutím tlačítka „Upravit text“ a nakonec text uložit do souboru transkripce.txt. Mezitím se nám do PC automaticky stáhne zvukový záznam WAV.

5.3. Aplikace Android – Google ASR

Pro Android existuje podobné rozhraní, jako jsme použili v případě HTML 5 aplikace. Toto Android Speech Input API je kostrbaté na použití a chová se jako tzv. „blackbox“ třída. Není možné ho nějak výrazně modifikovat a postrádá možnost zároveň nahrávat zvukový záznam, který je pro nás klíčový. V našem případě jsem tedy zvolil Google Speech API Full Duplex V1. Nahrávka je omezena délkou záznamu a ta je cca. 25 vteřin. Jak již bylo dříve zmíněno, je nutné k tomuto API mít vygenerovaný unikátní API key.

5.4. Získání API key

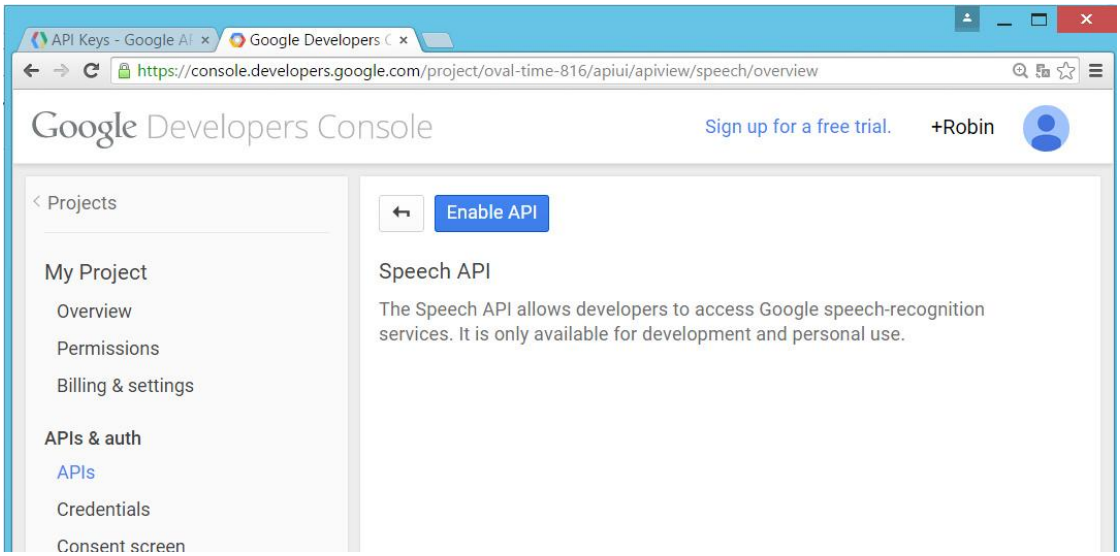
1) Je nutné mít zřízen účet na Googlu
2) Na adrese https://console.developers.google.com/project je třeba se přihlásit ke svému Google účtu a kliknout na tlačítko Create Project. Vyskočí nám nové okno, kam zadejte libovolný název projektu a stiskněte tlačítko Create pro potvrzení.



Obr. 5: Zobrazení Google konzole pro vývojáře.

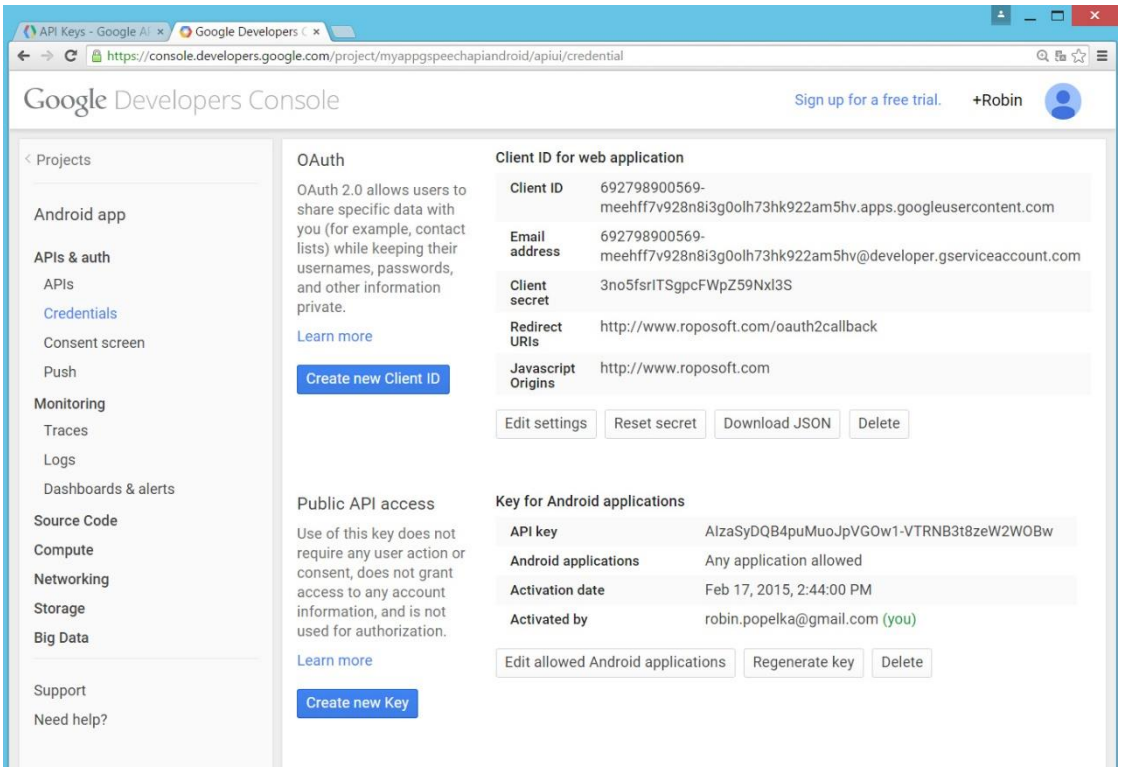
3) Chvíli vyčkejte, než se projekt vytvoří a poté v levém menu rozlikněte pod názvem APIs & Auth odkaz APIs. V horní části vyhledejte příslušné API, v našem případě

Speech API a poté jej kliknutím na Enable API aktivujte.



Obr. 6: Aktivování Speech API.

4) Dále rozklikněte v levém menu odkaz Credentials. Nejprve klikněte na tlačítko Create new Client ID a v novém okně vyberte typ aplikace: Installed application a potvrďte kliknutím na tlačítko Create client ID.



Obr. 7: Vytvoření nového ID v Google konzoli.

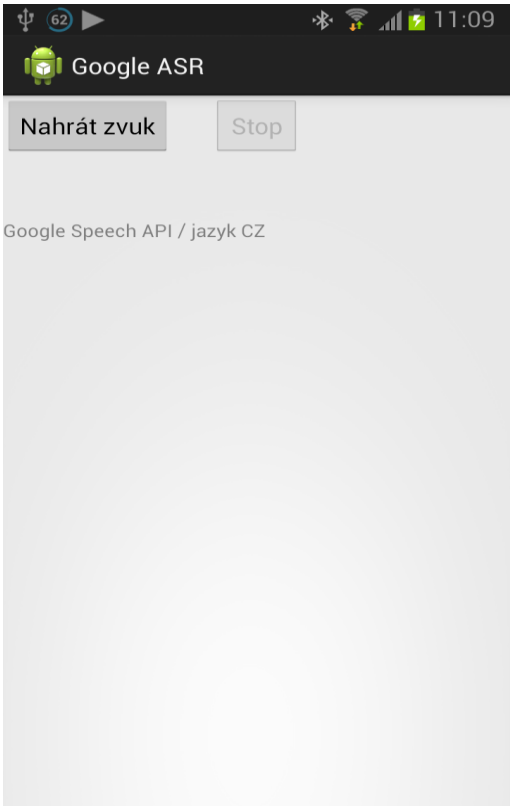
5) Teprve nyní můžeme vygenerovat nový API key kliknutím na tlačítko Create new key. V novém okně zvolte Android key a dále potvrďte tlačítkem Create. Tento klíč může být

použitelný pouze pro konkrétní aplikaci a package (balíček tříd) a Google kontroluje na vstupu, jestli souhlasí v aplikací vygenerovaný zašifrovaný certifikát SHA1. Pro naše účely tento certifikát nepotřebujeme a necháme tedy kolonku pro vyplnění certifikátu prázdnou. Nyní nám byl vygenerován nový API key pro androidí aplikaci.

Samotná Android aplikace pro záznam a odeslání zvuku na server Googlu je dostupná v příkladové ukázce na internetu na adrese https://github.com/katchsvartanian/voiceRecognition [12]. Tuto aplikaci jsme museli upravit tak, aby splňovala kritéria pro aplikaci v zadání této práce. Po naimportování aplikace do Eclipsu je možné vidět několik tříd. Jsou zde pro nás nyní nedůležité třídy pro záznam zvukového signálu tentokrát do bezztrátového formátu FLAC, který Google API vyžaduje. Tento je omezen svojí délkou cca. 25 vteřin. Záznam se ukládá do paměťové karty zařízení konkrétně do složky GoogleASR a spolu s ním je ukládán i výsledek rozpoznávání v textovém souboru. Třída, která nás bude zajímat je MainActivity.java. V této třídě je možné nastavit námi nově vygenerované API key tak, že jej nahradíme v proměnné API_key. Dále je zde možné nastavit jazyk rozpoznávání, v našem případě je zvolen opět český jazyk a je zde uvedena adresa Google API.

Ukázka kódu v aplikaci pro Android:

```
public class MainActivity extends Activity {
     int tmpNum = 1;
     // jazyk rozpoznavani
     String language = "cs_cz";
     // API klic
     String api_key = "AIzaSyDQB4puMuoJpVGOw1VTRNB3t8zeW2WOBw";
     // URL Google API
     String root = "https://www.google.com/speech-api/full-
duplex/v1/";
     String dwn = "down?maxresults=1&pair=";
     String API_DOWN_URL = root + dwn;
     String up_p1 = "up?lang=" + language
               + "&lm=dictation&client=chromium&pair=";
     String up_p2 = "&key=";
```



Obr. 8: Ukázka aplikace pro Android

6. Data pro vyhodnocení

6.1. Textové podklady

Pro vyhodnocení výsledků z rozpoznávače jsme si připravili několik různých sad promluv, které se od sebe výrazně liší svým obsahem, nebo délkou samotného textu. Textové podklady jsou zvoleny tak, abychom při zpracování výsledků dokázali analyzovat celkové schopnosti rozpoznávače, a říci v jakém prostředí je rozpoznávač vhodný či nikoli. Pro rozpoznávač jsme připravili promluvy obecného a specifického charakteru.

Obecné texty můžeme rozdělit do třech různých kategorií. Jedná se vždy o nějakou souvislou větu a každá kategorie obsahuje 10 vět. První kategorií jsou věty zaměřené na zpravodajství. Jedná se o náhodně vybrané věty přepsané z internetových zpravodajských portálů popř. z novin. Například věta: Dospělému pachateli hrozí v krajním případě trest odnětí svobody až na deset let. Další kategorií jsou nahodile vybrané zprávy z emailové konverzace nebo SMS zpráv. Například věta: Dobrý den, převzali jsme do přepravy balík. Poslední kategorií jsou přepsané popř. smyšlené věty, které se obsahově týkají kulturních akcí v Praze. Například věta: Centrum současného umění DOX představuje práci 12 předních fotografů.

Specifické texty jsou rozdělené obsahově do čtyř kategorií. V každé kategorii je celkem 15 slovních spojení nebo krátkých vět. První kategorií jsou texty zaměřené na datum a čas. Například věta: Minulý čtvrtek dopoledne. V další kategorii se zaměřujeme na názvy ulic v Plzni například: Mikulášské náměstí – Klatovská třída. Předposlední kategorií jsou slovní spojení dvou českých měst např.: Z Prahy do Plzně. Poslední kategorií jsou osobnosti českého sportu a název jejich sportu např.: Jaromír Jágr – hokejový útočník. Celý výčet textových podkladů můžete najít v příloze číslo 1.

6.2. Prostředí pro nahrávání

Celkem se jedná o 90 nahrávek různého zaměření. Nahrávky byly zaznamenány v několika různých prostředích a také několika různými lidmi. Všichni lidé zaznamenali data v domácím nerušeném prostředí v HTML 5 aplikaci pomocí běžného vestavěného mikrofonu, který je součástí notebooku. Pro nahrávky v zarušeném venkovním prostředí byla zvolena aplikace pro mobilní zařízení se systémem Android. Zdroj zvukového záznamu v tomto případě byl zvolen mikrofon, který je součástí mobilního zařízení, v našem případě vestavěný mikrofon na mobilním telefonu. V zarušeném prostředí bylo vybráno několik různých míst, kde byly nahrávky pořízeny. V prvním případě se jedná o frekventovanou křižovatku na ulici Klatovská a Americká v Plzni. Další místo, které bylo vybráno, je zarušené prostředí v restauraci a

posledním místem pro pořízení nahrávek byla zvolena jízda městskou hromadnou dopravou. Celkově byla vybrána reálná místa, ve kterých se běžně člověk pohybuje, a dobré výsledky takového rozpoznávání jsou důležitým kritériem pro reálné nasazení rozpoznávání řeči v praxi.

6.3. Anotace nahrávek

Veškeré zaznamenané nahrávky bylo nutné anotovat a přesně upravit podle zadaných kritérií, abychom je mohli následně jednoduše vyhodnotit v programu HTK Hresults. Jednotlivé výsledky rozpoznávání jsou uloženy v textových souborech. V obou aplikacích využívající různé API od Google bylo nutné vrácený výsledek upravit do správného formátu, který je definován následujícím způsobem.

Ukázka anotace:

```
"*/001.wav"
ukázkový
text
anotace
.
```

První řádek představuje název zvukového záznamu ve formátu WAV, nebo v případě aplikace pro Android ve formátu FLAC. Na dalších řádcích jsou od sebe oddělena jednotlivá rozpoznaná slova a konec jedné nahrávky označuje tečka na posledním řádku.

7. Experimenty

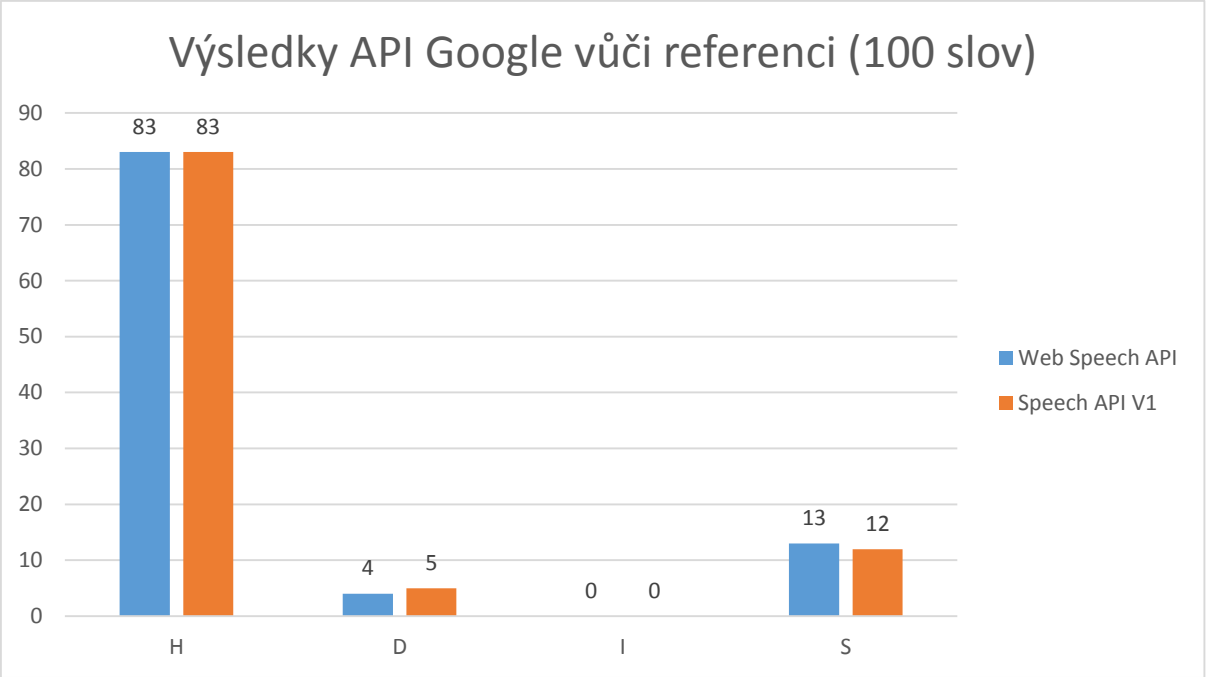
7.1. Porovnání API Google v HTML 5 a Android aplikaci

Vzhledem k tomu, že bylo použito odlišné API pro HTML 5 a Android aplikaci, tak se nyní podíváme na to, jak si jednotlivé API od Google dokáží poradit s rozpoznáváním. Půjde nám o to, jestli jsou konečné výsledky rozpoznávání u obou API stejné, nebo se od sebe liší a v případě, že budou odlišné, proč tomu tak je. Jak již bylo zmíněno, tak pro webovou aplikaci je použito Web Speech API napsané v Javascriptu a pro aplikaci pro Android je použito Speech API Full Duplex V1, dále jen Speech API V1. Tyto dvě API od Googlu se od sebe liší na první pohled v tom, jaký soubor je ke zpracování na server odeslán. U Web Speech API se soubor odesílá kontinuálně na server Google přímo přes webový prohlížeč a výsledek je ihned vrácen v textové podobě do okna webového prohlížeče. U rozhraní Speech API V1 je nutné nejprve nějakým způsobem pořídit záznam, který musí být v bezztrátovém formátu FLAC a ten lze teprve odeslat na server Google, který nám vrátí výsledek rozpoznávání v JSON objektu. *JavaScript Object Notation (JavaScriptový objektový zápis zkráceně JSON) je způsob zápisu dat (datový formát) nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Vstupem je libovolná datová struktura (číslo, řetězec, boolean, objekt nebo z nich složené pole), výstupem je vždy řetězec* [13].

Z výše uvedeného vyplývá, že nelze pro webovou aplikaci nejprve nahrát a až poté záznam odeslat na server Googlu, tak jsme pro porovnávání využili toho, že toto v aplikaci pro Android možné je a porovnávání obou API rozhraní bylo tedy provedeno tak, že nejprve byl pořízen záznam spolu s výsledkem ve webové aplikaci a poté stejný v aplikaci pro Android. Protože je u webové aplikace zvukový záznam uložen ve formátu WAV, který je také bezztrátový jako FLAC, bylo nutné ho před zpracováním v aplikaci pro Android upravit do FLAC formátu.

7.2. Výsledky porovnání API Google

Pro porovnání výsledků bylo náhodně vybráno patnáct různých textových podkladů o celkové délce 100 slov a vyhodnocení výsledků bylo provedeno v programu HTK HResult.



Z výsledků vyhodnocení můžeme z grafu vyčíst, že se od sebe jednotlivé API, byť jen minimálně liší. Výsledky jsou vyjádřeny procentem z celkového počtu 100 slov a jednotlivé API jsou porovnávány vůči referenčnímu dokumentu. V tomto konkrétním porovnání jsou celkové výsledky procentuálního vyčíslení rozpoznávání.

7.2.1. Ukázka chyb obou API vůči referenci

Reference	Z	HRADCE	KRÁLOVÉ	DO	MORAVSKÉ	TŘEBOVÉ
Web Speech API		ZAPNI	KRÁLOVÉ	DO	MORAVSKÉ	TŘEBOVÉ
Speech API V1		ZASE	KRÁLOVÉ	DO	MORAVSKÉ	TŘEBOVÉ

Z výše uvedené ukázky je patrné, že v obou případech se jedná o jednu chybu substituce (S) a jedno smazané slovo (D). Chybné vyhodnocení je v obou případech u stejných slov. V případě substituce se jedná o rozdíl v rozpoznaném slovu, kdy obě API rozpoznala slovo špatně.

7.2.2. Výsledky porovnání obou API mezi sebou

	Corr	Acc	H	D	S	I	N
Web Speech API Speech API V1	93.68 %	92.63 %	89	0	6	1	95

Výše uvedená tabulka vyjadřuje porovnání mezi oběma API od Googlu. Z tabulky můžeme vyčíst, že se výsledky shodují ve více jak 93 procentech. Z výsledků je patrné, že se

výsledky rozpoznávání od sebe liší v šesti substitucích a v jednom případě jde o jedno slovo v rozpoznaném dokumentu navíc tzv. insertion (vložení slova navíc oproti referenčnímu dokumentu).

Tato nepřesnost mezi oběma API se zdá být relativně veliká (7 %), ale ve většině případů se jedná o slova, která nebyla správně rozpoznána oběma API, jen jej každý rozpoznávač jinak vyhodnotil. Proto při výběru API je skoro jedno, jaký z nich v konečném rozpoznávání použijeme. Výsledky rozpoznávání se od sebe nebudou výrazně lišit.

7.3. Proč jsou výsledky API od sebe odlišné

Jelikož není možné tentýž identický zvukový záznam, který je bezprostředně odesílán na server Google stáhnout do počítače, tak bylo nutné vytvořit paralelně záznam totožného zvuku do formátu WAV a poté tento soubor stáhnout do PC pro další zpracování. Již samotný záznam do WAV formátu může vykazovat nepatrně odlišné datové informace a při následném převodu do FLAC formátu, ačkoliv se v obou případech jedná o bezztrátové formáty, může dojít k minimální ztrátě informace. Toto může mít výsledný vliv na odlišné výsledky v rozpoznávání. Dále je také možné, že Google používá jiné algoritmy při zpracování souborů v různých API a tím je možné, že dochází k rozdílnému vyhodnocení u námi použitých API. V každém případě, když jsou výsledky obou API porovnány, tak v celkovém výsledku správného rozpoznávání mezi nimi je minimální rozdíl a v zásadě je téměř jedno, jaký z nich použijeme.

7.3.1. Jaký výsledek vrací Google API

Zatímco u HTML 5 aplikace nám API od Googlu vrací výsledky ihned do webového prohlížeče, tak v aplikaci pro Android je nám vrácen výsledek v již zmíněném formátu JSON.

JSON vrácený výsledek:


```
{“results”:[{alternative”:[{“transcript”:”ukázka
rozpoznávání“,“confidence“:0.6204052},{“transcript”:”ukazka
rozpoznávání”},{“transcript”:”ukázka je
rozpoznávání“},{“transcript“:“ukázka rozpoznávání
řeči“},{“transcript”:”ukázka rozpoznávání
a”}],”final”:true}],”result_index”:0}
```


Google vrací v JSONu několik od sebe různých výsledků. My tedy z těchto výsledků vybereme ten s největší důvěrou (vždy uveden jako první v JSON objektu) a ten je poté zapsán do textového souboru spolu s názvem zvukového záznamu. Ostatní výsledky jsou pro účely

této práce nedůležité. Stejně tak uložíme výsledek vrácený v HTML 5 aplikaci, kdy výsledný text zpracujeme a uložíme do textového souboru.

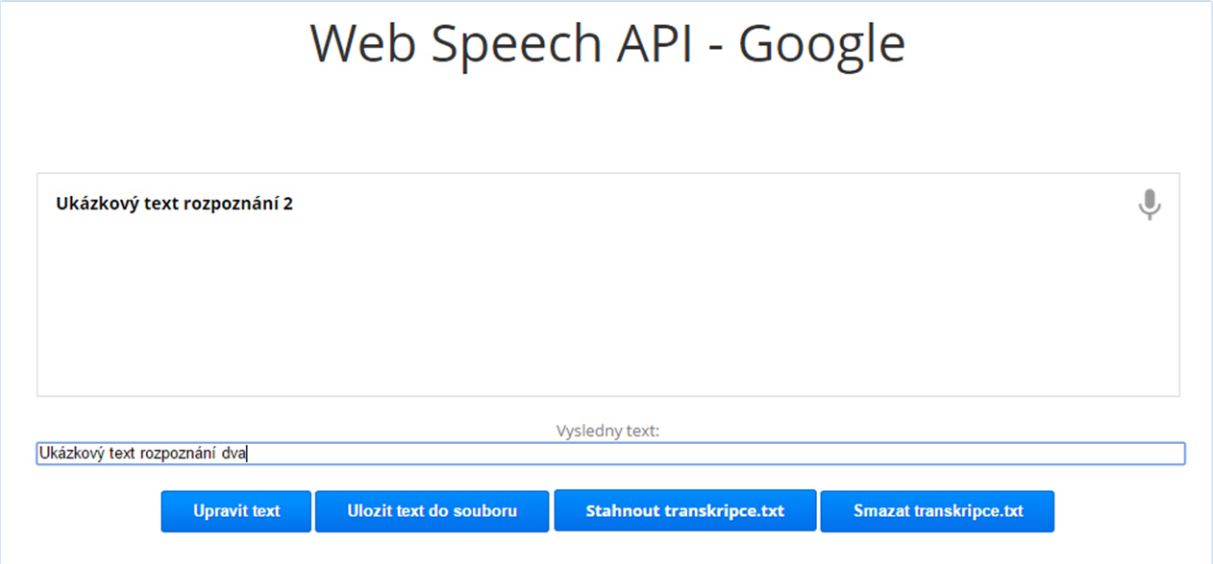
Ukázka zpracování vráceného výsledku do textového souboru v HTML 5 aplikaci:

```
<?php
$text = $_POST['data'];
$number = $_POST['number'];
$rozdelenyText = explode(" ",$text);
$File = "transkripce.txt";
if(filesize ($File)==0){
     $Handle = fopen($File, 'a');
     fwrite($Handle, "#!MLF!#\n");
     fclose($Handle);
}
     $Handle = fopen($File, 'a');
     fwrite($Handle, '"*/'.$number.'.wav"'."\n");
     for ($i = 0; $i < count($rozdelenyText); $i++) {
          if($i == 0){
               $first = lcfirst($rozdelenyText[$i]);
               fwrite($Handle, $first."\n");
          }else{
               fwrite($Handle, $rozdelenyText[$i]."\n");
          }
          $arrLength = count($rozdelenyText);
          if($i+1 == $arrLength){
               fwrite($Handle, ".\n");
          }
     }
fclose($Handle);
?>
```

7.4. Normalizace výsledku pro Google ASR

Z důvodu automatického vyhodnocení výsledků je v některých případech výsledný rozpoznaný text upravit. Např. při rozpoznávání čísel nám v některých případech Google vrací

výsledky napsané v číselné podobě. Program HTK HResults neumí čísla zpracovat a je potřeba převést čísla do psané podoby. Bylo by teoreticky možné vytvořit krátký program pro převod čísel do psané podoby, ale je zde problém v případě českého jazyka s různými podobami téhož čísla, kdy v případě, že je číslo napsané v číselné podobě, není možné poznat, jak bylo přesně číslo řečeno do mikrofonu. Například číslo 2. Toto číslo může mít psanou podobu „dvě“, ale také „dva“. Tento problém je tedy vyřešen tak, že výsledný text lze upravit ručně a až poté výsledek uložit. Dále bylo nutné převést veškerý text na malá písmena.



Obr. 9: Ukázka úpravy textu ve webové aplikaci HTML 5

7.5. Normalizace výsledku pro ZČU ASR

Další úpravy textových souborů byly nutné udělat pro ZČU rozpoznávač zvaný ZČU LVSCR. Tento rozpoznávač rozpoznává některá slova odlišně, přestože nabývají stejného významu a je nutné je poupravit do správné formy. U většiny případů se jedná o nějaké zkratky slov. Například se jedná o slovo kilometr, který rozpoznávač vyhodnotí jako slovo „km“. Pomocí krátkého skriptu napsaného v programovacím jazyce Python jsou tyto a další malé změny v textových souborech provedeny.

Skript v pythonu pro úpravu textových souborů:

```
import sys

x = sys.argv[1]

lines = [line.strip() for line in open(x+'.txt')]

lowercase = []
```

```
#vsechny radky v listu lines prevede na mala pismenka
for i in lines:
    c = i.replace("_", "\n");
    nahrady = c.replace("km", "kilometru");
    nahradaE = nahrady.replace("e-mailu", "mailu");
    d = nahradaE.lower()
    lowercase.append(d)
#toto prevede prvni radek na velka pismena
b = lowercase[0].upper()
#otevre a zapise do noveho lowercase.txt upraveny soubor
y = sys.argv[2] file = open(y+".txt", "w")
file.write(b + "\n")
for item in lowercase[1:]:
  file.write("%s\n" % item)
file.close()
print("soubor byl uspesne ulozen")
input()
```

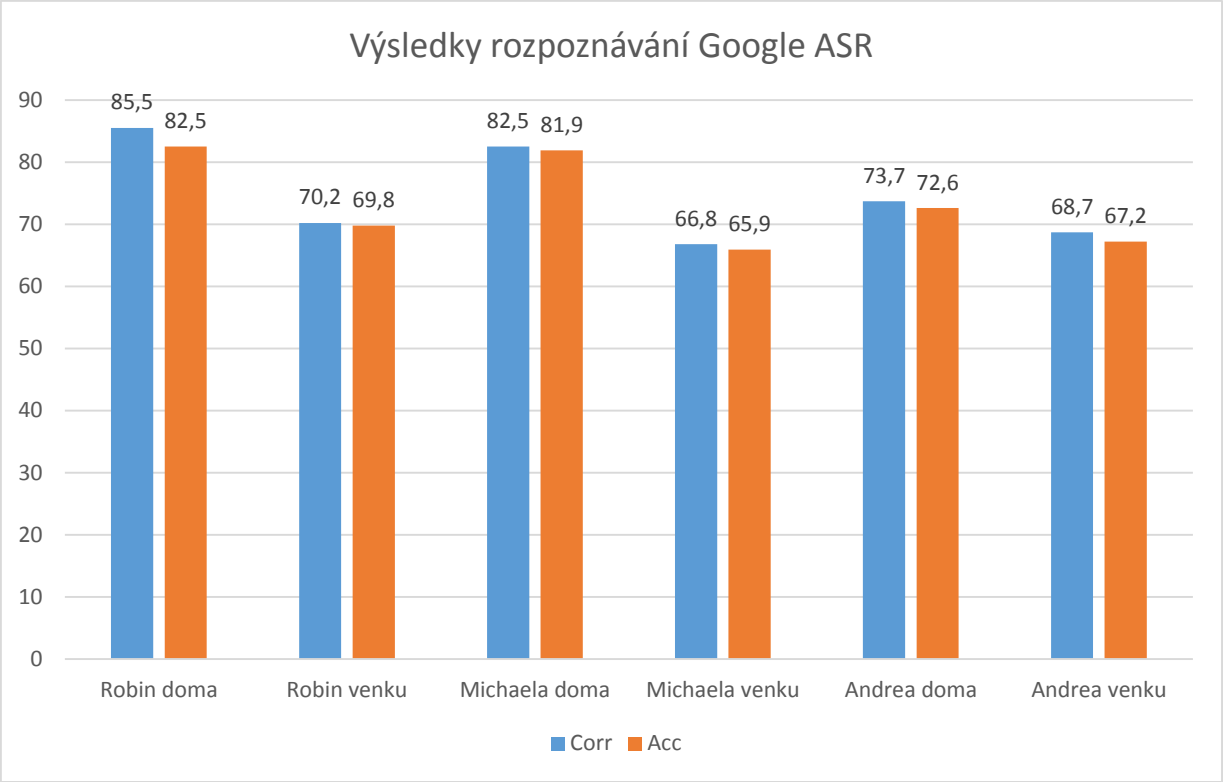
8. Vyhodnocení výsledků

Vyhodnocení výsledků rozpoznávání bylo provedeno jednou pro Google ASR a podruhé pro ZČU LVSCR rozpoznávač. Vyhodnocení probíhalo jako v případě porovnání API Googlu v programu HTK HResults. Nahrávky byly zaznamenány pro lepší vyhodnocení třemi lidmi ve dvou různých prostředích. Prvním prostředím bylo klidné domácí prostředí (HTML 5 aplikace) a druhým venkovní prostředí (Android aplikace). Nahrávky jsou namluveny jednou mužským hlasem a dvakrát hlasem ženským.

8.1. Výsledky Google ASR

Klidné prostředí	**Corr (%)**	**Acc (%)**	**H**	**D**	**S**	**I**	**N**
hlas Robin kvalitní mikrofon	88.06	87.69	472	23	41	2	536
hlas Robin	85.45	82.46	458	9	69	16	536
hlas Michaela	82.46	81.90	442	27	67	3	536
hlas Andrea	73.69	72.57	395	50	91	6	536
Celkem	**82,42**	**81,16**	1767	109	268	27	2144
Zarušené prostředí	**Corr (%)**	**Acc (%)**	**H**	**D**	**S**	**I**	**N**
hlas Robin	70.15	69.78	376	91	69	2	536
hlas Michaela	66.79	65.86	358	87	91	5	536
hlas Andrea	68.66	67.16	368	80	88	8	536
Celkem	**68,53**	**67,60**	1102	258	248	15	1608

Písmenem H označujeme celkový počet správných slov podle již výše uvedeného vzorce. Tento počet správných slov neuvažuje ve svém vzorci započítání počtu značené písmenem I (insertion). Ostatní písmena již byla vysvětlena dříve.



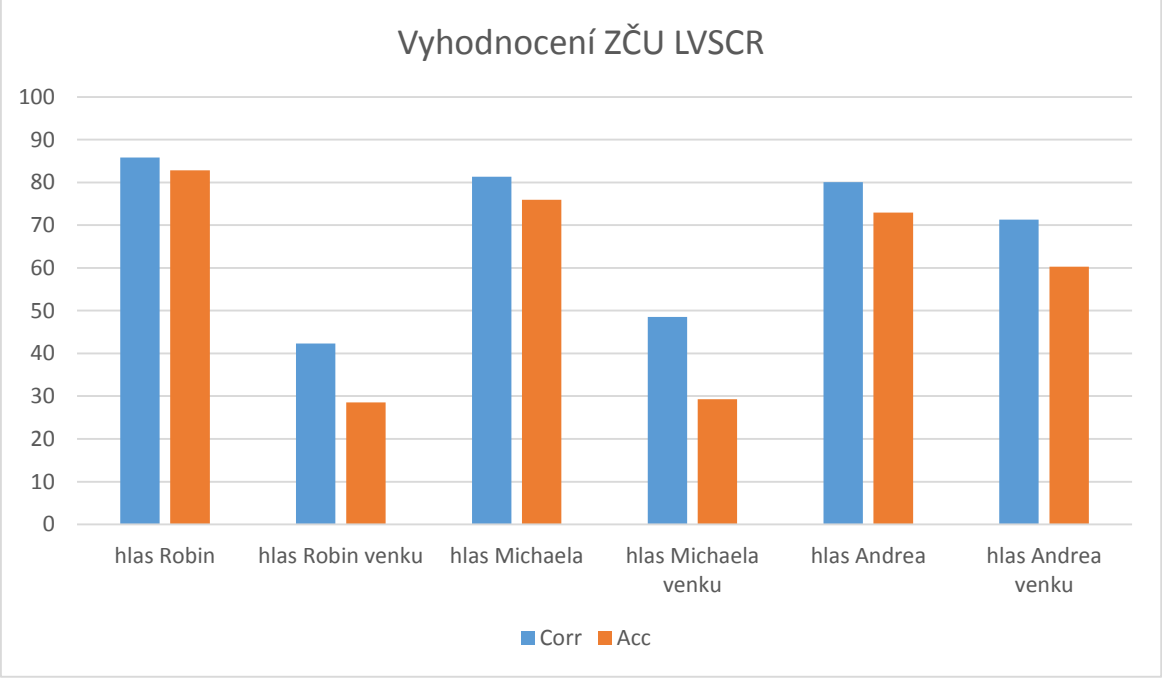
Z výsledků rozpoznávání pomocí Google ASR můžeme vyčíst, že jsou výsledky nahrávek pořízených v domácím prostředí lepší, než nahrávky pořízené v zarušeném prostředí. Rozdíl činí až 16 %. Toto číslo není zanedbatelné, ale je potřeba uvažovat v jakých podmínkách byly nahrávky pořízeny. Ve většině případů se jednalo o hodně zarušená místa. Nahrávky byly konkrétně nahrávány na frekventovaných ulicích Plzně, v městské hromadné dopravě či v restauracích. Je také třeba vzít v potaz, že nahrávky ve venkovním prostředí byly nahrány vestavěným mikrofonem mobilního telefonu. Je nutné poznamenat, že nahrávky byly nahrány v modelových situacích, které jsou v každodenním životě běžné a výsledky tak mají vypovídající reálnou hodnotu, co se rozpoznávání týče. V případě nahrávek hlasu Andrea činí rozdíl mezi prostředími cca 5 %. V tomto případě výsledky v domácím prostředí nedosahují stejné výše jako výsledky u ostatních hlasů, ale právě rozdíl mezi prostředími není veliký a je tak možné považovat výsledky rozpoznávání v různých prostředích pomocí Google ASR za přijatelné a použitelné v praxi. Nejvyššího výsledku dosahuje hlas, při kterém bylo použito kvalitního náhlavního mikrofonu Sennheiser PC230. Tento výsledek hlasu se v porovnání s vestavěným mikrofonem notebooku liší o cca. 3 % pro Currecy a více jak o 5 % pro Accuracy. Z tabulky je patrné, že rozdíl mezi celkovou správností a přesností není veliký.

8.2. Výsledky ZČU LVSCR

8.2.1. Technické údaje ZČU LVSCR

Akustický model je natrénován z 300 hodin čistých nahrávek diktovaných 800 řečníky do náhlavního mikrofonu. Data jsou zpracovávána na frekvenci 16 kHz. Využívány jsou GMM, konkrétně 4922 stavů HMM, každý stav je modelován 36 složkami. Trigramový jazykový model je natrénován přibližně z miliardy tokenů textových dat - novinových článků, přepisů rozhlasových a televizních pořadů, titulků a dalších textů. Celkem obsahuje přes milion slov a slovních tvarů a 50 milionů n-gramů.

Klidné prostředí	**Corr (%)**	**Acc (%)**	**H**	**D**	**S**	**I**	**N**
hlas Robin kvalitní mikrofon	91,60	89,93	491	5	40	9	536
hlas Robin	85,82	82,84	460	8	68	16	536
hlas Michaela	81.34	75.93	436	13	87	29	536
hlas Andrea	80,04	72,95	429	24	83	38	536
Celkem	**84,70**	**80,41**	1816	50	278	92	2144
Zarušené prostředí	**Corr (%)**	**Acc (%)**	**H**	**D**	**S**	**I**	**N**
hlas Robin venku	42,35	28,54	227	51	258	74	536
hlas Michaela venku	48,51	29,29	260	33	243	103	536
hlas Andrea venku	71,27	60,26	382	37	117	59	536
Celkem	**54,04**	**39,36**	869	121	618	236	1608



Výše uvedené výsledky rozpoznávače ZČU LVCSR můžeme považovat jako velmi dobré v případě klidného domácího prostředí. Lze říci, že dosahují stejného kvalitního rozpoznávání jako Google, v některých případech i vyšší. Výsledky ze zarušeného prostředí nejsou již tak dobré jako v případě Google. Důvodem proč tomu tak je, můžeme vyčíst z technických parametrů ZČU LVCSR. Akustický model tohoto rozpoznávače je natrénován na ničím nezarušených nahrávkách pomocí kvalitního mikrofonu, proto v případě použití kvalitního náhlavního mikrofonu jsou dosažené výsledky bez mála 92 %. Vzhledem k výsledkům rozpoznávače Googlu v zarušeném prostředí můžeme usoudit, že je akustický model v tomto případě natrénován i na zarušené prostředí s pomocí vestavěného méně kvalitního mikrofonu.

8.3. Shrnutí výsledků

Z níže uvedené tabulky lze porovnat výsledky jednotlivých rozpoznávačů. Oba dva rozpoznávače dosahují v případě kvalitních ničím nerušených nahrávek zhruba stejných výsledků. V případě nekvalitních nahrávek ve venkovním prostředí dosahuje rozpoznávač od Googlu lepších výsledků, protože jak již bylo řečeno, má k dispozici nahrávky z reálného prostředí z mobilních telefonů, kdežto rozpoznávač ZČU LVSCR disponuje pouze kvalitními čistými nahrávkami.

Souhrnné vyhodnocení	Corr	Acc
Google doma	**82,42**	**81,16**
Google venku	**68,53**	**67,60**
ZČU LVSCR doma	**84,70**	**80,84**
ZČU LVCSR venku	**54,04**	**39,36**

9. Závěr

Cílem této práce bylo zanalyzovat možné přístupy automatického rozpoznávání řeči na mobilních zařízeních, dále pro jednotlivé přístupy připravit ukázkové aplikace v HTML 5 a pro Android a otestovat schopnost rozpoznávání. Tento cíl byl v rozsahu zadání bakalářské práce splněn. Mezi hlavní úspěchy patří schopnost zároveň zaznamenávat a rozpoznávat stejný zvukový záznam, který můžeme použít pro další testování, ať už pro námi provedené vyhodnocení v ZČU LVCSR, nebo pro jiné účely. Toto je vyřešeno v HTML 5 aplikaci. Tento problém se objevil v průběhu řešení práce, kdy je problém nejprve nahrát zvukový záznam a ten poté odeslat na servery Googlu pro zpracování výsledku, aniž bychom nebyli svázáni omezujícími podmínkami, které limitují odesílání souborů na servery Googlu na 50/den. V jednotlivých aplikacích byly pomocí vytvořených textových podkladů zaznamenány nahrávky celkově třech lidí v klidném a zarušeném prostředí. Tyto nahrávky byly dále anotovány a pomocí programu HResults vyhodnoceny.

Zdroje

[1] Speech Recognition (Audio Processing) (Video Search Engines). *what-when-how* [on-line].[cit. 20.2.2015]. Dostupné z: http://what-when-how.com/video-search-engines/speech-recognition-audio-processing-video-search-engines/

[3] Titulkování živého vysílání ČT. Západočeská univerzita v Plzni a firma SpeechTech, s.r.o. Duben 2015. [cit. 13.2.2015]. Dostupné z: http://www.zivetitulky.cz/

[4] Rozpoznávání řeči. Wikipedie [on-line]. Únor 2015. [cit. 25.2.2015]. Dostupné z: http://cs.wikipedia.org/wiki/Rozpoznávání_řeči

[5] HTK Speech Recognition Toolkit. University of Cambridge [on-line]. Březen 2009. [cit. 25.4.2015]. Dostupné z: http://htk.eng.cam.ac.uk/download.shtml

[6] Mluvíme s počítačem česky. Psutka J., Müller L., Matoušek J., Radová V. Academia, Praha 2006. [cit. 10.2.2015].

[7] Zvuková karta. Wikipedia [on-line]. Únor 2015. [cit. 25.2.2015]. Dostupné z: http://cs.wikipedia.org/wiki/Zvuková_karta

[8] Jak se počítač učí rozpoznávat mluvenou řeč. Igor Szöke[on-line]. 14.07.2010. [cit. 02.03.2015]. Dostupné z: http://www.osel.cz/5152-jak-se-pocitac-uci-rozpoznavat-mluvenou-rec.html

[9] Hlas a počítač. Jan Nouza. 27.8.2009. [cit. 15.2.2015]. Dostupné z: http://www.msmt.cz/file/8005_1_1/

[10] IDC: Podíl Androidu mezi mobilními OS překročil v roce 2014 hranici 80 %. Matěj Čuchna. 26.02.2015. [cit. 18.4.2015]. Dostupné z: http://channelworld.cz/analyzy/idc-podil-androidu-mezi-mobilnimi-os-prekrocil-v-roce-2014-hranici-80-13351

[11] Webové stránky dostanou rozpoznávání řeči – přichází Web Speech API. Martin Hassman. 16.1.2013. [cit. 8.1.2015]. Dostupné z: http://www.zdrojak.cz/clanky/webove-stranky-dostanou-rozpoznavani-reci-prichazi-web-speech-api/

[12] katchsvartanian/voiceRecognition Github. Katchsvartanian. 17.04.2014. [cit. 21.4.2015]. Dostupné z: https://github.com/katchsvartanian/voiceRecognition

[13] JavaScript Object Notation. Wikipedia. 23. 2. 2015. [cit. 18.3.2015]. Dostupné z: http://cs.wikipedia.org/wiki/JavaScript_Object_Notation

Příloha č. 1

Kompletní seznam textových sad pro testování rozpoznávače

Obecné texty

10 vět ze zpráv

Dospělému pachateli hrozí v krajním případě trest odnětí svobody až na deset let.

Dálnice D2 ve směru na Brno byla v neděli několik hodin uzavřena.

K nehodě došlo před polednem zhruba na sedmém kilometru, kde vyjel řidič mimo silnici.

Už před třemi zavražděnými potomky mladá matka porodila dvě děti, o které se momentálně stará jejich otec a ženin bývalý partner.

Jeden desetiletý pionýr zasedl ke stolu a začal psát osobní dopis adresovaný prezidentu republiky Antonínu Zápotockému.

Na své si přijdou i menší návštěvníci, pro které bude nachystáno jedenáct komiksových panelů.

Oceněním chceme zviditelnit ty, kteří svou každodenní činností zajišťují odbornou přípravu mladé generace, ale také mnohdy rozhodujícím dílem přispívají k formování jejich osobnosti

Rozvoj moderních technologií dnes umožňuje pracovat v podstatě odkudkoli, kde je přístup k internetu.

Byt mezitím v dražbě prodal exekutor, peníze jen tak tak stačily pokrýt dluh u banky.

Klienti bank v Česku by si měli dávat velký pozor na to, co na pobočce finančního ústavu podepisují.

10 vět z SMS/mail

My taky v Lednici na Moravě. Jsme na zabíjačce.

Váš ověřovací kód Google je sedm osm pět tři dva devět.

Dobry den, převzali jsme do přepravy balík.

Předpokládaný termín doručení je osmnáctého druhý.

Upozorňujeme, že objednávkou domény není provedena její blokace či rezervace

dovolujeme si Vás upozornit, že se blíží konec výpůjční lhůty u níže uvedených titulů.

Vážení reprezentanti, věnujte pozornost tomuto mailu.

Tohoto turnaje se zúčastníte pod záštitou reprezentace

Službu jsme vám dnes aktivovali a účtovat vám ji začneme během nadcházejících 5 dnů.

Vážení studenti, zaktualizujte si prosím před začátkem semestru své osobní rozvrhy.

10 vět kulturní akce Praha

Výstava módní návrhářky Blanky Matragi.

Nový muzikál v divadle Broadway je Mýdlový Princ.

Vstupenky na představení Fantom Opery.

Eva Holubová ve svém nejlepším představení Hvězda.

Otevírací doba ateliéru Josefa Sudka.

Centrum současného umění DOX představuje práci 12 předních fotografů

Jediné historické oděvy můžeme vidět v Uprum muzeum.

60. léta v přestavení Andy Warhol v Praze.

Mecenášský klub Národního divadla slaví páté narozeniny.

Central Group představuje galavečer v čele s Gottem, Šporclem a Laffitou.

Specifické texty

15 vět zadání data

Prvního první dva tisíce patnáct.

Minulý čtvrtek dopoledne

Na Nový rok

Pozítří ráno

Předevčírem

Na Tři krále

Loni v létě

Navečer třetího března

V následujících týdnech

Před deseti lety

Na Vánoce

Druhou adventní neděli

Na konci měsíce

Popozítří

Včera v noci

15 vět ulice Plzeň

Mikulášské náměstí – Klatovská třída

Americká – Chodské náměstí

Sirková - U trati

Náměstí Generála Píky – Radyňská

Slovanská – Zborovská

Politických vězňů – Dobřanská

Vejprnická – Křimická

Nad ZOO – Na Chmelnicích

Kotíkovská – Elišky Krásnohorské

Studentská – Gerská

Plaská – U Velkého rybníka

Na Roudné – Zábělská

Popelnicová – Hřbitovní

V Malé Doubravce – Ke Špitálskému lesu

Rokycanská – Živnostenská

15 vět město - město ČR

Z Prahy do Plzně

Z Žatce do Pardubic

Z Českých Budějovic do Tábora

Z Příbrami do Brna

Z Opavy do Ostravy

Z Písku do Prachatic

Z Valašského Meziříčí do Kopřivnice

Z Pelhřimova do Humpolce

Z Vodňan do Českého Krumlova

Z Hradce Králové do Moravské Třebové

Z Poděbrad do Slanýho

Z Mariánských lázní do Karlových Varů

Z Aše do Stříbra

Z Trutnova do České Lípy

Z Jihlavy do Břeclavi

15 vět sportovní osobnosti ČR a jejich sport

Jaromír Jágr – hokejový útočník

Petra Kvitová – tenistka

Roman Šebrle – desetibojař

Martina Sáblíková – rychlobruslařka

Kateřina Neumannová – běžkyně na lyžích

Emil Zátopek – vytrvalostní běžec

Petr Jákl – judista

Petr Čech – fotbalový brankář

Josef Váňa – dostihový jezdec

Eva Samková – snowboardcrossařka

Ondřej Moravec – biatlonista

Jan Železný – atlet

Šárka Záhrobská – alpská lyžařka

Aleš Valenta – akrobatický lyžař

Pavel Nedvěd – fotbalový záložník