VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

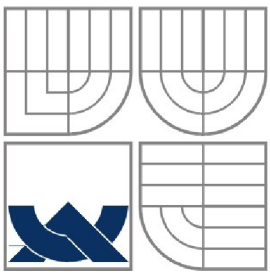
ROZPOZNÁVÁNÍ MLUVČÍHO VE SKYPE HOVORECH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

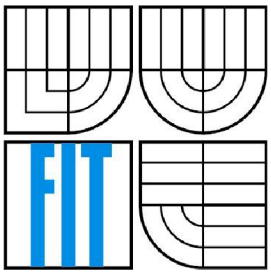
AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ KAŇOK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVÁNÍ MLUVČÍHO VE SKYPE HOVORECH

SPEAKER RECOGNITION IN SKYPE CALLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ KAŇOK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR SCHWARZ, Ph.D.

BRNO 2011

Abstrakt

Tato diplomová práce se zabývá problematikou strojové identifikace a verifikace řečníka, její teorií a aplikací. Vyhodnocuje existující implementaci dané problematiky skupinou Speech@FIT. Dále se zabývá problematikou tvorby zásuvných modulů do komunikačního programu Skype. Následně je navržen zásuvný modul pro Skype umožňující identifikaci a verifikaci řečníka. Ten je implementován a vyhodnocen. V závěru jsou uvedeny návrhy dalšího vývoje.

Abstract

This diploma thesis is concerned with machine identification and verification of speaker, it's theory and applications. It evaluates existing implementation of the subject by the Speech@FIT group. It also considers plugins for the Skype program. Then a functioning plugin is proposed which makes possible identification of the speaker. It is implemented and evaluated here. Towards the end of this thesis suggestions of future development are presented.

Klíčová slova

verifikace řečníka, identifikace rečníka, SkypeAPI, zásuvný modul, BSAPI, Speech@FIT, Qt

Keywords

speaker verification, speaker identification, SkypeAPI, plugin, BSAPI, Speech@FIT, Qt

Citace

Tomáš Kaňok: Rozpoznávání mluvčího ve Skype hovorech, diplomová práce, Brno, FIT VUT v Brně, 2011

Rozpoznávání mluvčího ve Skype hovorech

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Schwarze, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Bc. Tomáš Kaňok
9.5.2011

Poďakovanie

Chcel by som sa poďakovať môjmu školiteľovi Ing. Petrovi Schwarzovi, Ph.D., za vyčerpávajúcu pomoc. Ďalej mojim rodičom, bratovi, Bc. Romanovi Kantorovi, Bc. Martinovi Olšovi, Bc. Martinovi Krupovi a Eve Matejičkovej za podporu pri tvorbe tejto práce.

Obsah

Zoznam obrázkov

Zoznam tabuliek

1 Úvod

1.1 Problematika a využitie verifikácie a identifikácie rečníka

Problematike verifikácie rečníka sa do nástupu informačných systémov venovali len špecialisti z odboru fonoskopie. Tí boli a sú schopní na základe počuteľných a verbálnych informácií určiť, aká je pravdepodobnosť, že na dvoch rôznych náhravkach je rovnaká osoba. Pre svoje tvrdenia využívajú špecifické vady reči, nárečie, tempo, pomery dĺžok hlások, použitú slovnú zásobu, či intelektuálnu úroveň rečníka. Avšak vzhľadom na časovú náročnosť ich práce je ich využitie drahé a ich služby sú využívané najmä v oblasti súdnictva. Prirodzeným spôsobom teda vznikla myšlienka automatického spracovania nahrávok technikou. Jej možnú výpočtovú realizáciu umožnil dostatočný rast použiteľného výpočtového výkonu. Prvotné systémy boli nepraktické a v praxi využívané len ako poradné pre ich slabú úspešnosť. Od počiatku sa v oblasti spracovania vytvoril značný kus práce a jeho dôsledkom sú fungujúce systémy, ktoré, na rozdiel od počiatkov, ponúkajú presnosť reálne využiteľnú v praxi. Automatická identifikácia rečníkov počítačmi pracuje na úplne iných základoch ako fonoskopia.

Technické vyhodnocovanie nahrávok je postavené na extrahovaní akustických vlastností hlasu. Vychádza z myšlienky, že ľudské hlasové ústrojenstvo sa dá popísať ako zvukový filter, pričom parametre popisujúce tento filter sú u jednotlivcov výrazne odlišné. Preto môžu slúžiť na verifikáciu a identifikáciu. V tomto filtri pľúca zohrávajú úlohu zdroja energie. Tie produkujú prúd vzduchu, ktorý prechádza hlasivkami. Tie sa rytmicky rozkmitajú a tak určia základnú frekvenciu v hlase. Potom človek nastavením jednotlivých častí hlasového traktu upraví prechod vzduchu tak, aby dosiahol želaný výstup. Inak povedané, aby povedal to, čo povedať chce. Automatické spracovanie rečových signálov sa snaží nájsť túto jedinečnú konfiguráciu každého človeka, odselektovať ju od šumu a vhodne reprezentovať. Technický prístup sa teda snaží o fyziologickú analýzu hlasového traktu rečníka z nahrávky. Tú fonetici nedokážu vyhodnocovať odposluchom. Tá je u rečníka nemenná a na rozdiel od verbálneho prejavu sa nedá vedome upraviť na inú osobu a aj zdanlivá zmena hlasu je pre automatický rozpoznávač irelevantná. Preto je dnes bežnou praxou odborníka na fonoskopiu klásť relevanciu výsledku verifikátora rečníka na úroveň vlastného odposluchu. Dnešné úspešnosti sú mnohonásobne lepšie a dosiahli úroveň, kedy sa začínajú označovať za spoľahlivé. Svojou rýchlosťou spracovania a kompaktnou reprezentáciou rečníka už umožňujú nielen verifikáciu, ale aj identifikáciu v reálnom čase.

Až donedávna bolo vyhľadávanie párov v kapacitne veľkých databázach záznamov možné len za vysokú cenu. Vzhľadom na dnešnú kompaktnú reprezentáciu rečníka je možné prechádzanie veľkých databáz aj na bežných stolových počítačoch v prijateľnom čase. Možnosť rýchleho vyhľadávania konkrétnej osoby v zhluku nahrávok ponúka nasadenie najmä pre potreby bezpečnostných služieb vo vyhľadávaní záujmových skupín osôb v kvantách dát. Verifikácia zase dosiahla úroveň, ktorá je nepochybne využiteľná v komerčnej sfére. Napríklad ako verifikácia klienta hlasom v rôznych finančných inštitúciach alebo ako verifikácia osôb, ktoré nemajú iný možný spôsob overenia. Táto situácia môže vyvstať napríklad pri komunikácii medzi adekvátnymi oddeleniami pobočiek v nadnárodných korporáciách, kde neexistuje možnosť inej verifikácie. Alebo existuje, ale verifikácia postavená na hlase môže proces overenia identity urýchliť.

Tu je dôležité ujasniť si pojmy verifikácia a identifikácia. V tejto práci bude verifikácia reprezentovať problém rozhodnutia, či je osoba prehlasujúca sa za nejakú identitu naozaj ona. To znamená, že systém dostane dvojicu nahrávok, pričom o jednej bude vedieť, že je od danej osoby a musí rozhodnúť, či aj druhá je od tej istej osoby. Pojem identifikácia však bude označovať problém výberu, kedy je nutné stanoviť, kto a s akou mierou sa najviac zhoduje zo všetkých dostupných nahrávok osôb s neznámou osobou na druhej nahrávke. Čiže systém má k dispozícii veľké množstvo nahrávok a pre jednu neznámu má určiť, či hovoriaca osoba v nej existuje aj na niektorej z dostupných a overených nahrávok.

V rámci semestrálneho projektu sa táto práca snaží o teoretický úvod do spracovania audiosignálov pre potreby identifikácie rečníka tak, aby bol čitateľovi jasný všeobecný prechod od „surovej“ náhravky k reprezentácii konkrétneho rečníka. Práca zahŕňa aj postupy, ktoré sa už nepoužívajú, ale tvorili historický a logický vývoj a sú vhodné pre pochopenie pokročilejších prístupov. Ďalej sa práca venuje prehľadu existujúcich implementácií vyvíjaných skupinou Speech@FIT[1] na FIT VUT v Brne. Následne je spravený prehľad možností tvorby zásuvných modulov do komunikačného programu Skype. Na základe možností jednotlivých programov je navrhnutý zásuvný modul pre program Skype, ktorý poskytuje verifikáciu a identifikáciu rečníka. Modul je v rámci diplomovej práce naimplementovaný, zdokumentovaný a vyhodnotený. Prácu zakončuje zhodnotenie práce, ako aj návrhy na ďalší vývoj aplikácie.

1.2 Konvencie

V tejto práci používam slovenské preklady anglických výrazov, pokiaľ to neuberá na význame. Zastávam názor, že násilné prekladanie do slovenského jazyka v niektorých prípadoch môže byť na škodu aj pre čitateľa s absenciou znalosti anglického jazyka. Preto v týchto prípadoch uvádzam

1 http://speech.fit.vutbr.cz/

anglický pojem a za ním uvedený doslovný preklad do slovenčiny. Pre názvy metód a funkcií použitých v texte používam písmo z rodiny `Courier`. Ak nie je uvedené inak, autorom obrázkov som ja.

2 Spracovanie audiozáznamu

Táto kapitola si berie za úlohu uviesť čitateľa do problematiky prechodu od surového audiozáznamu k reprezentácii konkrétnej osoby využívanej pri identifikácii a verifikácii rečníka. Vzhľadom na obmedzený rozsah tejto práce poskytuje len náhľad do problematiky. Hlbšie pochopenie vyžaduje dodatočné štúdium jednotlivých algoritmov. U čitateľa predpokladám základné znalosti v oblasti tvorby a stavby nekomprimovaného audiozáznamu a klasifikácie.

Vo všeobecnosti je hlavným problémom pri identifikácii vysoká variabilita medzi nahrávkami. Tá plynie zväčša z okolia, kde bola nahrávka vytvorená, z prenosového média, kvality použitého mikrofónu, či aktuálnej nálady rečníka. Túto variabilitu budem v ďalšom texte označovať ako variabilita na kanáli. Ďalej uvedené algoritmy tvoria postupný prechod od nespracovanej nahrávky k jej vektorovej reprezentácii. Kategorizácia algoritmov súvisí s fázou spracovania, v ktorej sú aplikované.

Väčšina teoretického základu je prevzatá z prednášok kurzu SRE vyučovaného na FIT VUT [1].

2.1 Spracovanie na úrovni signálu

2.1.1 Additive noise suppression (Potlačenie prídavného šumu) – ANS

Aditívny šum vzniká ako dôsledok zvukov prostredia, kde je nahrávka vyrobená. Nenesie žiadnu informáciu, a preto je vhodné ho z nahrávky odstrániť. Svoje pomenovanie, aditívny, má z predpokladu, že jeho energia sa v priebehu nahrávky nemení, čiže je konštantne pridaný do záznamu. Potom rámce s tichom, tj. malé časti nahrávky, kde rečník nerozpráva, predstavujú samotný šum. Intuitívne konštantné odstránenie priemernej úrovne šumu po celú dobu nahrávky zdegeneruje informáciu o hlase. Pokročilejší spôsob ponúka pre každý rámec vypočítať filter, ktorý má v prvej fáze rovnaký efekt ako odčítanie priemernej hodnoty šumu. Avšak pred aplikovaním na signál sú parametre jednotlivých filtrov medzi sebou vyhladené v čase a tým poskytujú lepšie výsledky.

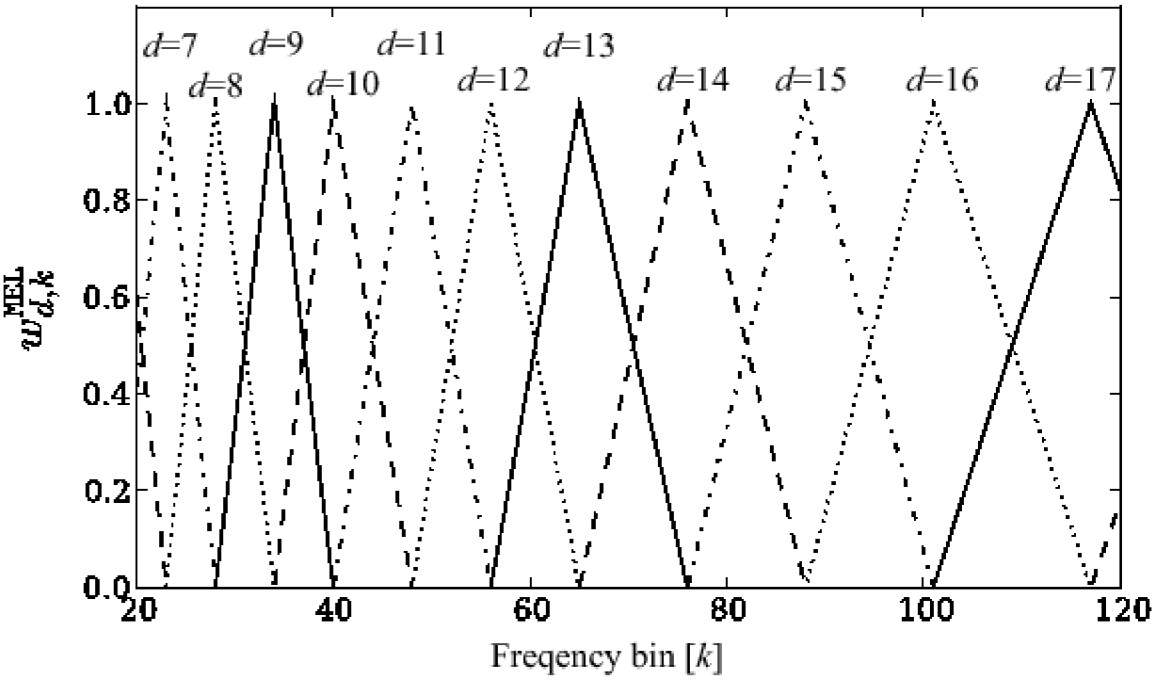
Algoritmus ANS je zložený z dvoch častí. Prvý má za úlohu detekovať jednotlivé rečové rámce a rámce s tichom (šumom). Druhý reprezentuje adaptívny Wienerov filter. Ten podľa typu rámca, frekvenčného spektra reči, frekvenčného spektra pozadia a predchádzajúcich snímkov širokopásmový šum vyfiltruje. Táto metóda sa používa v rozmanitých oblastiach ako predspracovanie signálu. Pre

problematiku identifikácie rečníka sa však nejaví ako najúčinnejšia a v dnešnej dobe existujú prístupy, ktoré tento šum filtrujú na úrovni príznakov [1].

2.2 Spracovanie na úrovni príznakov

2.2.1 Mel-frequency cepstral coefficients (Mel-frekvenčné cepstrálne koeficienty) – MFCC

Príznaky sú číselnou reprezentáciou charakteru pôvodného signálu. Najpoužívanejším pomyselným prechodom z nahrávky do redukovaného priestoru čísel v spracovaní reči sú MFC koeficienty. Vstupný signál, ktorý môže byť predspracovaný odfiltrovaním šumu, je rozdelený na malé, prekrývajúce sa časti. Tieto časti bývajú dlhé 20 ms s prekrytím 10 ms. Pre každý takýto rámec sa spočíta frekvenčné spektrum pomocou Fourierovej transformácie a oddelí sa z neho len magnitúdová časť. Tá je násobená prekrývajúcimi sa trojuholníkovými oknami, nazývanými aj banka filtrov, kde je pre každé prenásobenie spektra s oknom spočítaná energia.



Obrázok 1: Nenormalizovaná banka filtrov [2].

Trojuholníkové okná sú zvolené s ohľadom na relevanciu jednotlivých rozsahov frekvencií v spektre k identifikácii rečníka. Pretože je rozmanitosť medzi rečníkmi skrytá najmä v nízkych frekvenciách, sú okná v oblasti s nízkymi frekvenciami užšie ako v oblasti s vysokými frekvenciami.

Výsledné energie segmentov sú zlogaritmované a vpustené do kosínusovej transformácie. Amplitúdy výstupu kosínusovej transormácie sú mel-frekvenčné cepstrálne koeficienty. Tie sa využívajú v ďalšom spracovaní. Už MFC koeficienty nepredstavujú pri priamom prečítaní žiadnu intuitívnu informáciu pre človeka bez dodatočného strojového vysvetlenia [1].

2.2.2 Cepstral mean subtraction (cepstrálne odčítanie priemeru) – CMS

Vzhľadom na rôzne výrobné materiály a technickú konštrukciu mikrofónov je kvalita ich záznamu rôzna. Myšlienka CMS je postavená na predpoklade, že každý mikrofón sa správa ako filter s definovanou frekvenčnou charakteristikou. Ten potom mení vlastnosti vstupného signálu len na určitých frekvenciách podľa svojej impulznej odozvy. Napríklad rovnaké prehovorenie na dva rozličné mikrofóny spôsobí, že niektoré frekvencie jednotlivých spektier budú prenásobené impulznou odozvou daného mikrofónu. CMS odstraňuje túto nechcenú variabilitu tak, že spočíta priemernú hodnotu daného MFC koeficientu v čase už behom počítania MFC koeficientov. Potom je spočítaný priemer odpočítaný od daného koeficientu v každom čase. Tým sa stredná hodnota koeficientu v čase vyrovná nule a potlačí sa vplyv mikrofónu [1].

2.2.3 RASTA filtrovanie

RASTA filtrovanie sa používa ako alternatíva k CMS. Zo svojej podstaty v sebe CMS algoritmus zahŕňa a doplňuje ho. Rovnako ako CMS sa snaží o odfiltovanie zmien vo frekveknčnom spektre, ktoré nevytvára hlasový trakt. Hlavná myšlienka spočíva v predpoklade, že stavba hlasového traktu neumožňuje z fyziologického hľadiska vytvárať neobmedzene rýchle zmeny svojej konfigurácie. Preto ľudský hlasový trakt nie je schopný generovať rýchle zmeny vo frekvenciách väčších ako 25 Hz. Algoritmus prijíma na vstupe MFC koeficienty, ktoré filtruje s ohľadom na spomenuté skutočnosti v čase. Okrem odstránenia rýchlych zmien v koeficientoch odstraňuje aj veľmi malé zmeny z rovnakého predpokladu. Tým na väčších časových výsekoch nahrádza CMS filtrovanie. Idea predpokladá, že ani príliš pomalé zmeny nie sú dôsledkom zmien hlasového traktu [1].

2.2.4 Cepstral mean and variance normalization (Normalizácia priemeru a rozptylu) – MVN

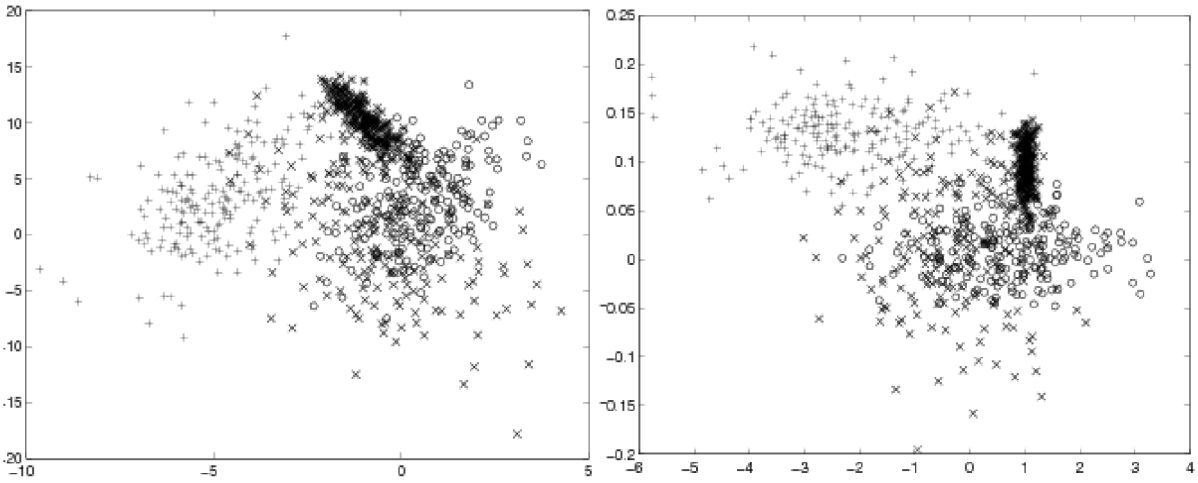
CMVN je znova určitým rozšírením CMS algoritmu. Zatiaľ čo šum spôsobený variabilitou mikrofónov spôsobuje konštantný posun MFC koeficientov a je odstrániteľný odčítaním strednej hodnoty koeficientu cez čas, aditívny šum v spektre vypĺňa „údolia“ priebehu koeficientu v čase. Pre jeho potlačenie sa používa normalizácia rozptylu koeficientu v čase. Tej je dosiahnuté podelením

daného koeficientu smerodajnou odchylkou v každom čase. Tým, že MVN normalizuje rozptyl, odstraňuje aditívny šum spomínaný v kapitole venovanej predspracovaniu signálu. Preto sa pri použití tejto techniky nevyužíva ANS z kapitoly 2.1.1 [1].

2.2.5 Feature warping (Rútenie príznakov) – FW

Ide o ďalší z prístupov, ktorý nahrádza predchádzajúce RASTA filtrovanie a kombinuje výhody predchádzajúcich algoritmov (MVN, RASTA). Rovnako ako pri RASTA filtrovaní sa používa okno (pri RASTA filtrovaní sa okno impluznej odozvy filtra konvoluje cez signál), v ktorom sa vyhodnocujú príslušné hodnoty koeficientov v čase. Zámerom FW je, aby mali nielen jednotlivé koeficienty nulovú strednú hodnotu a normalizovaný rozptyl, ale aby boli v okne lokálne modelované Gaussovým rozložením. To je vhodnejšie pre následné modelovanie pomocou Gaussových rozložení. Algoritmus pracuje tak, že postupne prechádza oknom centrovaným na jednotlivé hodnoty konkrétneho koeficientu v čase. V každom čase všetky body okna zoradí podľa veľkosti a následne premietne spracovávaný koeficient do inverznej distribučnej funkcie hustoty Gaussovského rozloženia okna. Zodpovedajúca hodnota na distribučnej funkcii nahradí pôvodnú veľkosť. Takto spočíta novú veľkosť pre všetky hodnoty koeficientu v čase [1].

2.2.6 Heteroscedastic linear discriminant analysis (Heteroskedastikná lineárna analýza diskriminantu) – HLDA



Obrázok 2: Ukážka prevodu príznakov pre lepšie spracovanie [3].

Tento prístup vychádza z myšlienky, že trénovacie dáta môžu byť korelované. Preto je ich popis Gaussovými krivkami s diagonálnymi kovariančnými maticami nevhodný a pre popis je nutné použiť väčší počet Gaussových kriviek. Tomu sa dá zabrániť, ak sú dáta najprv vhodne lineárne

transformované. Tento proces sa nazýva dekorelácia. Behom dekorelácie sa koeficienty vhodne transformujú tak, aby bola možná ich kompaktná reprezentácia Gaussovými krivkami s osami rovnobežnými s osami priestoru. Tým sa výrazne zjednodušia pamäťové nároky na reprezentáciu a s tým spojené výpočtové nároky [1].

2.3 Spracovanie na úrovni modelu

2.3.1 Speaker model synthesis (Syntéza modelu rečníka) – SMS

Problém reálnych trénovacích dát je, že je zväčša nemožné zabezpečiť nahrávku pre každý typ mikrofónu v každom prostredí, na každom dostupnom komunikačnom kanáli, pre každú osobu, v každom emotívnom rozpoložení. Dokonca sa ukazuje, že reálne použitia sú často obmedzené na jedinú krátku nahrávku. Preto bol navrhnutý systém SMS, ktorý je založený na myšlienke, že existuje vzťah medzi jednotlivými modelmi pre rôzne kanály. Preto je najprv vycvičený univerzálny model na všetkých nahrávkach, nehľadiac na rečníka a kanál. Ten je potom adaptovaný na nahrávky z jednotlivých kanálov. Následné vzájomné posunutie medzi dvoma modelmi pre konkrétne kanály určuje vzťah medzi zvolenými modelmi. Potom sa pri trénovaní modelu konkrétneho rečníka zisťuje, ku ktorému z natrénovaných jestvujúcich modelov kanálov prináleží. Vybraný model determinuje predpokladaný kanál, kde nahrávka vznikla. Následne sa tento model zadaptuje na trénovacie dáta daného rečníka, čím vzniká model rečníka vo vybranom kanáli.

Keď je systém postavený pred rozhodnutie, či príchodzie testovacie dáta patria natrénovanému rečníkovi, najprv systém stanoví, z akého kanála je nahrávka nahraná. Tento proces prebieha rovnako ako pri výbere kanála pri trénovaní rečníka. Potom sa podľa vzťahu medzi modelmi pre jednotlivé kanály vyberie posunutie z modelu kanála, ktorý slúžil na zadaptovanie na model konkrétneho rečníka, na model nahrávky. Týmto posunutím sa model rečníka zadaptuje na model kanálu príchodzích dát. Rozhodnutie, či príchodzie dáta patria konkrétnemu rečníkovi, je stanovené podľa pomeru vierohodností univerzálneho modelu kanálu príchodzích správ a adaptovaného modelu rečníka na kanál príchodzích správ [1].

2.3.2 Feature mapping (mapovanie príznakov) – FM

Feature mapping vznikol dôsledkom SMS a správnejšie by mal byť zaradený do katégorie o spracovaní príznakov. Pre zachovanie logickej náväznosti je však uvedený až tu. Vychádza

z myšlienky, že namiesto adaptácie modelu rečníka medzi kanálmi je vhodnejšie adaptovať už príznaky tak, aby vznikli príznaky nezávislé na kanáli. Postup tvorby modelu je nasledovný.

Najprv sa vytvorí univerzálny model na všetkých trénovacích dátach. Ten je potom zadaptovaný na model konkrétneho kanálu. Relevantná informácia je však na rozdiel od SMS prístupu ukrytá v adaptácii modelu kanálu na univerzálny model a nie v uchovaní posunu medzi dvoma konkrétnymi modelmi rôznych kanálov. Preto je pre každý kanál zaznamenaný len jeden posun, tj. posun na univerzálny model namiesto viacerých posunov na všetky ostatné kanály.

Proces tvorby modelov najprv vezme všetky nahrávky z jednotlivých kanálov a pre každý kanál natrénuje vlastný model. Následne je zo všetkých nahrávok vytvorený univerzálny model. V prípade príchodu trénovacích dát konkrétneho rečníka je najprv vybraný model kanála, z ktorého sú trénovacie dáta vytvorené. Potom sú dáta adaptované podľa adaptácie vybraného modelu kanálu na univerzálny model. Novovzniknuté príznaky sú týmto posunuté pod univerzálny model, tzn. nezávislé na kanáli.

V praxi však často chýbajú informácie o zdrojovom kanáli pre vytvorenie jednotlivých modelov kanálov. Tento fakt sa nahrádza vytvorením konštantného počtu kvázi-kanálov na všetkých dostupných trénovacích dátach. Tie sú potom pomocou iteratívnych algoritmov natrénované tak, aby čo najlepšie popisovali celú množinu trénovacích dát. Tieto modely sa ďalej úspešne používajú ako horeuvedené modely kanálov [1].

2.3.3 Eigenchannel adaptation (Adaptácia na vlastný kanál) – EA

Ako FM by mala byť aj EA uvedená v kategórii venujúcej sa spracovaniu príznakov, ale pre lepšiu následnosť je uvedená až tu.

S použitím FM sa objavili problémy, ktoré vyžadovali robustnejší prístup k príznakom. Ako hlavná nevýhoda FM sa ukázala nemožnosť vytvoriť model pre každú kombináciu variability v prenosovom médiu, kvality použitého mikrofónu, prostredia a emotívneho rozpoloženia rečníka. Ako nevýhoda sa tiež ukázalo, že natrénovený model pre danú kombináciu vplyvov zostal fixný a neumožňoval ďalší posun. To sa ukázalo ako nevhodné v situáciách, kedy sa menil napríklad len okolitý hluk. V skratke, nebolo možné vytvoriť pre všetky kombinácie vplyvov konkrétny model tak, aby sa prípadné prichádzajúce nahrávky dali adaptovať na nezávislé na kanáli.

Riešenie týchto problémov prináša Eigenchannel adaptation. Čerpá z predpokladu, že v mnohorozmernom priestore modelov sú smery, ktoré determinujú variabilitu medzi rečníkmi, medzi nahrávkami od jedného rečníka, atď. Objavením vektorov týchto posunov je možné model rečníka efektívne zadaptovať na testovaciu nahrávku, aj keď vznikla za diametrálne odlišných okolností.

Pri použití EA je trénovanie univerzálneho modelu rovnaké ako v prípade predchádzajúcich metód. Pre jednotlivé nahrávky jedného rečníka sa trénujú modely, ktoré sa premietnu do podpriestoru tak, aby sa dali kompaktne uložiť. Potom vzájomná poloha týchto parciálnych modelov určuje smery vnútrorečníckej variability. Pokiaľ pre všetkých rečníkov vykazujú modely variabilitu na kanáloch v rovnakých smeroch, vybrané smery sú označené ako smery s veľkou vnútrorečníckou variabilitou.

Po natrénovaní univerzálneho modelu sa model adaptuje na konkrétneho rečníka. Rozdiel oproti FM vyvstáva pri príchode testovacích dát. Modely jednotlivých rečníkov sú v smeroch veľkej vnútrorečníckej variability zadaptované na trénovacie dáta tak, aby sa maximalizovala ich vierohodnosť. V rovnakých smeroch je obdobne zadaptovaný na príchodzie dáta aj univerzálny model. Na rozdiel od FM sa posunutie v danom smere vykonáva o variabilnú dĺžku tak, aby zadaptovaný model čo najlepšie popisoval skúmané dáta. Po nájdení posunutia pre model jednotlivého rečníka a univerzálneho modelu sa pristupuje ku bežnej klasifikácii. Sú porovnané vierohodnosti optimálne zadaptovaného modelu rečníka a zadaptovaného univerzálneho modelu. Ich pomer určuje mieru, či testovacie dáta sú, alebo nie sú od vybraného rečníka [1].

2.3.4 Joint factor analysis (Analýza spoločných príznakov) – JFA

JFA vychádza z myšlienky, že množiny smerov s veľkou variabilitou môžu byť redukované bez výraznej straty na obecnosti. V reálnych nasadeniach sa redukujú priestory z rádu stotisícov na stovky. Redukciou smerov sa dosahuje zrobustnejšenie klasifikátora. Výsledkom JFA je potom sada vektorov popisujúcich variabilitu medzi rečníkmi a variabilitu medzi kanálmi. Následne je každý rečník reprezentovaný ako adaptácia univerzálneho modelu pomocou dvoch vektorov. Jeden popisuje variabilitu konkrétneho rečníka a druhý variabilitu prislúchajúceho kanálu [4].

2.3.5 Kompaktná reprezentácia rečníka – iVectors

Predpokladom iVectors prístupu je, že konkrétny rečník môže byť reprezentovaný nie dvoma vektormi ako pri použití JFA, ale iba jedným. Ten reprezentuje adaptáciu univerzálneho modelu pozadia na rečníka len v smeroch s najväčšou variabilitou. Pri reprezentácii rečníka pomocou iVectors dochádza k redukcii reprezentácie rečníka zo státisícového vektora na vektor rádu stoviek hodnôt. Toto výrazné zredukovanie umožňuje nielen uchovávať hlasový podpis rečníka s minimálnou pamäťovou náročnosťou, ale umožňuje aj porovnávanie dvojíc v extrémne krátkom čase. To otvára priestor pre použitie systémov nielen pri verifikácii, kedy je systému ponúknutá dvojica nahrávok o ktorých sa má rozhodnúť, či sú od rovnakého autora, ale aj pri identifikácii rečníka. Pokiaľ sa vytvorí

databáza veľkého množstva rečníkov, systém je po extrakcii hlasového odtlačku schopný nájsť najväčšiu zhodu v okamihu [4].

Pre vyhodnocovanie zhody sa využíva PLDA (Probabilistic Linear Discriminant Analysis).

2.4 Spracovanie na úrovni skóre

2.4.1 Z-Norm

Veľká variabilita v kvalite natrénovaných modelov, zapríčinená napríklad množstvom trénovacích dát, spôsobuje rozdielne rozsahy výsledných skóre testovacej nahrávky pre jednotlivé modely. Tento stav vzniká napríklad v prípadoch, kedy je na trénovanie jednotlivých modelov rečníkov dostupný príliš veľký alebo príliš malý počet trénovacích dát. Aby bolo možné určiť spoločný prah pre celý klasifikátor, tzn. pre všetky modely rečníkov, je nevyhnutné tieto skóre normalizovať. Z-Norm normalizuje skóre tak, že každý model oskóruje množinou nahrávok takou, ktorá neobsahuje nahrávky od rečníka daného modelu. Potom sa vytvorené rozloženie normalizuje, a to podelením svojou smerodajnou odchýlkou a odčítaním strednej hodnoty. Tým sa normalizuje skóre pre nevalídne nahrávky do prekrývajúceho sa rozloženia a adekvátne k tomu sa upraví aj skóre pre validné nahrávky od daného rečníka. To umožňuje nastaviť jednotný prah pre všetky modely [1].

2.4.2 T-Norm

T-Norm vychádza z myšlienky, že vo všeobecnosti existuje skupina testovacích nahrávok, ktorá produkuje lepšie skóre pre všetky modely rečníkov. Naopak, existuje aj skupina, ktorá vo všeobecnosti produkuje na všetkých modeloch horšie výsledky. Samotná normalizácia je konštruovaná tak, že sa natrénujú modely rečníkov, ktorí zaručene nie sú medzi rečníkmi natrénovanými v systéme. Potom sa pre každú testovaciu nahrávku vypočíta skóre pre sadu modelov nevalídnych rečníkov a model konkrétneho rečníka. Vypočítané rozdelenie skóre sa normalizuje ako v predchádzajúcej kapitole a potom sa určí výsledné ohodnotenie pre danú nahrávku na daný model rečníka [1].

2.4.3 ZT-Norm a TZ-Norm

ZT-Norm je normalizácia, kedy je najprv aplikovaná Z-Norm a potom T-Norm. V systémoch sa využíva aj jej obrátená forma, zapisovaná ako TZ-Norm, kedy sa normalizácie aplikujú v opačnom poradí. ZT-Norm produkuje najlepšie výsledky z celej štvorice normalizácií [1].

3 Speech@FIT

Vývojová skupina Speech@FIT pracujúca na Ústave počítačovej grafiky a multimédií FIT VUT sa už niekoľko rokov venuje spracovaniu rečových záznamov. Svoje kvality dokázala už niekoľkokrát na medzinárodnej úrovni a spolupracuje na vývoji aplikácie Brno Speech Core pre výstavbu rýchlych rečových rozpoznávačov [5].

3.1 Brno Speech Core - BSC

Brno Speech Core zahŕňa implementáciu množstva algoritmov využívaných pri spracovaní reči. Rovnako aj algoritmy používané pre prácu so súbormi, prístup k mikrofónu, dávkové spracovanie súborov, extrakciu a transformáciu príznakov, klasifikáciu, dekódovanie, fonémové rozpoznávače naviazané na rečové rozpoznávače, vyhľadávanie kľúčových slov a jazykové rozpoznávače. Z pohľadu tejto práce sa najdôležitejší komponent venuje identifikácii rečníka.

Jadro BSC (BSCORE) je aplikácia naprogramovaná v jazyku C++ a poskytuje svoje služby pomocou aplikačného rozhrania s názvom Brno Speech Application Interface (ďalej len BSAPI). To je objektovo orientované a pre každý algoritmus poskytuje vlastnú triedu. Tieto triedy sú prístupné pomocou rozhrania, pričom každá z nich ho má vlastné, implementované ako plne virtuálne [5].

Od verzie 1.0.42 už obsahuje podporu pre kompaktnú reprezentáciu rečníka.

Čas pri trénovaní nového modelu sa pri použití BSC pohybuje približne na rýchlosti 155-krát vyššej ako reálny čas. To znamená, že 155 minút záznamu sa spracúva približne minútu strojového času (testované na 3 GHz Intel CPU, 64 bit Linux) [6]. Na tomto príklade je vidieť, že extrakcia hlasového odtlačku je pomerne náročný proces, avšak toto spracovanie je nutné len raz pre jednu nahrávku. Skórovanie jednotlivých odtlačkov je následne výrazne rýchlejšie a umožňuje porovnávať obrovské zoznamy v krátkom čase. Toto spracovanie umožňuje už nielen verifikáciu rečníka, ale aj identifikáciu neznámeho rečníka z veľkého množstva už extrahovaných odtlačkov.

4 Tvorba pluginov pre Skype klienta

4.1 Program Skype

Skype je voľne dostupný a hojne rozšírený program primárne určený na hlasovú a video komunikáciu na Internete. Svojou cenou, kvalitou a nenáročnosťou na prenosové pásmo sa rýchlo po uvoľnení masívne rozšíril a už tri roky po vydaní dosiahol 100 miliónov používateľov. Jeho rast sa nezastavil a v treťom kvartáli roku 2009 evidoval Skype 521 miliónov účtov s 27,7 miliardami pretelefonovaných minút vo vlastnej sieti a 3,1 miliardami minút mimo vlastnú. Vzhľadom na medzinárodný charakter s 13% podielom na medzinárodnom hlasovom trhu sa v rebríčku umiestnil na prvom mieste ako najväčší medzinárodný poskytovateľ hlasových služieb [7].

Napriek tomu, že Skype sieť nemá zverejnené zdrojové kódy, ponúka pripojenie do svojej siete pomocou rozhrania prítomného na každom klientovi. To je jednoducho prístupné a umožňuje pomocou klienta využiť možnosti Skype siete. To otvára cestu vývojárom vytvárať vlastné aplikácie, ktoré môžu dopĺňať a rozširovať funkcionalitu pomocou Skype protokolu[2]. Túto snahu Skype preferuje a snaží sa vývojárov podporovať a pomáhať im distribuovať rozšírenia v rámci svojich možností na vlastných stránkach.

Pod pojmom „Skype“ bude v ďalšom texte myslený oficiálny klient pre operačné systémy MAC OS, MS Windows a Unixové OS. Hlavným zdrojom informácií v tejto kapitole je verejná referenčná príručka SkypeAPI [8].

4.2 SkypeAPI

SkypeAPI je rozhranie prístupné na oficiálnom klientovi. Umožňuje externým aplikáciam a zariadeniam používať funkcie Skype rozhrania a implementovať dodatočné alebo vylepšiť stávajúce vlastnosti klienta. Pretože je Skype dostupný pre viaceré operačné systémy, je aj jeho API rozdelené na dve vrstvy podľa závislosti na operačnom systéme. Prvá časť, nazývaná aj komunikačná vrstva, je závislá na operačnom systéme a ponúka mechanizmus pre externé aplikácie na komunikáciu s klientom. Druhá časť, nazývaná protokolová vrstva, je nezávislá na operačnom systéme a zahŕňa jazyk príkazov pre komunikáciu s klientom.

2 V dobe písania tejto práce Skype chystá uvoľnenie SDK s názvom SkypeKIT pre priamy prístup do svojej siete.

4.2.1 Komunikačná vrstva

Komunikačná vrstva má zo svojej podstaty rozličnú implementáciu na jednotlivých operačných systémoch. Jej hlavná funkcia je vytvoriť transportný nosič medzi klientom a externou aplikáciou. Po ňom sa následne prenášajú príkazy protokolovej vrstvy Skype a prijímajú odpovede.

V operačnom systéme Windows je komunikačná vrstva vytvorená s použitím WinAPI správ. Pre vytvorenie spojenia je nutné zaregistrovať dve správy v systéme, pomocou funkcie `RegisterWindowMessage` prítomnej vo WinAPI. Názvy týchto správ sú pevne definované, menovite: `SkypeControlAPIDiscover` a `SkypeControlAPIAttach`. Pre otvorenie komunikácie je nutné odoslať prvú správu s parametrom `wParam`, obsahujúcim ukazovateľ na seba. Skype odpovedá pomocou druhej správy s adresátom `wParam` z predchádzajúcej. Podľa návratovej hodnoty Skype oznamuje svoj stav. Možné hodnoty sú:

- 0 - Modul je pripojený a ukazovateľ na Skype je odoslaný ako parameter.
- 1 - Skype spracúva žiadosť o pripojenie, predložil žiadosť používateľovi a čaká na rozhodnutie.
- 2 - Používateľ explicitne zakázal prístup ku Skype.
- 3 - SkypeAPI nie je prístupné, napríklad z dôvodu, že žiaden z používateľov nie je prihlásený.

Pokiaľ je API znova použiteľné, vyšle správu všetkým oknám v systéme s hodnotou `0x8001`. Jednotlivé správy sú očakávané vo formáte UTF-8, zakončené bežným ukončovacím znakom, tj. nulou. Jednotlivé správy teda nemôžu byť zreťazené, ale dĺžka jednotlivej správy nie je obmedzená. Pokiaľ API spracúva príkaz viacej ako jednu sekundu, spojenie preruší. Preto je dobré pamätať na vhodné kontrolovanie nadviazania spojenia behom plánovaného dlhotrvajúceho spojenia.

Rovnako musia byť obe aplikácie, tzn. Skype klient a zásuvný modul, spustené s rovnakými právami. Pokiaľ je jedna z aplikácií spustená s administrátorskými právami a druhá s používateľskými, spojenie sa nepodarí vytvoriť.

Vzhľadom na plánovanú implementáciu zásuvného modulu v operačnom systéme Windows, nezabieha táto práca do detailov implementácie v iných operačných systémoch.

4.2.2 Protokolová vrstva

Protokolová vrstva definuje jazyk možných príkazov Skype a jeho možné odpovede. Vzhľadom na paralelný vývoj jednotlivých klientov pre rôzne OS sú najnovšie príkazy, tj. posledná verzia Skype protokolu, dostupné len na Skype klientovi pre Windows. Preto možno prenesením zásuvného modulu na inú platformu prísť o časť funkcionality a to najmä pri využití príkazov z poslednej verzie.

Jednotlivé možné príkazy sú synchrónne alebo asynchrónne. V prvej skupine je po príkaze Skype očakávaná odpoveď. Do tejto skupiny patrí väčšina príkazov, napríklad príkazy zisťujúce dostupnosť služieb u kontaktov alebo overovanie aktívnosti spojenia. Druhý prípad zahŕňa zvyšné správy, ktoré Skype posiela bez žiadosti. Tie vznikajú pri mimoriadnych situáciách, ako je napríklad zmena stavu niektorého z kontaktov.

Skype podporuje identifikáciu jednotlivých príkazov a odpovedí naň. Pokiaľ je príkaz predchádzaný mriežkou s číselným označením, odpoveď naň je rovnakým spôsobom a číslom označená.

Protokolová vrstva ponúka príkazy na správu:

- audio alebo video rozhovorov a konferencií,
- SMS správ a textovej komunikácie v rámci Skype siete,
- prenosu súborov,
- kontaktov a ich nastavení,
- vzhľadu okna Skype a jeho nastavení.

V rozhraní existujú objekty, ktoré abstrahujú rôzne logické štruktúry. Menovite Skype ponúka abstrakciu pre používateľa, profil, telefonát, správu, konverzáciu, účastníka konverzácie, správu konverzácie, hlasovú správu, SMS, aplikáciu, skupinu a presun súborov. Inštancie týchto objektov obsahujú vlastné dátové štruktúry, ktoré sú prístupné externým aplikáciám. Ich úprava a úprava všeobecných nastavení klienta je možná pomocou príkazov `GET`, `SET` a `ALTER` so zrejmým použitím.

Chybové hlásenia Skype vyhodnocuje a chybu aj s jednoznačným číselným označením vracia zásuvnému modulu.

Je zaujímavé, že po povolení prístupu zásuvného modulu má modul prístup do všetkých vrstiev Skype klienta. Z toho plynie, že napríklad aj jednoduchá aplikácia pracujúca s čítaním správ má umožnený prístup k všetkým dostupným častiam. Nasleduje ukážka komunikácie pomocou SkypeAPI.

```
> CONNSTATUS ONLINE
> CURRENTUSERHANDLE user1
> USERSTATUS ONLINE
> CONTACTS FOCUSED user2
> WINDOWSTATE MINIMIZED
> WINDOWSTATE NORMAL
> CONTACTS FOCUSED echo123
> CALL 254317 STATUS ROUTING
> CALL 254317 STATUS RINGING
> CALL 254317 STATUS INPROGRESS
> CALL 254317 DURATION 0
> CALL 254317 VIDEO_RECEIVE_STATUS AVAILABLE
> CALL 254317 DURATION 1
> CALL 254317 DURATION 2
> CALL 254317 STATUS FINISHED
> CONTACTS FOCUSED user2
> MESSAGE 254441 STATUS SENDING
> GROUP 298 NROFUSERS 10
> CHAT #user1/$user2;abfa74dc0c3cc917 POSTERS user1
> GROUP 298 NROFUSERS 10
> CHAT #user1/$user2;abfa74dc0c3cc917 ACTIVITY_TIMESTAMP 1304703242
> CHAT #user1/$user2;abfa74dc0c3cc917 FRIENDLYNAME Sir Inkognito | spam :]
> MESSAGE 254441 STATUS SENT
> CHATMEMBER 254127 IS_ACTIVE TRUE
> CHAT #user1/$user2;abfa74dc0c3cc917 ACTIVEMEMBERS user2 user1
> MESSAGE 254473 STATUS RECEIVED
> CHAT #user1/$user2;abfa74dc0c3cc917 POSTERS user2 user1
> CHAT #user1/$user2;abfa74dc0c3cc917 ACTIVITY_TIMESTAMP 1304703249
```

Obrázok 3: Ukážka komunikácie pomocou SkypeAPI.

5 Návrh systému

Behom dôkladného preskúmania dostupných zásuvných modulov pre klienta Skype som nenašiel žiaden modul, ktorý by spracúvaval reč. Niektoré sa vydávajú za vedecké analyzátory reči s cieľom odhaliť, či hovoriaci klame, ale s ohľadom na nadobudnuté vedomosti v oblasti spracovania reči si dovolím tvrdiť, že ich popis je zavádzajúci a testovanie ukázalo, že pravdepodobne vyhodnocujú len intenzitu zvuku. Žiaden z dostupných zásuvných modulov sa však nevenuje verifikácii hovoriaceho analýzou reči, rovnako ani jeho identifikácii.

Pri návrhu som chcel zachovať systém čo najjednoduchší. To hlavne preto, aby bolo jeho použitie čo najprirodzenejšie a mohol ho tak obsluhovať aj používateľ, ktorý má len elementárne informatické vedomosti. Preto som každý možný ovládací prvok prehodnocoval, či je jeho použitie nutné a či sa nedá nahradiť prednastaveným správaním sa. V súlade s uvedeným som vypracoval niekoľko náležitostí, ktoré musí zásuvný modul spĺňať.

Zásuvný modul musí vyhodnocovať nahrávku priebežne. Z používateľského hľadiska je informácia, či bol partner, s ktorým používateľ komunikoval, málo zaujímavá, pokiaľ ju modul poskytne až po ukončení hovoru. Alternatíva priebežného upresňovania skóre je užitočnejšia.

Zásuvný modul musí svoje vyhodnotenia poskytovať zrozumiteľne bežnému používateľovi. To znamená, že skóre, ktoré bežne vracia Brno Speech Core, musí byť prevedené do slovného popisu alebo obmedzenej škály. Napríklad na percentuálnu mieru zhody.

Modul musí poskytovať rozširovanie svojej databázy priamo z hovorov. Používateľ musí mať možnosť pridať do svojej databázy vyextrahovaný hlasový odtlačok osoby, o ktorej vie, že je verifikovaná.

Na druhú stranu, musí modul poskytovať rozhranie pre správu odtlačkov. Pokiaľ používateľ zanesie do databázy nekvalitný hlasový podpis partnera, ten by mohol spôsobovať chybné výsledky a to tak, že by napríklad spôsoboval zhodu s iným rečníkom, ale hlavne by znižoval mieru zhody s valídnym rečníkom a tým by znižoval funkčnosť celého systému.

Zásuvný modul by mal poskytovať informácie o svojom stave. Vzhľadom na predpokladanú časovú náročnosť výpočtu je vhodné, aby používateľ vedel, v akej fáze je modul. Najmä na slabších počítačoch by mohol používateľ vyhodnotiť dlhú časovú odozvu ako nefunkčnosť.

Okrem rutinného povolenia prístupu ku klientu Skype nemôže zásuvný modul vyžadovať ďalšie dodatočné nastavovanie.

Všetky horeuvedené podmienky som sa pokúsil zakomponovať do návrhu a v rámci diplomovej práce naimplementovať, otestovať a zhodnotiť.

5.1 Predpokladaná úspešnosť

Pri kalkulácii predpokladanej úspešnosti som využil predpripravený príklad distribuovaný s BSC, ktorý umožňuje dávkovo vyextrahovať hlasové odtlačky z nahrávok a následne vypočítať ich vzájomné zhody. Ako testovacie nahrávky som použil nahrávky, ktoré som ukladal behom bežnej prevádzky Skype klienta s rôznymi ľuďmi. Následne som povyberal nahrávky, ktoré boli dlhšie ako dve minúty. Celkovo som mal k dispozícií 23 nahrávok s dĺžkou väčšou ako dve minúty od 11-tich rôznych používateľov. Tie som rozdelil na 10, 30, 60 a 120 sekundové časti. Pre potreby testovania som zachoval vždy len prvú časť po rozdelení. Výsledných 92 nahrávok bolo rozdelených do štyroch skupín podľa dĺžky záznamu. Pre každú nahrávku som vyextrahoval hlasový odtlačok a vyhodnotil jeho zhodu s ostatnými v skupine. Mieru jednotlivých zhôd zhŕňajú nasledujúce tabuľky.

Rozhodovací prah som stanovil tak, aby bol čo najväčší, ale zároveň správne detekoval všetky zhody medzi nahrávkami od rovnakého používateľa. Inak povedané, aby bola pravdepodobnosť nenájdenia korektného rečníka nulová. Po stanovení prahu pre každú tabuľku osobitne sú v tabuľkách vyznačené žltou farbou miesta, kde došlo ku korektnému zamietnutiu identity. Zelenou sú označené miesta, kde došlo k správnemu nájdeniu rečníka a červenou sú označené miesta, kde systém nesprávne označil rečníka za zhodného.

Z priestorových dôvodov sú tabuľky otočené o 90°.

Tabuľka 1: Miery zhôd pre 10 sekundové nahrávky.

	usr1	usr1	usr2	usr2	usr3	usr3	usr4	usr4	usr5	usr5	usr5	usr6	usr6	usr7	usr7	usr8	usr8	usr9	usr9	usr10	usr10	usr11	usr11
usr1	90	-46	-97	-185	-95	-103	-105	-133	-56	-56	-51	-85	-71	-60	-78	-95	-128	-92	-99	-192	-231	-109	-59
usr1	-46	123	-132	-191	-72	-134	-153	-153	-81	-75	-59	-127	-92	-134	-79	-119	-138	-110	-138	-176	-271	-150	-147
usr2	-97	-132	113	-88	-128	-152	-95	-79	-106	-105	-118	-69	-85	-17	-34	-97	-88	-81	-48	-154	-138	-24	-70
usr2	-185	-191	-88	129	-181	-214	-77	-151	-115	-124	-190	-106	-75	-139	-110	-159	-131	-178	-77	-130	-60	-90	-150
usr3	-95	-72	-128	-181	118	-18	-138	-165	-114	-131	-124	-137	-142	-123	-75	-109	-124	-119	-125	-216	-230	-154	-162
usr3	-103	-134	-152	-214	-18	117	-151	-203	-92	-180	-134	-164	-142	-92	-42	-141	-137	-113	-122	-321	-250	-144	-120
usr4	-105	-153	-95	-77	-138	-151	105	-4	-91	-83	-132	-66	-102	-103	-109	-42	-67	-100	-35	-88	-42	-66	-68
usr4	-133	-153	-79	-151	-165	-203	-4	115	-112	-77	-122	-90	-148	-104	-139	-36	-62	-90	-51	-62	-106	-57	-100
usr5	-56	-81	-106	-115	-114	-92	-91	-112	96	-39	-25	-69	-60	-102	-72	-66	-78	-74	-78	-156	-149	-88	-67
usr5	-56	-75	-105	-124	-131	-180	-83	-77	-39	97	-34	-67	-76	-112	-94	-67	-98	-63	-72	-74	-165	-107	-96
usr5	-51	-59	-118	-190	-124	-134	-132	-122	-25	-34	95	-102	-101	-114	-108	-80	-96	-101	-115	-171	-199	-136	-96
usr6	-85	-127	-69	-106	-137	-164	-66	-90	-69	-67	-102	111	-3	-108	-103	-85	-94	-70	-54	-130	-155	-42	-90
usr6	-71	-92	-85	-75	-142	-142	-102	-148	-60	-76	-101	-3	110	-130	-99	-128	-118	-91	-75	-209	-166	-47	-84
usr7	-60	-134	-17	-139	-123	-92	-103	-104	-102	-112	-114	-108	-130	117	-29	-108	-99	-100	-52	-179	-177	-91	-66
usr7	-78	-79	-34	-110	-75	-42	-109	-139	-72	-94	-108	-103	-99	-29	123	-119	-114	-73	-69	-218	-190	-85	-85
usr8	-95	-119	-97	-159	-109	-141	-42	-36	-66	-67	-80	-85	-128	-108	-119	108	-33	-92	-62	-71	-114	-78	-59
usr8	-128	-138	-88	-131	-124	-137	-67	-62	-78	-98	-96	-94	-118	-99	-114	-33	103	-106	-69	-126	-122	-80	-91
usr9	-92	-110	-81	-178	-119	-113	-100	-90	-74	-63	-101	-70	-91	-100	-73	-92	-106	107	-49	-121	-198	-46	-88
usr9	-99	-138	-48	-77	-125	-122	-35	-51	-78	-72	-115	-54	-75	-52	-69	-62	-69	-49	99	-114	-91	-69	-55
usr10	-192	-176	-154	-130	-216	-321	-88	-62	-156	-74	-171	-130	-209	-179	-218	-71	-126	-121	-114	120	-66	-145	-181
usr10	-231	-271	-138	-60	-230	-250	-42	-106	-149	-165	-199	-155	-166	-177	-190	-114	-122	-198	-91	-66	126	-133	-152
usr11	-109	-150	-24	-90	-154	-144	-66	-57	-88	-107	-136	-42	-47	-91	-85	-78	-80	-46	-69	-145	-133	120	-48
usr11	-59	-147	-70	-150	-162	-120	-68	-100	-67	-96	-96	-90	-84	-66	-85	-59	-91	-88	-55	-181	-152	-48	119

Tabuľka 2: Miery zhôd pre 30 sekundové nahrávky.

	usr1	usr1	usr2	usr2	usr3	usr3	usr4	usr4	usr5	usr5	usr5	usr6	usr6	usr7	usr7	usr8	usr8	usr9	usr9	usr10	usr10	usr11	usr11
usr1	84	4	-114	-124	-55	-89	-67	-88	-41	-71	-41	-76	-73	-59	-58	-36	-75	-78	-84	-125	-124	-103	-88
usr1	4	86	-110	-141	-35	-56	-80	-118	-33	-56	-45	-67	-70	-54	-43	-45	-70	-62	-63	-119	-123	-62	-68
usr2	-114	-110	125	-60	-114	-139	-116	-136	-101	-134	-135	-46	-42	-56	-82	-97	-104	-66	-77	-120	-109	-80	-80
usr2	-124	-141	-60	113	-124	-172	-72	-89	-109	-149	-166	-69	-32	-92	-78	-58	-82	-41	-59	-62	-77	-123	-104
usr3	-55	-35	-114	-124	97	-32	-92	-133	-81	-80	-77	-82	-55	-87	-83	-83	-90	-100	-104	-112	-128	-91	-127
usr3	-89	-56	-139	-172	-32	118	-87	-143	-79	-62	-81	-95	-84	-87	-79	-61	-114	-77	-92	-165	-139	-78	-84
usr4	-67	-80	-116	-72	-92	-87	97	24	-51	-66	-72	-33	-34	-43	-70	8	13	-22	-17	-19	-27	-47	-56
usr4	-88	-118	-136	-89	-133	-143	24	105	-72	-87	-83	-49	-57	-44	-88	-17	-19	-48	-31	-33	-43	-82	-83
usr5	-41	-33	-101	-109	-81	-79	-51	-72	86	-23	-2	-54	-54	-66	-61	-37	-64	-64	-46	-82	-69	-61	-66
usr5	-71	-56	-134	-149	-80	-62	-66	-87	-23	96	-15	-40	-46	-82	-78	-38	-61	-83	-48	-108	-119	-50	-41
usr5	-41	-45	-135	-166	-77	-81	-72	-83	-2	-15	89	-84	-76	-97	-105	-59	-82	-83	-77	-128	-88	-82	-75
usr6	-76	-67	-46	-69	-82	-95	-33	-49	-54	-40	-84	98	19	-42	-51	-14	-38	-25	-10	-71	-75	-40	-51
usr6	-73	-70	-42	-32	-55	-84	-34	-57	-54	-46	-76	19	91	-32	-54	-12	-24	-13	-18	-65	-57	-53	-28
usr7	-59	-54	-56	-92	-87	-87	-43	-44	-66	-82	-97	-42	-32	103	-2	-44	-33	-45	-46	-65	-90	-59	-75
usr7	-58	-43	-82	-78	-83	-79	-70	-88	-61	-78	-105	-51	-54	-2	111	-65	-74	-70	-44	-130	-126	-61	-63
usr8	-36	-45	-97	-58	-83	-61	8	-17	-37	-38	-59	-14	-12	-44	-65	80	25	-10	-2	-16	-40	-60	-30
usr8	-75	-70	-104	-82	-90	-114	13	-19	-64	-61	-82	-38	-24	-33	-74	25	99	-19	-24	-14	-49	-74	-63
usr9	-78	-62	-66	-41	-100	-77	-22	-48	-64	-83	-83	-25	-13	-45	-70	-10	-19	88	27	-53	-33	-75	-44
usr9	-84	-63	-77	-59	-104	-92	-17	-31	-46	-48	-77	-10	-18	-46	-44	-2	-24	27	83	-34	-40	-66	-27
usr10	-125	-119	-120	-62	-112	-165	-19	-33	-82	-108	-128	-71	-65	-65	-130	-16	-14	-53	-34	100	4	-80	-87
usr10	-124	-123	-109	-77	-128	-139	-27	-43	-69	-119	-88	-75	-57	-90	-126	-40	-49	-33	-40	4	103	-78	-91
usr11	-103	-62	-80	-123	-91	-78	-47	-82	-61	-50	-82	-40	-53	-59	-61	-60	-74	-75	-66	-80	-78	98	-22
usr11	-88	-68	-80	-104	-127	-84	-56	-83	-66	-41	-75	-51	-28	-75	-63	-30	-63	-44	-27	-87	-91	-22	98

Tabuľka 3: Miery zhôd pre 60 sekundové nahrávky.

	usr1	usr1	usr2	usr2	usr3	usr3	usr4	usr4	usr5	usr5	usr5	usr6	usr6	usr7	usr7	usr8	usr8	usr9	usr9	usr10	usr10	usr11	usr11
usr1	68	29	-49	-58	-5	-39	-41	-69	-14	-20	-34	-29	-25	-39	-29	-22	-42	-53	-46	-66	-90	-43	-57
usr1	29	78	-61	-66	-8	-23	-49	-75	-23	-17	-37	-38	-43	-46	-26	-32	-46	-41	-47	-85	-107	-40	-47
usr2	-49	-61	97	8	-84	-69	-79	-92	-63	-78	-95	-23	-7	-44	-73	-54	-61	-63	-54	-82	-97	-36	-52
usr2	-58	-66	8	96	-80	-85	-43	-60	-64	-62	-108	-20	-11	-44	-45	-14	-34	-23	-12	-52	-66	-37	-38
usr3	-5	-8	-84	-80	88	-3	-77	-90	-42	-37	-59	-57	-43	-72	-48	-47	-67	-71	-72	-84	-98	-71	-99
usr3	-39	-23	-69	-85	-3	94	-56	-81	-27	-19	-48	-69	-56	-65	-31	-45	-77	-75	-57	-93	-87	-48	-62
usr4	-41	-49	-79	-43	-77	-56	94	30	-40	-39	-67	-18	-28	-37	-48	13	5	-6	4	-4	-25	-26	-45
usr4	-69	-75	-92	-60	-90	-81	30	92	-51	-56	-69	-36	-41	-28	-69	1	-3	-12	-14	-7	-25	-41	-63
usr5	-14	-23	-63	-64	-42	-27	-40	-51	75	18	4	-50	-50	-62	-58	-23	-48	-70	-43	-53	-57	-25	-68
usr5	-20	-17	-78	-62	-37	-19	-39	-56	18	75	5	-37	-37	-66	-31	-28	-46	-61	-30	-59	-64	-7	-37
usr5	-34	-37	-95	-108	-59	-48	-67	-69	4	5	92	-68	-71	-81	-98	-52	-74	-91	-69	-93	-89	-50	-72
usr6	-29	-38	-23	-20	-57	-69	-18	-36	-50	-37	-68	84	40	-34	-59	-1	-27	-3	-8	-35	-56	-18	-30
usr6	-25	-43	-7	-11	-43	-56	-28	-41	-50	-37	-71	40	89	-30	-44	-7	-24	-10	-13	-51	-52	-19	-29
usr7	-39	-46	-44	-44	-72	-65	-37	-28	-62	-66	-81	-34	-30	97	-12	-27	-38	-27	-32	-41	-82	-36	-48
usr7	-29	-26	-73	-45	-48	-31	-48	-69	-58	-31	-98	-59	-44	-12	94	-40	-56	-72	-48	-93	-119	-56	-46
usr8	-22	-32	-54	-14	-47	-45	13	1	-23	-28	-52	-1	-7	-27	-40	71	29	13	9	-5	-30	-39	-18
usr8	-42	-46	-61	-34	-67	-77	5	-3	-48	-46	-74	-27	-24	-38	-56	29	92	1	-2	-7	-34	-50	-45
usr9	-53	-41	-63	-23	-71	-75	-6	-12	-70	-61	-91	-3	-10	-27	-72	13	1	86	38	-22	-36	-61	-35
usr9	-46	-47	-54	-12	-72	-57	4	-14	-43	-30	-69	-8	-13	-32	-48	9	-2	38	75	-5	-24	-28	-29
usr10	-66	-85	-82	-52	-84	-93	-4	-7	-53	-59	-93	-35	-51	-41	-93	-5	-7	-22	-5	83	24	-42	-62
usr10	-90	-107	-97	-66	-98	-87	-25	-25	-57	-64	-89	-56	-52	-82	-119	-30	-34	-36	-24	24	100	-68	-90
usr11	-43	-40	-36	-37	-71	-48	-26	-41	-25	-7	-50	-18	-19	-36	-56	-39	-50	-61	-28	-42	-68	89	10
usr11	-57	-47	-52	-38	-99	-62	-45	-63	-68	-37	-72	-30	-29	-48	-46	-18	-45	-35	-29	-62	-90	10	98

Tabuľka 4: Miery zhôd pre 120 sekundové nahrávky.

	usr1	usr1	usr2	usr2	usr3	usr3	usr4	usr4	usr5	usr5	usr5	usr6	usr6	usr7	usr7	usr8	usr8	usr9	usr9	usr10	usr10	usr11	usr11
usr1	72	35	-29	-49	-8	-11	-41	-51	-22	-12	-29	-12	-14	3	3	-27	-31	-49	-42	-80	-83	-30	-37
usr1	35	72	-51	-56	-13	-15	-38	-40	-21	-15	-29	-27	-24	0	-5	-19	-25	-38	-35	-73	-72	-29	-33
usr2	-29	-51	92	36	-73	-57	-73	-72	-56	-64	-82	-23	-7	-27	-42	-27	-46	-43	-38	-78	-83	-34	-25
usr2	-49	-56	36	95	-70	-75	-58	-58	-53	-47	-85	-18	0	-52	-45	-23	-49	-26	-18	-57	-60	-12	-22
usr3	-8	-13	-73	-70	79	22	-55	-56	-25	-19	-22	-35	-29	-47	-28	-27	-40	-57	-45	-72	-71	-60	-64
usr3	-11	-15	-57	-75	22	81	-46	-54	-6	-4	-20	-45	-42	-33	-15	-36	-50	-68	-45	-91	-76	-41	-39
usr4	-41	-38	-73	-58	-55	-46	94	41	-29	-34	-47	-13	-9	-39	-56	13	5	3	8	-2	-2	-13	-26
usr4	-51	-40	-72	-58	-56	-54	41	89	-18	-27	-36	-24	-22	-39	-62	11	10	-5	2	-3	-4	-11	-30
usr5	-22	-21	-56	-53	-25	-6	-29	-18	72	30	15	-32	-35	-37	-36	-23	-38	-57	-36	-51	-50	-18	-42
usr5	-12	-15	-64	-47	-19	-4	-34	-27	30	69	28	-20	-30	-29	-22	-28	-34	-50	-28	-57	-53	-6	-33
usr5	-29	-29	-82	-85	-22	-20	-47	-36	15	28	85	-47	-43	-52	-48	-50	-57	-86	-52	-77	-62	-37	-56
usr6	-12	-27	-23	-18	-35	-45	-13	-24	-32	-20	-47	85	35	-25	-35	4	-12	1	-5	-25	-36	-8	-10
usr6	-14	-24	-7	0	-29	-42	-9	-22	-35	-30	-43	35	75	-32	-35	-2	-16	2	5	-40	-38	-9	-7
usr7	3	0	-27	-52	-47	-33	-39	-39	-37	-29	-52	-25	-32	100	44	-27	-35	-41	-35	-84	-87	-38	-32
usr7	3	-5	-42	-45	-28	-15	-56	-62	-36	-22	-48	-35	-35	44	93	-36	-45	-51	-37	-92	-90	-38	-28
usr8	-27	-19	-27	-23	-27	-36	13	11	-23	-28	-50	4	-2	-27	-36	70	44	17	6	-3	-6	-19	-6
usr8	-31	-25	-46	-49	-40	-50	5	10	-38	-34	-57	-12	-16	-35	-45	44	86	16	7	-1	-11	-31	-25
usr9	-49	-38	-43	-26	-57	-68	3	-5	-57	-50	-86	1	2	-41	-51	17	16	77	46	-7	-17	-16	-22
usr9	-42	-35	-38	-18	-45	-45	8	2	-36	-28	-52	-5	5	-35	-37	6	7	46	76	-4	-13	2	-14
usr10	-80	-73	-78	-57	-72	-91	-2	-3	-51	-57	-77	-25	-40	-84	-92	-3	-1	-7	-4	86	46	-36	-53
usr10	-83	-72	-83	-60	-71	-76	-2	-4	-50	-53	-62	-36	-38	-87	-90	-6	-11	-17	-13	46	88	-38	-51
usr11	-30	-29	-34	-12	-60	-41	-13	-11	-18	-6	-37	-8	-9	-38	-38	-19	-31	-16	2	-36	-38	74	30
usr11	-37	-33	-25	-22	-64	-39	-26	-30	-42	-33	-56	-10	-7	-32	-28	-6	-25	-22	-14	-53	-51	30	89

Pre jednotlivé tabuľky boli stanovené prahy postupne na úroveň -89, -60, -13 a 14. Z výsledkov vyplýva, že pre bežné nahrávky s dĺžkou menšou ako dve minúty sú jednotlivé skóre ešte výrazne nepresné. Od nahrávok s dĺžkou väčšou ako dve minúty je možné považovať systém za dôveryhodný, aj keď je vidieť, že aj tam sa vyskytujú chyby. Budem vychádzať z predpokladu, že bežné telefónne rozhovory trvajú viac ako spomínané dve minúty, a preto usudzujem, že po implementácii bude systém pracovať úspešne. Ak by aj nastal prípad, kedy by rozhovor trval kratšie ako dve minúty, pre nastavený prah by mal systém tvrdiť, že rečník je neidentifikovaný.

6 Skype Speaker Verification Plugin

V rámci diplomovej práce som postavil zásuvný modul rozširujúci možnosti klienta Skype o identifikáciu rečníka. Ten by mal vedieť určiť, či osoba, s ktorou používateľ komunikuje, je naozaj tá, za ktorú sa vydáva alebo prípadne odhaliť identitu rečníka na druhej strane.

Alternatívu som v dobe písania tejto práce medzi dostupnými rozšíreniami aplikácie Skype nenašiel a preto považujem vývoj za opodstatnený s možným rozšírením medzi používateľmi Skype. V ďalšom texte je Skype Speaker Verification Plugin skrátene označovaný ako plugin.

6.1 Implementácia

Pretože je BSC napísané v jazyku C++, bolo vhodné zvoliť tento jazyk ako implementačný. Vzhľadom na dostupnosť najnovšieho klienta a predpokladanú najväčšiu základňu používateľov Skype v operačnom systéme Windows, bol plugin navrhnutý a implementovaný pre OS Windows. Plugin používa funkcie WinAPI pre komunikáciu s klientom Skype. Pre vytvorenie používateľského rozhrania som zvolil voľne dostupnú knižnicu Qt od Nokia [9]. Tá je dostupná pod licenciou LGPL, a preto umožňuje vytvorenie pluginu s možnosťou zverejnenia. V implementácii som použil knižnicu Qt verzie 4.7.1. Projekt bol vyvíjaný vo vývojovom prostredí Visual Studio 2010, dostupnom pod študentskou licenciou. Pre vývoj Qt aplikácie som použil program pre integráciu Qt do Visual Studio s názvom „Visual Studio Add-in“. Ten zjednodušil tvorbu a kompiláciu pluginu.

6.2 Popis implementácie

Vzhľadom na vysokú pamäťovú náročnosť je aplikácia viacvláknová. To preto, aby mohli jednotlivé komponenty aplikácie pracovať samostatne a navzájom sa nebrzdili. Plugin sa skladá zo štyroch hlavných vlákien. Jednotlivé vlákna sú implementované pomocou funkcií z knižnice Qt a majú rôzne polia pôsobnosti.

Hlavné vlákno sa spustí pri štarte aplikácie, zavedie používateľské rozhranie a naštartuje ostatné vlákna. Rovnako sa stará o komunikáciu medzi jednotlivými vláknami a obsluhuje ich. Druhé vlákno obsluhuje TCP server, ktorého funkcia bude vysvetlená neskôr. Tretie vlákno umožňuje a spravuje komunikáciu s klientom Skype. Posledné vlákno komunikuje s BSAPI.

Po tom, ako sa zavedie hlavné vlákno a spustí sa používateľské rozhranie, je spustený TCP server v osobitnom vlákne. TCP server je nevyhnutný na odchytávanie rozhovoru. Skype umožňuje dvojaký prístup k nahrávkam. Jedna z možností je požiadať Skype klient (ďalej len klient), aby nahrávku uložil do súboru sám. Problém pri tomto spôsobe prenosu spočíva v tom, že nahrávka je

prístupná až po skončení hovoru, čo je pre kontinuálne vyhodnocovanie nahrávky nevhodné. Druhý spôsob spočíva na kontinuálnom odosielaní rozhovoru pomocou sieťového protokolu TCP/IP. Klientovi stačí oznámiť miesto, tzn. IP adresu a port počúvajúceho servera a on následne odosiela zvolený príchodzí alebo odchodzí tok rozhovoru na daný server. Z tohto dôvodu je po spustení programu vytvorený TCP server na jednom z používateľsky dostupných portoch, kde čaká na začiatok prenosu. Jeho port je vždy pri začatí hovoru odoslaný klientovi.

Po spustení servera sa spustí ďalšie vlákno komunikujúce s klientom. Vlákno hneď pri štarte vytvorí neviditeľné okno a zaregistruje Skype definované správy. Menovite `SkypeControlAPIAttach` a `SkypeControlAPIDiscover`. Po vytvorení okna sa toto pokúsi kontaktovať klienta odoslaním správy `SkypeControlAPIDiscover`. Na túto správu reaguje klient tak, že upozorní používateľa, že sa k nemu snaží plugin pripojiť. Ak používateľ povolí prístup, klient potvrdí pluginu obdržané povolenie a začne ho informovať a načúvať.

Pokiaľ vznikne takto nastavené spojenie, plugin triedi príchodzie správy a reaguje až na správu `CALL <id> <status>`. Tá je klientom odoslaná vtedy, keď nastala zmena v stave volania. Id je číslo určujúce konkrétny hovor, pričom všetky správy prislúchajúce k danému rozhovoru majú toto id rovnaké. Jednotlivé správy, obsluhované pluginom, sú:

- `STATUS UNPLACED` - hovor nebol uskutočnený,
- `STATUS ROUTING` - hovor sa spojuje,
- `STATUS FAILED` - hovor sa nepodarilo spojiť,
- `STATUS RINGING` - hovor bol spojený, čaká sa na druhú stranu na zahájenie hovoru,
- `STATUS INPROGRESS`h - hovor začal a trvá,
- `STATUS FINISHED` - hovor skončil,
- `STATUS MISSED` - hovor ostal neprijatý,
- `STATUS REFUSED` - hovor bol odmietnutý druhou stranou,
- `STATUS BUSY` - druhá strana je zaneprázdnená a nemôže prijímať ďalšie hovory.

Na väčšinu z nich plugin reaguje oboznámením používateľa o vzniknutej situácii. Pokiaľ sa však podarí vytvoriť spojenie, plugin si vyžiada meno a jedinečnú Skype prezývku pomocou volania príkazu `GET CALL <id> PARTNER_HANDLE`. Tá je neskôr použitá ako jedinečný identifikátor.

V momente, kedy sa podarí vytvoriť spojenie medzi dvoma stranami, tzn. plugin obdrží správu `CALL <id> STATUS INPROGRESS`, plugin prikáže klientovi, aby príchodzí tok preposielal na už bežiaci TCP server. To sa vykoná pomocou príkazu `ALTER CALL <id> SET_OUTPUT PORT=<port>`, kde `<port>` určuje port, na ktorom TCP server počúva. Od tohto momentu je možné dáta čítať a spracúvať behom rozhovoru.

Tento audio tok je v nekomprimovanom formáte, vzorkovaný na frekvencii 16 KHz, s veľkosťou 16 bit na vzorku. Vzhľadom na povahu telefónneho rozhovoru je audio tok jednokanálový. BSC nepodporuje spracovávanie audia v takomto formáte, a preto je nevyhnutné tento záznam skonvertovať. Najbližší formát podporovaný BSC je nekomprimované audio so vzorkovacou frekvenciou 8 KHz a 16 bit na vzorku. Preto pred spracovaním signálu v BSC je prijímaná zvuková stopa redukovaná tak, že posledných 16 bit z každých 32 bit je zahodených. Tým je dosiahnuté podvzorkovanie z 16KHz na 8 KHz a umožnené spracovanie v BSC.

Behom rozhovoru TCP server čaká pevne stanovenú dobu, ktorá zodpovedá piatim sekundám. Keď sa jeho vyrovnávacia pamäť naplní, TCP server ju prečíta, pridá do pamäti k už načítaným dátam z minulých cyklov, podvzorkuje a odošle do vlákna spravujúceho komunikáciu s BSC s príznakom, že hovor stále prebieha alebo už skončil. Ak hovor skončil, TCP server posiela aktuálny obsah vyrovnávacej pamäti a nečaká na jej naplnenie. Vlákno obsluhujúce BSC sa spustí vždy, keď sa má spracovať zvuková stopa a po spracovaní sa ukončí na rozdiel napr. od vlákna komunikujúceho s klientom alebo vlákna s TCP serverom, ktoré bežia bez zastavenia.

Vlákno, ktoré spracúva nahrávku, najprv zaregistruje licenčný súbor. Následne vytvorí inštanciu extraktoru, načíta konfiguráciu pre extrakciu a podľa nej sa pokúsi extrahovať hlasový odtlačok (voiceprint). Tento proces je výrazne pamäťovo a výpočtovo náročný a preto je najužším hrdlom aplikácie. Na slabších strojoch môže dôjsť k neúmerne dlhému výpočtu, čo vedie k overeniu identity až neskoro po skončení hovoru alebo dokonca k úplne chybnej extrakcii voiceprintu. Pokiaľ sa však podarí voiceprint extrahovať, je dočasne uložený. Následne je načítaný obsah adresára s už existujúcimi odtlačkami, kde každý obsahuje autora nahrávky, ako aj ďalšie informácie vysvetlené nižšie. Po načítaní adresára sú jednotlivé hlasové odtlačky rozdelené podľa pôvodu. Každý hlasový odtlačok je asociovaný s konkrétnym účtom, presne v tvare, ako je registrovaný u Skype. Keď sú odtlačky rozdelené do skupín, potom sa pre každú skupinu spočíta zhoda s práve extrahovaným hlasovým odtlačkom a ak existuje viac odtlačkov od jednoho zdroja, sú jednotlivé miery zhody aritmeticky spriemerované. Skóre je následne prevedené na percentá podľa vzorca:

$$p = \frac{\mathrm{e}^{\frac{S-b}{s}}}{1+\mathrm{e}^{\frac{S-b}{s}}} \cdot 100 \quad , \tag{1}$$

kde S reprezentuje vypočítané skóre, s určuje strmosť prevodu, b určuje tvrdý práh, podľa ktorého verifikátor určí svoje rozhodnutie. Pre takto odvodenú pravdepodobnosť potom platí, že verifikátor označí rečníka za verifikovaného, ak presiahne hranicu 50%. Konštanty b a s sú nastavené nemenne v zdrojovom kóde a to ako dôsledok testovania na hodnoty $s = 30$ a $b = 20$.

Pretože plugin prechádza celou databázou hlasových odtlačkov, dokáže oznámiť nielen pravdepodobnosť, že človek na druhej strane je naozaj ten, za koho sa vydáva, ale aj vypísať identitu používateľa, ktorý má najväčšiu zhodu. Prípad, kedy je zhoda s predpokladaným kontaktom malá a s iným evidovaným kontaktom veľká, upozorňuje na využívanie Skype účtu inou osobou, aká je s daným účtom asociovaná.

Po vyhodnotení jednotlivých pravdepodobností je hlavné vlákno oboznámené o výsledku jednotlivých porovnaní a vlákno sa ukončí. Ak prišla nahrávka s príznakom poslednej časti, vlákno pred ukončením hlasový odtlačok uloží a do jeho upravovateľných používateľských dát vloží čas vytvorenia a dĺžku záznamu, z ktorého sa odtlačok extrahoval a prezývku volaného. Potom pomocou hlavného vlákna ponúkne používateľovi rozhodnutie, či hlasový odtlačok uložiť alebo zahodiť. Ak sa používateľ rozhodne daný hlasový odtlačok ponechať, ten je prenesený z dočasného adresára do adresára s používanými odtlačkami v tvare `<skype_id>-<čas_vzniku>.vp`.

Okrem spomenutých funkcií patria do hlavného vlákna aj funkcie obsluhujúce grafické rozhranie. V rámci toho ponúkajú jednoduchého grafického správcu odtlačkov v databáze.

Jednotlivé vlákna používajú súbor `error.log` na informatívny a chybový výpis.

Použitá konfigurácia BSC bola behom implementácie zmenená z dôvodu veľkých výpočtových nárokov a z toho plynúcich chýb s alokáciou. Po zmene konfigurácie na menej náročný systém sa problémy s alokáciou miesta v pamäti vyriešili.

6.3 Popis jednotlivých tried a ich grafická reprezentácia

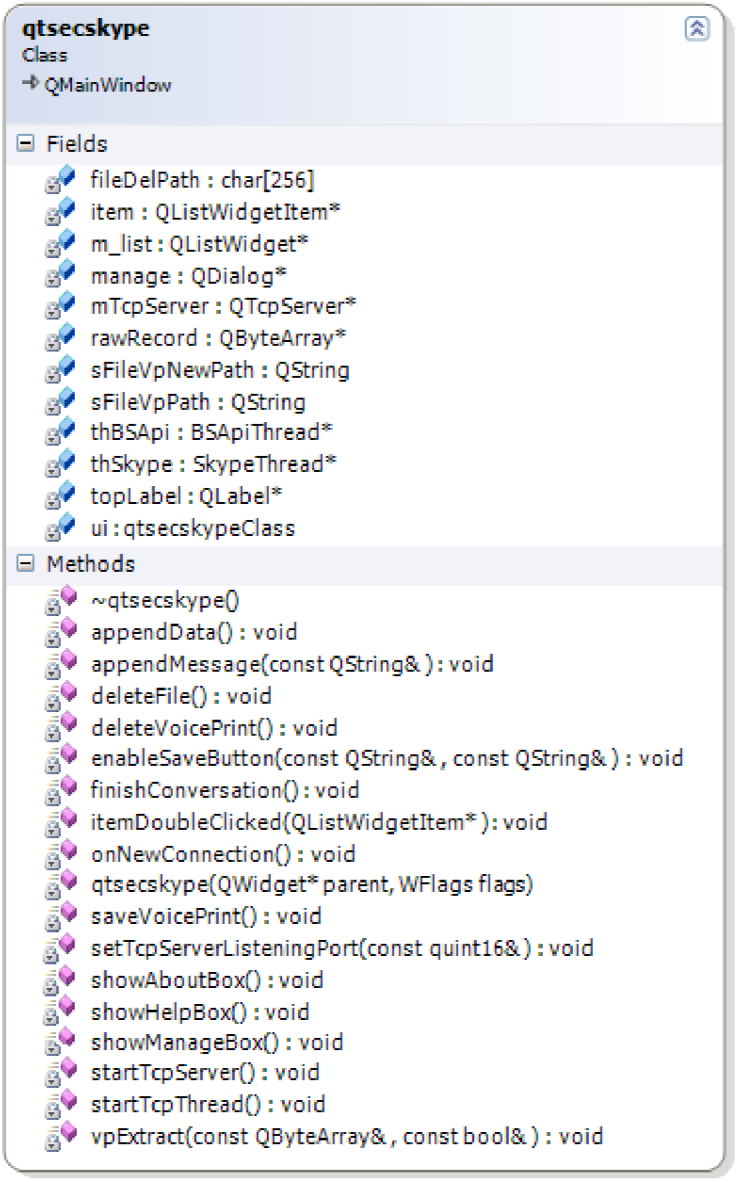
Jednotlivé triedy sú porozdeľované do viacerých súborov tak, že každý súbor obsahuje triedy pre obsluhu jednotlivých častí pluginu. Väčšina súborov so zdrojovým kódom má príponu .cpp a k nemu príslušný hlavičkový súbor .h. Menovite sú to:

- main.cpp - obsahuje jedinú funkciu, ktorá vytvorí okno pluginu a spustí ho vo vlastnom vlákne,
- qtsecskype.cpp s hlavičkovým súborom qtsecskype.h - obsahuje triedy na správu grafického rozhrania, výpisy a definuje komunikačné väzby medzi jednotlivými vláknami,
- SkypeThread.cpp s hlavičkovým súborom SkypeThread.h - obsahuje triedy umožňujúce a spravujúce komunikáciu so Skype klientom,
- BSApiThread.cpp s hlavičkovým súborom BSApiThread.h - obsahuje triedy na komunikáciu s BSC,

– Hlavičkový súbor bsapi.h - obsahuje deklarácie verejných metód prístupných z dynamickej knižnice bsapi.dll,

– qtsecskype.ui - obsahuje základné rozloženie prvkov v grafickom rozhraní, súbor využívaný knižnicou Qt.

6.3.1 Trieda qtsecskype



Obrázok 4: Grafická reprezentácia triedy qtsecskype.

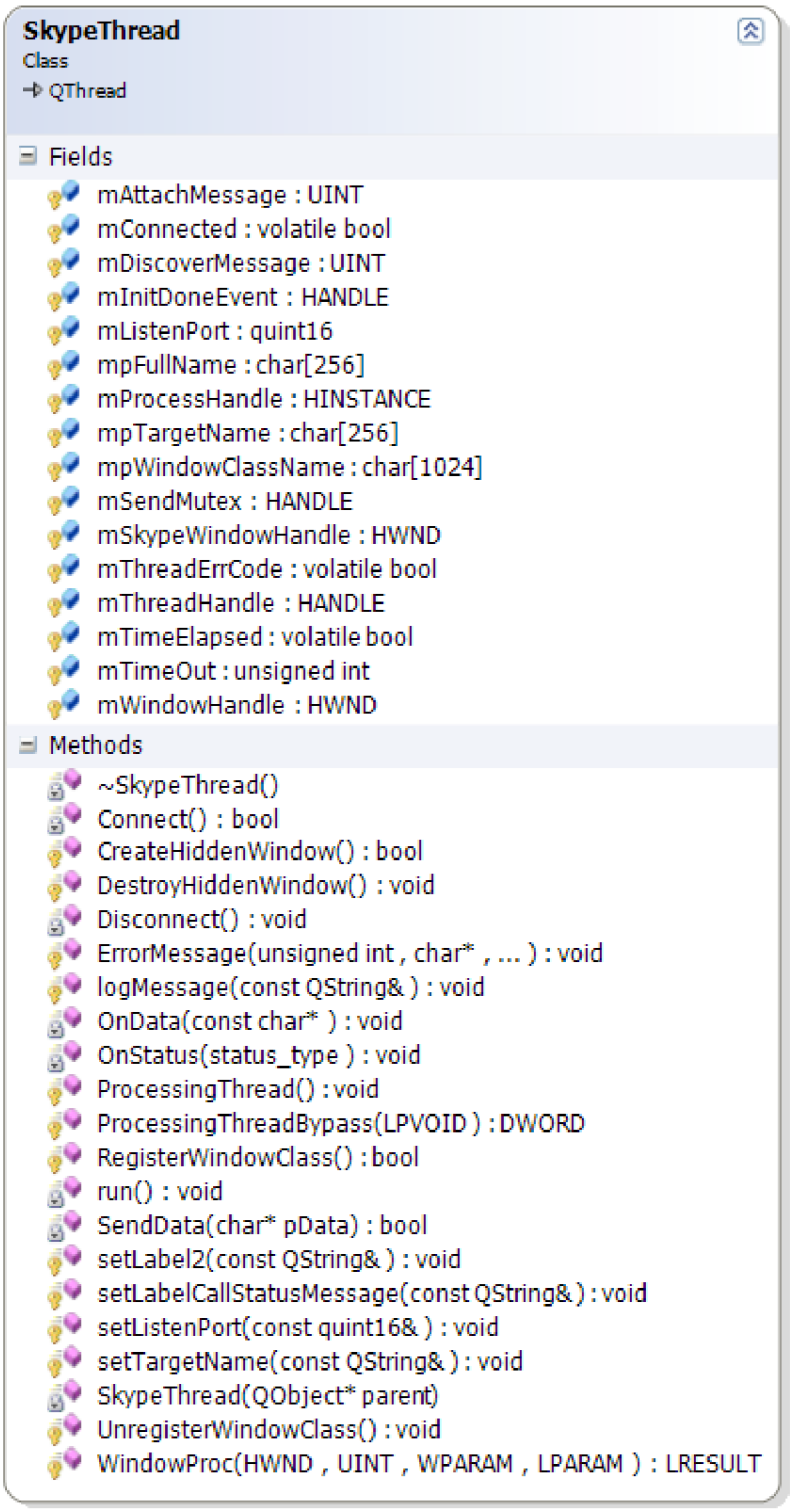
Je hlavnou triedou, ktorej inštancia sa spúšťa pri spustení aplikácie. V konštruktori vytvára spojenie medzi jednotlivými vláknami, zavádza TCP server a ponúka funkcie pre obsluhu používateľského rozhrania. Jednotlivé funkcie:

- `appendMessage()` - prijíma textový reťazec a zapisuje ho do odlaďovacej konzoly. Používaná len pri vývoji.
- `startTcpThread()` - vytvorí a spustí vlákno s TCP serverom.
- `startTcpServer()` - vytvorí inštanciu TCP servera na niektorom z voľných portov a jeho hodnotu odošle do vlákna obsluhujúceho komunikáciu so Skype klientom.
- `onNewConnection()` - volaná pri príchode dát na počúvajúci TCP server. Nastaví callback funkcie na príchodzie dáta a ukončenie prenosu.
- `appendData()` - vyhodnocuje príchodzie dáta na TCP server a pri prekročení stanoveného počtu ich podvzorkuje a odošle na spracovanie do vlákna komunikujúceho s BSAPI na spracovanie.
- `finishConversation()` - podobné správanie ako `appendData()` s rozdielom, že pri žiadosti na spracovanie pridáva informáciu o ukončení prenosu.
- `enableSaveButton()` - dostupná pre ďalšie vlákna, umožňuje aktivovať *Save* a *Discard* tlačidlá.
- `saveVoicePrint()` - volaná pri rozhodnutí používateľa, že chce uložiť vyextrahovaný hlasový odtlačok. Premiestni uložený odtlačok z dočasného adresára do adresára s odtlačkami.
- `deleteVoicePrint()` - volaná pri rozhodnutí používateľa, že chce zmazať vyextrahovaný hlasový odtlačok. Vymaže uložený odtlačok z dočasného adresára.
- `showAboutBox()` - zobrazí informačné okno s popisom pluginu a o autorovi.
- `showManageBox()` - zobrazí okno zobrazujúce všetky dostupné hlasové odtlačky a dvojklik na niektorý z nich prepojí s funkciou `itemDoubleClicked()`.
- `showHelpBox()` - zobrazí okno vysvetľujúce význam *Self score* a *Best score* z grafického rozhrania.
- `itemDoubleClicked()` - zobrazí informácie o zvolenom hlasovom odtlačku v separátnom okne. Zobrazuje Skype prezývku autora, čas vytvorenia a dĺžku nahrávky, z ktorej bol odtlačok vyextrahovaný. Ponúka aj možnosť daný podpis vymazať.
- `deleteFile()` - vybraný súbor vymaže.

Signály, používané v medzivláknovej komunikácii:

- `setTcpServerListeningPort()` - potom, ako je známy port TCP servera, je tento port odoslaný do vlákna komunikujúceho so Skype klientom. Ten toto číslo zasiela klientovi vždy pri začatí nového volania.
- `vpExtract()` - slúži na odoslanie aktuálnej nahrávky na spracovanie v BSC. Druhý parameter nesie príznak, či sa jedná o poslednú žiadosť, tzn. či prebiehajúci hovor skončil.

6.3.2 Trieda SkypeThread



Obrázok 5: Grafická reprezentácia triedy SkypeThread.

Je podtriedou triedy QThread a ponúka funkcie pre komunikáciu pomocou správ implementovaných v OS Windows. Správa sa autonómne k ostatným vláknam, to znamená, že vytvorí spojenie so Skype klientom pri inicializácii a v reakciách na príchodzie správy odosiela príkazy klientovi samostatne. Tieto správy spracúva a relevantné informácie odosiela do ostatných vlákien. Jednotlivé funkcie sú:

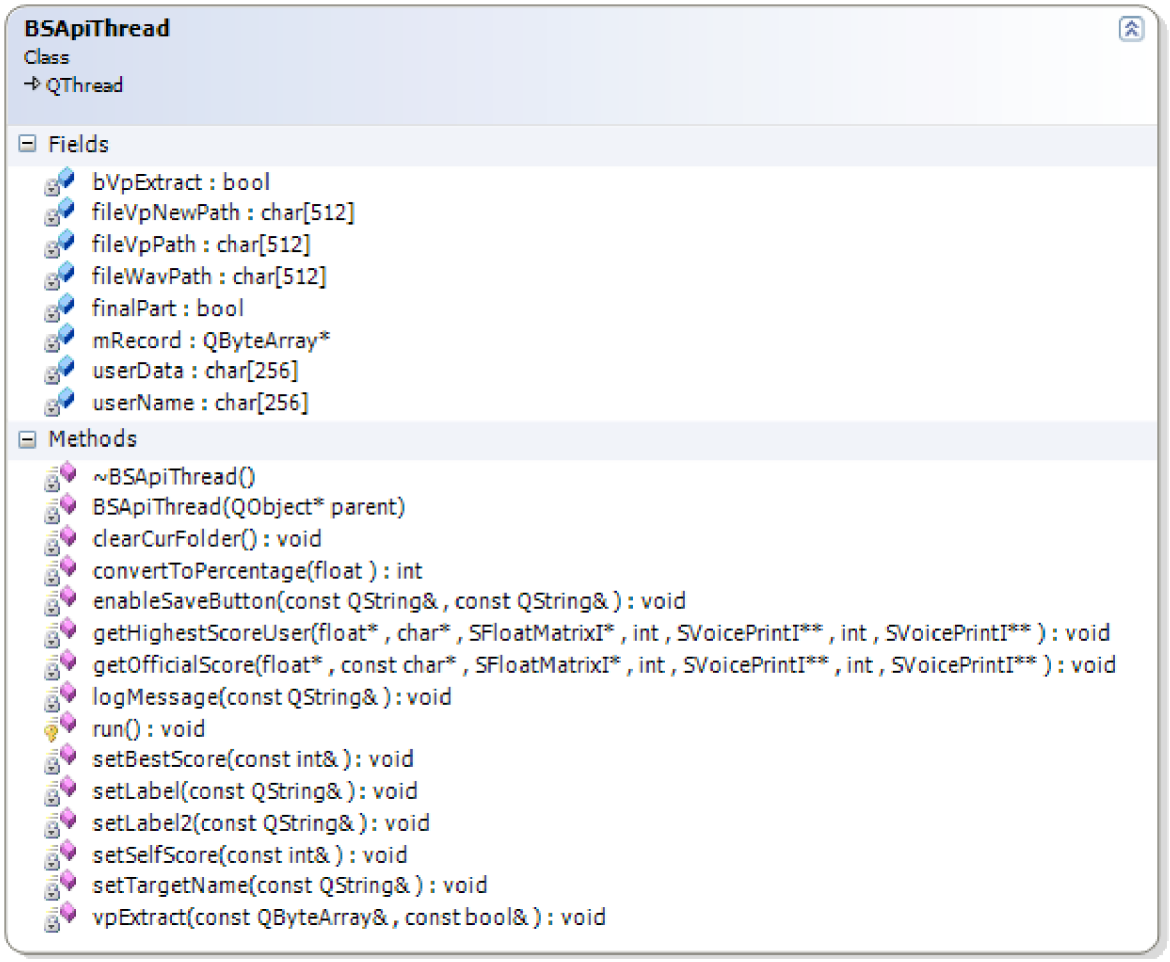
- `run()` - zdedená funkcia, spustená pri spustení vlákna.
- `SendData()` - dáta prijaté ako parameter odošle klientovi pomocou správ v OS Windows.
- `Connect()` - pretože Skype klient po určitej dobe nečinnosti uzavrie otvorené spojenie, je nutné vždy pred odoslaním dát skontrolovať, či je vytvorené spojenie. Ak nie je, je nevyhnutné ho vytvoriť. Funkcia `Connect()` existujúce spojenie ukončí a vytvorí nové.
- `Disconnect()` - ak je vytvorené spojenie s klientom Skype, ukončí ho.
- `OnStatus()` - callback funkcia volaná, pokiaľ sa zmení stav Skype klienta. Ide najmä o odhlásenie používateľa, jeho prihlásenie či ukončenie klienta.
- `RegisterWindowClass()` - funkcia, ktorá zaregistruje v OS Windows triedu neviditeľného okna, ktoré je využívané na komunikáciu pomocou správ v OS Windows.
- `UnregisterWindowClass()` - opak funkcie `RegisterWindowClass()`.
- `CreateHiddenWindow()` - vytvorí neviditeľné okno z triedy definovanej pomocou `RegisterWindowClass()`.
- `DestroyHiddenWindow()` - odstráni okno vytvorené pomocou `CreateHiddenWindow()`.
- `ProcessingThread()` - je volaný pri vytvorení nového vlákna vo funkcii `Connect()`. Vytvorí okno a v nekonečnej slučke spracúva príchodzie správy za využitia `WindowProc()`.
- `ProcessingThreadBypass()` - pomocná funkcia, nutná pre korektné volanie `ProcessingThread()`.
- `WindowProc()` - spracúva správy systému Windows určené pre vytvorené neviditeľné okno. Najmä však správy od Skype klienta.
- `ErrorMessage()` - funkcia spracujúca chybové výstupy.
- `setListenPort()` - vlákno si uloží číslo TCP servera, bežiaceho v inom vlákne.

Signály pre komunikáciu s ostatnými vláknami v rámci pluginu:

- `logMessage()` - prepojený s odlaďovacou konzolou v grafickom rozhraní. Využívaný pri vylaďovaní.
- `setLabelCallStatusMessage()` - nastaví text v hlavnom okne grafického rozhrania.
- `setLabel2()` - nastaví pomocný text v hlavnom okne grafického rozhrania.

– `setTargetName()` - nastaví prezývku hovoriaceho na druhej strane pre potreby identifikácie a verifikácie.

6.3.3 Trieda BSApiThread



Obrázok 6: Grafická reprezentácia triedy BSApiThread.

Jedná sa o triedu, ktorá je rovnako ako SkypeThread potomkom triedy QThread. Pokytuje najmä vybrané funkcie BSC pre hlavné vlákno, ako je extrakcia hlasového odtlačku a počítanie miery zhody s ostatnými hlasovými odtlačkami. Jednotlivé funkcie sú:

– `run()` - zdedená funkcia od QThread, ktorá je volaná pri spustení vlákna.
– `vpExtract()` - vykonáva všetku komunikáciu s BSAPI. Zaregistruje a vytvorí nevyhnutné štruktúry pre extrakciu a porovnanie hlasových odtlačkov. Spočítané miery zhôd vyhodnotí a podľa nich informuje používateľa. Ak je nahrávka na konci, získaný odtlačok je uložený do dočasného adresára.

- `setTargetName()` - uloží do privátnej premennej prezývku práve volaného partnera. Pri ukladaní hlasového odtlačku je táto premenná použitá v názve uloženého súboru a vo vnútornej štruktúre.
- `getOfficialScore()` - spočíta aritmetický priemer zhôd extrahovaného hlasového odtlačku so všetkými dostupnými odtlačkami od rovnakého partnera.
- `getHighestScoreUser()` - spočíta aritmetický priemer zhôd extrahovaného hlasového odtlačku so všetkými dostupnými odtlačkami od ostatných zdrojov. Vráti najväčšiu zhodu zo všetkých.

Signály pre komunikáciu s ostatnými vláknami:

- `logMessage()` - prepojená s odladovacou konzolou. Využívaná je len vo fáze vývoja na odladovacie výpisy.
- `setLabel()` - nastavenie hlavného textu v hlavnom okne.
- `setLabel2()` - nastavenie pomocného textu v hlavnom okne.
- `enableSaveButton()` - aktivuje ovládacie tlačidlá pre uloženie a zahodenie extrahovaného hlasového odtlačku.
- `setSelfScore()` - nastaví úroveň zhody v grafickom rozhraní pre prvý grafický ukazovateľ zhody. Ten reprezentuje mieru zhody extrahovaného hlasového odtlačku s odtlačkami, ktoré prináležia účtu, s ktorým prebieha hovor.
- `setBestScore()` - nastaví úroveň zhody v grafickom rozhraní pre druhý grafický ukazovateľ zhody. Ten reprezentuje najväčšiu zhodu zo všetkých dostupných hlasových odtlačkov.

6.4 Závislosti a inštalácia

Program využíva funkcie z viacerých knižníc. Okrem štandardných knižníc operačného systému Windows sa program odkazuje na tri dynamické knižnice prostredia Qt. Okrem nich využíva funkcie z BSC. Vymenované nevyhnutné knižnice sú priložené k spustiteľnému súboru. Jedná sa o:

- bsapi.dll,
- mkl.dll,
- QtCore4.dll,
- QtGui4.dll,
- QtNetwork4.dll.

Vzhľadom na to, že bol program vyvíjaný a skompilovaný vo vývojovom prostredí Visual Studio 2010, potrebuje pre svoj chod niektoré z jeho komponentov. Pre osobné počítače bez nainštalovaného Visual Studio 2010 ponúka Microsoft na svojich stránkach voľne stiahnuteľný balík s názvom

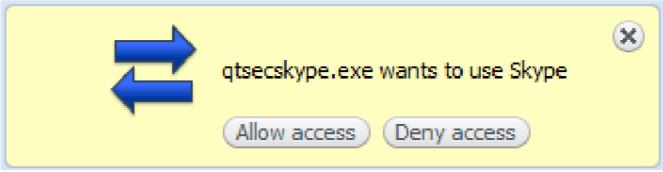
Microsoft Visual C++ 2010 Redistributable Package so spomínanými komponentami. Je platformovo závislý a v dobe písania tohto textu stiahnuteľný na stránkach Microsofu pre 32 bit a 64 bit systémy.

Okrem spomínaných komponentov je pre správny chod nevyhnutný prístup na Internet z dôvodu overenia licencie k funkciám z BSC. Licencia je očakávaná v súbore s názvom license.dat a je tiež priložená k spustiteľnému súboru.

Program vyžaduje pre správny chod prístup do zanorených podadresárov.

Program je spustiteľný pomocou súboru qtsecskype.exe. Pri spustení sa očakáva už bežiaca inštancia Skype klienta s rovnakými právami, s akými bude spustený plugin. Plugin bol testovaný so Skype klientom verzie 5.1.0.112, ale využíva štandardné funkcie a preto by mal pracovať so všetkými dostupnými verziami pre OS Windows.

Po spustení je používateľ v okne Skype klienta vyzvaný, aby povolil prístup pluginu k aplikácii. Nasledujúci obrázok popisuje pravdepodobný vzhľad v okne Skype klienta.

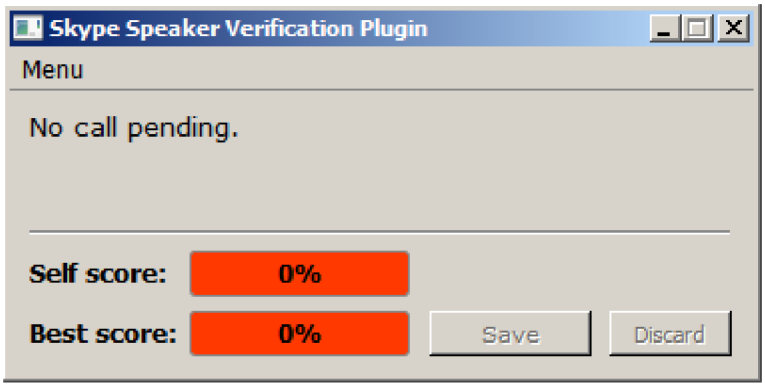


Obrázok 7: Výzva Skype klienta na povolenie prístupu pluginu.

Pokiaľ používateľ povolí prístup, je komunikácia nadviazaná a plugin je úspešne nainštalovaný a pripravený k použitiu.

6.5 Scenár použitia

Po spustení sa na obrazovke objaví okno podobné nasledujúcemu.

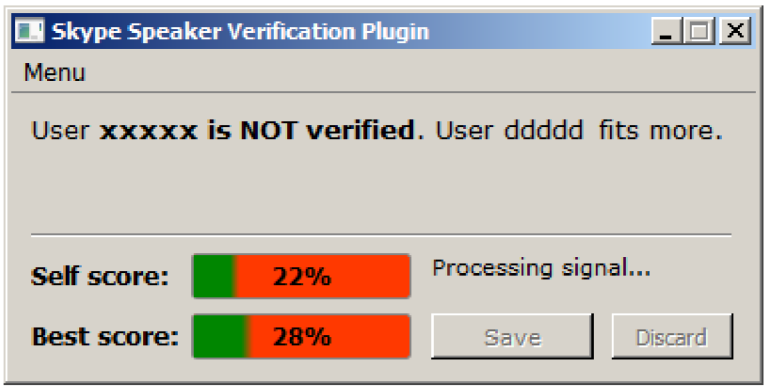


Obrázok 8: Snímok grafického výstupu 1.

Pokiaľ je plugin povolený v Skype klientovi, nie je už potrebné nič nastavovať. V momente, kedy používateľ zavolá niektorý z kontaktov alebo príde výzva na spojenie zvonka, plugin túto zmenu automaticky detekuje. Keď sa nadviaže spojenie, rozhovor sa začne nahrávať pre potreby spracovania. Po prvých piatich sekundách plugin začne s kontinuálnym spracovaním nahrávky. O výsledkoch priebežne informuje používateľa. V prvom grafickom ukazovateli miery zhody informuje plugin používateľa o miere podobnosti k odtlačkom asociovaných k účtu, s ktorým práve prebieha rozhovor. V druhom ponúka používateľovi maximálnu mieru zhody zo všetkých odtlačkov asociovaných k dostupným účtom. Z toho plynie, že miera podobnosti v prvom riadku nikdy nemôže presiahnuť úroveň v druhom riadku.

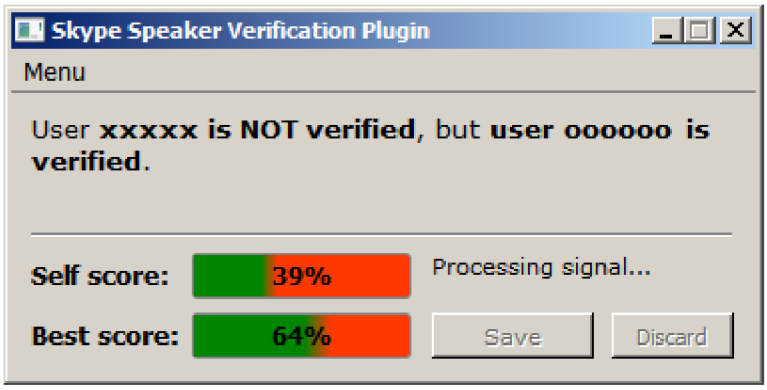
Ako tvrdý rozhodovací prah je zvolená úroveň 50%. Pokiaľ jedna z hodnôt prekročí túto úroveň, je príslušný rečník verifikovaný. Najčastejšie výstupy pluginy sú teda nasledovné:

- volaný kontakt nie je verifikovaný a rovnako nie je nájdený nikto v databáze, kto by bol verifikovateľný.



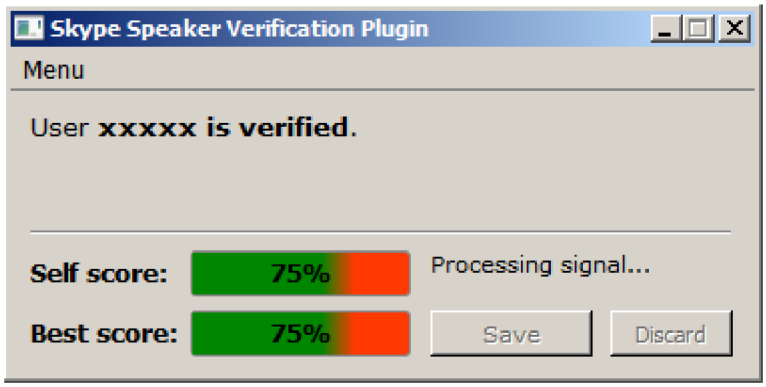
Obrázok 9: Snímok grafického výstupu 2.

- volaný kontakt nebol verifikovaný, ale iná osoba z dostupnej databázy bola verifikovaná.



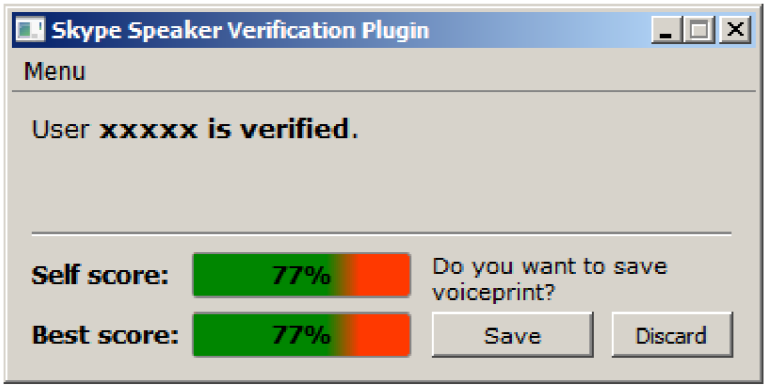
Obrázok 10: Snímok grafického výstupu 3.

- volaný kontakt bol verifikovaný.



Obrázok 11: Snímok grafického výstupu 4.

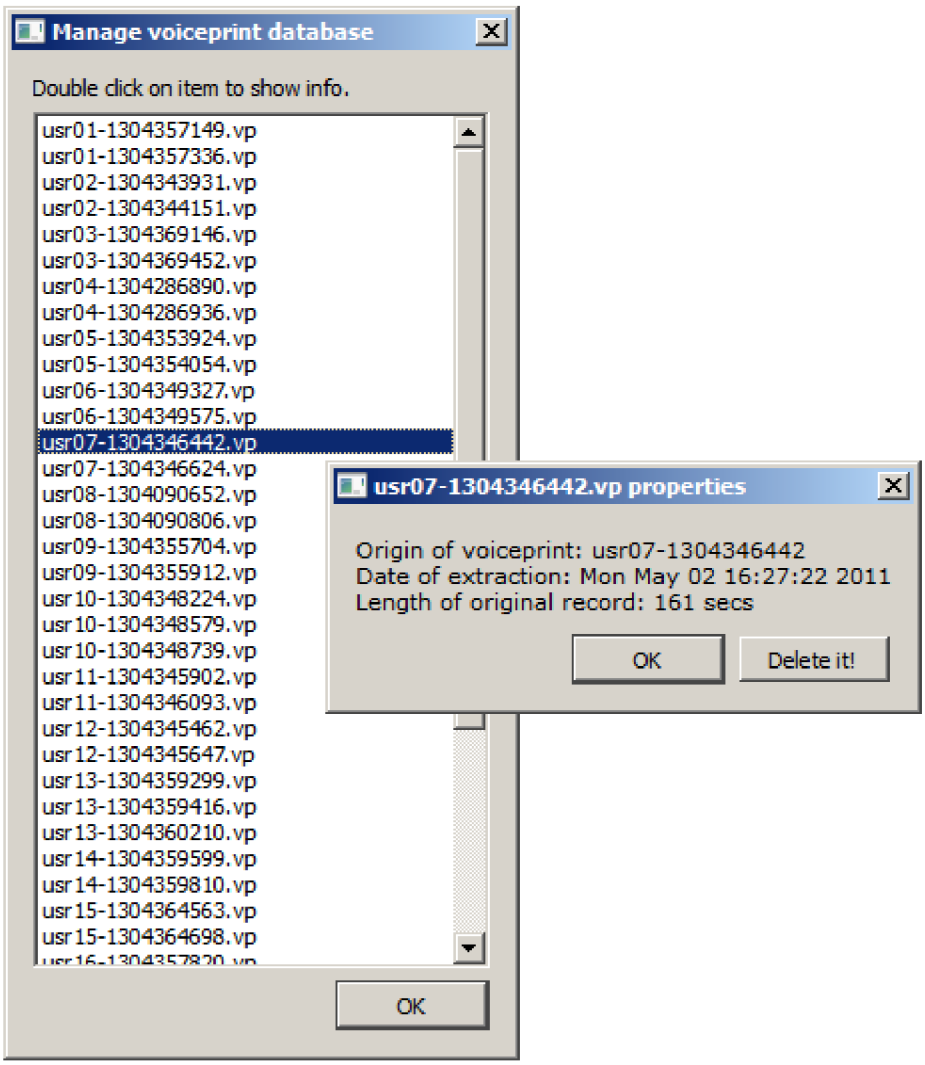
Po skončení rozhovoru je nahrávka spracovaná a používateľ sa môže rozhodnúť, či vyextrahovaný hlasový podpis uloží do databázy.



Obrázok 12: Snímok grafického výstupu 5.

Pokiaľ sa používateľ rozhodne pre uloženie hlasového odtlačku, tak je tento uložený do databázy a behom všetkých ďalších nahrávok zahrnutý do výpočtu.

Plugin ponúka jednoduchú správu už uložených hlasových odtlačkov. Rozhranie pre zobrazenie a správu databázy je prístupné cez *Menu* v položke *Manage*. Po rozkliknutí sa v novootvorenom okne objaví zoznam všetkých dostupných odtlačkov. Po dvojkliku na vybraný prvok sa zobrazia detaily o vybranom odtlačku s možnosťou jeho odstránenia z databázy.



Obrázok 13: Snímok grafického výstupu 6.

Ďalšie dostupné okno s názvom *Help* ponúka vysvetlenie pojmov *Self score* a *Best score* použitých v grafickom rozhraní.

6.6 Vyhodnotenie v reálnych podmienkach

Testovanie prebiehalo na počítači s nainštalovaným operačným systémom Windows 7 s procesorom Intel Core 2 Quad Q9550 s 4GB operačnej pamäte. Postupne bolo volaných 20 ľudí s dĺžkou hovorov od jednej až po štyri minúty. Všetky nahrávky boli zaznamenané a dodatočne oskórované systémom každý s každým. Nasledujúca tabuľka zhŕňa jednotlivé skóre, tak ako ich vracia Brno Speech Core. To znamená, pred konverziou do percentuálnej škály. V tabuľke sú zaokrúhlené hodnoty a červenou označené polia, kde je skóre menšie ako dvadsať a zelenou polia, kde je skóre väčšie ako dvadsať. Výsledky sú prekvapujúco dobré a predpokladám, že by boli poznateľne horšie za predpokladu, ak by

došlo k väčšej variabilite na kanáli. Napríklad ak by jednotlivé nahrávky prebiehali cez iný mikrofón alebo za iného stavu prenosového kanála.

	u01	u01	u02	u02	u03	u03	u04	u04	u05	u05	u06	u06	u07	u07	u08	u08	u09	u09	u10	u10	u10
u01	71	39	-30	-44	-36	-34	-78	-83	-12	-33	2	0	-35	-46	12	12	-41	-22	-19	-14	-32
u01	39	72	-48	-46	-31	-42	-62	-85	-24	-44	-5	-9	-31	-37	12	10	-41	-13	-17	-17	-34
u02	-30	-48	91	46	-42	-26	-46	-55	-31	-42	-45	-43	-57	-60	-57	-46	-34	-28	-53	-64	-80
u02	-44	-46	46	92	-33	-23	-39	-51	-29	-47	-56	-58	-42	-54	-65	-56	-36	-35	-50	-50	-86
u03	-36	-31	-42	-33	73	20	-37	-45	-10	-20	-37	-44	-17	-23	-43	-38	-20	-17	-55	-56	-75
u03	-34	-42	-26	-23	20	72	-31	-38	-12	-15	-34	-34	-6	-13	-34	-33	-12	-7	-40	-34	-59
u04	-78	-62	-46	-39	-37	-31	108	52	-51	-57	-62	-66	-41	-32	-76	-67	-50	-45	-81	-69	-104
u04	-83	-85	-55	-51	-45	-38	52	103	-63	-56	-83	-88	-59	-42	-96	-83	-47	-50	-90	-85	-113
u05	-12	-24	-31	-29	-10	-12	-51	-63	83	34	-17	-28	2	-1	-23	-24	-30	-20	-16	-20	-40
u05	-33	-44	-42	-47	-20	-15	-57	-56	34	83	-21	-23	18	10	-42	-45	-19	-5	-24	-27	-36
u06	2	-5	-45	-56	-37	-34	-62	-83	-17	-21	71	50	-24	-20	1	-2	-25	-21	4	10	-4
u06	0	-9	-43	-58	-44	-34	-66	-88	-28	-23	50	74	-24	-28	3	-3	-27	-17	-4	0	-20
u07	-35	-31	-57	-42	-17	-6	-41	-59	2	18	-24	-24	86	45	-33	-36	-11	6	-28	-26	-39
u07	-46	-37	-60	-54	-23	-13	-32	-42	-1	10	-20	-28	45	87	-31	-33	-6	-1	-22	-18	-34
u08	12	12	-57	-65	-43	-34	-76	-96	-23	-42	1	3	-33	-31	87	47	-30	-7	-7	-6	-29
u08	12	10	-46	-56	-38	-33	-67	-83	-24	-45	-2	-3	-36	-33	47	80	-17	-13	-6	2	-22
u09	-41	-41	-34	-36	-20	-12	-50	-47	-30	-19	-25	-27	-11	-6	-30	-17	82	27	-26	-23	-37
u09	-22	-13	-28	-35	-17	-7	-45	-50	-20	-5	-21	-17	6	-1	-7	-13	27	74	-18	-21	-39
u10	-19	-17	-53	-50	-55	-40	-81	-90	-16	-24	4	-4	-28	-22	-7	-6	-26	-18	68	42	26
u10	-14	-17	-64	-50	-56	-34	-69	-85	-20	-27	10	0	-26	-18	-6	2	-23	-21	42	68	32
u10	-32	-34	-80	-86	-75	-59	-104	-113	-40	-36	-4	-20	-39	-34	-29	-22	-37	-39	26	32	84
u11	-17	-29	-23	-16	-9	5	-49	-55	9	-5	-23	-29	-4	-11	-16	-25	-10	-6	-25	-20	-44
u11	-13	-17	-7	-1	-10	-7	-55	-61	1	-5	-18	-31	3	-8	-15	-16	-12	-14	-30	-26	-44
u12	10	8	-28	-40	-45	-33	-62	-81	-23	-32	-21	-21	-19	-27	13	3	-21	-9	-22	-15	-38
u12	5	-1	-23	-33	-54	-36	-72	-76	-29	-46	-18	-16	-43	-43	4	0	-29	-25	-28	-14	-43
u13	-26	-20	-25	-26	4	-5	-28	-49	-5	1	-18	-23	16	18	-32	-28	-6	6	-32	-30	-52
u13	-29	-37	-19	-34	-22	-16	-35	-45	-21	-9	-19	-22	9	7	-38	-33	-10	-6	-37	-34	-50
u13	-22	-22	-43	-46	-2	-12	-39	-62	-14	-6	-10	-20	11	20	-28	-30	-12	-3	-33	-23	-48
u14	-48	-40	-30	-21	-14	-4	-5	-21	-20	-26	-44	-48	-17	-17	-54	-48	-29	-23	-54	-46	-59
u14	-40	-28	-19	-7	1	-1	-2	-26	-6	-20	-38	-47	-16	-8	-38	-36	-21	-15	-45	-42	-57
u15	-50	-41	-38	-26	13	-7	-12	-14	-7	-9	-45	-55	10	11	-42	-44	-13	-4	-60	-51	-84
u15	-40	-31	-29	-15	8	0	4	-5	-18	-15	-22	-30	10	13	-33	-29	-7	-7	-43	-30	-50
u16	-71	-80	-38	-44	-16	-3	-68	-55	-25	1	-47	-43	-20	-25	-65	-53	-23	-21	-48	-49	-67
u16	-54	-70	-51	-58	-24	1	-51	-40	-22	4	-34	-30	-8	-16	-71	-56	-13	-7	-38	-37	-45
u16	-46	-62	-39	-41	-24	-5	-54	-55	-3	7	-28	-27	5	1	-52	-40	-24	-12	-29	-25	-46
u17	-81	-74	-66	-48	-15	-16	-38	-48	-25	-13	-44	-55	9	10	-56	-64	-27	-18	-55	-50	-70
u17	-86	-75	-74	-50	-17	-19	-45	-56	-33	-21	-42	-56	3	5	-58	-62	-34	-20	-56	-52	-61
u18	-27	-23	-20	-7	-17	-1	-19	-23	-10	-10	-28	-29	-16	-5	-27	-27	-1	2	-35	-27	-46
u18	-34	-27	-13	-11	-14	-3	-8	-18	-13	-9	-32	-29	-15	-6	-32	-27	3	2	-40	-33	-54
u19	-46	-45	-37	-38	-4	-12	-37	-53	-8	-14	-43	-39	5	-12	-43	-45	-30	-10	-57	-56	-79
u19	-51	-51	-32	-33	-8	-17	-15	-38	-1	-5	-57	-58	4	-9	-53	-52	-29	-16	-57	-60	-86
u20	-71	-68	-81	-86	-20	-21	-51	-66	-25	-12	-46	-55	-1	4	-70	-81	-37	-31	-73	-66	-87
u20	-82	-83	-81	-89	-16	-23	-55	-57	-23	-16	-56	-73	-18	-3	-67	-82	-45	-42	-85	-93	-107

Tabuľka 5: Vyhodnotenie úspešnosti na reálnych dátach - časť prvá.

	u11	u11	u12	u12	u13	u13	u13	u14	u14	u15	u15	u16	u16	u16	u17	u17	u18	u18	u19	u19	u20	u20
u01	-17	-13	10	5	-26	-29	-22	-48	-40	-50	-40	-71	-54	-46	-81	-86	-27	-34	-46	-51	-71	-82
u01	-29	-17	8	-1	-20	-37	-22	-40	-28	-41	-31	-80	-70	-62	-74	-75	-23	-27	-45	-51	-68	-83
u02	-23	-7	-28	-23	-25	-19	-43	-30	-19	-38	-29	-38	-51	-39	-66	-74	-20	-13	-37	-32	-81	-81
u02	-16	-1	-40	-33	-26	-34	-46	-21	-7	-26	-15	-44	-58	-41	-48	-50	-7	-11	-38	-33	-86	-89
u03	-9	-10	-45	-54	4	-22	-2	-14	1	13	8	-16	-24	-24	-15	-17	-17	-14	-4	-8	-20	-16
u03	5	-7	-33	-36	-5	-16	-12	-4	-1	-7	0	-3	1	-5	-16	-19	-1	-3	-12	-17	-21	-23
u04	-49	-55	-62	-72	-28	-35	-39	-5	-2	-12	4	-68	-51	-54	-38	-45	-19	-8	-37	-15	-51	-55
u04	-55	-61	-81	-76	-49	-45	-62	-21	-26	-14	-5	-55	-40	-55	-48	-56	-23	-18	-53	-38	-66	-57
u05	9	1	-23	-29	-5	-21	-14	-20	-6	-7	-18	-25	-22	-3	-25	-33	-10	-13	-8	-1	-25	-23
u05	-5	-5	-32	-46	1	-9	-6	-26	-20	-9	-15	1	4	7	-13	-21	-10	-9	-14	-5	-12	-16
u06	-23	-18	-21	-18	-18	-19	-10	-44	-38	-45	-22	-47	-34	-28	-44	-42	-28	-32	-43	-57	-46	-56
u06	-29	-31	-21	-16	-23	-22	-20	-48	-47	-55	-30	-43	-30	-27	-55	-56	-29	-29	-39	-58	-55	-73
u07	-4	3	-19	-43	16	9	11	-17	-16	10	10	-20	-8	5	9	3	-16	-15	5	4	-1	-18
u07	-11	-8	-27	-43	18	7	20	-17	-8	11	13	-25	-16	1	10	5	-5	-6	-12	-9	4	-3
u08	-16	-15	13	4	-32	-38	-28	-54	-38	-42	-33	-65	-71	-52	-56	-58	-27	-32	-43	-53	-70	-67
u08	-25	-16	3	0	-28	-33	-30	-48	-36	-44	-29	-53	-56	-40	-64	-62	-27	-27	-45	-52	-81	-82
u09	-10	-12	-21	-29	-6	-10	-12	-29	-21	-13	-7	-23	-13	-24	-27	-34	-1	3	-30	-29	-37	-45
u09	-6	-14	-9	-25	6	-6	-3	-23	-15	-4	-7	-21	-7	-12	-18	-20	2	2	-10	-16	-31	-42
u10	-25	-30	-22	-28	-32	-37	-33	-54	-45	-60	-43	-48	-38	-29	-55	-56	-35	-40	-57	-57	-73	-85
u10	-20	-26	-15	-14	-30	-34	-23	-46	-42	-51	-30	-49	-37	-25	-50	-52	-27	-33	-56	-60	-66	-93
u10	-44	-44	-38	-43	-52	-50	-48	-59	-57	-84	-50	-67	-45	-46	-70	-61	-46	-54	-79	-86	-87	-107
u11	83	43	-13	-23	5	2	-3	-11	3	-8	-6	-17	-16	6	-14	-24	-11	-11	-19	-19	-36	-26
u11	43	77	-8	-21	5	11	-2	-3	8	1	4	-10	-24	1	-26	-31	-6	-7	-21	-21	-29	-23
u12	-13	-8	84	61	-7	-16	-10	-34	-30	-35	-21	-56	-50	-36	-70	-72	-22	-22	-29	-37	-65	-82
u12	-23	-21	61	91	-27	-23	-27	-43	-34	-40	-21	-65	-56	-40	-83	-85	-27	-32	-35	-49	-84	-93
u13	5	5	-7	-27	72	44	47	-2	14	15	7	-24	-20	-9	-2	-1	-14	0	7	18	6	5
u13	2	11	-16	-23	44	86	46	-7	2	5	-1	-33	-20	-6	-10	-13	-17	-5	-9	-6	2	-6
u13	-3	-2	-10	-27	47	46	79	-11	-6	16	12	-28	-19	-10	4	-4	-19	-16	-1	1	13	9
u14	-11	-3	-34	-43	-2	-7	-11	88	57	-1	-5	-30	-29	-24	-15	-15	-25	-25	-26	-21	-37	-22
u14	3	8	-30	-34	14	2	-6	57	81	10	-4	-23	-29	-15	-5	-4	-18	-12	-13	-1	-39	-17
u15	-8	1	-35	-40	15	5	16	-1	10	79	42	-7	-10	-5	-6	-12	-8	-4	18	20	14	19
u15	-6	4	-21	-21	7	-1	12	-5	-4	42	76	-14	-11	-10	-3	-9	-2	-2	-3	-4	-7	-7
u16	-17	-10	-56	-65	-24	-33	-28	-30	-23	-7	-14	92	38	30	-30	-24	-32	-32	-14	-21	-42	-33
u16	-16	-24	-50	-56	-20	-20	-19	-29	-29	-10	-11	38	87	43	-31	-33	-28	-31	-8	-20	-29	-29
u16	6	1	-36	-40	-9	-6	-10	-24	-15	-5	-10	30	43	87	-20	-19	-24	-25	-9	-10	-15	-17
u17	-14	-26	-70	-83	-2	-10	4	-15	-5	-6	-3	-30	-31	-20	86	68	-39	-35	-34	-15	-22	-11
u17	-24	-31	-72	-85	-1	-13	-4	-15	-4	-12	-9	-24	-33	-19	68	91	-36	-37	-40	-20	-30	-13
u18	-11	-6	-22	-27	-14	-17	-19	-25	-18	-8	-2	-32	-28	-24	-39	-36	77	57	-39	-34	-36	-50
u18	-11	-7	-22	-32	0	-5	-16	-25	-12	-4	-2	-32	-31	-25	-35	-37	57	84	-25	-17	-25	-49
u19	-19	-21	-29	-35	7	-9	-1	-26	-13	18	-3	-14	-8	-9	-34	-40	-39	-25	94	52	-7	-10
u19	-19	-21	-37	-49	18	-6	1	-21	-1	20	-4	-21	-20	-10	-15	-20	-34	-17	52	90	0	6
u20	-36	-29	-65	-84	6	2	13	-37	-39	14	-7	-42	-29	-15	-22	-30	-36	-25	-7	0	102	54
u20	-26	-23	-82	-93	5	-6	9	-22	-17	19	-7	-33	-29	-17	-11	-13	-50	-49	-10	6	54	111

Tabuľka 6: Vyhodnotenie úspešnosti na reálnych dátach - časť druhá.

Ako vidieť z tabuľky, hodnota 20 najlepšie popisuje úroveň, kedy sa v testovacích dátach dá s istotou určiť identita rečníka. Preto je pre potreby pluginu táto hodnota zvolená ako tvrdý rozhodovací prah a po prevode do percentuálnej škály zodpovedá 50-tim percentám.

Je nutné podotknúť, že väčšina nahrávok bola dlhšia ako dve minúty. To sa tiež nepochybne podpísalo pod vynikajúci výsledok testovania. Takýto výsledok bol očakávaný behom testovania vo fáze návrhu systému. Krátke nahrávky by teda nepochybne zhoršili výsledky, ako naznačovali predimplementačné testy.

Výsledky testovania hovoria, že pre databázu 20-tich používateľov je možné považovať výsledok pluginu po druhej minúte za spoľahlivý na identifikáciu a verifikáciu druhej osoby.

6.7 Používateľské testy

Pre testovanie bol pripravený archív obsahujúci plugin a všetky závislosti. Daný archív bol rozdistribuovaný medzi jednotlivých používateľov s návodom na spustenie. Na testovanie bolo vybraných 10 používateľov. Potom bol používateľom vysvetlený význam pluginu a boli ponechaní na testovanie. Po určitom čase dostali všetci testovaní jednotnú sadu otázok na zodpovedanie. Jednotlivé otázky boli nasledujúce:

- Vedel si plugin používať od začiatku do konca? Ak nie, kde si sa zastavil?
- Ako hodnotíš intuitívnosť modulu?
- Vieš si predstaviť jeho využitie tebou samým?
- Čo by si zmenil alebo pridal?
- Skús vysvetliť pojmy *Self score* a *Best score* z rozhrania.

Vo všetkých prípadoch prebehla inštalácia bez problémov. Po tom, ako boli používatelia oboznámení s významom pluginu, dokázali sami určiť, že je jeho funkčnosť závislá od prebiehajúceho rozhovoru. Intuitívnosť ovládania bola vyhodnotená ako dobrá. Žiaden z používateľov neoznačil plugin za použiteľný ním samým. To by mohlo znamenať, že plugin si môže nájsť svoje uplatnenie len v úzko špecifikovanej skupine používateľov.

V prvej fáze testovania sa vyskytlo viacero nezrovnalostí pri vysvetlení pojmu *Self score* a *Best score* z grafického rozhrania. Preto sa vyskytli požiadavky na pridanie vysvetľujúceho textu týchto hodnôt. Ten bol pred ďalšími testami doimplementovaný formou vyskakovacieho okna prístupného z menu. V ňom sú obe hodnoty vysvetlené aj pomocou jednoduchého príkladu. Výsledky testov po doplnení informačného okna už obsahovali správnu odpoveď na poslednú otázku.

Po zapracovaní pripomienok považujem plugin za funkčný a použiteľný.

6.8 Návrhy na ďalší vývoj aplikácie

Vytvorený plugin ponúka základnú funkčnosť a obraz o možnostiach systémov s verifikáciou a identifikáciou rečníka. Tá bola demonštrovaná a zhodnotená. Pre použitie v komerčnej sfére plugin vyžaduje zvýšenie zabezpečenia jednotlivých súborov obsahujúcich hlasové odtlačky. Tie by mohli byť jednoducho podvrhnuté a plugin by mohol nesprávne vyhodnocovať na základe falošných informácií. Takéto zabezpečenie by vyžadovalo pravdepodobne inú reprezentáciu odtlačkov ako vo forme jednotlivých súborov. Rovnako by vyžadovalo aj bezpečné uloženie. Napríklad pomocou kryptografie.

Terajší systém podporuje vytváranie hlasových podpisov len behom vlastných rozhovorov s partnermi. To znamená, že si používateľ nemôže naimportovať už extrahované hlasové odtlačky. Jedinou možnosťou je ich manuálne nakopírovanie do príslušného adresára, ale to môže neskúsenému používateľovi spôsobovať nemalé problémy. Preto navrhujem začleniť do implementácie grafický modul, ktorý by umožňil import a export hlasových podpisov.

Pre potreby autorizovanej komunikácie si môže scenár vyžiadať, aby si komunikujúce strany vymenili navzájom podpisy pred začatím rozhovoru. Preto by rozhranie mohlo ponúkať možnosť vytvorenia si hlasového podpisu z nahrávky alebo priamo rozprávaním do mikrofónu. Vtedy by si mohol používateľ v kľude vytvoriť svoj podpis a predať ho druhej strane.

Sofistikovanejší systém by mohol predstavovať sieťový server, ktorý by spracúval väčšie množstvo hlasových odtlačkov a buď by priamo vyhodnocoval zhodu s jednotlivými ľudmi z databázy alebo by ponúkal synchronizáciu lokálnej databázy s databázou centralizovanou. Takýto server by našiel uplatnenie napríklad vo väčších firmách, kde by stačilo pre každého zamestnanca vložiť podpis len naň a odtiaľ by sa rozkopíroval na všetky relevantné stanice. Tým by sa značne zjednodušila réžia na udržiavanie aktuálnych podpisov a rovnako by sa ušetrila práca jednotlivým používateľom na správu vlastnej databázy.

Plugin v aktuálnej konfigurácii vyhodnocuje nahrávky v konštantných odstupoch a to 5 sekundových. To spôsobuje problémy na slabších počítačoch, ktoré nie sú schopné vyextrahovať hlasový odtlačok behom tejto doby a dochádza k preťaženiu systému, pričom výsledky prichádzaju s veľkým oneskorením. Preto by stálo za zváženie rozumnejšie rozhodovanie o obnovovacej frekvencii podľa aktuálneho zaťaženia.

V prípade väčšieho rozšírenia by bolo možné plugin preložiť do iných jazykov a rozšíriť o možnosť výberu jazyka.

Ak bude rozhodnuté plugin uvoľniť pre verejnosť, bude nutné vytvoriť alternatívny systém licencovania pre použitie funkcií BSC.

7 Záver

Zadaním práce bolo naštudovať problematiku spracovania rečových nahrávok za účelom identifikácie rečníka a následná implementácia zásuvného modulu do programu Skype ponúkajúceho verifikáciu a identifikáciu rečníka.

Behom práce som si musel:

- naštudovať teóriu z oblasti spracovania rečových signálov,
- naštudovať komunikačné rozhranie aplikácie Brno Speech Core,
- naštudovať komunikačné rozhranie programu Skype,
- naštudovať funkcie ponúkané knižnicou Qt najmä z oblasti tvorby používateľských rozhraní a správy vlákien,
- rozšíriť svoje vedomosti z oblasti objektového programovania.

Zásuvný modul bol naimplementovaný, zdokumentovaný, otestovaný a funguje.

Literatúra

[1] Burget, L. Systémy zpracování řeči. (prednáška) FIT: VUT, 02.12.2010.

[2] Yotaro KUBO, Regularized Discrimination of High-Dimensional Signal Representations for Automatic Speech Recognition, dissertation, Waseda University, 2010, [online], [08-01-2011], http://yota.ro/dthesis/dthesis.pdf.

[3] Nagenda KUMAR, Investigation of Silicon Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition, dissertation, Johns Hopkins University, Baltimore, Maryland, 2007, [online], [09-05-2011], http://128.220.117.40/~kumar/thesis.ps.

[4] Najim Dehak, Redah Dehak, Patrick Kenny, Niko Brummer, Pierre Oullet, and Pierre Dumouchel. Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification. In Interspeech 2009, Brighton, UK, 2009.

[5] Brno Speech Application Interface Documentation, [online], [09-05-2011], http://www.phonexia.com/docs/bsapi/index.html.

[6] Dokumentácia k SID modulu distribúcie BSC verzie 1.0.40.

[7] Článok o Skype na Wikipédii, [online], [09-05-2011], http://en.wikipedia.org/wiki/Skype.

[8] Referenčná príručka k SkypeAPI, [online], [09-05-2011], http://developer.skype.com/resources/public_api_ref.zip.

[9] Online dokumentácia knižnice Qt, [online], [09-05-2011], http://doc.qt.nokia.com/.

Zoznam príloh

Príloha A: Obsah priloženého CD.

Príloha B: CD so zdrojovými kódmi, spustiteľným zásuvným modulom a textom diplomovej práce.

Príloha A: Obsah priloženého CD

Priložené CD k práci obsahuje v adresári:

- src - zdrojové kódy implementovaného zásuvného modulu,
- qtsecskype - projekt Visual Studio 2010 vrátane zdrojových súborov,
- doc - text tejto práce vo formátoch .odt a .pdf,
- plugin - spustiteľný zásuvný modul s knižnicami potrebnými pre jeho beh,
- vcredist - *Microsoft Visual C++ 2010 Redistributable Package* pre 32 bit a 64 bit OS Windows.